Low energy operation in WSNs: A survey of preamble sampling MAC protocols

C. Cano *, B. Bellalta, A. Sfairopoulou, M. Oliver

NeTS Research Group, Department of Information and Communication Technologies, Universitat Pompeu Fabra, C/Tànger 122-140, 08018 Barcelona, Spain

ABSTRACT

The limited energy resources of sensor nodes are among the most important constraints in Wireless Sensor Networks (WSNs). Consequently, the Medium Access Control (MAC) layer design is crucial, due to its influence on the transceiver, which is the most energy-consuming component of a sensor node. Among the different MAC protocols designed for WSNs, preamble sampling techniques provide extremely low energy consumption at low loads and have a notably simple operation and a lack of synchronisation requirements, which are characteristics that are especially appealing to WSNs. In this work, a survey of the different types of MAC protocols designed for WSNs is presented with a special focus on preamble sampling MAC protocols. The aim of this work is to give a detailed overview and classification of the most relevant preamble sampling MAC protocols, being motivated by the extremely large number of MAC protocols designed for WSNs in recent years. Moreover, a simple set of guidelines for matching the most suitable MAC protocol category to a given application is provided in this work.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Wireless Sensor Networks (WSNs) are networks formed by small and low-capability devices that are able to sense environmental metrics and to communicate them wirelessly to a central unit, known as a sink [1]. There is a wide range of potential applications in various areas, including industrial, military, environmental, health and home automation applications, which can profit from using WSNs to collect data. However, the large number of constraints on the sensor nodes and the special characteristics and uses of WSNs impose several challenges on the design of a WSN. Some of these characteristics are summarised next.

The most important characteristic of WSNs is their high application dependence [1]. Different requirements and constraints are directly imposed by the application. This issue complicates the design of a general protocol stack to be used in different deployments and applications and leads to the definition of different approaches. What follows is a review of the general characteristics and constraints of these networks; however, it is important to note that they all depend on the application and deployment scenario.

The deployment of dense WSNs in large, remote and difficult-to-access areas requires keeping the size and cost of the sensor nodes as low as possible. This constraint implies that the energy, computational and memory resources of the sensor nodes are usually limited. Therefore, simplicity and low energy consumption are important requirements in the design of WSN protocols. In addition, the limited energy resources of sensor nodes, combined with variable environmental conditions, cause a high level of network dynamics, which should also be taken into account.

Another interesting feature is the traffic patterns that are usually found in WSNs. In these networks, it is common that all of the sensor nodes periodically transmit
information (the value of the metric of interest) to the central unit, causing a communication pattern notably different from the traditional point-to-point approach. There is also the case of event reporting (for example, when a metric surpasses a given threshold) to the sink. In this instance, only those nodes that detect the event will try to send a notification to the central unit. Finally, the sink can also query the network about certain information. For instance, the sink can ask the network in which location a given metric has surpassed a certain value. In this case, the data flows from the sink to all of the sensor nodes in a data-centric approach, in contrast to the more common communication pattern based on destination addresses.

The scalability of WSNs is also a challenge because deployments can be formed by a large number of sensor nodes. This challenge also causes extra difficulty in some scenarios in which a global identifier should be assigned to each device.

To summarise, the constraints and characteristics of WSNs are the following:

- High application dependence
- Large, remote, dense and difficult-to-access deployments
- Low-cost and small devices
- Devices with limited memory, transmitting and computing resources
- Devices with limited energy resources
- High network dynamics
- Periodic event-based and query-based communication patterns
- Data flowing from all or a group of sensors to the central unit and vice versa

Among the different constraints, limitations on energy resources are the most important because this limitation directly affects the network lifetime. This relationship is particularly important given that, in typical WSN deployments, it is too costly or even impossible to access the sensor nodes to replace the batteries. In such a situation, it is obvious that the WSN protocol stack must be designed to obtain the highest possible energy savings and thus extend the network lifetime, while maintaining the performance of the target application. Therefore, the design of the Medium Access Control (MAC) layer is of crucial importance because it controls the most energy consuming component of a sensor node: the transceiver. Asynchronous MAC protocols and, particularly, preamble sampling are especially appealing to WSNs because of their simplicity and lack of synchronisation requirements.

In this work, a survey of the basic preamble sampling technique and its extensions is provided. There are other general surveys of MAC protocols for WSNs presented in the literature, such as the review performed in 2006 by Dermikol et al. [2], the extensive survey performed in 2007 by Kredo et al. [3], the survey performed by Langeendoen in 2008 [4] and the more recent review presented by Bachir et al. [5]. The main difference between this work and those articles is in the specific focus placed here on the preamble sampling technique and its extensions. That focus allows us to provide a deeper study on this specific and prolific area. Moreover, this detailed study has been extended with a discussion of the benefits and drawbacks of each approach to provide a set of guidelines to help researchers and developers of WSN protocols and applications to select the most suitable MAC category for a given application.

The rest of this paper is organised as follows: Section 2 describes the importance of the MAC layer design in a WSN. Next, in Section 3, a general overview of the different categories of MAC protocols for WSNs is provided. In Section 4, the basic preamble sampling and its extensions are presented. A suggested procedure to select the MAC protocol based on the application is subsequently provided in Section 5. Finally, some conclusions are drawn.

2. The importance of the MAC layer in WSNs

The MAC layer is responsible for coordinating transmissions to a shared channel by defining how and when a node will attempt transmission. The medium access procedure can be based on a schedule assignment (which can be fixed during the entire network lifetime or during a certain amount of time) or, conversely, can be based on a random procedure in which the transmission attempt time is decided independently at each node. It is also possible to define hybrid approaches that combine both techniques.

In WSNs, the MAC protocol design faces several limitations, such as the low computational and synchronisation capabilities of sensor nodes, as well as their low memory capacities [5]. Moreover, in this type of network in which the energy consumption is a major constraint, the MAC layer design becomes an important issue, as it directly controls the transceiver operation, which is the most energy consuming component in a sensor node [1]. Therefore, the MAC layer must focus on the reduction of energy consumption of the sensor nodes. The different sources of energy waste can be classified into the following [3]:

1. Idle Listening: Idle listening occurs when a node listens to the channel for a possible reception, but nothing is received. Idle listening has been identified as the major source of energy waste in a sensor node, mainly due to the low traffic loads commonly found in WSNs.
2. Collisions: When two (or more) nodes simultaneously transmit, the recipient may be unable to decode any of the packets involved. This problem implies that the senders waste energy through transmitting and that the receiver expends energy receiving without obtaining any benefit, as senders may eventually retry transmission. In the presence of hidden terminals, this effect is even more important because transmitters cannot sense each other.
3. Overhearing: This source of energy dissipation occurs when a sensor node receives energy receiving a packet that is intended for a different destination.
4. Overhead: Data packets in WSNs are usually small; therefore, headers and other types of overhead (such as control messages) imply a high level of energy waste for WSNs.

Please cite this article in press as: C. Cano et al., Low energy operation in WSNs: A survey of preamble sampling MAC protocols, Comput. Netw. (2011), doi:10.1016/j.comnet.2011.06.022
MAC protocols designed for Wireless Local Area Networks (WLANs) or mesh networks, such as the well-known Distributed Coordination Function (DCF) defined in the IEEE 802.11 standard [6], are not suitable for the special requirements of WSNs for several reasons. One of the reasons is that they do not consider the different constraints of WSNs. Usually, these protocols assume that nodes are full-capability devices with high memory footprints and precise synchronisation hardware and, most importantly, unlimited energy resources. Consequently, the protocols do not address the sources of energy waste previously described. The common behaviour of these protocols is to continuously listen to the channel for possible reception. However, in WSNs, this design will rapidly consume the energy resources of a sensor node, due to idle listening. Moreover, as nodes remain constantly listening to the channel, overhearing is common in that type of protocol. The overhead is also considerable in traditional MAC protocols because the length of the data packets of the applications that they are addressed to are significantly higher than the lengths found in WSNs. The only source of energy waste that these MAC protocols address is collisions because they (apart from wasting energy) considerably reduce the performance of the network. Another reason for their unsuitability is that traditional MAC protocols are commonly focused on throughput or latency, while, in WSNs, these metrics are traded off for a decrease in energy consumption. Although sporadic increases in the network load can happen because of, for example, event detections, WSNs usually work at extremely low loads (with delay-tolerant traffic). Therefore, it is not always necessary to provide a high throughput or reduced delay to the application.

3. A general overview of MAC protocols for WSNs

In previous years, a large number of MAC protocols that are specially designed for WSNs have been defined [3]. To save energy, especially the energy wasted because of idle listening, the most common approach is to put the transceiver into sleep mode for as much time as possible because sleep mode consumes substantially less energy than the other available modes (idle, transmitting or receiving). In sleep mode, a sensor node is not able to receive/transmit packets from/to the medium. This solution greatly reduces the idle listening energy waste. However, specific mechanisms should be defined to ensure that a sensor node will be awake if a node sends something to it. There are three general categories to address this issue [5], as follows:

- **Scheduled (Time Division Multiple Access (TDMA)-like) protocols** [7,8]: in this category, sensor nodes are assigned to specific slots in a frame to transmit and/or receive; then, the nodes only wake up in those slots and sleep in the rest. A node must know the time slot in which a neighbour will be awake before it can send data to it. A schematic representation is shown in Fig. 1(1); observe that nodes transmit on their assigned slots and wake up to receive on the slots of their neighbours. This type of protocol provides good performance (controlled delays, high throughput and reduced or zero collision probabilities) but requires good synchronisation among sensor nodes. Moreover, there is always an overhead cost associated with creating and maintaining the slot assignment and keeping the synchronisation among the nodes.

- **Protocols with Common Active Periods** [9,10]: these protocols organise sensor nodes to wake up and go to sleep at the same time. A sensor node must wait for the next wake-up period to send data. Observe Fig. 1(2), which represents the functionality of one of the most well-known protocols with common active periods: the S-MAC protocol. It can be seen that sensor nodes ready to transmit wait until the active period. Moreover, as an extension to the previously described behaviour of S-MAC presented in [11], the authors of S-MAC define in [9] that a first handshake (using RTS/CTS) is performed in the active period, but data communication is done in the sleep period, allowing the overhearing nodes to go to sleep. Although a certain degree of synchronisation is also needed in these protocols, this requirement is not as stringent as in scheduled TDMA-like protocols. Additionally, protocols with common active periods have a cost associated with the creation and maintenance of the schedule (when to sleep and when to wake up).

- **Asynchronous MAC protocols** [12,13]: in this type of protocol, no synchronisation is required, because each node goes to sleep and wakes up independently of the others. These protocols implement different mechanisms to ensure that a receiver will be awake to receive the data. The protocols can be divided into preamble sampling and receiver-initiated approaches. In preamble sampling, sensor nodes wake up only for checking whether the channel is busy, in which case they remain awake. Next, a sensor node wanting to transmit first sends a long preamble to overlap with the channel sampling time of the receiver and subsequently sends the data. This behaviour is represented in Fig. 1(3); it can be observed that the long preamble makes both the receiver and an overhearing node stay awake to finally receive the message. Conversely, in receiver-initiated protocols, sensor nodes send an advertisement when they wake up. Then, a sensor node with a packet to transmit waits until it receives the advertisement sent by the receiver.

Among the different MAC protocols designed for WSNs, asynchronous approaches and preamble sampling provide some interesting features that make them especially appealing to WSNs. Observe that, compared to protocols with common active periods, preamble sampling provides less complexity and cost because no common schedule needs to be created, communicated and maintained. In addition, preamble sampling consumes less energy compared to protocols with common active periods when the traffic load is low. This difference is caused by the extremely short channel sampling time, which allows a sensor node to return immediately to sleep if the medium is found idle. On the contrary, in protocols with common active periods, the active period of the duty cycle is commonly
large because it should give time for transmitting synchronism messages, as well as control and/or data packets. If compared to TDMA-like MAC protocols, preamble sampling is much less complex, because no schedule should be computed and distributed [3]. Table 1 shows a summary of the characteristics that each category of the MAC protocols provide. This table shows how asynchronous MAC protocols are able to reduce energy consumption while taking into consideration the lack of synchronisation and simplicity requirements. It is also important to note that none of the basic MAC protocols in each category is able to provide traffic load adaptability by itself.

The special suitability of preamble sampling for WSNs has motivated this work, which provides a deep survey of the basic preamble sampling technique and its extensions.

4. Preamble sampling MAC protocols

In the basic preamble sampling technique, sensor nodes sleep and periodically wake up only to sample the channel. If the channel is determined to be busy, the nodes remain awake; otherwise, they return to sleep. The time between channel samples, called the check interval ($T_{ci}$), is fixed and is known by all of the nodes in the network. To ensure

Please cite this article in press as: C. Cano et al., Low energy operation in WSNs: A survey of preamble sampling MAC protocols, Comput. Netw. (2011), doi:10.1016/j.comnet.2011.06.022
the correct reception of packets, each message is preceded by a long preamble transmission that must overlap with the listening time of the receiver; thus, its duration must be at least equal to $T_c$ (see Fig. 2(1)).

El-Hoiydi [14] and Hill et al. [15] presented the preamble sampling technique at approximately the same time in 2002. The former combined this technique with Aloha and called it preamble sampling, while the latter combined it with Carrier Sense Multiple Access (CSMA) and called it Low Power Listening (LPL).\footnote{In this work, \textit{preamble sampling} and \textit{LPL} will be used interchangeably to refer to the same technique: sending a long preamble before each data transmission.}

Later, Polastre et al. defined \textit{Berkeley MAC (B-MAC)} [12], a preamble sampling MAC protocol with an improved Clear Channel Assessment (CCA). Using CCA, the noise floor is estimated by taking samples when the channel is supposed to be free (for instance, immediately after transmitting a packet). Moreover, the decision of whether the channel is clear is made based on the detection of outliers (channel energy significantly below the noise floor) because a valid packet could never have an outlier. This technique reduces the number of false negatives compared with taking only one sample and comparing that sample with the noise floor. B-MAC also defines a set of interfaces to control the protocol parameters: back-off (BO), enable/disable CCA, set the value of the duty cycle and enable/disable the acknowledgement (ACK).

Waking up to sample the channel allows preamble sampling MAC protocols to consume small amounts of energy when the traffic load is small (which is the usual case in WSNs). The main reason for this energy waste reduction

---

**Fig. 2.** Schematic representation of preamble sampling with short packet burst extensions.
is the decrease in idle listening. However, basic preamble sampling protocols have several disadvantages, due to the transmission of the long preamble, as follows:

1. High overhead: The long preamble transmission depends on the sleep period of the duty cycle. Therefore, if a long sleep period is set, a high overhead is inserted at each transmission due to the required length of the preamble. This construct results in a low saturation throughput and in a high transmitting and receiving cost.

2. Costly collisions: If there is a collision, then the entire long preamble is involved, causing a high collision duration (the long preamble can be in the order of a thousand milliseconds). Moreover, if Aloha is used or if hidden terminals exist, then the vulnerability time is equal to the duration of the long preamble and the packet transmissions. This effect further reduces the achieved saturation throughput.

3. High overhearing cost: Using the basic preamble technique, overhearing nodes receive the entire preamble and the data message because they are not able to identify the destination of the transmission until the data packet is received. This effect results in a considerable waste of energy consumption.

4. Incompatibility with newer radios: Sending a long preamble with a variable size is not compatible with modern radios, such as the CC2420 [16], in which only fixed short preambles can be transmitted.

Moreover, the basic preamble sampling technique lacks traffic load adaptability because the duty cycle is fixed for all of the sensor nodes independent of their traffic load requirements. It has to be considered that variations in the duty cycle of a sensor node must be known by its neighbours for them to adapt the duty cycle accordingly.

Several approaches exist that address some of the previously described disadvantages of the basic preamble sampling technique. These approaches can be divided into three different categories: i) mechanisms that divide the long preamble into a burst of short packets, ii) approaches that take advantage of synchronisation information and iii) protocols that adapt the duty cycle.

4.1. Division of the long preamble into a burst of short packets

This category of preamble sampling MAC protocols divides the long preamble into a series of short packets with the main goal of including some useful information in them. Moreover, this approach is fully compatible with modern packetised radios.

The most common approach is to include the destination identifier (ID) in the short packets, thus allowing overhearing nodes to return to sleep for the purpose of reducing the overhearing cost (see Enhanced B-MAC (ENB-MAC) [17]). This type of protocol is represented in Fig. 2(2). Observe that when the destination information is included, the overhearing node can immediately go to sleep.

Other proposals also include information about when the data transmission will start; by doing this, the destination can go to sleep after receiving the short packet and can wake up to receive the data (see also Fig. 2(3), where this behaviour is represented). Examples of this behaviour are the BMAC+ [18], SpeckMAC-B [19], Divided Preamble Sampling MAC (DPS-MAC) [20], Synchronous Wake-Up Frame (SyncWUF) [21] and Signalling-Embedded Short Preamble MAC (SES-P-MAC) [22] protocols.

4.1.1. Transmission of an early acknowledgement

Some approaches also permit the transmission of an early ACK that stops the transmission of the burst. To allow the transmission of the early ACK, a gap should be allowed between short packets. This approach implies that an extra mechanism should be defined to avoid a receiver’s going back to sleep in case it wakes up in a gap of the burst. Most of the existing approaches define that sensor nodes must listen during at least the gap period each time they wake up to sample the medium. These protocols are represented in Fig. 2(4) in which the increased wake-up period can be observed. Some examples of this behaviour are the MAC protocols Transmitter Initiated Cyclical Receiver (TICER) [23], Minimum Preamble Sampling MAC (CSMA-MPS) [24], X-MAC [25], Adaptive Schedule MAC (AS-MAC) [26], Patterned Preamble MAC [27], Asynchronous Real-time Energy-efficient and Adaptive MAC (AREA-MAC) [28], Preamble Sampling with State Information [29] and 1-hop MAC [30]. Other approaches, in contrast, define a double check mechanism in which the channel is sampled twice each time that a sensor node wakes up. The interval between channel samples is such that, even if the first sample is taken in a gap, the second sample detects the short packet burst. Refer also to Fig. 2(5) in which this mechanism is represented and in which it can be seen that the first wake-up does not detect the short preamble, while the second does. The first definition of this technique appeared in Dual Preamble Sampling MAC (DPS-MAC) [31] and is also used in Convergent MAC (CMA) [32].

4.1.2. Repetitions of the data packet

Other approaches, instead of sending a burst of short packets, including some useful information, send repetitions of the data message. By sending repetitions, the destination can immediately receive the packet when it wakes up. These approaches are usually combined with the early ACK technique to reduce the overhearing cost. Examples of this approach are the MX-MAC [33] and the Preamble Sampling with State Information [29] protocols. On the contrary, SpeckMAC-D [19] sends repetitions of the data message until it is ensured that the destination has woken up and received the message. In this case, the sender does not request any type of ACK from the recipient.

4.1.3. Packet-dependent behaviour

Other protocols, such as Multi-mode Hybrid MAC (MH-MAC) [34], DPS-MAC [31], MIX-MAC [35] and Traffic Aware MAC (TrawMAC) [36], perform different operations, depending on the type of packet to be sent. In the MH-MAC and DPS-MAC protocols, the early ACK technique is allowed for unicast messages, while the time until the data transmission start is included in broadcast messages, allowing receivers to go to sleep until the beginning of the transmission. MIX-MAC selects from a pool of MAC...
protocols (compatible among them) the best to be used for each packet transmission. For instance, for small packets and a low sending rate versus receiving rate, the X-MAC protocol (based on short preambles) is used, while for the rest of the cases, except broadcast messages, the MX-MAC protocol (based on repetitions of the data message) is used. In both cases, the early ACK is allowed. For broadcast messages, MIX-MAC uses SpeckMAC-D. In contrast, TrawMAC uses short packet preambles for long data messages and repetitions of the data packet for small messages.

4.1.4. Other approaches

It is worth mentioning the Micro-Frame Preamble MAC (MFP-MAC) [37]. This protocol includes a hash of the data in the short packets to allow nodes to identify whether they have already received the message, which is useful for reducing the load of broadcast communications.

More elaborate approaches are the protocols CMAC [32], 1hopMAC [30] and RA-MAC [38]. In CMAC and 1hopMAC, anycast transmission is used. Then, those nodes with a better routing cost to the sink reply before using an early ACK. In CMAC, the early ACK stops the preamble transmission, while, in 1hopMAC, the sender selects the destination based on the early ACK responses of the neighbourhood. A different approach is defined in RA-MAC, which addresses the problem of concurrent data transmissions and defines that sensor nodes sending a short packet burst as a preamble are capable of aggregating bursts from other nodes if they overhear their burst transmissions. The destination thus creates a schedule for data transmission, alleviating the collision probability.

A completely different technique is the SEESAW [39] protocol. Using SEESAW, nodes willing to transmit send a number of advertisements during their active time that eventually overlap with the listening time of the receiver. Moreover, with the goal of increasing the network lifetime (defined by the authors as the time until the first node runs out of battery), the overhead cost is balanced among sensor nodes. Each node adapts its own parameters (listening time and number of advertisements sent per active time) based on the remaining energy, the listening time and the number of advertisements of its neighbours.

4.2. Taking advantage of synchronisation information

This category of MAC protocols combines preamble sampling techniques with a synchronisation mechanism and has the main goal of reducing the length of the long preamble. These protocols can be divided into two main groups: protocols that make sensor nodes learn the wake-up time of the receivers and protocols that synchronise the wake-up time of all of the neighbouring nodes.

4.2.1. Remembering the wake-up time of the receivers

The first group of MAC protocols make sensor nodes remember the wake-up time of the receivers with the goal of reducing the long preamble duration specifically to account for the possible synchronisation error. This technique first appeared in the definition of Wireless Sensor MAC (WiseMAC) [13], which was presented by El-Hoiydi et al. in 2004. In this protocol, sensor nodes include the information of the next channel sampling time in ACK messages. In this way, transmitters can store this information and use it the next time that new data is available for that node. Knowing the channel sampling time allows the sender to become synchronised with the receiver and to transmit a preamble of length $L_p = \min \left(4\theta T_{\text{sync}}, T_c\right)$ where $\theta$ is the clock drift, $T_{\text{sync}}$ is the time since the last synchronisation happened and $T_c$ is the sampling period. The computed value of the preamble accounts for the possible clock drifts of the sender and the destination with the maximum value of the usual long preamble duration set to the listen plus sleep time. Observe in Fig. 3, where a representation of the WiseMAC protocol is provided, how these techniques are able to considerably reduce the long preamble transmissions. However, note that one disadvantage of these approaches is that short preambles cannot be used to send broadcast messages. In this case, long preambles should be used instead.

This approach is easily combined with the division of the long preamble into a burst of short packets. Immediately before the sampling time of the receiver, the sender starts transmitting the burst of short packets that can be stopped by an early ACK or that can include information of when the data transmission will start. Examples of this behaviour are the protocols CSMA-MPS [24], SyncWUF [21] and TrawMAC [36], which were already seen in the previous section. CSMA-MPS, SyncWUF and TrawMAC adapt the length of the short packet burst according to the time since the last synchronisation information from the receiver is known. Moreover, they allow the reception of an early ACK that makes the transmitter stop the burst, further reducing the transmitting and receiving costs compared with WiseMAC.

One disadvantage of this category of MAC protocols is that they can suffer from continuous collisions when sensor nodes share the same wake-up time. The MAC protocol Asynchronous Scheduled MAC (AS-MAC) [40] tries to address this problem by forcing nodes in a neighbourhood to select a different wake-up time. It allows for a reduction in contention, but there is a cost associated with the advertisement of the schedule because each node broadcasts an advertisement if it wakes up and nothing is received. Moreover, broadcast communication is poorly supported (repetitions of the message should be performed or a long preamble should be transmitted). The authors of WiseMAC also define a technique to address this issue in [13]. Before transmitting the short preamble, a medium access reservation preamble of randomised length is sent. Then, collisions to destinations that share the same wake-up time by nodes using nearly equal short preambles are reduced.

4.2.2. Synchronising wake-up schedules

Conversely, the second group of MAC protocols that take advantage of synchronisation information schedule sensor nodes to wake up at the same instant and use preamble sampling techniques to decrease the impact of the synchronisation error. This type of solution is compatible with broadcast communication because all nodes wake up at nearly the same moment; however, they normally suffer from an increased channel contention. In these protocols, if compared with the approaches that remember the wake-up time of the receivers, there is a reduction of the information that is re-
quired to be stored (because all nodes share the same wake-up time), but, in contrast, there is an increased cost of synchronisation to maintain the schedule.

One of the most well-known approaches is the Scheduled Channel Polling MAC (SCP-MAC) [41] protocol. In SCP-MAC, the maintenance of the common schedule is achieved by periodically sending synchronisation messages and/or piggybacking synchronisation information in data packets, depending on the rate at which packets are sent (i.e., if the rate is low, then explicit synchronisation messages are needed). It is important to note that piggybacking synchronisation information allows for a reduction in the cost of maintaining the schedule. Moreover, the authors of SCP-MAC address the increased contention by defining two contention phases per packet, one for the preamble and the other for the data message. In this way, only the winners of the first contention (those nodes that were able to transmit the preamble, i.e., no busy channel detected before transmission) try to send the data message; therefore, the collision probability is reduced. A similar approach is followed in Crankshaft [42] where a preamble is used to reduce the contention regarding the data message. This protocol belongs to the TDMA-like category because sensor nodes have slots assigned to listen for data from their neighbouring nodes. However, the same slot can be owned by different sensor nodes. This reason motivates the use of long preambles to reduce contention.

Another approach of this group is the MIX-MAC [35] protocol, which takes advantage of the information of the wake-up times of the receivers to synchronise all nodes in a path and to stagger their transmissions with the goal of reducing the end-to-end delay of a burst of packets from a sensor node to the sink.

4.3. Duty cycle adaptation

There are different approaches in the literature to adapting the duty cycle and, also, the preamble duration to achieve a specific goal, usually to handle increases in the traffic load.

4.3.1. Request-based

Some protocols adopt the duty cycle based on requests from the neighbourhood (normally embedded in packet transmissions). For instance, the WiseMAC more bit [13] is used by senders to indicate to the receiver to stay awake (not follow its duty cycle) when they have more packets to send. An extension to this approach is the Stay Awake Promise introduced in [43]. This extension dictates that when a sensor node receives a message with the more bit set to 1, it replies with an ACK with the stay awake promise bit also set. As a result, other nodes willing to transmit to the same destination can contend for the channel immediately after the current data transmission. This approach is useful for handling bursts of packets from different nodes; however, it is not effective for different sensor nodes that want to send only one message. A similar approach is defined in the Burst-Aware Energy-Efficient Adaptive MAC (BEAM) [44]: the sender marks a bit in the header of a packet if it has more packets to send. If the bit is marked, the destination doubles its duty cycle, allowing the sender to reduce the long preamble and thus increase the throughput. Another example is the AS-MAC [26] protocol. In AS-MAC, transmitters include the expected next packet transmission time in the messages, so that the receiver can set its duty cycle accordingly. If the sender indicates that there are no more packets to transmit, the receiver sets the sleep time to the maximum. On the contrary, if the transmitter has another packet pending, the receiver sets the sleep time to the minimum, which is defined as the time needed to forward the current received packet. Also, in AREA-MAC [28], sensor nodes adapt their duty cycle based on specific requests of their neighbours, which are performed to handle real-time data to minimise the delay from source to sink. Finally, the last example is the Asynchronous Sensor MAC (AS-MAC”) [45], in which the transmitter indicates the duty cycle to follow by the receiver when there is a burst of packets to be sent. Receivers can have requests from different nodes in that case. AS-MAC” defines that the request with the smallest sleep interval should be considered.

Please cite this article in press as: C. Cano et al., Low energy operation in WSNs: A survey of preamble sampling MAC protocols, Comput. Netw. (2011), doi:10.1016/j.comnet.2011.06.022

![Diagram](image.png)
4.3.2. Based on the traffic load

Another technique is to base the duty-cycle adaptation on the observed traffic load. An example is the work presented in [46], which proposes an extension of B-MAC+ and defines that, after detecting an increase in the number of incoming packets, a receiver must increase its duty cycle. After that, the sensor node informs its neighbours about that change. The authors specify that the notification is included in the short packets of the burst sent before transmitting a packet. A set of predefined listening modes associated with an incoming rate must be defined and known by all of the sensor nodes. A similar approach is the BoostMAC [47] protocol. In this approach, sensor nodes increase their sleep times in additive steps (until a certain value is reached) if they find the medium idle when they wake up. Otherwise, if a high level of traffic is observed, they decrease the sleep time in a multiplicative manner. Instead of announcing the sleep times that each node is using, senders decide which preamble length is used based on their own traffic and the cost of sending at different preamble lengths, given that the probability that the packet must be retransmitted because the receiver is sleeping (this probability is estimated and fine tuned throughout the lifetime of the network). Another example is the Maximally Traffic-Adaptive MAC (MaxMAC) [48] protocol, which provides an extension for WiseMAC-like protocols. Using MaxMAC, nodes double their duty cycle if the number of received packets exceeds a threshold. They double the duty cycle once again if a second threshold is surpassed. Finally, after exceeding a third threshold, the sensor nodes start working in a CSMA fashion without going to sleep. ACK messages are used to inform about the change in the duty cycle and the duration of the new configuration; therefore, neighbouring nodes can adapt the length of the short packet burst accordingly. A similar approach is the Low Power Listening with Wake-Up after Transmissions MAC (LWT-MAC) protocol [49] in which all nodes that overhear a packet transmission wake up at the end of it and are allowed to send packets (to any destination) without using the long preamble first. In this protocol, sensor nodes behave in a CSMA operation as soon as the network load increases, which improves the network performance. In a similar way, SCP-MAC [50] adds extra channel samples immediately after the reception of a message. The sensor node returns to its regular duty cycle if none of these extra channel samples is useful, i.e., no more messages are received.

Two more approaches that fit in this category are the control algorithms used for setting the sleep schedule presented in [50]: the Asymmetric Additive Duty Cycle Control (AADCC) and the Dynamic Duty Cycle Control (DDCC). In AADCC, each node increments its sleep time in 100 ms increments when it successfully sends 5 packets to the destination. On the contrary, the sleep time is decreased 250 ms if there is a single packet failure. The goal of this algorithm is to react to an increased channel contention: if 5 consecutive packets are correctly transmitted, then the channel contention is low and nodes can sleep for longer periods to save energy. In contrast, a packet failure can indicate a high contention for the channel; therefore, nodes should sleep for shorter periods. This algorithm can work better than a static sleep time; however, the algorithm does not take into account the energy consumption or the required data rate. To this aim, the authors define DDCC. The goal of DDCC is to regulate the sleep interval to accommodate the number of packets to be sent and to minimise the energy consumption. The algorithm is well-suited for single-hop networks; however, in multi-hop sensor networks, there are several key aspects that complicate its design: the variability of the one-hop delay of a path should be reduced and sensor nodes must know information about the entire path to the destination. Both in AADCC and DDCC, nodes must exchange information about their current sleep time. Finally, another approach is the Zero Configuration Algorithm (ZeroCal) [51], in which sensor nodes select the sleep interval that provides the required bandwidth and minimises their energy consumption and the energy consumption of their children. To compute that sleep interval, an energy consumption model is used. It is assumed that sensor nodes will run the energy consumption model to compute the sleep interval to use. The computation is made at fixed intervals or when the number of sent packets of a child exceeds a given threshold. The sleep interval that each node is using is piggybacked in data packets. A similar approach is the traffic adaptation mechanism of X-MAC [25], which allows sensor nodes to select the sleep interval to use based on their estimation of the traffic load. However, in this case, sensor nodes do not need to compute an analytical model; they just look up values and interpolate them from a table of pre-computed values.

A completely different approach appears in the work presented in [52]. Sensor nodes compute their next wake-up time based on the expectation of receiving a new packet with the goal of minimising the delay or the energy consumption. To compute the expectation of receiving a packet, the distribution of the packet inter-arrival times should be known, although they present a learning algorithm to approximate it.

4.3.3. Based on topology information

There are other approaches that adapt the duty cycle based on the position of the node in the topology. For instance, Energy Aware Adaptive LPL (EA-ALPL) [53] assumes a tree topology and allows each sensor node to set its own duty cycle according to its offered load and its number of descendants in the routing tree. Nodes learn the number of children by counting the number of messages they have forwarded in a given interval. The duty cycle information is exchanged among neighbours and stored. Similarly, in Preamble Sampling with State Information [29], each node selects its duty cycle depending on its position on the tree. Nodes that are closer to the sink set a small sleep time (decreasing the energy of sending preambles), while nodes in the leaves configure a longer sleep time. However, the authors leave open how to set the values of the sleep intervals depending on the position in the tree.

4.4. Summary

As a summary of the previous section, the classification of the different MAC protocols is shown in Table 2.
In Table 3, the characteristics of the different extensions of preamble sampling are reviewed. In the case of short preamble burst solutions, the only modification that needs to be introduced to achieve a notable reduction of the overhearing energy consumption is the division of the long preamble into short bursts and the insertion of extra information into them; as a result, the low level of complexity is maintained. If the early ACK is enabled, the complexity is increased but not in a notable manner. The extensions that take advantage of synchronisation information are able to reduce substantially the energy waste by slightly increasing the synchronisation requirements and the complexity. Finally, the approaches that adapt the duty cycle are able to adapt to the traffic load at the cost of an increased complexity. Observe that both the traffic load adaptation and the complexity are higher in those protocols that adapt the duty cycle based on the estimation of the traffic load.

5. Selection guideline

As previously pointed out, the selection of the MAC protocol to use strictly depends on the application to be provided. The usual recommendation is to use TDMA-like MAC protocols for high traffic loads, protocols with common active periods for applications with periodic traffic and asynchronous protocols for low traffic loads [5]. In this section, a more elaborated guideline to select the best MAC category is provided. The guideline takes into account the preamble sampling extensions reviewed in the previous section.

In Fig. 4, the recommended guideline is presented in a flowchart format. The most suitable MAC protocol is suggested, given the most important requirement of an application:

- It is recommended to use TDMA-like protocols for applications with high traffic load requirements and protocols with common active periods in cases of low traffic load and periodic traffic profiles only, as suggested in [5].
- Also, for applications that require the network to support increases in the traffic load (due to event-based traffic) with high QoS requirements, TDMA-like approaches are also recommended. Having slots assigned to sensor nodes enables the network to easily handle increases in the network load and, therefore, maintain the required QoS for the application.
- Current preamble sampling with adaptive duty cycles cannot provide strict QoS to the application because, although there is a reduction of the time to send a packet when the network load is high, contention can still degrade the performance of the network. For this reason, these protocols are recommended when increases in the network load caused by events are expected, but no strict QoS requirements are needed by these messages.
- The last part of the flowchart provides the suggested protocols when the traffic load is low, when it is aperiodic and when increases in the network load are not expected. In these cases, the basic preamble sampling or any of the extensions that divide the long preamble into short bursts or that take advantage of synchronisation can be used. However, a fine election is provided here based on some trade-offs and network configurations, as follows:
  - First, it is recommended to use preamble sampling with some type of technique that exploits synchronisation information in which the complexity of the solution can be traded off for a reduction of energy consumption. As previously discussed, these solutions can substantially reduce the energy consumption but at the cost of a slightly increased complexity. Taking advantage of synchronisation information requires sensor nodes to either obtain and store the wake-up time of the receivers or to create and maintain a scheduled wake-up time (depending on the subcategory used). This construct

<table>
<thead>
<tr>
<th>Category</th>
<th>Energy awareness</th>
<th>Lack of sync. requirements</th>
<th>Simplicity</th>
<th>Traffic load adaptability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short preamble burst</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>–</td>
</tr>
<tr>
<td>Taking advantage of sync. info.</td>
<td>++</td>
<td>+</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Adaptive (requests)</td>
<td>+</td>
<td>++</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Adaptive (load)</td>
<td>+</td>
<td>++</td>
<td>–</td>
<td>++</td>
</tr>
</tbody>
</table>
implies an increased complexity if compared to the protocols that divide the long preamble into bursts of short packets and basic preamble sampling.

- Next, it has been suggested that one of the major advantages of the division of the long preamble into a short packet burst is the reduction of the overhearing energy consumption. For this reason, this category has been recommended for networks with high density (a high number of neighbouring sensor nodes).

- Finally, the use of basic preamble sampling is suggested in cases in which simplicity is more important than a reduction in energy consumption and the density is not high.

6. Concluding remarks

Given the high application dependence of WSNs, there is not one MAC protocol able to satisfactorily work in every deployment. However, preamble sampling is able to pro-
vide some interesting capabilities, which are especially appealing to WSNs that usually work in low traffic load conditions and that are formed by low-capability and energy-constrained devices.

The benefits and disadvantages of basic preamble sampling have been outlined and used as the basis for the extended review of the different preamble sampling protocols that are presented here.

The energy consumption of the basic preamble sampling technique can be further reduced in protocols that divide the long preamble into short burst packets and in protocols that exploit synchronization information. The adaptive approaches, in contrast, apart from attempting to keep the energy consumption low, aim to adapt to the traffic load, improving the system throughput and reducing the delay of the generated packets. These last approaches also imply a greater complexity compared with the basic preamble sampling technique, especially the adaptive techniques that are based on estimating the traffic load.

Acknowledgment

This work has been partially supported by the Spanish Government under projects TEC2008-06055 (Plan Nacional I+D) and CSD2008-00010 (Consolider-Ingenio Program), and by the Catalan Government (SGR2009 #00617).

References
