

MAC Essentials for Wireless Sensor Networks

Abdelmalik Bachir*

Imperial College, London, UK

a.bachir@imperial.ac.uk

Mischa Dohler

CTTC, Barcelona, Spain

mischa.dohler@cttc.es

Thomas Watteyne

CITI Lab, A4RES INRIA, France

thomas.watteyne@ieee.org

Kin K. Leung

Imperial College, London, UK

kin.leung@imperial.ac.uk

MAC Essentials for Wireless Sensor Networks

Abdelmalik Bachir, *IEEE Member*, Mischa Dohler, *IEEE Senior Member*,
Thomas Watteyne, *IEEE Member*, Kin K. Leung, *IEEE Fellow*

Abstract

The wireless medium being inherently broadcast in nature and hence prone to interferences requires highly optimized medium access control (MAC) protocols. This holds particularly true for wireless sensor networks (WSNs) consisting of a large amount of miniaturized battery-powered wireless networked sensors required to operate for years with no human intervention. There has hence been a growing interest on understanding and optimizing WSN MAC protocols in recent years, where the limited and constrained resources have driven research towards primarily reducing energy consumption of MAC functionalities.

In this paper, we provide a comprehensive state-of-the-art study in which we thoroughly expose the prime focus of WSN MAC protocols, design guidelines that inspired these protocols, as well as drawbacks and shortcomings of the existing solutions and how existing and emerging technology will influence future solutions.

In contrast to previous surveys that focused on classifying MAC protocols according to the technique being used, we provide a thematic taxonomy in which protocols are classified according to the problems dealt with. We also show that a key element in selecting a suitable solution for a particular situation is mainly driven by the statistical properties of the generated traffic.

1 Introduction to WSN MACs

Sensor networks have been researched and deployed for decades; their wireless extension, however, has witnessed a tremendous upsurge in recent years. This is mainly attributed to the unprecedented operating conditions of wireless sensor networks (WSNs). As of today, a major problem in deploying WSNs is their dependence on limited battery power. A main design criterion is to extend the lifetime of the network without jeopardizing reliable and efficient communications from sensor nodes to other nodes as well as data sinks. A prominent example of today's non-optimized WSN deployment experiences is that the start-up alone costs the network a third of its battery power [1][2].

Optimizing every facet of the communication protocols is therefore vital and imperative. Such stringent design requirements can be met by a plethora of approaches, e.g. using cross-layer design paradigms, collaborative protocols, etc. This has led to copious novel distributed signal processing algorithms, energy-efficient medium access control and fault-tolerant routing protocols, self-organizing and self-healing sensor network mechanisms, reliable data aggregation algorithms, etc. These solutions stipulated first commercial activities as well as standardization approaches, including WOSA [3], KNX [4], IEEE 802.15.4 [5], IETF 6LowPan [6], IETF ROLL [7], etc.

The glue to and thread through all these solutions, however, is the medium access control (MAC) [8]. It is part of the link layer in the OSI layer model and is central to the proper functioning of any communication system, and hence focus of this survey paper. The prime role of the MAC is to coordinate access to and transmission over a medium common to several nodes. In the wireless context this is complicated by the fact that the common medium is the wireless channel which is broadcast in nature, i.e. any ongoing transmission interferes with any other transmission within communication range. Interference may lead to packet losses which need to be catered for with suitable retransmission mechanisms. Appropriate MAC rules have hence to be put in place to minimize interference and packet collisions.

This is traditionally achieved by optimizing the channel access, packet transmission and retransmission methods; packet¹ lengths (trading throughput with the probability of collision over the packet transmission duration); modulation and coding schemes (trading throughput with the reliability to achieve error free reception and hence avoiding re-transmission); transmission powers (trading communication with interference range); etc.

These techniques, however, are not necessarily suitable to WSNs. The peculiarities of WSNs, i.e.

- applications: very dispersed (\neq any wireless system);
- control: often decentralized (\neq cellular, broadcast, satellite);
- data: low load but highly directed (\neq ad hoc);
- links: volatile due to channel and dynamicities (\neq many wireless system);
- nodes: huge amounts, low complexity, energy limited (\neq any wireless system);
- run-time: very long (\neq any wireless system);

required a serious paradigm shift in MAC designs [2]:

The dispersed set of applications encompassed by WSNs ranges from small-size low-latency industrial monitoring applications to large-scale energy-constrained urban monitoring applications [7]. Since the scheduling and associated delays happen at the MAC, each application may require a different solution at the link layer.

The decentralized nature of a typical WSN roll-out complicates any attempt to achieve a network wide synchronization and hence common scheduling pattern. Decentralized mechanisms hence need to be invoked which – as will be exposed subsequently – are known to be less efficient than centralized ones.

The data load in WSNs is generally very low; however, the generated traffic is usually highly directed from many nodes towards one or a few data-processing sink units thereby exhibiting a convergecast communication pattern. This requires special attention in the design process as nodes closer to the sink need to handle more traffic than nodes in the perimeters.

The links between sensor nodes are very volatile. This is mainly the result of packet errors due to wireless channel effects; packet errors due to imperfect MACs; packet errors due to interference from other systems; and link unavailability due to network dynamicity, i.e. appearing and disappearing nodes. Suitable retransmission mechanisms hence need to be in place to mitigate this unreliability.

The number of nodes in the entire network as well as the number of (two-hop) neighboring nodes plays a central part in the WSN MAC design; the former influences the convergecast behavior as discussed above and the latter the MAC properties at local scale. The low

¹ Although it would be more correct to refer to frames in the context of MAC protocols, we reserve the term frame to TDMA MACs in subsequent exposures.

complexity of the nodes, reflected in low processing capabilities and very limited buffers, influences average scheduling delays and hence end-to-end data reporting times.

The largest design constraint, however, is the limited energy budget of a sensor node together with the requirement of long network runtimes. For instance, having a node continuously powered on drains an AA battery of 3000mAh in about 4 days [9], which is well below the typically required decade of operation. On the other extreme, even if a node is switched off all the time, inherent current leakages in the battery limits the battery lifetime to 10-15 years in dependency of the operating temperature; this renders network lifetime maximization with battery-powered nodes beyond this lifespan useless.

In typical sensor applications, the energy consumption is dominated by the node's radio consumption. From Fig. 1 it can be concluded that power consumption in sleeping mode is negligible to the power consumption in active mode. Since the radio is controlled by the MAC, the MAC is central in optimizing the WSN's lifetime.

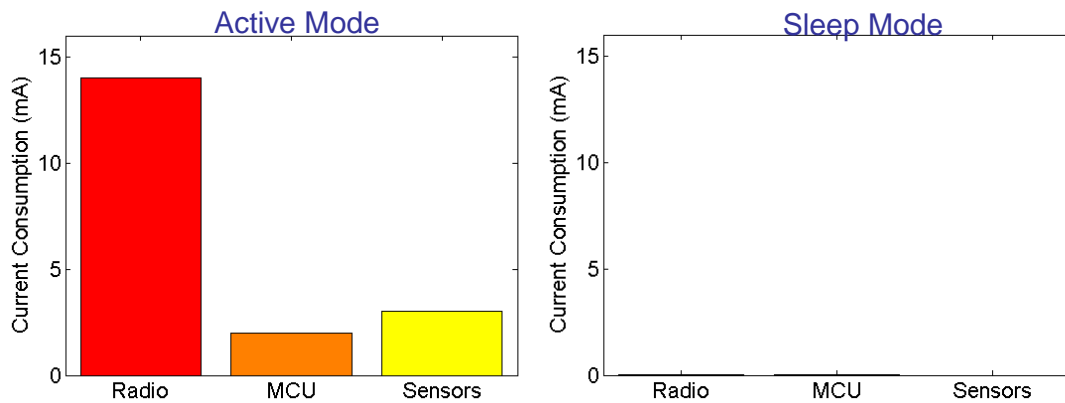


Fig. 1. Energy consumption of typical node components. Measurements taken with a node using a CC2500 radio chip and MSP 430 MCU and typical accelerometers [10].

In addition, note that different node powering mechanisms are available, such as non-rechargeable battery; rechargeable battery with regular recharging (e.g. sunlight); rechargeable battery with irregular recharging (e.g. opportunistic energy scavenging); capacitive/inductive energy provision (e.g. active RFID); etc. This has also an influence on the choice and design of the MAC protocol.

In conclusion, the aim of a WSN design is to guarantee its longevity under the given energy and complexity constraints. The MAC plays a central part in this design since it controls the active and sleeping state of each node. The MAC protocols hence needs to trade longevity, reliability, fairness, scalability and latency; throughput is rarely a primary design factor.

Previous milestone surveys [103], [104], and [105] have focused on classifying MAC protocols according to traditional MAC techniques used. Usually, such a classification distinguishes between protocols that use reservation for medium access and those that use contention. This classification, however, lacks useful and viable guidelines to network designers and field engineers because it often does not satisfactorily determines the MAC protocols that are the most suitable for a given set of circumstances. In this paper we cope with this lack in the literature by providing rather a thematic classification which is more suitable to the needs of WSNs. We center our classification on traffic patterns which we show

to have a major impact onto the right choice of MAC protocols. In addition, instead of classifying MAC protocols according to the specific sub-techniques used, we propose a classification according to the problems they target to solve, thus allowing efficient understanding of such solutions.

2 Taxonomy & Design-Drivers

The aim of this section is to categorize known medium access protocols and position them w.r.t. canonical solutions for WSNs. We argue that the traditional MAC taxonomy is not suitable to WSNs since obeying different design drivers when compared to other known wireless networks.

2.1 Traditional MAC Families

There are two main approaches for regulating access to a shared wireless medium: contention-based and reservation-based approaches. Thus, any derived MAC protocol is based on one of those two approaches or a combination thereof. In the following, we discuss the key features of these approaches and whether they are suitable to WSNs.

2.1.1 Reservation-Based Protocols

This approach requires the knowledge of the network topology to establish a schedule that allows each node to access the channel and communicate with other nodes. The schedule may have various goals such as ensuring fairness among nodes, or reducing collisions by avoiding that two interfering nodes or more access to the channel and transmit at the same time. TDMA (Time Division Multiple Access) is a representative example for such a reservation-based approach.

In TDMA, time is divided into frames and each frame is divided into slots. During a frame, each node is assigned a unique slot during which it has the right to transmit. As a consequence, transmissions do not suffer from collisions², which guarantees finite and predictable scheduling delays and also increases the overall throughput in highly loaded networks. The throughput is usually hard-limited, i.e. it cannot be increased beyond the utilization of all available slots. TDMA schemes also ensure fairness among nodes as each node is assigned a unique slot in each frame.

Although TDMA schemes have appealing features, they have some shortcomings resulting from their dependency on network topology and time synchronization. A given network topology is used to establish a collision-free arrangement and tight synchronization to ensure a common schedule among nodes. Both knowledge of topology and strict synchronization requires large overheads and/or expensive hardware and hence renders TDMA solutions less attractive in large-scale rollouts.

2.1.2 Contention-Based Protocols

This approach is fairly simple compared to reservation-based protocols, mainly because neither global synchronization nor topology knowledge is required. In a contention-based approach, nodes compete for the use of the wireless medium and only the winner of this competition is allowed to access to the channel and transmit. ALOHA and CSMA (Carrier Sense Multiple Access) are canonical representative schemes of contention-based approaches.

² We make a distinction between collisions caused by neighbors using the same channel and transmission errors resulting from interference with other signals. We call the former collisions and the latter transmission errors.

In CSMA, for instance, a node having a packet to transmit first senses the channel before actually transmitting. In the case that the node finds the channel busy, it postpones its transmission to avoid interfering with the ongoing transmission. In the other case that the node finds the channel clear, it starts transmitting (after possibly having waited a random time). CSMA does not rely on a central entity and is robust to node mobility, which makes it intuitively a good candidate for networks with mobility and dynamicity.

Despite its success, contention-based protocols suffer from degraded performance in terms of throughput when the traffic load increases [11]. In addition, the distributed nature prevents them to achieve the same efficiency as ideal reservation-based protocols. A comparative study is shown in Fig. 2, where reservation-based protocols clearly outperform the contention-based ones; however, at the price of requiring a network-wise synchronization.

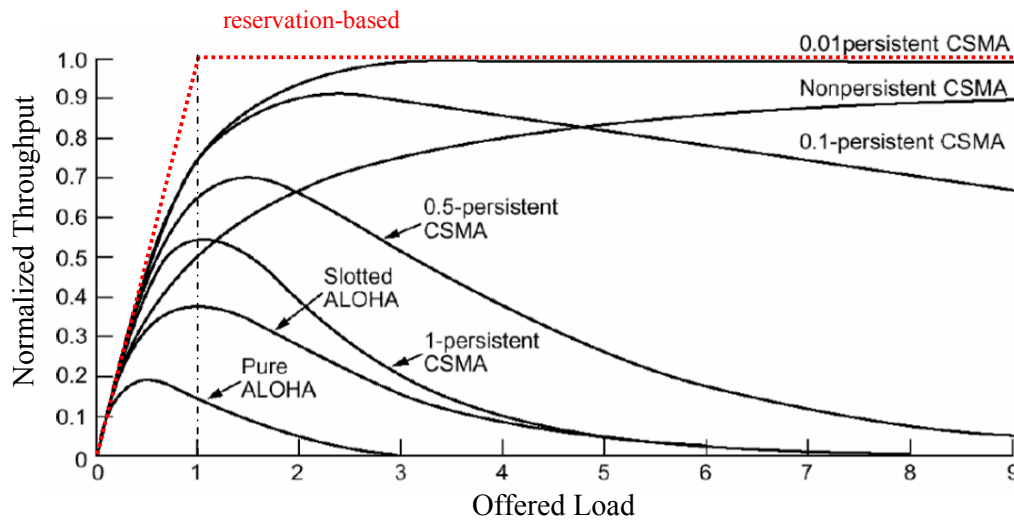


Fig. 2. Qualitative throughput comparison between reservation (red dotted line) and contention based (black solid lines, [12]) MAC protocols.

2.1.3 Suitability to WSNs

Whilst there had been extensive research efforts in the past for optimizing both MAC families as well as obtaining hybrid solutions thereof, these protocols are not necessarily suited to WSNs. As already alluded to in the introduction, WSNs are highly energy constrained and also normally operate at low loads. In the low load region of the MACs in Fig. 2, all protocols behave the same since all offered load can be accommodated by the wireless medium. Plotting for this low throughput region the load versus the energy consumption of the protocols, however, yields very different insights as exemplified in Fig. 3 [13].

Here, the power consumption is plotted versus the packet inter-arrival time (which can effectively be translated to throughput). Clearly, the CSMA contention-based protocol operates very far from an ideal protocol which uses a genie-like scheduler. This clearly motivated the research and development into protocols which perform closer to the ideal bound.

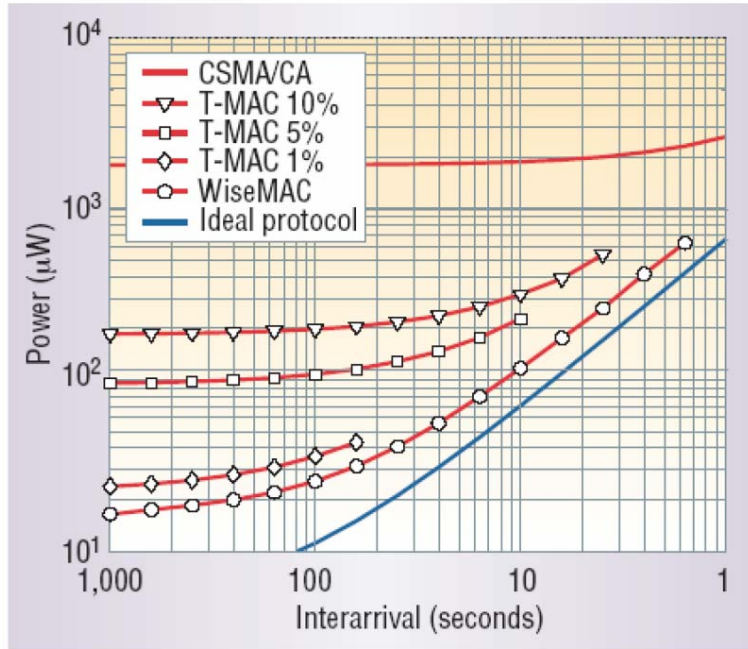


Fig. 3. Qualitative throughput comparison between an ideal MAC protocol and various contention based MAC protocols (taken from [13]).

2.2 Design-Drivers for WSN MAC Protocols

The design of MAC protocols for WSNs is mainly impacted by a high energy constraint but also by a low complexity of the nodes, their low computational capabilities and low memory footprints as well as poor synchronization capabilities. A functional MAC for WSNs hence ought to be highly energy-efficient but also ensure high reliability, low access delay and throughput given above impairments. Throughout this paper, we focus on energy efficiency as the prime design driver whilst also describing its consequences on the other design drivers such as reliability, delay and throughput. Designing an energy-efficient MAC protocol requires a thorough understanding of the main causes of energy dissipation, only after which techniques can be found which alleviate their effects.

2.2.1 Main Causes of Energy Consumption

The following effects seriously impair the energy dissipation of each sensor node:

Collisions. They may happen when a node is within the transmission range of two or more nodes that are simultaneously transmitting so that it does not capture any frame. The energy drained in the transmission and reception of collided frames is just wasted. Due to the large impact of collisions on protocols performance, MAC protocols should feature techniques to reduce or even avoid them.

Overhearing. It happens when a node drains energy receiving irrelevant packets or signals. Irrelevant packets may be for example unicast packets destined to other nodes or redundant broadcast packets. Irrelevant signals include the preambles used in some low power MAC protocols to occupy the communication channel (see Section 5).

Overhead. Protocol overhead may result in energy waste when transmitting and receiving control packets. For example, RTS and CTS control packets used in some protocols do not

carry any useful data to applications although their transmission consumes energy. For example, the exchange of RTS/CTS induces high overheads in the range of 40% to 75% of the channel capacity, because data frames are typically very small in sensor networks [14], [15].

Idle Listening. It happens when a node does not know when it will be the receiver of a frame, which is generally the situation. In this case, the node keeps its radio on while listening to the channel waiting for potential data frames. The amount of energy wasted whilst the radio is on is considerable even when it is neither receiving nor transmitting frames, as shown in Fig. 4.

Radio (sleep)	900 nA
Radio (idle)	1.5 mA
Radio (transmit)	22 mA
Radio (receive)	14 mA
Microcontroller (active)	8 mA
Microcontroller (idle)	2 mA

Fig. 4. Current consumption of main status of a typical radio (CC 2500) [16].

Sensor network applications usually generate low traffic load, thus the communication channel is expected to be idle most of the time. Under such circumstances, idle listening is the most significant source of energy dissipation. Without any specific energy management, nodes waste considerable amounts of energy as they keep their radios on for large time intervals while listening to an idle channel. Due to the importance of energy consumption of idle listening, energy-efficient MACs should make nodes sleep for long periods of time instead of enabling them to be permanently active. The main techniques used for reducing idle listening will be detailed below as well as in Sections 3 – 6.

2.2.2 Canonical Solutions

Based on these causes of energy consumption, a few canonical approaches have been proposed for wireless systems in general which are either common to or can easily be added to a large variety of WSN MAC protocols.

Reducing Collisions. Whilst the avoidance of collisions is naturally achieved in reservation-based MACs, contention-based MACs require special attention. Elements of the MAC protocols listed below are typically used in the context of WSNs.

- **CSMA/CA** (CSMA/Collisions Avoidance) is among the most widely used technique to reduce collisions in wireless networks. It is based on the exchange of RTS (Request To Send) and CTS (Clear To Send) mini packets prior to data transmission. In CSMA/CA, the transmitter starts by sending a RTS packet to the receiver. As RTS packets are small in size, the probability of them colliding is low. When the receiver receives a RTS, it replies by sending a CTS packet. The role of the CTS is to reserve the channel around the receiver so that interfering nodes in the receiver's vicinity refrain from transmitting so as not to collide with the active transmission. Although the RTS/CTS procedure efficiently reduces collisions in traditional wireless networks, it has some drawbacks in wireless sensor networks. First, data packet sizes are also usually small in sensor networks so that their collision probability is in the same order as for RTS packets. Therefore, its use does not improve but often even deteriorates performance. In addition, the use of RTS/CTS increases the energy consumption of the protocol. Finally, RTS/CTS packets can only be used for unicast transmissions.

- **MACA** (Multiple Access Collision Avoidance) [17] improves CSMA/CA by adding a random backoff time before the transmission of RTS packet to avoid collisions resulting from synchronized forwarding by multiple neighbors. Usually, the backoff time is picked according to a uniform distribution which, as per [18] and [19], is not the optimum choice.

CSMA/CA and MACA aim at reducing collisions by equally trying to protect all packets. In sensor networks, however, packets have different importance. For example, in surveillance applications, all nodes detecting an intrusion send an alert to the sink which generates peak traffic around the intrusion region. In this case, it is more important to protect the first packets so that they reach the sink rapidly.

- **Sift** [19] considers this issue by exploiting the spatio-temporal correlation that exists between events generated by nodes. Sift's goal is to reduce collision of the first R of N potential report from nodes to the sink.

Reducing Overhead. The above contention-based CSMA protocol family can be further optimized w.r.t. protocol overhead.

- **CSMA/ARC** (Adaptive Rate Control) [14] is a MAC protocol that avoids using RTS/CTS altogether to reduce the overhead while ensuring fairness between the forwarded traffic and the generated traffic. CSMA/ARC modifies the basic CSMA/CA protocol by omitting the RTS/CTS exchange and applying a backoff that is shifted according to the application periodicity, which reduces both backoff time and collisions. CSMA/ARC reduces the overhead further by avoiding explicit use of ACK packets: a data packet is considered successfully received when the upstream node sends it forward to its upstream node.

Reducing Overhearing. Overhearing irrelevant unicast packets can be avoided through a filtering based on the packet's destination addresses.

- **PAMAS** (Power-Aware Multi-Access with Signaling) [20] is based on MACA but differs from it by using a separate channel for RTS/CTS exchange. PAMAS makes use of the RTS/CTS exchange to inform nodes that receive them about the source, the destination, and the duration of the ongoing transmission. Thus a node that is not concerned with the transmission may switch off its radio to avoid overhearing unneeded transmissions. PAMAS has been designed for ad hoc networks with high traffic loads, in which transmissions and thus energy saving operations are frequent. In sensor networks, the situation is different: traffic loads are fairly low and RTS/CTS are mostly omitted. As a consequence, PAMAS does not realize high energy savings in wireless sensor networks.

There are other ways of avoiding other forms of overhearing related to the preamble used by some low power MAC protocols. These are described in detail in Section 5.3.6.

Reducing Idle Listening. The key idea for energy savings in wireless networks is to put nodes to sleep as long as possible while avoiding deafness and reducing overhearing and overhead. In networks with a pre-existing fixed infrastructure, such as WLANs, energy savings through reduced idle listening is less difficult because the infrastructure can be powered on continuously. The infrastructure can thus manage putting nodes in sleep and active modes. The IEEE 802.11 PSM (Power Save Mode) [21] for BSS (Basic Service Set) is a representative example of such an idea:

- **IEEE 802.11 PSM** targets energy saving in WLANs where all nodes are able to reach each other with a direct transmission. In PSM, nodes enter the sleep mode to avoid

draining energy in being continuously in active mode. Depending on the presence of an access point or not, two cases can be distinguished: PSM with a BSS (Basic Service Set) and PSM with an IBSS (Independent BSS).

With an access point (BSS), each node that wants to save energy sends a packet to inform the access point (AP). When it receives a positive response from the AP, the node starts its PSM procedure. In PSM, a node spends most of the time in sleep mode. It wakes up periodically to receive beacons from the AP. When a node is in a PSM mode, the AP cannot transmit packets to it according to the traditional procedure because the node is in sleep mode most of the time. Therefore, the AP buffers all packets destined to nodes that are in PSM modes. The beacons transmitted by the AP include a TIM (Traffic Indication Map) containing information about pending packets. When a node receives a beacon, it can know whether there are pending packets for it. To receive pending packets, the node has two options depending on whether the packets are unicast or broadcast. If the packet is unicast, the node chooses when to receive it. However, if the packet is broadcast, it is the AP that decides when it will be transmitted and it is to the node to decide whether to receive the packet or not. For unicast packets, the node determines when it wants to receive the packets and transmits a poll packet to inform the AP that it is ready for reception. Once the AP receives the poll packet, it starts sending packets to the node.

Without an access point (IBSS), all nodes maintain synchronization in a distributed way through the periodic transmission of beacon packets. In PSM, a source node that wants to transmit a packet has to know prior to transmission whether the destination node is in power save mode or not. This information is obtained from ATIM (Announcement Traffic Indication Message) transmitted during the period when all nodes are awake. If the destination node is in PSM then the source node announces the packet to be transmitted during the ATIM window, which is the period during which all nodes are awake. When the destination node receives the announcement message, it replies by sending an ACK back to the transmitter and stays awoken to receive the forthcoming data packet. Once the transmitter receives the ACK packet, it starts transmitting the data packet after the ATIM window according to the conventional backoff procedure described in [21]. Note that for broadcast/multicast packets, no ACK packets are expected from potential receivers.

The IEEE 802.11 PSM is an efficient way to save energy; however, it has two drawbacks. First, it is not suitable for multihop networks and, second, it introduces latency in the traffic exchange. Note that the latency issue can be alleviated because the node can switch to normal mode when it expects high activity periods.

In multihop ad hoc networks with no infrastructure, there is no central node being active in permanence to coordinate and manage sleeping periods of the other nodes. This complicates the task of letting nodes sleep without causing deafness.

Table 1: Summary of MAC improvements for canonical solutions.

Function	Protocols
Reducing Collisions	CSMA/CA [12], MACA [17], Sift [19]
Reducing Overhead	CSMA/ARC [14]
Reducing Overhearing	PAMAS [20]
Reducing Idle Listening	PSM [21]

From the above discussed protocols, which have been summarized in Table 1, it becomes clear that WSNs require very specific MAC protocols that ought to be designed taking their peculiarities into account. In the remainder of the paper, we explore the key ideas and the main energy efficient protocols available in the literature. Since WSNs are highly application tailored [7], we first dwell on Scheduled Protocols in Section 3 which are optimized for periodic high-load traffic (typical to e.g. multimedia applications); second, we deal with Protocols with Common Active Periods in Section 4 that are well suited for medium-load traffic scenarios (typical to e.g. industrial applications); third, we discuss Preamble Sampling Protocols that are the most convenient for rare reporting events in Section 5 (typical to e.g. metering applications); and finally we will also expose some hybrid protocols in Section 6 which combine the benefits of several protocols. Since we divert from the typical MAC taxonomy, we also expose switching points between these protocols according to our traffic-dependent taxonomy in Section 6.2.6.

3 Scheduled Protocols

3.1 Basic Idea

Periodic and high-load traffic is most suitably accommodated by means of reservation-based protocols, i.e. those which build a specific schedule. Generally, in the context of WSNs, such protocols are variants of TDMA (Time Division Multiple Access) combined with FDMA (Frequency Division Multiple Access) where different time slots and frequency channels can be used by different nodes. TDMA is attractive because – once the schedule is set up – there are no collisions, no overhearing, and minimized idle listening. In addition, TDMA offers bounded latency, fairness and good throughput in loaded (but not saturated) traffic conditions.

The central concern of TDMA type protocols is how to set up and maintain a specific schedule. To this end, three methods are used in the context of WSNs:

- **Scheduling of communication links:** This fairly traditional approach sets up a unique slot dedicated to a specific sender and specific receiver, thereby minimizing idle listening and eliminating collisions and overhearing. Since transmitter and receiver know exactly when to wake up, this is the most energy efficient solution given the schedule is set up and that packets need to be transmitted; however, varying traffic conditions, imprecise clocks and network dynamics require new schedules to be set up which incurs large overheads.
- **Scheduling of senders:** In this approach the slot is specified which is used by the sender which requires all receiving nodes to listen. It hence minimizes idle listening, eliminates collisions and reduces overheads to a certain extent (since any changes at the receiving side remain transparent to the established schedule); however, overhearing remains a problem of such an approach. A node, however, may minimize overhearing further through header filtering, i.e. when the packet is destined to another node, the receiver goes back sleeping during that slot.
- **Scheduling of receivers:** Here, the receiving slots are specified. Overhearing is eliminated, idle listening minimized and overheads are reduced (since network dynamics at the transmitting side are transparent to the schedule). However, collisions between various transmissions can potentially occur if more than one transmitter wishes to reach a specific receiver; suitable contention resolution methods are hence needed.

The first two variants of TDMA are suited to periodic, delay sensitive and fairly high-load traffic, the third to periodic and medium-load traffic. Whilst many variants of above protocols

exist, such as the beacon-enabled guaranteed time slot transmission during the collision free period of the IEEE 802.15.4 MAC to be exposed in Section 6, we shall discuss the recently emerged Time Synchronized Mesh Protocol (TSMP) [22] to exemplify its functioning.

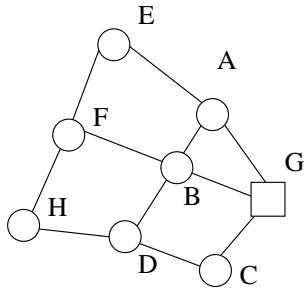
TSMP is TDMA-based and hence requires network-wide synchronization. Access is controlled by means of a tunable amount of timeslots which form a frame. The protocol is designed such that a node can participate in multiple frames at once allowing it to have multiple refresh rates for different tasks. TSMP employs in addition FDMA and frequency hopping, i.e. different links use differing frequency slots and the same link hops during its life time across different frequency slots. This yields high robustness against interference and other channel impairments.

A traditional approach to facilitate synchronization is beaconing, where longer frame lengths decrease the refresh rate at which synchronization is performed and hence power consumption and shorter frame lengths conversely invoke the opposite. TSMP does refrain from doing so because it requires long listening windows which consume power. Instead, TSMP nodes maintain a precise sense of time and exchange only offset information with neighbors to ensure alignment. These offset values are exchanged during active periods together with the usual data and acknowledgement packets hence invoking negligible overhead. TSMP nodes are active in three states: 1) sending a packet to a neighbor; 2) listening for a neighbor to talk; and 3) interfacing with an embedded hardware component. The duration of active periods, i.e. the duty cycling, is very flexible in TDMA; typical applications require duty cycles of less than 1%.

When applied, the sink typically retrieves the list of nodes, their neighbors and their requirements in terms of traffic generation. From this information, it constructs a scheduling table in both time and frequency. When implementing TSMP on IEEE 802.15.4 compatible hardware, 16 frequency channels are available. Exemplified by means of the scheduling table of Fig. 5, the TSMP link establishment and maintenance rules are simple:

- never put two transmissions in the same time/frequency slot;
- at a given time, a given node should not receive from two neighbors nor have to send to two neighbors.

Assuming that slots are 10ms long and node H sends a packet following route $H \rightarrow F \rightarrow B \rightarrow G$, then H send to F in slot [t5, ch.6], thereafter $F \rightarrow B$ in [t10, ch.11], then $B \rightarrow G$ at [t8, ch.8]. Latency is hence in this particular case 13 slots (130ms) and in general always guaranteed to be bound by a finite value which depends on the particular design of the time-frequency pattern.



ch.15								E->A		
ch.14	A->G			G->C						
ch.13						D->H				
ch.12		F->E					B->A			
ch.11					C->D				F->B	
ch.10			G->B							
ch.9										
ch.8	E->F					G->A	B->G			
ch.7				D->B					A->E	
ch.6					H->F					
ch.5		D->C					C->G			
ch.4								B->D		
ch.3										
ch.2	H->D					B->F				
ch.1			F->H							
ch.0					A->B					
	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10

Fig. 5. Possible schedule for given connectivity graph.

Fig. 5 shows that successive packets sent between two nodes are sent using different frequencies, following a preset hopping sequence. Therefore, the main aim of TSMP is not to increase network throughput, but rather to increase robustness against narrowband interference. See Section 3.3.8 for other implementations of Multichannel MAC protocols with different goals.

3.2 Discussion

The use of scheduled protocols generates the problems discussed below.

3.2.1 Complexity in Infrastructure-less Networks

The lack of a central access point that is one hop away from sensors is the crux of the problem which generally results in elevated complexity and high cost of

- maintaining tight synchronization; and
- distributing slots with good proprieties such as collision free slots.

Whilst distributed TDMA scheduling works well for medium sized networks, to determine a collision free schedule for huge networks becomes quickly infeasible, which clearly impacts scalability.

3.2.2 Scalability

The scalability of collision free slot assignment is a serious issue. Finding a collision-free schedule is a two-hop coloring problem. According to the Brook and Vizing Theorem, this requires $d(d-1) + 1$ distinct colors, where d is the degree of the graph.

3.2.3 Broadcast Communication

A further problem is broadcast: Unless the protocol is sender scheduled, the transmission of broadcast packets requires the repetition of the very same packet several times which is clearly not very energy efficient.

3.2.4 Reduced Flexibility

A schedule is usually set up given certain traffic generation and flow requirements. When these change, the entire schedule needs to be re-calculated. This class of protocols is hence also not adapted to highly dynamic topologies which occur in mobile environments.

3.2.5 Memory footprint

Collision free scheduling requires the knowledge of the two-hop neighborhood topology [23], which uses a large memory footprint. Maintaining memory status consumes energy that scales with memory size.

3.3 Principal Protocols

Subsequently, we will discuss some canonical protocols which aim at improving on some of the above listed shortcomings. For convenience, they are summarized in Table 2 at the end of this section.

3.3.1 Centralizing Scheduling at the Sink

In essence, **Arisha** [24] is very similar to TSMP in that the sink node gathers topology information about the network topology and assigns slots to all the nodes. Unlike TSMP however, Arisha details the algorithm used to compute the scheduling table where the slot assignment can be done according to a breadth first search or a depth first search. In the *breadth first search*, slot assignment starts at the leaf nodes of the tree. Leaf nodes having the same parent are assigned contiguous transmission slots. Other nodes sharing another parent are assigned the subsequent slots, etc. After that, parents are assigned the subsequent slots according to the same procedure. The advantage of this technique is that the parent minimizes switching times as it keeps listening to its children during a contiguous interval. As switching the radio consumes energy, this technique is energy-efficient. This technique also facilitates data aggregation. The drawback of this technique is that it introduces some delay as packets are backlogged in intermediate nodes; a parent first receives all its children packets before forwarding them. Backlogging packets at intermediate nodes may lead to other problems such as buffer overflow, which leads to packet loss and thus lowers throughput. In the *depth first search*, slot assignment starts at a leaf node. The next slot is assigned to its parent; the after next slot is assigned to its grand parent; etc; until reaching the sink. After that, the same is run for the next leaf node. This model avoids backlogging packets in intermediate nodes, which reduces end-to-end delay, packet loss thus increasing the throughput. However, it also involves larger energy consumption as intermediate nodes experience more frequent status switching.

The main issue of this protocol is its scalability. The described method does not describe a space re-use of un-overlapped slots. Each link has a network-wide unique slot. Another issue is that, as this is a centralized method, the entire network fails when the sink fails.

In **PEDAMACS** (Power Efficient and Delay Aware Medium Access Control protocol for Sensor networks) [25], the sink gathers information about traffic and topology during the setup phase. Based on this gathered information, the sink calculates a global scheduling and sends it to the entire network. The protocol assumes that the sink is powerful enough so that it can reach all nodes when it transmits. The uplink communications follows the TDMA scheme established by the sink. The collision-free scheduling is based on coloring the original conflict graph. The topology collection phase is based on CSMA to send information to the sink. In the topology learning phase, the sink node starts flooding *topology-learning* packets. At the

end of this phase, a spanning tree is constructed and the sink has the knowledge of the entire topology.

The main issue of this method is the traffic pattern is always convergecast. Other patterns thus cannot not be supported. In addition, the assumption that the sink reaches all nodes is not always satisfied. For nodes that do not receive the schedule transmitted by the sink, they wait for the next topology-learning phase and piggyback in their reply packets to the sink that they have not received the sink's transmission. Therefore, during the scheduling phase, intermediate nodes forward schedule packets to their neighbors that did not receive the sink's schedule.

The **BitMAC** [26] protocol constructs a spanning tree rooted at the sink. Each node acts as an access point for its direct children. This protocol is similar to the **IEEE 802.15.4** protocol, to be discussed below.

G-MAC (Gateway MAC) [27] uses a similar approach as above protocols; it defines a nodes acting as a gateway for a certain time, and then rotates nodes so as to balance the load among them. The TDMA frame of G-MAC contains three periods: the collection period, the traffic indication period and the distribution period. During the collection period, nodes contend for the channel and send packets expressing their future upload traffic. In the traffic indication period all nodes wake up and listen to the channel to receive the gateway traffic indication message (GTIM). The GTIM maintains synchronization among nodes and sets up slot owners among nodes having data to be sent to the gateway.

Although G-MAC reduces TDMA rigidity by allowing time-critical packets to be sent during the collection phase, it has some drawbacks. The main drawback of G-MAC is the large overhead experienced by the gateway node, especially during the collection phase that should be large enough to decrease the number of collisions among contenders. Even if G-MAC envisages rotation among nodes, the collection phase introduces a large overhead that increases the mean energy consumption per node. Another drawback is that nodes communicate directly with the gateway, which requires more gateways in the network and thus reduces the energy-efficiency of the protocol.

3.3.2 Distributed Scheduling

By using a local scheme, the drawback of transmitting information to a central node and getting back slots assignment is avoided. **SMACS** (Self-organizing Medium Access Control for Sensor networks) [28] allows nodes to establish a communication infrastructure between neighboring nodes by defining transmission and reception slots. SMACS is localized and distributed, that is, there is no need for a master node. It contains two phases: neighbor discovery and channel assignment. In SMACS, a channel is assigned to a neighbor if discovered. Each link works on a different channel, i.e. a different frequency randomly chosen from a given set, to reduce collisions. To find its neighbor, a node wakes up and listens for a given time to receive invitation packets. If it does not receive such a packet, it starts inviting others by sending an invitation packet. To save energy, nodes sleep and wake up randomly. There is, however, a non-vanishing probability that two nodes never meet.

When a link is formed between two nodes, they establish transmission-reception slots. These slots are used periodically to exchange data between nodes. Outside these slots, nodes sleep to save energy.

The advantage of this method is that it is simple to implement, because slots are formed on the fly. The drawbacks are: the energy consumption, the low degree of connectivity of the

network, and the difficulty of finding optimal routes. Furthermore, broadcast is not naturally supported since replaced by a series of unicast packets.

3.3.3 Using Localized Collision-Free Scheduling

TRAMA (TRAffic Adaptive Medium Access protocol) [31] determines a collision-free scheduling and performs link assignment according to the expected traffic. The protocol contains two phases: localized topology formation and scheduled channel access. The scheduled channel access allows each node to wake up only to transmit or to receive, which reduces idle listening and overhearing to zero.

The main issue with TRAMA is complexity and the assumption that nodes are synchronized network-wide.

FLAMA (FLow-Aware Medium Access protocol) [32] improves on TRAMA by avoiding periodic exchange of information between two-hop neighbors.

μMAC (micro MAC) [33] uses a similar idea as TRAMA: it also relies on external clock synchronization based on a beacon source. It also has a contention period for two-hop topology construction and a contention free period for data exchange. The contention period is also slotted; during this period, nodes transmit packets in slots. As there are no acknowledgments, retransmissions ensure reliability.

EMACs (Energy Efficient MAC) [34] defines three types of nodes: active, passive and dormant. Only active and passive nodes participate in communication, dormant nodes may be those that run out of energy or those that are recharging their batteries.

Active nodes participate in all of the operations and the passive nodes can only exchange information with a corresponding active node. This is similar to FFD and RFD defined by IEEE 802.15.4. Time is divided into frames that contain slots. Each slot contains three parts: CR (Communication Request), TC (Topology Control), and DATA. The TDMA style used is node-scheduled: each active node owns a slot. In the CR part of its slot, an active node listens to incoming requests from the passive nodes attached to it. In the TC part, an active node transmits acknowledgments to its passive nodes, synchronization information, and a table containing the schedule information.

A passive node attaches itself to an active node and follows it. The passive node spends its time in sleep mode to save energy; it only wakes up to transmit a CR to its corresponding active node or to receive from it a TC. As a passive node determines its corresponding active node independently of the others, more than one passive node may choose the same active node. This may result in a collision when more than one passive node sends a CR packet to the same active node at the same time. When a collision occurs, the active node indicates in the ACK packet that a collision has occurred. If it has no data to transmit during its DATA part, the active nodes allows nodes that experience collision to contend for transmitting data during its DATA part.

Transmissions from an active node are announced in the TC packets. All active nodes listen to all their neighboring active nodes. The active nodes should form a connected dominating set to ensure connectivity. A dominating set means that each node in the network is either a member of the dominating set or it has a neighbor that belongs to the dominating set.

Slot assignment is distributed: each active node transmits the used slots by itself and by its neighbors in bitmapping style to its neighbors. This allows neighbors to construct a local topology and to select slots that are not being used in the two-hop neighborhood to avoid collisions.

The first drawback of EMACs is that active nodes are permanently active and thus cannot save energy. Although EMACs envisages rotating roles, the resulting dominating sets with such forced rotation might not be optimal, thus leading to more energy consumption by more nodes.

PMAC (Position-enabled MAC) [36] uses a similar idea as EMAC. It defines three types of nodes: dynamic, passive and sleep. It adds a new period in the slot structure to enable nodes to request position estimate from the dynamic nodes.

3.3.4 Rotating Node Roles

In **PACT** (Power Aware Clustered TDMA) [29], the beginning of the TDMA frame contains mini-slots used for exchanging control information. The remainder of the frame contains the transmission slots of nodes according to some node assignment. The main drawback is the establishment of the node slot assignment, which is not energy efficient. During the control slot, each node declares upcoming transmissions so that nodes that will not be receivers go to sleep to save energy. PACT is based on passive clustering to organize the network into clusters and gateways connecting the cluster. To balance the working load, rotation is executed according to the residual energy of the nodes and traffic pattern that is piggybacked in traffic exchanged during the control phase.

The drawbacks are the complexity of the algorithm and the overhead of the control phase.

BMA (Bit-Map Assisted) [30] is also another protocol in which nodes alternate in acting as access points controlling TDMA slots of their neighbors.

3.3.5 Handling Node Mobility

MMAC (Mobility adaptive MAC) [35] caters for both weak dynamicity (due to topological changes, node (dis)association) as well as strong dynamicity (due to concurrent node (dis)association and physical node mobility). It builds a collision-free schedule based on estimates of traffic flow, mobility and dynamicity patterns. The core of the addressed problem is the static frame length, which makes mobile nodes wait for often unacceptable times before being able to communicate and also renders schedules obsolete. A dynamic frame time, that is inversely proportional to level of mobility, has hence been proposed. MMAC assumes that the sensor nodes are aware of their location and hence are able to estimate/predict mobility patterns. The proposed tradeoff was shown to outperform other comparable protocols.

The main disadvantage is the requirement on the knowledge of the position, which is often either not feasible or too energy consuming.

FlexiMAC (Flexible MAC) [37] is able to cope with some network dynamics and node mobility. It defines a contention period in which nodes exchange packets to build a data-gathering tree rooted at the sink. The tree is based on small links to reduce interference and increase spatial reuse. The slot distribution follows a depth first search (DFS), as explained above. The slot numbering starts with number 2 because slot number 1 is reserved for management of network dynamics; hence the name (FTS: Fault Tolerant Slot). The FTS is a large slot in which access is contention-based; it is envisaged for managing network dynamics: for example when nodes move, appear and disappear. Orphan nodes or new nodes use this slot to ask for communication slots.

Slot assignment is done according to a tree. Two kinds of slots exist: data-gathering slots and multifunctional slots. Data-gathering slots are used for uplink traffic from nodes toward the sink. Multifunctional slots are used for downlink traffic and synchronization.

Each node uses three lists: RSL (Receive Slot List), TSL (Transmit Slot List) and CSL (Conflict Slot List) for slot assignment and maintenance. The conflict slot list contains the slots of two-hop neighboring nodes with which the node interferes.

Although the idea is appealing, the tree structure lacks robustness and optimality. When a link fails a tree reconstruction even localized is necessary. Only parent-child and child-parent communications are optimal: a node that wants to communicate with a neighbor should pass by its nearest common parent. The period envisaged to cope with network dynamics may consume large amounts of energy. This method could be optimized, by letting nodes only listen to detect preamble during their reception slots.

3.3.6 Adapting to Traffic Changes

PMAC (Pattern MAC) [38] is a TDMA-based protocols in which time is divided into frames which in turn are divided into slots. Each node maintains a *schedule* according to which it either remains active to exchange data or goes to sleep mode to save energy. Before generating a schedule, each node establishes a *pattern* in which it announces its active and sleep slots. The pattern depends on the local traffic surrounding each node. If there is no traffic in a node's neighborhood, then the node doubles its sleeping slots to quickly maximize its sleeping duration. When the number of sleeping slots is larger than a certain predetermined threshold, the node continues to increase the number of its sleep slots linearly. After this, it broadcasts the generated pattern to its neighbors so that each neighbor can generate its schedule pattern that is used for determining the actual sleep/active slots. The resulting schedule depends on the pattern of the node (let it be the transmitter) and that of the receiver and whether there actually is data to be transmitted. For example, if the pattern of both the transmitter and the receiver indicate that they are both in active slots, and the transmitter has data to be transmitted, then the schedule of the transmitter will indicate that the considered slot will be active. Another example, if the pattern node of the transmitter indicates an active slot and the pattern bit of the receiver is indicating a sleep slot, then the schedule for this slot at the transmitter will be partially active mode. In the latter mode, the transmitter does not stay active during the whole slot; it only wakes up during the beginning of the channel to check whether it is receiving data from another node. If the node is not receiving data, then it goes to sleep mode for the rest of the partially active slot.

Although PMAC was developed to adapt to traffic changes by dynamically increasing the number of sleep slots, it does not reduce idle listening inside the active slots, which may be significant as each slot is designed to be long enough to handle a complete data transmission. In addition, using a fixed size slot may result in high collision rate when there is high contention during that slot. Moreover, like the majority of TDMA-based protocols, PMAC suffers from mobility of nodes and the overhead generated to maintain synchronization and to distribute slots.

3.3.7 Using Receiver Oriented Slot Assignment

O-MAC (Off MAC) [39] is a TDMA protocol that is similar to **Crankshaft**, to be explained below. Each node independently selects its reception slot. The node wakes up during its reception slot and senses the channel. If it detects a preamble, then it remains on until it receives the whole frame. Otherwise, it goes back to sleep to save energy. The key idea of O-MAC is to find receiver non-overlapping slots to avoid waking up other nodes uselessly. O-MAC also proposes to define broadcast slots for broadcast reception as using a set of unicasts to replace a broadcast is costly.

3.3.8 Using Different Communication Frequencies

PicoRadio [40] is a multichannel CDMA MAC using simple tone radio to wakeup neighbors. In PicoRadio each node listens to a common control channel (CCC) and broadcasts a channel assignment packet (CAP) to inform its neighbors about its channel. It also keeps track of all of its one and two-hop neighbors' channels to avoid choosing an overlapping channel with them. During this channel set up period, nodes execute the following procedure for channel assignment. Each node that wakes up listens for a certain time to gather information about its neighborhood's used channel. Then, it selects another unused channel from the channel pool and broadcasts its neighbors channel and its chosen channel. If a conflict occurs, the node that first detects the conflict switches to another unused channel. The quick convergence of such an algorithm is shown by simulation. Two techniques for channel assignment can be used: sender based or receiver based. In the sender based, the sender transmits packets during its transmission channel. In the receiver based, the receiver wakes up to receive packets. PicoRadio uses a sender based approach to avoid collisions. PicoRadio uses an ultra low power radio on an ultra low power channel to wake up nodes that will be addressed packets. The concept of having a second ultra low power radio just to wake up nodes has many limitations to be implemented in practice. First, the channel used by the ultra low power radio should have similar characteristics to the communication channel to ensure link maintenance. Next, to be efficient, the wake up radio should be very simple and thus may be very sensitive, which involves false positives caused by noise activating the data radio unnecessarily. Moreover, the ultra low power radio usually uses a tone to wake up nodes, and thus it is probably not able to carry address information, which may wake up a large number of nodes uselessly [41]. In PicoRadio, the wake-up radio is assumed to be capable of carrying information [42]. When a transmitter wants to send a packet to a node that is sleeping, it uses a wakeup signal addressed to the receiver. In the PicoRadio case, the address is the channel number of the receiver.

Wavenis [3] is another implementation of FHSS. Wavenis is based on a tree structure and thus supports convergecast traffic. Wavenis ensures high communication reliability at ultra low cost due to its relax synchronization scheme similar to **SCP MAC**, to be discussed below. However, due to its tree structure, communication patterns other than convergecast may not be optimal.

f-MAC [43] (framelet MAC) essentially targets reducing collisions and minimizing interference. It decomposes a packet into several framelets and sends them at a given frequency. As each potentially interfering node operates at a different frequency, the probability of collision is reduced. However, f-MAC may not realize large energy saving because nodes wake up frequently to check the channel and to receive the framelets.

Multichannel LMAC [44] proposes to enhance the TDMA-based LMAC with multi-channel support. In **LMAC** (Lightweight MAC) [45], nodes in a 2-hop neighborhood decide on a TDMA schedule in a distributed way, assigning different slots to different nodes. When the density of the network gets very high (i.e. a two-hop neighborhood is composed of tens of nodes) all slots end up assigned, and new nodes may end up without slots thus unable to communicate. [44] proposes for those nodes to pick a slot on another frequency. The number of potential slots is roughly multiplied by the number of frequency channels. This protocol hence allows more nodes to communicate in LMAC.

MMSN (Multi-Frequency Media Access Control for Wireless Sensor Networks) [46] uses an initial frequency assignment phase. Frequencies are assigned evenly to the nodes of a 1-hop neighborhood, with nodes learning their neighbors' frequencies. A node now needs to use the

destination's frequency when transmitting, and its own frequency when receiving. During network run-time, nodes are synchronized and time is sliced up into slots. A backoff-based CSMA algorithm solves contention between nodes in a given frequency/time slot. An interesting proposal is to use toggle snooping/transmission, in which nodes listen/transmit on both their own and the destination's frequency to make sure they are not receiving a packet while transmitting to a neighbor.

Y-MAC [47] is primarily designed to decrease latency. Nodes are synchronized and reception slots are assigned to each node, all on a common base channel. In case multiple packets need to be sent between neighbor nodes, successive packets are sent each on a different frequency following a pre-determined hopping sequence. This hopping sequence starts at the base channel. As a result, bursts of messages ripple across channels, significantly reducing latency. The presented implementation results serve as proof-of-concept for the multi-channel MAC approach.

Practical Multichannel MAC [48] is a multi-channel MAC protocol that does not assume nodes are synchronized. The protocol dynamically assigns channel to nodes, and groups nodes sharing a channel into clusters. As in [44] and [47], nodes all start at the same base channel. By periodically exchanging status messages measuring the loss ratio, nodes can detect when too much contention/interference is experienced on their channel. Clusterheads then take the initiative to hop on the next available channel, followed by the other nodes in its cluster. Inter-cluster communication is done by temporarily changing to the destination's channel. The resulting mechanism is transparent to the application and routing layers and it efficiently minimizes cross-channel communication while maximizing same-channel traffic. Throughput increase of as much as 50% is reported in dense networks. Although nodes do not need to be synchronized, they need to broadcast status messages to their neighbors frequently (every 1 second in the presented experiment). Energy-efficiency may thus be an issue under this setting. Other than increasing throughput, the solution elegantly copes with narrowband long lasting interference. In case of a permanent interference on a given channel, bad status reports will cause the nodes to hop to another channel, increasing the nodes' robustness.

3.3.9 Other Protocols

LMAC (Lightweight MAC) [45] uses two-hop collision free node scheduling. **AI-LMAC** (Adaptive Information-centric Lightweight MAC) [49] improves on LMAC and proposes that each node accesses to more slots. In **SS-TDMA** (Self-Stabilizing MAC) [50], traffic is scheduled in a fixed sequence of rounds (north, east, west, south).

RMAC (Randomized MAC) [51] belongs to on-demand slot assignment TDMA algorithms in which nodes ask the sink for slot assignments. RMAC does not consider multihop communication between sensor nodes; it assumes that sensor nodes are one-hop to the sink. Being an on-demand slot assignment algorithm, RMAC defines three periods in its frame: a contention period during which sensor nodes send requests to reserve communication slots, an acknowledgment period in which the sink sends acknowledgments to nodes that succeed during the contention, and a communication period in which each node uses the slot assigned by the sink to transmit its data. The main contribution of RMAC is to find the optimal number of minislots to be used during the contention period. The motivation of this is that a large contention period reduces collision but increases per-packet latency and a small contention period increases collisions. Based on the observation of the number of empty minislots, collided minislots, and successful minislots, RMAC estimates the number of competing nodes by using a maximum likelihood estimate. It also predicts the number of competing nodes in

the next round using a simple linear predictor. Finally, it finds the number of minislots to be used in the next round by optimizing the expected delay of a packet.

Table 2: Summary of MACs and their prime roles for scheduled protocols.

Function	Protocols
Canonical Solutions	TSMP [22], IEEE 802.15.4 [5]
Centralized Scheduling	Arisha [24], PEDAMACS [25], BitMAC [26], G-MAC [27]
Distributed Scheduling	SMACS [28]
Localization-Based Scheduling	TRAMA [31], FLAMA [32], μ MAC [33], EMACs [34], PMAC [36]
Rotating Node Roles	PACT [29], BMA [30]
Handling Node Mobility	MMAC [35], FlexiMAC [37]
Adapting to Traffic Changes	PMAC [38]
Receiver Oriented Slot Assignment	O-MAC [38]
Using Different Frequencies	PicoRadio [40], Wavenis [3], f-MAC [43], Multichannel LMAC [44], MMSN [46], Y-MAC [47], Practical Multichannel MAC [48]
Various Functionalities	LMAC [45], AI-LMAC [49], SS-TDMA [50], RMAC [51]

4 Protocols with Common Active Periods

In this approach, nodes define common active/sleep periods. The active periods are used for communication and the sleep ones for saving energy. This approach requires that nodes maintain a certain level of synchronization to keep active/sleep periods common to all nodes. During the active periods, nodes contend for the channel using contention-based approaches, such as pure CSMA, IEEE 802.11 DCF, etc.

The contention-based approach achieves its highest performance in applications in which traffic is periodic such as monitoring and applications in which keep-alive packets are periodically exchanged to ensure network reliability. However, the use of common active/sleep periods may not be suitable for applications with irregular traffic, because nodes use contention inside active periods, which would be prohibitive when nodes wake up without communicating, and may cause collisions when there is high traffic that cannot be absorbed by the initially envisaged size of the active periods.

4.1 Basic Idea

SMAC (Sensor MAC) [52] is a seminal work in this area; we take it as a representative protocol. SMAC copes with idle listening by repeatedly putting nodes in active and sleep periods. Nodes turn off their radios in sleep periods to save energy and they turn them on in active periods to exchange packets. Active periods are of fixed size whereas the length of sleep periods depends on a predefined duty-cycle parameter.

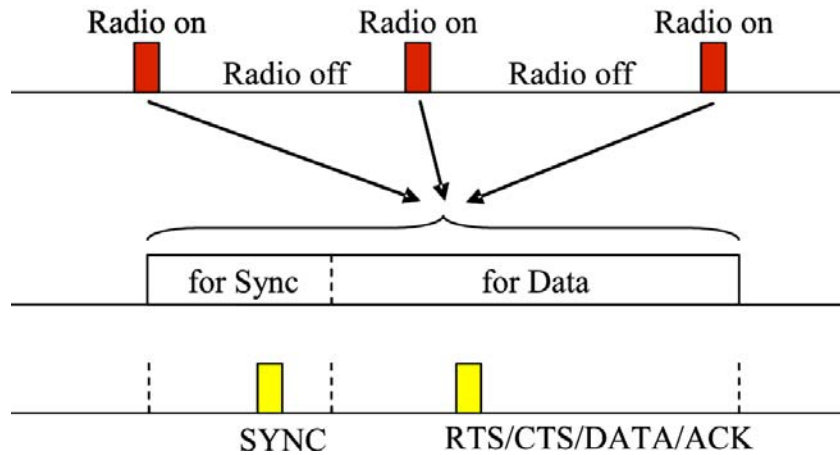


Fig. 6. SMAC alternates turning on and off the radio. SMAC splits the active period into two sub-periods: one for exchanging sync packets and the other for exchanging data packets. Data packet exchange may require RTS, CTS and ACK [52].

SMAC deals with deafness by making nodes share common active periods. Using common active periods requires synchronization establishment and maintenance among nodes. SMAC splits active periods into two sub periods: one for exchanging *SYNC* packets and the other one for exchanging data packets (as shown in Fig. 6). Each sub period is divided into mini-slots³. In both these sub-periods, nodes contend for the channel in a similar way as the IEEE 802.11 DCF [21] nodes perform a carrier sense first and then transmit in the next mini-slot if the channel is sensed free.

Each node using SMAC should have a schedule according to which it determines when it turns on its radio and when it turns it off. When deployed for the first time, a node starts by looking whether there are existing schedules in the network. The node keeps continuously listening to the channel for a duration of at least one active period plus one sleep period. If the node receives a *SYNC* packet, then it adopts the schedule carried by that packet. If it does not receive any *SYNC* packet, the node chooses its own schedule and follows it. Once a node has a schedule, it disseminates it throughout the neighborhood by broadcasting a *SYNC* packet with that schedule. Some of the node's neighbors receive the *SYNC* packet; these neighbors adopt the schedule and continue disseminating it throughout the network.

Nodes that follow the same schedule form a virtual cluster. A network is most likely to contain several virtual clusters. Transmission errors, collisions, large end-to-end delays, simultaneous self schedule selection and other factors may result in different *SYNC* packets with different schedules being transmitted in the network. Some nodes may receive several *SYNC* packets with various schedules. These nodes are called *border nodes*. Border nodes should adopt all the schedules they receive and thus keep their radios on during all the active periods determined by these schedules. Border nodes sustain network connectivity by ensuring packet passing from one cluster to another. Some implementations suggest that border nodes adopt only some schedules to reduce the time during which the radio is on. Although this further saves energy, it may cause network fragmentation as some virtual clusters may be isolated.

³SMAC implementation in NS2 (Network Simulator 2) [53] uses 31 mini slots for the *SYNC* sub-period and 63 mini-slots for the data period.

Applications may want to send packets while nodes are in sleep periods. SMAC postpones these transmissions until the next active period. As nodes sleep most of the time, strong contention for the channel is expected in the active periods. Specifically, nodes are implicitly synchronized at the beginning of active periods and there is a significant chance they simultaneously access to the channel at the beginning of active periods. SMAC copes with this kind of collisions by having nodes backoff for a random duration before transmission. SMAC also copes with collisions through the use of traditional mechanisms such as RTS/CTS exchange and virtual listening according to a NAV (Network Allocation Vector). The NAV contains a value that tells the node if there is an ongoing transmission and, if so, when it ends. NAV sets this value from overhearing headers of RTS, CTS and data frames — these headers carry information about transmission durations.

In SMAC, nodes do not transmit long packets in a single packet because this involves the retransmission of the whole packet in the case of a collision, even when only a few bits are corrupted. Instead, nodes fragment each long packet into many independent small packets and transmit them in a burst. Nodes use RTS/CTS only before transmitting the first small packet. The RTS/CTS exchange, in this case, reserves the channel for the whole burst duration instead of reserving it only for the subsequent packet as usual. Although this is unfair from a per-hop MAC level, it saves the energy of using RTS/CTS exchange before each small packet transmission.

4.2 Discussion

The use of common/sleep periods of a fixed size generates the problems below.

4.2.1 Rigidity

Determining the optimal size of active periods requires taking into account two parameters: idle listening and collisions. Short active periods reduce idle listening, but they increase contention and thus collision rates. Long active periods do the opposite; they reduce contention at the cost of increased idle listening. SMAC uses a fixed pre-calculated size for active periods that is optimized for an expected workload. This makes SMAC rigid, as nodes have no means to dynamically change their duty-cycle to meet time-varying or spatially non-uniform traffic loads. Note that variable workloads are expected in sensor networks as some nodes may be involved in relaying traffic more than others. For instance, nodes that are closer to a sink are most likely to relay more traffic than border nodes. Event-based reporting is the most pathological case for this rigidity. In a nutshell, the difficulty is to find a suitable length for active slots.

4.2.2 Sleep Delay

Sleep periods do save energy; however, they introduce extra end-to-end delay called *sleep delay*. Sleep delay increases communication latency in multihop networks, as intermediate nodes on a route do not necessarily share a common schedule. In a nutshell, the difficulty is to make a trade off between sleep delay and optimal active periods.

4.3 Principal Protocols

Subsequently, we will discuss some canonical protocols which aim at improving on some of the above listed shortcomings. For convenience, they are again summarized in Table 3 at the end of this section.

4.3.1 Increasing Flexibility

TMAC (Timeout MAC) [54] follows up on the basic idea introduced by SMAC that consists in using common active/sleep schedules: nodes determine their active/sleep schedules in a way similar to SMAC. TMAC alleviates SMAC's rigidity by proposing an adaptive duty-cycle in which the duration of active periods is no longer fixed but varies according to the traffic. The key idea of TMAC consists in making a node predict channel activity during an active period so that it can switch its radio off before the active period ends, in case it does not expect any traffic. Fig. 7 shows the overall operation of TMAC and its difference compared to SMAC.

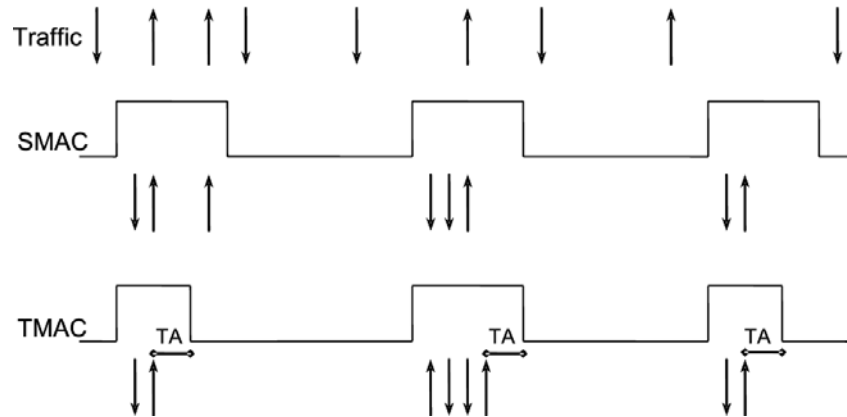


Fig. 7. TMAC downsizes active period lengths to further save energy. The arrows indicate transmitted and received frames. Both TMAC and SMAC move the traffic that comes during a sleep period to the subsequent active period. TMAC prematurely ends an active period if no traffic occurs for a duration of TA [54].

By downsizing active period lengths, TMAC saves more energy than SMAC. The proportion of this saving depends on the amount of time cut back on the initial active period duration; the more nodes sleep during active periods, the larger the saving. To optimize the sleep period durations, TMAC moves all communications to a burst at the beginning of active periods. Therefore, a node can determine that there will be no communications in the remainder of an active period if no *activation event* occurs within the duration of TA . An activation event may be, for instance, the reception of a frame or sensing some noise considered as collision. The minimum duration of TA should be long enough to span the maximum contention duration and the RTS/CTS exchange (see Fig. 8).

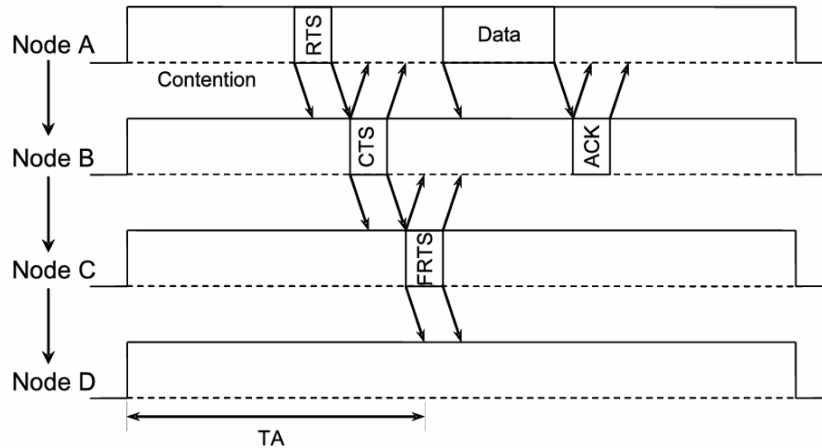


Fig. 8. The transmission of FRTS aims at keeping node *D* awake. FRTS control frames make it possible for TMAC to achieve a transmission over three hops within a single active period. This technique reduces the end-to-end latency [54].

By having nodes ending their active periods prematurely, TMAC partially breaks the synchronization among nodes within a virtual cluster, which leads to the *early sleep problem*. The early sleep problem happens when a third hop node, supposed to be the next relay of an ongoing transmission, prematurely goes to sleep. TMAC copes with this by using the FRTS (Future Request To Send) frames sent to the third hop node before its *TA* timer expires. Thus, the third hop node stays active and then receives the next transmission right away instead of receiving it in the next active period in case FRTS was not used.

In variable workloads, TMAC saves about five times more energy than SMAC does. However, this is achieved at the cost of an increased latency and thus reduced throughput. Although TMAC improves on SMAC's energy savings, it still suffers from the main problem of the high cost of maintaining common active/sleep schedules via exchanging *SYNC* packets.

E2MAC [55] uses a similar idea as TMAC except that it accumulates packets in a node until they reach a certain buffer limit size. When the buffer is almost full, the node transmits data in a burst. Although, this increases energy savings, it also increases latency and is not suitable for delay-critical applications.

SWMAC (Separate Wakeup MAC) [56] targets reducing the length of active periods by dividing it into slots; each node is assigned a reception slot during which it wakes up to receive data. Slot distribution is performed according to nodes' addresses. Each slot is long enough to handle a full transmission, including RTS, CTS, DATA and ACK packets. Therefore, SWMAC will suffer from the same drawbacks as SMAC.

4.3.2 Minimizing Sleep Delay

Adaptive Listening [57] suggests the use of overhearing to reduce the sleep delay. In adaptive listening, the node that overhears its neighbor's transmission and learns from it when that transmission ends may sleep in the meantime and then wakes up just when the transmission ends. This idea has also been proposed in **nanoMAC** [58]. Then, the node wakes up after that transmission even if it might happen during its sleep period. This makes it possible for the node's neighbor to immediately send data to it, instead of waiting for the node's next scheduled active time.

Technically speaking, nodes can learn about when a transmission ends if they receive the frame header that indicates the frame length or if they receive RTS or CTS that precede the data frame — RTS and CTS frames indicate the transmission duration of the data frame.

In slotted protocols that use common active/sleep schedules, the sleep delay is a serious drawback because it increases communication latency. Improvements such as adaptive listening and TMAC only affect the next hop and the next two-hop nodes respectively. The following protocols aim at minimizing the sleep delay further.

DSMAC (Dynamic SMAC) [59] protocol dynamically changes each node's duty-cycle to meet applications' demands. A node increases its duty cycle by adding extra active periods when it requires less latency or when it observes an increasing traffic load. DSMAC assumes that all nodes start with the same duty cycle. Then, when a node needs to increase its duty cycle, it sends a *SYNC* packet to its neighbors to inform them about its additional active schedule. After receiving the *SYNC* packet, each neighbor locally decides whether to increase its duty cycle to meet the announced schedule or not. The key idea that makes DSMAC work even with nodes that do not increase their schedules is that, initially, active periods never get changed; nodes only insert their new active schedules in the middle of the sleep period. Note that nodes can also decrease their duty-cycles by removing the added active periods.

FPA (Fast Path Algorithm) [60] makes nodes wake up for an additional time, even during their pre-scheduled sleep periods, to ensure timely relaying of frames. A node uses its hop-distance from the sender to estimate when its upstream neighbor will send a frame to it. Then, the node wakes up at the estimated time only to receive and potentially forward the frame to its downstream neighbor. The node sets these additional wakeup times from information piggybacked in the first data packet on that path.

DMAC (Data-gathering MAC) [61] considers the situation where many sources send data to a sink through a unidirectional tree, called convergecast communication. Nodes exploit this tree to determine their active schedules. A node determines its active schedules according to the traffic load and to its depth in the tree. The active periods of DMAC are similar to the additional active periods of FPA. DMAC mainly targets stationary networks as it does not envisage common global active periods. Thus, dynamicity may decrease DMAC's performance drastically.

Q-MAC (Query MAC) [62] proposes a scheduled MAC that adapts to traffic conditions. It is similar to DMAC. It defines a static sleep schedule for all nodes. Nodes that are one hop away from the sink use their initial sleep schedule. Nodes that are two hops from the sink shift their sleep wakeup periods to reduce latency. Q-MAC uses a common overlapping active period to allow communication between one-hop and two-hop neighbors with minimum latency. The shifted periods aim at reducing idle listening and alleviating contention to reduce collisions. In contrast to DMAC, Q-MAC defines shifted active periods in the sense of downlink traffic carrying queries from sink to source nodes. For uplink traffic carrying data from sources to sink, Q-MAC has two options depending on whether the sink knows the route length to sources or not. If the sink knows the route length then intermediate nodes can expect the instant of data transmission from sources to the sink. In this case, each intermediate node wakes up only to receive data and forward it back to its uplink neighbor. In the second case, the sink and thus intermediate nodes do not know the route depth from sources to the sink. In this case, Q-MAC lets each node remain active from the instant of the query reception to the instant of data reception and forwarding. The goal of this idea is to reduce data forwarding latency.

Although Q-MAC reduces latency, its energy efficiency is questionable as nodes spend long times in idle listening, especially in the second case when route lengths are not known in advance.

4.3.3 Handling Mobility

The initial SMAC protocol targets stationary sensor networks and it does not envisage specific optimizations to handle node mobility efficiently. With mobility, the node's schedule is no longer valid whenever the node moves to another virtual cluster. To re-establishes a new schedule, a node keeps continuously listening for a duration of an active period plus a sleep period to receive a sync. Mobility decreases SMAC's energy savings as mobile nodes waste extra energy in establishing new schedules. Furthermore, mobility increases communication latency as mobile nodes spend additional time to establish a new schedule and setup a connection.

The **MSMAC** (Mobility-aware SMAC) [63] proposes a mechanism that adapts the duty cycle of SMAC to improve connection setup times in mobile environments. Nodes measure changes in received signal levels of the periodic *SYNC* packets and use them to estimate the mobility speed. A node first estimates the speed with each one of its neighbors. Then it informs its neighbors about the maximum estimated speed by including that speed in *SYNC* packets. When the node's neighbors receive the *SYNC* packets, they create an *active zone* around it. In active zones, nodes increase their active periods by staying awake longer to reduce the connection setup time.

4.3.4 Minimizing Number of Schedules

Multiple active schedules lower SMAC's energy saving rates as nodes spend more time in active periods. Experiments with motes reported in [60] and results of simulations reported in [64], show that more than half of the nodes have more than one active schedule. The **GSA** (Global Schedule Algorithm) [60] focuses on minimizing the number of active schedules by making all nodes within a sensor network converge to a common global schedule. The GSA uses the schedule's age to determine which schedule to keep; when a node has to select between many schedules, it selects the oldest one. Results reported in [60] show that the GSA converges to one schedule in a network of 40 nodes organized in a linear topology. A refinement to this mechanism has been proposed in [65].

4.3.5 Using Statistical Techniques

RL-MAC (Reinforcement Learning MAC) [66] tackles the problem of optimizing active and sleep periods with the double aim of increasing throughput and saving energy. RL-MAC uses MDP (Markov Decision Process) to model the process of active time reservation. The main drawback of this protocol is that it relies on a constant traffic load over a long period of time.

U-MAC (Utilization-based MAC) [67] does not adapt the same duty cycle for each node. In U-MAC, each node uses a utilization function to tune its duty cycle. The utilization function is the ratio of the actual transmissions and receptions performed by the node over the whole active period. The utilization function is less than 1. If the utilization function is low then the node is experiencing a long idle period within the active period. In U-MAC a node maintains two values, U_{max} and U_{min} , and adapts its activity period to let its current utilization function in this interval. U-MAC's basic idea is akin to T-MAC's and thus they share the same drawbacks.

4.3.6 Using a Wakeup Radio

RMAC (Reliable MAC) [68] targets increasing communication reliability in sensor networks. It proposes techniques to reduce collisions and congestions through the use of implicit ACKs, adaptive retransmissions, and transmission rate control. For example, a sender does not transmit another packet immediately but waits for its parent's end of transmission to avoid colliding with it.

E2RMAC (Energy-Efficient Reliable MAC) [69] improves the energy-efficiency of RMAC by reducing idle listening and overhearing. It is based on the use of an additional wakeup radio. When a node wants to send a packet, it sends a tone on its low power radio. As this tone wakes up all the neighbors, the node sends a filter packet on the data radio to let the non-concerned nodes go to sleep⁴. E2RMAC integrates the same features of RMAC like immediate data forwarding to save on ACK transmissions. After sending a packet, a node goes to sleep mode for a duration equal to the time needed by the receiver to forward the packet, which is similar to adaptive listening.

Table 3: Summary of MACs and their prime roles for protocols with common active periods.

Function	Protocols
Canonical Solution	SMAC [52]
Increasing Flexibility	TMAC [54], E2MAC [55], SWMAC [56]
Minimizing Sleep Delay	Adaptive Listening [57], nanoMAC [58], DSMAC [59], FPA [60], DMAC [61], Q-MAC [62]
Handling Mobility	MSMAC [63]
Minimizing Schedules	GSA [60]
Statistical Approaches	RL-MAC [66], U-MAC [67]
Using Wake-Up Radio	RMAC [68], E2RMAC [69]

5 Preamble Sampling Protocols

5.1 Basic Idea

Preamble sampling protocols do not use common active/sleep schedules; instead, they let each node choose its active schedule independently of other nodes around. In preamble sampling protocols, a node spends most of the time in sleep mode; it wakes up only for a short duration to check whether there is a transmission on the channel. To avoid deafness, each data frame is preceded by a preamble that is long enough to make sure that all potential receivers detect the preamble and then get the data frame.

⁴ Note that recent radio chips such as the Chipcon CC 2500 implements this option in hardware, i.e. the radio chip is able to perform address filtering without the need for microcontroller processing, which is faster and reduces the overhead as the microcontroller can stay in low-power mode.

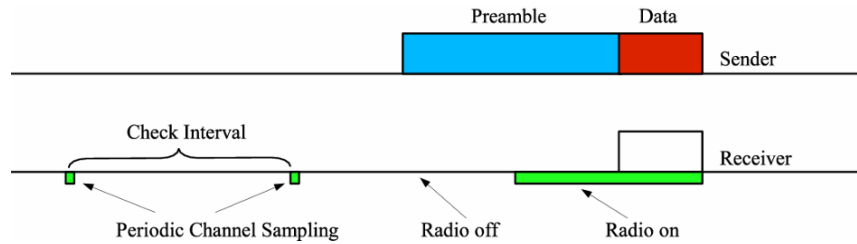


Fig. 9. Preamble sampling.

Fig. 9 shows an example of a preamble sampling protocol operation. According to the duty-cycle parameter, nodes periodically switch their radios on to sample the channel. If a node finds that the channel is idle, it goes back to sleep immediately. However, if it detects a preamble transmission on the channel, then it keeps its radio on until it receives the subsequent data frame. Right after the reception of the data frame, the node sends an ACK frame, if needed, and goes back to sleep afterwards. To be effective, the duration of the preamble transmission needs to be at least as long as the *Check Interval (CI)* defined as the period between two consecutive instants of node wakeup. In this way, a node makes sure that all potential receivers are awake during its preamble transmission so that they get the subsequent data frame. The preamble sampling technique has been combined with Aloha in [70] and with CSMA in [71].

We can find in the literature other terminologies that refer to a similar approach, e.g. Cycled Receiver [72], LPL (Low Power Listening) [73] and Channel Polling [74]. Hereafter, these protocols are collectively referred to as preamble sampling protocols.

5.2 Discussion

By reducing synchronization overhead, preamble-sampling protocols realize larger energy savings; however, this comes at the cost of a longer preamble. The use of a longer preamble causes two major problems which are discussed below.

5.2.1 Costly Collisions

The preamble sampling technique shifts the cost of coping with idle listening from the receiver to the transmitter. The receiver uses less energy as it wakes up only for a very short time, whereas the transmitter uses more energy as it transmits a long preamble before each data frame. This is highly beneficial for applications in which transmission is not frequent, such as surveillance. However, in some applications when traffic is quiescent, collisions are frequent.

The high transmission cost counteracts the energy efficiency of preamble sampling protocols in situations with high collision rates. When a collision occurs, it very likely implies retransmission, which is costly. Preamble sampling protocols should have robust mechanisms in place for avoiding collisions.

As a packet contains a long preamble, transmission and thus collisions/retransmission are very costly. It is a good idea to use techniques (coding for e.g.) to make the transmission of data more error-resilient.

5.2.2 Limited Duty Cycle

In order to extend nodes' lifetime, applications need to save more energy by lowering the duty cycle. Lowering the duty cycle implies putting nodes in sleep mode for larger periods, which means extending the check interval.

While using a larger check interval reduces the cost of idle listening at the receiver, it increases the transmission cost as the transmitter uses a longer preamble. Thus, nodes cannot indefinitely extend their check intervals with the aim of saving more energy.

There is an optimal value for the check interval beyond which nodes waste more energy in transmission than they save in reception. Finding the optimal check interval depends upon several parameters such as transmission power, reception power, traffic load and switching times of the radio chip. Fig. 10 shows that there is an optimal value for the check interval that maximizes the lifetime. This value also depends on the traffic load. Therefore, preamble-sampling protocols have a limited duty cycle that is determined by the optimal check interval value.

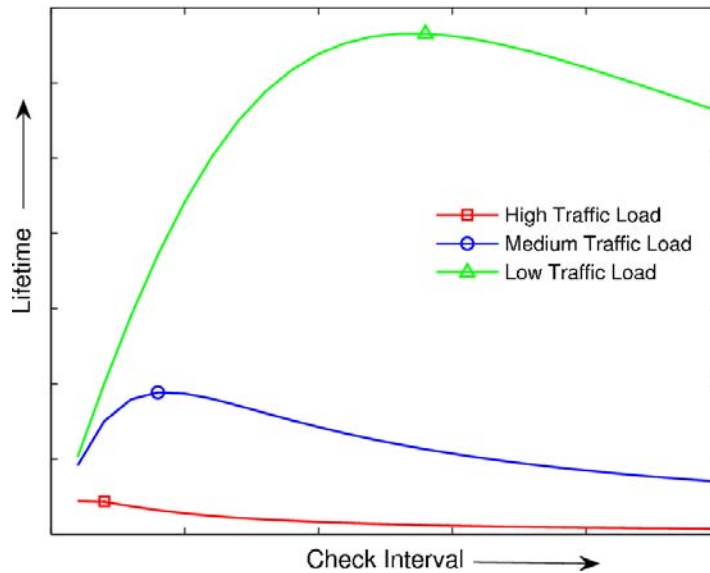


Fig. 10. Lifetime of nodes according to different check intervals and traffic loads.

Reducing the duty cycle involves extending the check interval. On one side, this saves energy because the receiver will sleep for longer periods. On the other side, this drains more energy because the transmitter will use larger preambles.

5.3 Principal Protocols

Subsequently, we will discuss some canonical protocols which aim at improving on some of the above listed short comings. For convenience, they are again summarized in Table 4 at the end of this section.

5.3.1 Improving CCA

CCA (Clear Channel Assessment) based on thresholding is a poor technique for estimating the channel status. It may result in false positives, which reduces throughput. False positives may induce the receiver to keep receiving a noise taken for a preamble.

Clear Channel Assessment (CCA) is the operation that determines whether the channel is clear. In CSMA (Carrier Sense Multiple Access), a node performs a CCA before transmitting a frame to avoid causing a collision in case it transmits while the channel is busy. A common method used to performing a CCA is thresholding. As used in the IEEE 802.15.4, thresholding consists in measuring the power of a received signal and comparing it to the noise floor. The channel is considered clear only if the measured signal is below the noise floor. Thresholding generates a large number of false positive because of the significant variance in channel energy [73]. False positives lower the effective channel bandwidth, thus they should be reduced.

Instead of thresholding, the **BMAC** (Berkeley MAC) [73] protocol proposes a technique based on outlier detection to improve the quality of CCA. In this technique, a node searches for outliers in the received signal such that the channel energy is significantly below the noise floor. If the node detects an outlier during channel sampling, then it declares the channel is clear because a valid signal has outliers significantly below the noise floor with low probability only. If the node does not find any outlier within five samples, then it declares the channel to be busy. Outlier detection substantially outperforms thresholding as reported in [73].

The outlier detection technique depends upon the accuracy of the noise floor estimation. BMAC uses automatic gain control for estimating the noise floor to adapt to ambient noise changes. Each node takes signal strength samples at times when the channel is supposed to be clear, such as immediately after transmitting a frame. From these values, each node calculates an average value and uses it as a simple low pass filter for the noise floor estimate.

Apart from collision avoidance and good channel utilization, accurate CCA has additional benefits. It makes it possible for a node, listening to the preamble while waiting to receive the data frame, to determine whether the channel is still busy. In the case the node detects that the channel is back to idle before it receives the data, it stops listening and goes back sleeping. By avoiding this reception, an accurate CCA further improves preamble-sampling performance.

5.3.2 Adapting Duty Cycle

Determining the optimal check interval in preamble sampling protocols requires knowing applications' traffic load *a priori* because nodes have no means for adapting their check interval to traffic load changes. This constraint makes preamble sampling inflexible for applications with highly fluctuating traffic loads. BMAC proposes to alleviate such rigidity through the use of a versatile low power listening in which each node has an interface for dynamically configuring several MAC layer parameters, such as the check interval. BMAC proposes eight standard listening modes corresponding to eight different check intervals. A node can dynamically switch from one listening mode to another to meet applications' new and changing demands.

EA-ALPL (Energy-Aware Adaptive Low Power Listening) [75] exploits BMAC's reconfiguration interfaces to adapt to traffic changes; that is, each node sets its listening mode according to its current and past forwarding loads. EA-ALPL also makes use of these different listening modes to influence routing decisions. For example, an overused node that does not want to forward other nodes' traffic anymore, voluntarily increases its listening mode to encourage its neighbors to choose another node to continue relaying their traffic. In energy-efficient routing, a node selects the next hop with the minimum check interval because transmitting to that node consumes less energy as it requires a shorter preamble. **SEESAW** [76] is another protocol that uses the similar idea. It allows a node to change its duty cycle according to its traffic load.

5.3.3 Using Two Separate Channels

The **STEM** (Sparse Topology and Energy Management) protocol [41] makes use of two channels: a wakeup channel and a data channel. The wakeup channel is used to organize a meeting between the transmitter and the receiver to avoid deafness, whereas the data channel is only used for data exchange once the meeting occurs. To ensure a meeting between the transmitter and the receiver, nodes follow a preamble sampling approach: the receiver periodically samples the wakeup channel and the transmitter sends preambles on the wakeup channel before sending the data on the data channel.

STEM has two preamble variants: **STEM-T** (STEM Tone) and **STEM-B** (STEM Beacon). In STEM-T, the preamble consists of a simple busy tone. Thus STEM-T is very similar to traditional preamble sampling protocols except for using two separate channels instead of only one. In STEM-B, the preamble consists of a series of beacons, each beacon carrying the MAC addresses of the transmitter and of the receiver. The node that wakes up to sample the channel expects to receive a beacon. When the node receives a beacon, it learns from the MAC addresses carried therein whether it is the destination of the data frame. If so, the node sends an acknowledgment back to the sender (note that beacons are not transmitted contiguously), inter-beacon blanks being intentionally inserted to let the receiver send its acknowledgment. When the transmitter receives an acknowledgment, it stops transmitting beacons and switches to the data channel to send the data frame. After sending the acknowledgment, the receiver also switches to the data channel to receive data.

STEM-B has the advantage of cutting back the preamble length, as transmitters do not require always using a full-length preamble. However, STEM-T uses a simpler transceiver on the wakeup channel, which can be significantly cheaper and less energy consuming than a transceiver used for data communication.

5.3.4 Reducing Preamble Length by Packetization

Many protocols such as **CSMA-MPS** (CSMA with Minimum Preamble Sampling) [77], **TICER** (Transmitted Initiated Cycled Receiver) [72], **X-MAC** [78], and **MH-MAC** (Multimod-Hybrid MAC) [79] use techniques similar to STEM-B, but with a single channel: beacons are transmitted on the same data channel. These protocols using preambles split into packets with a gap between consecutive packets have the advantage of not always requiring the full-length preamble. In the case of unicast transmissions, the receiver sends the ACK in the gap between the frames, thus stopping the preamble transmission. However, in very lightly loaded networks, these protocols do not guarantee optimal energy savings, because they increase idle listening at the receivers. When there is a gap between frames, nodes should stay in receive mode for a duration larger than the gap to sample the channel. Therefore, the sampling duration increases and thus nodes waste more energy in sampling.

Another similar class of protocols replaces the preamble by actual data packets, examples of which include **WOR** (Wake On Radio) [16], **SpeckMAC-D** [106] and **MX-MAC** [107], and terminate transmission once successful reception is acknowledged.

DPS-MAC (Dual Preamble Sampling MAC) [80] improves STEM-B-inspired protocols by reducing the sampling duration of nodes. As shown in Fig. 11, a node that wakes up to sample the channel does not need to be awoken for a duration larger than the inter-preamble packet gap. In DPS-MAC, the sampling duration is the minimum of time needed to check whether a preamble is being transmitted (i.e. RSSI check). If the node wakes up and finds that the channel is free for the first wakeup, then it needs to wake up again to ensure that the channel is really clear, because the first wakeup might have happened during the gap period. The duration between the first and the second wakeup is $[t_{P1}, t_{P2}]$ and the duration of the second

channel sampling t_{p2} can be found in [80] DPS-MAC further reduces idle listening of STEM-B-inspired protocols, which results in further energy savings, especially in lightly loaded networks. However, the efficiency of the method depends on switching times of the radio being used.

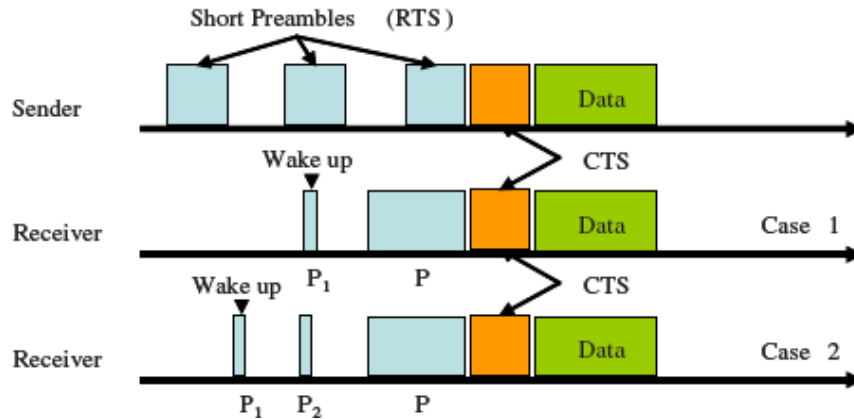


Fig. 11. Principle of DPS-MAC [80].

CMAC (Convergent MAC) [81] uses a similar technique as X-MAC, which will be discussed below. Its preamble is composed of a burst of RTS packets. The gap between two consecutive RTS packets is large enough to contain 3 CTS slots. These CTS slots are sorted to let the best next-hop node reply first to the transmitter. The best next-hop node is the geographically furthest one from the transmitter as described by **GeRAF** (Geographic Random Forwarding) [82]. Note that this idea is similar to that introduced before for the 1-Hop MAC [83]. As there is a possibility to have two best next-hop nodes (nodes from the same region), a collision may occur. CMAC use a contention window within each CTS slot to alleviate collisions. The gaps introduced imply that potential receivers have elaborated channel samplings to avoid missing transmission (this happens when a node samples the channel during a gap, i.e. CTS slots). CMAC makes each node sample the channel five times, each duty cycle period, to make sure that it does not miss a RTS/CTS transmission. Although the idea is interesting as it is a sort of cross link-routing protocol, it has a non-negligible overhead resulting from sampling the channel many times to check whether there is an active transmission on the channel. The authors of CMAC are using XSM radios [84] that perform rapid channel check. CMAC has another phase in addition to the anycast phase described above. After receiving a data packet, the receiver keeps its radio on waiting for potential additional data; this phase is called the convergence phase. CMAC converges from unsynchronized anycast to synchronized unicast. In the synchronized unicast, the sender and the receiver synchronize to exchange information. Once, they are no longer exchanging information, they go back to the unsynchronized anycast.

The **RICER** (Receiver Initiated Cycled Receiver) [72] shifts communication initiation from the transmitter side to the receiver side. When the receiver wants to receive a frame, it sends a beacon to announce that it is awoken. Right after beacon transmission, the receiver monitors the channel for a certain time waiting for a response from the sender. If the receiver gets a response, it transmits the data just afterwards, otherwise, it goes back to sleep. To send a data frame, the transmitter stays awake and monitors the channel waiting for a beacon from the receiver. Once the transmitter receives the beacon, it transmits the data frame and waits for an ACK to end the session. RICER achieves high energy savings for unicast and anycast communications. However, it cannot be used for broadcast and multicast communications,

because it is receiver-initiated. The idea of RICER is similar to preamble sampling; however, the transmitter keeps receiving instead of transmitting a full-length preamble. The receiver periodically sends frames to announce it is ready to receive frames and monitors the channel thereafter to receive the ACK and the transmission. This overhead is large in lightly loaded networks as the receiver does this periodically.

5.3.5 Reducing Preamble Length by Piggybacking Synchronization Information

Large preambles decrease the performance of preamble sampling protocols because nodes drain significant energy in transmission. **WiseMAC** (Wireless Sensor MAC) [85] alleviates this drawback by making it possible for nodes to use short preambles for some unicast transmissions.

In preamble sampling, a node that wants to transmit a data frame uses a preamble that is as long as the check interval hereafter referred to as full-length preamble. The transmitting node uses a full-length preamble because it does not know when the receiver wakes up. To save the transmitter the overhead of using a full-length preamble, WiseMAC aims at letting each node learn about its neighbors' wakeup times; if the transmitter knows the wakeup time of the receiver, then it can timely start its transmission just to meet the receiver wakeup. Clock drifts may make the transmitter lose accuracy about the receiver's wakeup time. In such a case, the transmitter uses a preamble that is just long enough to make up for the estimated maximum clock drift. The length of the preamble used in this case depends on clock drifts: the smaller the clock drift, the shorter the preamble the transmitter has to use. Fig. 12 shows an example of short preamble utilization.

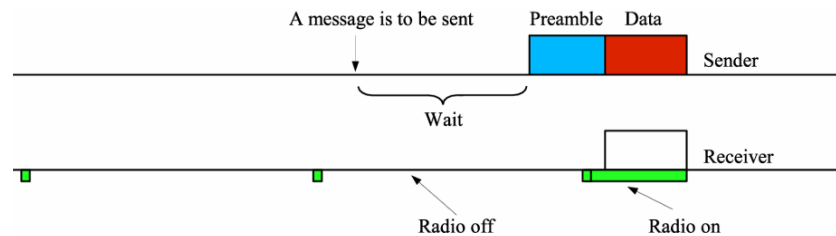


Fig. 12. WiseMAC improves on preamble sampling energy savings through the use of short preambles. In WiseMAC, each transmitter knowing the wakeup time of the receiver sends its transmission just on time to meet the receiver wakeup. The transmitter may use a short preamble to cover clock drifts about the receiver's wakeup time [85].

Each node running WiseMAC makes use of an internal table to store its neighbors' wakeup times. To keep maintenance and construction of such tables low-cost, nodes adopt a passive approach: each node declares its wakeup time by piggybacking it on the ACK frames used to acknowledge a successful reception. When a node receives an ACK frame, it updates its table with the wakeup time of the node that transmitted the ACK. Note that a node may have no information about the wakeup time of a neighbor to which it wants to transmit a frame; in that case, the node uses a full-length preamble.

Many other protocols use a similar idea to WiseMAC to shorten the length of the transmitted preamble in the case of unicast packets. **RATE EST** (Rate Estimation MAC) [86] is similar to WiseMAC except that it is used for nodes with two radios (thus similar to STEM-T too). **SP** (Sensornet Protocol) [15] also features this preamble length optimization. **SyncWUF** (Synchronized Wake Up frame) [87] combines it with the Packet Preamble technique for further energy savings.

5.3.6 Avoiding Unnecessary Receptions

MFP (Micro Frame Preamble) [88] identifies two kinds of unneeded receptions: one is general and concerns the reception of redundant frames in the case of a network-wide broadcast, and the other is specific to sampling protocols and concerns the reception of the remainder of the preamble until the data. To avoid these unneeded receptions, MFP replaces the continuous long preamble (tone) by a series of small packets called microframes. Each microframe contains an indicator on the data frames contents such as destination address or a digest of the data field. In addition, each microframe contains a sequence number that indicates the number of microframes to be transmitted before the data packet. A similar idea has been proposed in [89].

When a node wakes up to perform sampling and receives a microframe, it learns from the sequence number when the forthcoming data will arrive. After the reception of a microframe the nodes goes back sleeping to avoid receiving the subsequent microframes. As the node knows when the data frame arrives, it wakes up again to receive the data frame in the case it considers that the data frame is worth receiving. Data frames whose reception is not required are unicast frames destined to another node or broadcast frames that have already been received. Redundant data frames appear when a network-wide broadcast is executed. For example, when nodes flood a packet, a node may receive multiple copies of the same packet resulting from the systematic forwarding of the broadcast packet by its neighbors.

The information carried in a microframe allows a node to identify those unneeded receptions. The destination address allows a node to filter out unicast packets destined to other nodes and the digest field allows it to filter out redundant broadcast. The authors in [64] show that the digest field may be a hash of the data payload or a unique sequence number.

Reducing unneeded receptions, especially the remainder of the preamble, makes the energy savings by MFP substantial compared to basic preamble sampling protocols. This idea can also be used to further enhance all improvements on preamble-sampling protocols, such as proposed for the **1-hopMAC** [83].

Table 4: Summary of MACs and their prime roles for preamble sampling protocols.

Function	Protocols
Canonical Solutions	Preamble-Sampling ALOHA [70], Preamble-Sampling CSMA [71], Cycled Receiver [72], LPL [73], Channel Polling [74]
Improving CCA	BMAC [73]
Adaptive Duty Cycle	EA-ALPL [75]
Reducing Preamble Length by Packetization	CSMA-MPS [77], TICER [72], WOR [16], X-MAC [78], MH-MAC [79], DPS-MAC [80], CMAC [81], GeRAF [82], 1-hopMAC [83], RICER [72], SpeckMAC-D [106], MX-MAC [107]
Reducing Preamble Length by Piggybacking Synchronization Information	WiseMAC [85], RATE EST [86], SP [15], SyncWUF [87]
Using Separate Channels	STEM [41]
Avoiding Unnecessary Receptions	MFP [88], 1-hopMAC [83]

6 Hybrid Protocols

6.1 Basic Idea

Protocols of this category combine above discussed protocols to take advantage of their characteristics to achieve high performance under variable traffic patterns. When a small number of nodes transmit, contention-based approaches yield sufficient performance; however, when a large number of nodes transmit, then scheduled protocols are a better choice. The main protocols using this hybrid technique are presented below.

6.2 Principal Protocols

We now discuss key hybrid protocols, which are summarized in Table 5 at the end of this section for convenience.

6.2.1 Flexible MAC Frame Structure

The MAC proposed by the IEEE 802.15.4 [5] is very flexible by allowing to switch between many different modes. It supports two device classes, i.e. 1) the full function device (FFD) which can form any type of topology and can play the role of a network coordinator; and 2) the reduced function device (RFD) which can only form star topologies by connecting to the network coordinator.

Since devices are expected to be very heterogeneous in requirements, complexity and power availability, a flexible MAC has been designed which allows for two basic communication modes, i.e. 1) beacon-enabled; and 2) non-beacon modes. Since the latter is essentially reduced to a CSMA/CA MAC, it is of interest in networks composed of very low complexity nodes which are only able to perform simple sensing. The former is used in networks where FFDs are able to send beacons and hence maintain a slot structure, similar to the scheduled protocols discussed before.

Most of the pertinent features of IEEE 802.15.4 are in the beacon-enabled mode and are hence subsequently discussed. With reference to Fig. 13, the beacon-enabled mode uses a superframe structure where the MAC frame is split in an active and an inactive period; tuning the length of each of them allows to dynamically adjust the duty cycle. Each active period is preceded by a beacon frame which is emitted by the coordinator at an interval BI adjustable to 15ms to 245s. The length of the active period is determined by SD ; the duty cycle can hence be calculated as $2^{SD}/2^{BI}$. The active period is further split into 16 time slots of duration BP . The active period consists of 1) the beacon; 2) a Contention Access Period (CAP); and 3) a Collision Free Period (CFP). The CFP is only available if Guaranteed Time Slots (GTS) are allocated by the coordinator; each GTS can occupy multiple timeslots among the 16 available, but only 7 GTS are allowed in the CFP.

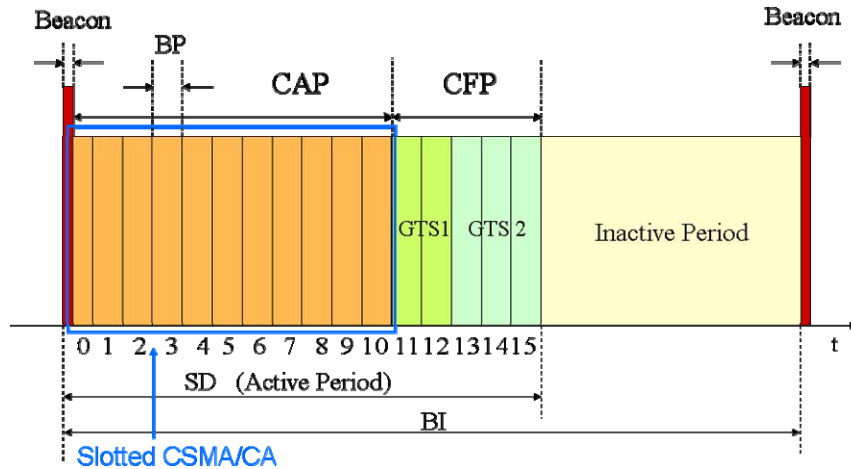


Fig. 13. Superframe structure of the IEEE 802.15.4 MAC [5].

A node therefore listens to the beacon first to understand whether a GTS has been reserved by the coordinator or not. If it has, then it remains powered off until its GTS is scheduled to transmit the data. If no GTS is reserved, then it uses CSMA/CA during CAP where the typical back-off procedures are applied (note that to save energy, the backoff exponent can be limited to 0-2).

As discussed already with the scheduled protocols, the main issue here is the synchronization of the entire network as well as maintaining this synchronization. The IEEE 802.15.4 MAC envisages two methods of synchronization, i.e. 1) the beacon tracking mode; and 2) the non-tracking mode. In the former case, the devices synchronize to the first beacon and then use therein inserted information to switch on just before the next beacon taking clock-drifts into account. In the latter case, synchronization is only done once when some data needs to be transferred; however, this requires that the synchronizing radio is on for half of the superframe in average. The authors of [90] have quantified the tradeoff in energy consumption and data rate for both tracking methods and found that whilst the tracking approach is almost independent of the data rate, the non-tracking mode almost linearly depends on the rate. There is a cross-over point which suggests that below a given data rate the non-tracking mode and above this rate the tracking-mode ought to be used.

6.2.2 CSMA inside Large TDMA Slots for Variable Traffic

ZMAC (Zebra MAC) [91] finds a solution to increase the throughput in networks with variable traffic patterns. ZMAC defines a hybrid method that runs CSMA in low traffic and switches to TDMA in high traffic conditions. At the initial phase, nodes run a distributed slot allocation algorithm DRAND [92] to assign a slot to each node. The algorithm takes care to distribute slots in a way that avoids hidden node collisions which may happen when a node and its two-hop neighborhood share the same time slot. The slots assigned to each node are large enough to allow transmission of multiple packets. Thus, there is no need for strong synchronization. Once the TDMA schedules are established, each node uses the slot assigned to it for transmission. If a node needs more than one slot, it attempts to utilize its neighbors' unused slots. To utilize a non pre-assigned slot, a node starts a random back-off timer at the beginning of that slot. When the random backoff expires and the slot is still unused by its owner, the nodes appropriates the slot and starts transmitting. Note that the random backoff is large enough to let the slot owner have access to its slots before the others.

ZMAC experiences schedule drifts [93] and thus it would be necessary to periodically re-run DRAND to resolve the schedule drift, which reduces its energy efficiency.

6.2.3 Spatial Separation to Mitigate Funneling Effect

Funneling MAC [93] uses TDMA in regions close to the sink and CSMA elsewhere, as shown in Fig. 14. As most traffic patterns in wireless sensor networks are convergecast, nodes in regions close to the sink experience higher traffic loads. Traffic intensity in those regions is high so that more than 80% of packet loss happens in the two-hop neighborhood of the sink when a CSMA-based MAC protocol is used. As collisions reduce both energy efficiency and bandwidth, Funneling MAC proposes to use a hybrid protocol TDMA/CSMA in these regions. Outside these regions in which traffic is less intense, Funneling MAC uses pure CSMA to have better energy/throughput performance.

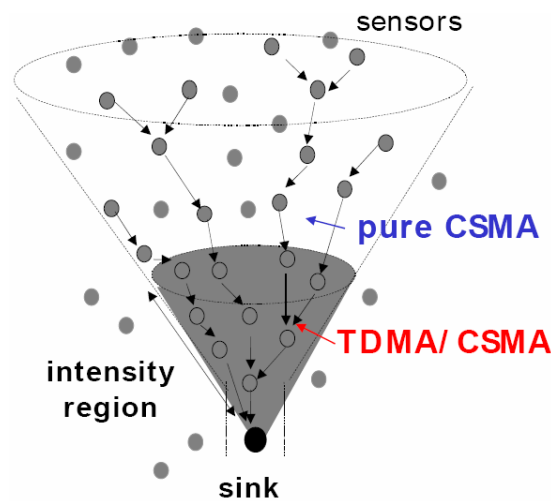


Fig. 14. Funneling MAC [93].

The sink periodically sends on-demand beacons to dynamically drive the depth of the intensity region and to synchronize the nodes inside the intensity region. At the bootstrap of the network, the power used for beacon transmission is the same as that of sensor nodes. Nodes that receive a beacon consider themselves as inside the intensity region and become f-nodes. The f-nodes that are at the boundary of the intensity region become path-heads. An f-node becomes a path-head in two cases: 1) when it receives a data packet from another node outside the intensity region; or 2) when it originates a data packet. Path-heads send their identities to the sink so that it knows their number and how many hops away they are from it. This information is sent according to passive registration to avoid exchanging extra packets. The passive registration uses a dedicated field (path information field) in the MAC header containing the path-head ID and the number of hops.

When the node receives the identities of path-heads and their relative depth, it runs a slot distribution algorithm. The sink node sends a schedule announcement packet in which it includes the path-heads and the number of slots assigned to each path-head. As the schedule is broadcast, all intermediate f-nodes can figure out their schedules.

The sink determines the optimal depth [94] of the intensity region to maximize throughput and minimize packet loss. To construct an intensity region with an optimal depth, the sink uses a depth-tuning algorithm in which it varies the transmission power according to the obtained depth. If the sink realizes that the obtained depth is smaller than the optimal depth, it

increases the transmission power of the beacons. However, if it finds that the obtained depth is greater than the optimal depth, then it reduces the transmission power of beacons.

Funneling MAC envisages CSMA periods for many reasons [93]. For example, CSMA slots can be used by nodes that are inside the intensity region but lose synchronization or by those nodes that are new in the intensity region due to mobility. These nodes do not have allocated TDMA slots so CSMA periods allow them to communicate and perform a registration at the sink. Intermediate nodes should listen to all CSMA periods. Listening to TDMA and to CSMA slots are both performed according to the preamble sampling technique. Funneling MAC defines a superframe as TDMA slot followed by a CSMA slot. Beacons are sent between several superframes, and schedules announcement packets are transmitted from time to time in the slot following beacon transmission.

As any TDMA schemes, Funneling MAC may suffer from network dynamics such as node mobility which requires the sink to recalculate the path-heads and the traffic on each path. MH-MAC [79] is another protocol similar to Funneling MAC allowing nodes to switch from asynchronous mode (preamble sampling) to synchronous mode (common active/wakeup schedules with contention based access) to full-on mode to adapt to varying traffic conditions.

6.2.4 Hybrid Slotted and Sampling to Reduce Preamble Cost

SCP (Scheduled Channel Polling) [74] synchronizes neighboring nodes on a common schedule like SMAC. However, it avoids idle listening within the active periods through the use of sampling techniques. When a node wakes up during the common active schedule and does not find a preamble, it goes back to sleep like in sampling protocols.

SCP reduces idle listening and it also adapts to traffic changes. The basic idea of adaptability to traffic is to detect bursty traffic. When the traffic is high additional wakeups are added in the interval between two regular wakeups as shown in Fig. 15.

As nodes have synchronized wakeups (wake up at the same time), and each node uses a short preamble during the wakeup, a strong contention is expected. To alleviate this contention and to reduce collisions, SCP uses a two-phase contention. It defines two contention windows CW1 and CW2. In CW1, nodes contend to send a wakeup tone: each node that wants to transmit a packet randomly selects a slot in CW1; if the channel is idle, then it sends a preamble to wake up the receiver; the preamble should have started before the sampling period so that it meets the receiver. In the second phase, only nodes that have succeeded in transmitting the preamble compete for the second contention for actual data transmission. In the second contention window CW2, nodes randomly select a slot in CW2 and then perform a carrier sense; if they find the channel idle then they transmit the data. The rationale behind using two contention periods CW1 and CW2 instead of using one of similar length (CW1 + CW2) is that the probability of collision in the former case is lower than that of the latter. If we assume that the length of each one of the contention windows is $m/2$, then the probability of having a collision in the latter case is $1/m$. However, the probability of collision in the latter case is $1/(m/2) * 1/(m/2)$, which is equal to $4/m^2$. Therefore, the second case is more advantageous when $m > 4$. In the SCP implementation, the authors use 8 slots for the contention for preamble transmission and 16 slots for the transmission of data. This is to stress on reducing collisions during data transmission more than during preamble transmission while having low global collision probability.

To reduce synchronization cost, SCP piggybacks synchronization information in exchanged data packets. In the case data rates are lower than synchronization periods, explicit synchronization packets are sent.

By synchronizing all nodes on a common schedule, SCP allows efficient broadcast communications compared to sampling protocols. However, the adaptive channel polling option may increase idle listening in case the information transmitted can fit in a single packet, because a node that receives a packet during its regular channel polling systematically wakes up in the n subsequent adaptive slots as shown in Fig. 15.

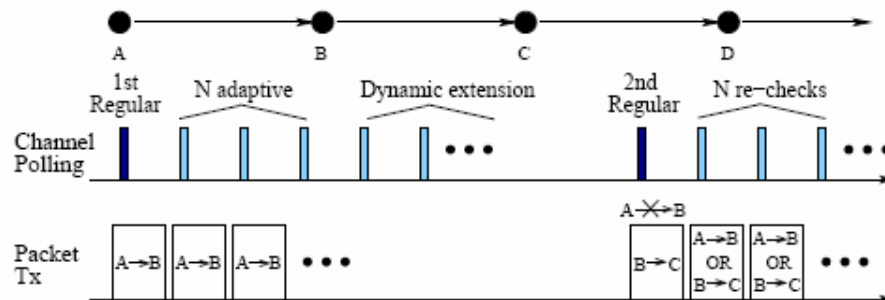


Fig. 15. SCP Adaptive Slots [74].

6.2.5 Receiver Based Scheduling to Avoid Overhearing

Crankshaft [95] decomposes time into frames. Each frame contains slots. There are slots dedicated for unicast traffic at the beginning of the frame and others destined to broadcast traffic at the end of the frame. The assignment of slots is receiver-based, i.e. each node has its own slot during which it wakes up to receive potential data. In broadcast slots, all nodes wake up. Each node owns a slot according to its MAC address; if there are n unicast slots, the node owns slot i that is the result of its MAC address modulo n . It is tolerated that two neighbors share the same slot. Although there are more elaborated collision-free slot assignment methods, this way of assigning slots aims at reducing complexity. As many nodes may want to transmit a frame to a node at the same time, Crankshaft uses a contention mechanism prior to transmission and each transmission is followed by an acknowledgement packet. The duration of a slot should be large enough to contain contention, data, and acknowledgment. Contention is similar to WiseMAC and SCP, transmitters start at random moments transmitting a preamble in the contention period. The transmitter that starts before the others wins the contention because when they want to transmit they perform a carrier sense before transmitting. When they find the channel busy, they stop the contention. The contention is performed according to the Sift technique to reduce collisions. In the Sift technique (described above), starting transmission are shifted to the end of the contention window, which reduces the amount of the transmitted preamble and thus increases energy saving. The maintenance of synchronization is achieved according to in-band signaling in which each transmitted packet contains information about synchronization in its header.

Although Crankshaft reduces idle listening, it has some overhead resulting from sampling all the broadcast slots, which increases the energy drained. Another drawback is the use of TDMA, which decreases the flexibility of the protocol especially with sporadic traffic between some nodes.

6.2.6 Optimum Protocol Switching Point

As has become clear in prior exposure, different protocol families perform better under different traffic conditions. Notably, whilst preamble sampling based MAC protocols perform well under very light loads, protocols with common active periods are suitable to medium

in many useless wakeups. Furthermore, the wakeup and the data radios are in different channels, thus a transmitter node might not be able to wake up a node with which it wants to communicate. Finally, the wakeup radio may increase the overall cost of a sensor node.

7.2 Packet-Based vs Bit-Based Radio

This is a question of versatility (flexibility of programming) versus performance; it is essentially also the question of narrowband versus wideband radios.

Narrowband radios are more flexible as the microcontroller controls most of the radio functionalities: every bit transmitted over the air interface can be controlled by the MCU (including the physical preamble and CRC). These radios have also very small switching times thus they offer large energy savings as switching times cause major energy waste in optimized protocols. However, the main issue of these radios is the high bit error rate as they use simple modulation schemes, and no spreading codes.

Wideband radios are robust to noise but this is at the cost of a less flexibility. Modulation schemes (such as DSSS) are more robust to noise but consume more power. These radios are packet-based and the transmission of a packet is done by hardware because the microcontroller cannot afford the overhead of these operations. These radios have dedicated circuits for synchronization, channel coding and even security. In these radios, the microcontroller has no control on each single transmitted bit over the air, it sends data to the radio that transforms it into a packet and sends it to the air interface. Major operation performed by these radios may include coding, physical preamble transmission, encryption, CRC, address filtering, automatic ACK transmission. Some radios such as the CC2500 [16] are also able to perform a periodic channel sampling without the need for the MCU. For some specific radios, it is not possible to control the transmitted packet length, e.g., the maximum frame length that could be transmitted by an IEEE 802.15.4 radio is 127 bytes. Implementing some protocols with such radios may be challenging (see [74] for implementing SCP on CC2420 radio [97]). These radios have higher bit rates thus their circuits are more complex, and additionally require longer switching times.

In conclusion, selecting the most suitable radio depends on the parameters required by the application, which includes: robustness to interference, switching times, flexibility, bandwidth, power consumption of the different modes, programmability, cost, etc.

7.3 Frequency-Agile MAC Protocols

Current low power radio chips enable the center frequency to be programmed within a range of several tens of MHz. As an example, on a CC2500 radio chip the 250 kBaud MSK modulation occupies 296kHz of bandwidth [16]; adjacent carrier frequencies can thus be safely positioned as close as 500MHz apart. With the operating frequency range of 2400-2483.5MHz, there is potentially room for over 150 non-overlapping channels.

In single-channel MAC protocols, all nodes are configured to use a single frequency all the time. Frequency-agile MAC protocols switch between multiple frequencies during run-time. Recent radio chips are able to switch between frequency channels fast (e.g. in less than 100 μ s). We envision that multi-channel hardware support will have a significant impact on WSN MAC protocol design, and that it will become a standard means of decreasing collisions, increasing throughput and improving reliability.

Multi-channel hardware support largely impacts MAC layer design. It increases the network's throughput by reducing contention; additional channels can be introduced when a common

base channel becomes too crowded [44], [47], and [48]. Multi-channel can also be used to increase robustness against narrowband long-lasting [48] and transient interference [22]. Whereas most current multi-channel solutions assume nodes are synchronized and time is split into time slots [22], [44], [46], and [47], this assumption is broken in recent solutions [48]. Proof-of-concept experiments [48] and [47] and commercial products [22] show the great potential of multi-channel MAC protocols for WSNs.

7.4 Memory Size and Usage

The size of RAM should not be too large because it increases current consumption and the status of the RAM should be maintained to prevent losing the stored data. Instead, additional circuitry should be developed to reduce code size in RAMs. This solution also trades performances for flexibility.

Typical sensors have in the order of 4kB of available RAM (examples include the MICA family from Crossbow). Out of these 4kB, typical TinyOS applications require 2-3kB of RAM; therefore, 1kB can be allocated for packet buffering. Under the assumption that data packets are about 100 bytes long, a typical buffer can hold at most 10 packets which greatly influences MAC performance. As per [96], the lower bound on the packet generation interval is shifted from 8 seconds to 12 seconds when taking this constraint into account.

7.5 Transmit Power Control

Power control has been put forward in various recent contributions to improve the energy efficiency of the wireless sensor network; see, for example, [98], [99], [100], and [101]. Whilst it is a useful tool to control traffic flows, congestion and interference levels, the power savings due reduced transmission powers are negligible since the radio's power consumption is in the range of the transmit power levels typically employed for embedded WSN nodes.

8 Conclusions

As has become apparent, a large amount of work has been produced over the past two decades on wireless sensor networks. Since these networks operate wirelessly, the medium access control is of pivotal importance. We have classified key MACs which were put forward in past years, which we have summarized for convenience in Table 6. The time stamps on the publications indicate clearly that early realizations of WSN MACs relied on scheduled MACs, often assuming a one-hop link between sensors and sink; thereafter, protocol families with common active periods appeared since traffic load conditions relaxed; lately, preamble sampled MACs appeared which are very appealing for light traffic conditions in multihop networks. Whilst the state-of-the-art is seemingly complete, we believe that numerous issues are still wide open and summarize these below.

First, a complete system-wide quantification of throughput, delay, energy consumption, etc., taking the multi-hop nature and different applications into account has not been developed to date. First attempts have been undertaken in [102] for a specific MAC family, but outcomes are far from final nor sufficiently general.

Furthermore, no MAC as of today is proven to be highly scalable as well as facilitate network ramp-up and auto-organization/configuration/healing. This is particularly of importance, when sensor nodes arrive in a box of several thousands of nodes and are being switched on for deployment. Since this large quantity of nodes are within their one-hop radio neighborhood, any MAC described above will experience serious operational problems.

There is also a strong need for security in WSNs, which usually is not taken care of in MAC mechanisms. However, the fact that WSNs are operating at their energy limit, the MAC can be of help and hinder communication with nodes which are deemed to be malicious. It should be able to detect attempts to make a node uselessly enter the state of idle listening, overhearing, etc. It should also be able to avoid or minimize jamming-style DoS attacks, unfairness selfish behavior, and packet injection aiming at exhausting nodes' energy.

On the deployment side, in the future, less nodes will really be equipped with batteries. It is expected that the majority of the WSN nodes will be relying on power harvesting. This has a profound impact on the MAC design, including its schedules. For instance, if power can be harvested every 24h only, then the MAC protocol needs accordingly be adapted to provide a high activity level during the time the node is energized.

Table 6: Summary of MACs belonging to the discussed MAC families.

MAC Family	Protocols
Scheduled Protocols	TSMP [22], IEEE 802.15.4 [5], Arisha [24], PEDAMACS [25], BitMAC [26], G-MAC [27], SMACS [28], TRAMA [31], FLAMA [32], μ MAC [33], EMACs [34], PMAC [36], PACT [29], BMA [30], MMAC [35], FlexiMAC [37], PMAC [38], O-MAC [38], PicoRadio [40], Wavenis [3], f-MAC [43], Multichannel LMAC [44], MMSN [46], Y-MAC [47], Practical Multichannel MAC [48], LMAC [45], AI-LMAC [49], SS-TDMA [50], RMAC [51]
Protocols with Common Active Period	SMAC [52], TMAC [54], E2MAC [55], SWMAC [56], Adaptive Listening [57], nanoMAC [58], DSMAC [59], FPA [60], DMAC [61], Q-MAC [62], MSMAC [63], GSA [60], RL-MAC [66], U-MAC [67], RMAC [68], E2RMAC [69]
Preamble Sampling Protocols	Preamble-Sampling ALOHA [70], Preamble-Sampling CSMA [71], Cycled Receiver [72], LPL [73], Channel Polling [74], BMAC [73], EA-ALPL [75], CSMA-MPS [77], TICER [72], WOR [16], X-MAC [78], MH-MAC [79], DPS-MAC [80], CMAC [81], GeRAF [82], 1-hopMAC [83], RICER [72], WiseMAC [85], RATE EST [86], SP [15], SyncWUF [87], STEM [41], MFP [88], 1-hopMAC [83]], SpeckMAC-D [106], MX-MAC [107]
Hybrid Protocols	IEEE 802.15.4 [5], ZMAC [91], Funneling MAC [93], MH-MAC [79], SCP [74], Crankshaft [95]

References

- [1] Christophe Dugas. Configuring and managing a large-scale monitoring network solving real world challenges for ultra-low powered and long-range wireless mesh networks. *International Journal of Network Management*, 15:269–282, 2005.
- [2] M. Dohler et al. The ARESA Project: Facilitating Research, Development and Commercialization of WSNs. In *Proceedings of IEEE SECON*, San Diego, CA, June 2007.
- [3] <http://www.wavenis-osa.org>.
- [4] <http://www.knx.org/>.
- [5] IEEE 802.15.4. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). 2003.
- [6] <http://www.ietf.org/html.charters/6lowpan-charter.html>.
- [7] <http://tools.ietf.org/wg/roll/>.
- [8] H. Wu and Y. Pan. *Medium Access Control in Wireless Networks*. Nova Science Publishers, 2008.
- [9] K. Langendoen. Energy-Efficient Medium Access Control. Book chapter in *Medium Access Control in Wireless Networks*, H. Wu and Y. Pan (editors), Nova Science Publishers, 2008.
- [10] S. Plancoulaine, A. Bachir, and D. Barthel. WSN Node Energy Dissipation. France Telecom R&D, internal report, July 2006.
- [11] 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):1400–16, December 1975.
- [12] H. Takagi and L. Kleinrock. Throughput Analysis for Persistent CSMA Systems. *IEEE Transactions On Communications*, 33(7):627–38, July 1985.
- [13] C. Enz, A. El-Hoiydi, J. Decotignie, V. Peiris. WiseNET: An Ultralow-Power Wireless Sensor Network Solution. *IEEE Computer*, 37(8):62–70, August 2004.
- [14] A. Woo and D. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proceedings of ACM Mobicom*, Rome, Italy, July 2001.
- [15] J. Polastre. A Unifying Link Abstraction for Wireless Sensor Networks. PhD thesis, University of California, Berkeley, October 2005.
- [16] Chipcon Corporation. CC2500 Single Chip Low Cost Low Power RF Transceiver, Data Sheet. 2005.
- [17] P. Karn. MACA- a new channel access method for packet radio. in *ARRL/CRRL Amateur Radio 9th Computer Networking*, pages 134–40, 1990.
- [18] G. Brandner C. Bettstetter and R. Vilzmann. On Colliding First Messages in Slotted ALOHA. In *Proceedings of IEEE PIMRC*, Cannes, France, September 2008.
- [19] Y. C. Tay, Kyle Jamieson, Hari Balakrishnan. Collision-Minimizing CSMA and its Applications to Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 22(6):5–26, 2004.
- [20] S. Singh, C. S. Raghavendra. PAMAS—power aware multi-access protocol with signaling for ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 28(3):1048–57, 1998.

- [21] IEEE 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 1999.
- [22] Dust Networks. Technical Overview of Time Synchronized Mesh Protocol, White Paper, <http://www.dustnetworks.com/>. 2008.
- [23] L. Bao and J.J. Garcia-Luna-Aceves. Collision-Free Topology-Dependent Channel Access Scheduling. In proceedings of IEEE MILCOM, Los Angeles, CA, October 2000.
- [24] K. Arisha, M. Youssef and M. Younis. Energy-Aware TDMA-Based MAC for Sensor Networks. In proceedings of IEEE IMPACCT, New York City, NY, May 2002.
- [25] S. Coleri-Ergen and P. Varaiya. PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks. IEEE Transactions on Mobile Computing, 5(7):920–30, 2006.
- [26] M. Ringwald and K. Roemer. BitMAC: A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks. In proceedings of IEEE EWSN, 2005.
- [27] M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis IV. Wireless Sensor Network Energy-Adaptive MAC Protocol. In proceedings of IEEE CCNC, 2006.
- [28] K. Sohrabi et al. Protocols for Self-Organization of a Wireless Sensor Network. IEEE Personal Communications, 7(5):16–27, 2000.
- [29] G. Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. In proceedings of IEEE MILCOM, 2001.
- [30] J. Li and G. Lazarou. A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks. In proceedings of IEEE IPSN, 2004.
- [31] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. In Proceedings of ACM Sensys, pages 181–92, Los Angeles, CA, November 2003.
- [32] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka. Energy-Efficient, Application-Aware Medium Access for Sensor Networks. In Proceedings of IEEE MASS, 2005.
- [33] A. Barroso, U. Roedig, and C. Sreenan. uMAC: an energy-efficient medium access control for wireless sensor networks. In Proceedings of IEEE EWSN, 2005.
- [34] L. van Hoesel, P. Havinga. Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs. In Proceedings of IEEE VTC, 2004.
- [35] M. Ali, T. Suleman and Z. Uzmi. MMAC: A Mobility-Adaptive, Collision-Free MAC Protocol for Wireless Sensor Networks. In Proceedings of IEEE IPCCC, 2005.
- [36] P. Cheong and I. Oppermann. An Energy-Efficient Positioning-Enabled MAC Protocol (PMAC) for UWB Sensor Networks. In Proceedings of EURASIP IST, Dresden, Germany 2005.
- [37] W. Lee, A. Datta, and R. Cardell-Oliver. FlexiMAC: A flexible TDMA-based MAC protocol for fault-tolerant and energy-efficient wireless sensor networks. In Proceedings of IEEE ICON, Singapore, September 2006.
- [38] T. Zheng, S. Radhakrishnan and V. Sarangan. PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. In Proceedings of IPDPS, April 2005.
- [39] H. Cao, K. Parker, and A. Arora. O-MAC: A Receiver Centric Power Management Protocol. In Proceedings of IEEE ICNP, Santa Barbara, CA, November 2006.
- [40] J. Rabaey et al. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. IEEE Computer Magazine, 33(7):42–48, 2000.

- [41] C. Schurgers et al. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.
- [42] L. C. Zhong C. Guo and J. Rabaey. Low power distributed MAC for ad hoc sensor radio networks. In *Proceedings of IEEE GLOBECOM*, pages 2944–48, San Antonio, TX, November 2001.
- [43] A. Barroso U. Roedig and C. Sreenan. f-MAC: A Deterministic Media Access Control Protocol Without Time Synchronization. In *Proceedings of EWSN*, 2006.
- [44] S. Dulman O. Incel and P. Jansen. Multi-channel Support for Dense Wireless Sensor Networking. In *Proceedings of EuroSSC*, 2006.
- [45] L. van Hoesel, P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing Preamble Transmissions and Transceiver State Switches. In *Proceedings of International Conference on Networked Sensing Systems*, Tokyo, Japan 2004.
- [46] G. Zhou, et al. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, Barcelona, Spain, April 2006.
- [47] Y. Kim, H. Shin, and H. Cha. DY-MAC: An Energy-efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks. In *Proceedings of IPSN*, pages 53–63, St. Louis, MI, April 2008.
- [48] H. Le, D. Henriksson, and T. Abdelzaher. A Practical Multi-Channel Media Access Control Protocol for Wireless Sensor Networks. In *Proceedings of IPSN*, pages 70–81, St. Louis, MI, April 2008.
- [49] S. Chatterjea, L. van Hoesel, P. Havinga. AI-LMAC: An Adaptive, Information-centric and Lightweight MAC Protocol for Wireless Sensor Networks. In *Proceedings of IEEE ISSNIP*, 2004.
- [50] S. Kulkarni and M. Arumugam. SS-TDMA: A self-stabilizing MAC for sensor networks. *Sensor network operations*, ieee press edition, 2006.
- [51] S. Datta. RMAC: a randomized adaptive access control algorithm for sensor networks. In *Proceedings of SANPA*, Boston, MA, August 2006.
- [52] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, pages 1567–76, New York, NY, July 2002.
- [53] <http://www.isi.edu/nsnam/ns/>.
- [54] T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of ACM Sensys*, pages 171–80, Los Angeles, CA, November 2003.
- [55] P. Havinga and G. Smit. E2MaC: an energy efficient MAC protocol for multimedia traffic. Technical Report, University of Twente, Netherlands, 1998.
- [56] J. Son J. Pak and K. Han. A MAC Protocol Using Separate Wakeup Slots for Sensor Network. In *Proceedings of ICCSA*, pages 1159–68, Glasgow, UK, May 2006.
- [57] W. Ye, J. Heidemann, D. Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. *IEEE Trans. on Networking*, 12(3):493–506, June 2004.
- [58] P. Mahonen J. Haapola J. Ansari, J. Riihijarvi. Implementation and Performance Evaluation of nanoMAC: A Low-Power MAC Solution for High Density Wireless Sensor Networks. In *Proceedings of IEEE ICC*, Istanbul, Turkey, June 2006.

- [59] P. Lin, C. Qiao and X. Wang. Medium Access Control with a Dynamic Duty Cycle for Sensor Networks. In Proceedings of IEEE WCNC, Atlanta, GA, March 2004.
- [60] Y. Li, W. Ye, and J. Heidemann. Energy and Latency Control in Low Duty Cycle MAC Protocols. In Proceedings of IEEE WCNC, New Orleans, LA, March 2005.
- [61] G. Lu, B. Krishnamachari and C. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks. In Proceedings of the WMAN, Santa Fe, NM, April 2004.
- [62] N. Vasanthi and S. Annadurai. Energy Efficient Sleep Schedule for Achieving Minimum Latency in Query based Sensor Networks. In Proceedings of IEEE SUTC, 2006.
- [63] Huan Pham and Sanjay Jha. An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC). In Proceedings of IEEE MASS, pages 588–60, Fort Lauderdale, FL, October 2004.
- [64] A. Bachir, D. Barthel, M. Heusse and A. Duda. Abstract Frames for Reducing Overhearing in Wireless Sensor Networks. In Proceedings of the IFIP Networking, Coimbra, Portugal, May 2006.
- [65] A. Tyrrell and G. Auer. Imposing a reference timing onto firefly synchronization in wireless networks. In Proceedings of IEEE VTC, Dublin, Ireland, April 2007.
- [66] Z. Liu, I. Elhanany. RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks. InderScience International Journal of Sensor Networks, 1(3/4):117–24, 2006.
- [67] S-H. Yang H-W. Tseng, E. Wu, and G-H. Chen. Utilization based duty cycle tuning MAC protocol for wireless sensor networks. In Proceedings of IEEE GLOBECOM, 2005.
- [68] R. Biswas, V. Jain, C. Ghosh, and D. Agrawal. On-Demand Reliable Medium Access in Sensor Networks. In proceedings of IEEE WoWMoM, Buffalo, NY, June 2006.
- [69] V. Jain, R. Biswas and D. Agrawal. Energy Efficient and Reliable Medium Access for Wireless Sensor Networks. In proceedings of IEEE WoWMoM, Helsinki, Finland, June 2007.
- [70] A. El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks. In Proceedings of IEEE ICC, New York, NY, April 2002.
- [71] J. Hill and D. Culler. Mica: a wireless platform for deeply embedded systems. IEEE Micro, 22(6), 2002.
- [72] E-Y. Lin, J. Rabaey, A. Wolisz. Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks. In Proceedings of IEEE ICC, Paris, France, June 2004.
- [73] J. Polastre, J. Hill and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In Proceedings of ACM SenSys, 2004.
- [74] W. Ye, F. Silva and J. Heidemann. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. In Proceedings of ACM SenSys, Boulder, CO, November 2006.
- [75] R. Jurdak, P. Baldi, and C. V. Lopes. Energy-Aware Adaptive Low Power Listening for Sensor Networks. In Proceedings of INSS, pages 24–29, San Diego, CA, June 2005.
- [76] A. Silberstein R. Braynard and C. Ellis. Extending Network Lifetime Using an Automatically Tuned Energy-Aware MAC Protocol. In Proceedings of EWSN, 2006.

- [77] S. Mahlknecht and M. Boeck. CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks. In Proceedings of IEEE Workshop on Factory Communication Systems, Vienna, Austria, September 2004.
- [78] M. Buettner et al. X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Networks. In Proceedings of ACM SenSys, Boulder, CO, November 2006.
- [79] M. Pereira M. Macedo P. Pinto L. Bernardo, R. Oliveira. A Wireless Sensor MAC Protocol for Bursty Data Traffic. In Proceedings of IEEE PIMRC, Athens, Greece, September 2007.
- [80] F. N-Abdesselam H. Wang, X. Zhang and A. Khokhar. DPS-MAC: An Asynchronous MAC Protocol for Wireless Sensor Networks. In Proceedings of HIPC, Goa, India, December 2007.
- [81] S. Liu, K-W. Fan, and P. Sinha. CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks. In Proceedings of IEEE SECON, San Diego, CA, June 2007.
- [82] M. Zorzi and R. Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance. IEEE Transactions on Mobile Computing, 2(4):349–365, 2003.
- [83] T. Watteyne et al., 1-hopMAC: An Energy-Efficient MAC Protocol for Avoiding 1-hop Neighborhood Knowledge. In Proceedings of IWWAN, New York, NY, June 2006.
- [84] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events. In Proceedings of IPSN, pages 302–11, Los Angeles, CA, April 2005.
- [85] A. El-Hoiydi, J-D. Decotignie, C. Enz, E. Le Roux. Poster Abstract: WiseMac, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Networks. In proceedings of ACM SenSys, pages 302–3, Los Angeles, CA, November 2003.
- [86] M. Miller and N. Vaidya. A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio. IEEE Transactions on Mobile Computing, 4(3):228–42, May/June 2005.
- [87] X. Shi, and G. Stromberg. SyncWUF: An Ultra Low-Power MAC Protocol for Wireless Sensor Networks. IEEE Transactions on Mobile Computing, 6(1): 115-25, January 2007.
- [88] A. Bachir, D. Barthel, M. Heusse and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. In Proceedings of IEEE ICC, Istanbul, Turkey, June 2006.
- [89] X. Shi, G. Stromberg, Y. Gsottberger, and T. Sturm. Wake-Up-Frame Scheme for Ultra Low Power Wireless Transceivers. In Proceedings of IEEE Globecom, Dallas, TX, November 2004.
- [90] G. Lu, B. Krishnamachari, and C. Raghavendra. Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks. In Proceedings of IEEE IPCCC, Phoenix, AZ, April 2004.
- [91] I. Rhee et al. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. In Proceedings of ACM SenSys, San Diego, CA, November 2005.
- [92] I. Rhee et al. DRAND: Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks. In Proceedings of ACM MobiHoc, Florence, Italy, May 2006.

- [93] G-S. Ahn, E. Miluzzo, A. Campbell, S. Hong, and F. Cuomo. Funneling-MAC: A Localized, Sink Oriented MAC for Boosting Fidelity in Sensor Networks. In Proceedings of ACM Sensys, pages 293–306, Boulder, Colorado, November 2006.
- [94] A. Campbell S. Hong G-S. Ahn, E. Miluzzo and F. Cuomo. Funneling-MAC: A Localized, Sink Oriented MAC for Boosting Fidelity in Sensor Networks. Columbia University. Technical Report, CU/EE/TAP-TR-2006-08-003, 2006.
- [95] G. Halkes and K. Langendoen. Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks. In Proceedings of EWSN, 2007.
- [96] T. Watteyne et al,. Analyzing MAC protocols for Wireless Sensor Networks. In preparation.
- [97] Chipcon Corporation. CC2420 Single Chip Low Cost Low Power RF Transceiver, Data Sheet. 2005.
- [98] H. Hu, and Z. Yang. The study of power control based cooperative opportunistic routing in wireless sensor networks. In Proceedings of ISPACS, 2007.
- [99] M. Zawodniok and S. Jagannathan. Predictive Congestion Control Protocol for Wireless Sensor Networks. IEEE Transactions on Wireless Communications, 6(11), 2007.
- [100] J. Jeong, D. Culler, and O. Jae-Hyuk. Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks. In Proceedings of INSS, 2007.
- [101] M. Vuran and I. Akyildiz. Cross-Layer Analysis of Error Control in Wireless Sensor Networks. In Proceedings of IEEE SECON, Reston, VA, September 2006.
- [102] T. Karveli K. Voulgaris, I. Ashraf and A. Aghvami. Energy Efficiency of p-Persistent CSMA under Variable Traffic Conditions. In Proceedings of IEEE PIMRC, Cannes, France, September 2008.
- [103] Bhaskar Krishnamachari. Networking Wireless Sensors – Chapter 6. Cambridge University Press, 2006.
- [104] I. Demirkol, C. Ersoy, and F. Alagoz. MAC Protocols for Wireless Sensor Networks: a Survey. IEEE Communications Magazine, 44(4):115–21, April 2006.
- [105] Kurtis Kredo II and Prasant Mohapatra. Medium Access Control in Wireless Sensor Networks. Computer Networks, 51(4):961–94, March 2007.
- [106] K.-J. Wong, D. Arvind, “Speckmac: Low-power decentralized MAC protocol low data rate transmissions in Specknets,” IEEE Int. Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN’06), May 2006.
- [107] C. J. Merlin, W. B. Heinzelman, “Network-aware adaptation of MAC scheduling for Wireless Sensor Networks,” Conf. on Distributed Computing in Sensor Systems (DCOSS’07 Poster Session), June 2007.