Energy Efficiency of MAC Protocols in Low Data Rate Wireless Multimedia Sensor Networks: A Comparative Study

Tarek AlSkaif[a,*], Boris Bellalta[b], Manel Guerrero Zapata[a], Jose M. Barcelo Ordinas[a]

[a]*Department of Computer Architecture*
Universitat Politecnica de Catalunya (UPC)
08034 Barcelona, Spain
[b]*Universitat Pompeu Fabra (UPF)*
08018 Barcelona, Spain

Abstract

Some new application scenarios for Wireless Sensor Networks (WSNs) such as urban resilience, smart house/building, smart agriculture and animal farming, among others, can be enhanced by adding multimedia sensors able to capture and transmit small multimedia samples such as still images or audio files. In these applications, Wireless Multimedia Sensor Networks (WMSNs) usually share two conflicting design goals. On the one hand, the goal of maximizing the network lifetime by saving energy, and on the other, the ability to successfully deliver packets to the sink. In this paper, we investigate the suitability of several WSNs MAC protocols from different categories for low data rate WMSNs by analyzing the effect of some network parameters, such as the sampling rate and the density of multimedia sensors on the energy consumption of nodes. First, we develop a general multi-class traffic model that allows us to integrate different types of sensors with different sampling rates. Then, we model, evaluate and compare the energy consumption of MAC protocols numerically. We illustrate how the MAC protocols put some constraints on network parameters like the sampling rates, the number of nodes, the size of the multimedia sample and the density of multimedia nodes in order to make collisions negligible and avoid long queuing delays. Numerical results show that in asynchronous MAC protocols, the receiver-initiated MAC protocols (RI-MAC and PW-MAC) consume less energy than the sender-initiated ones (B-MAC and X-MAC). B-MAC outperforms X-MAC when the sampling rates of multimedia nodes is very low and the polling periods are short. PW-MAC shows the lowest energy consumption between the selected asynchronous MAC protocols and it can be used in the considered WMSNs with a wider range of sampling rates. Regarding synchronous MAC protocols, results also show that they are only suitable for the considered WMSNs when the data rates are very low. In that situation, TreeMAC is the one that offers the lowest energy consumption in comparison to L-MAC and T-MAC. Finally, we compare the energy consumption of MAC protocols in four selected application scenarios related to Smart Cities and environment monitoring.

Keywords: Wireless sensor networks; Wireless Multimedia Sensor Networks; Medium Access Control; Energy efficiency; Smart Cities

1. Introduction

Wireless Sensor Networks (WSNs) are considered as the building blocks of new network paradigms and application scenarios in Smart Cities [1]. In this context, applications such as structural health and urban resilience, smart house/building, smart agriculture and animals surveillance, among others, can be enhanced by adding multimedia sensors (MMSs) able to capture and transmit small multimedia samples such as still images or audio files.

In these networks, namely Wireless Multimedia Sensor Networks (WMSNs) [2], maximizing the network lifetime is of a paramount importance. To achieve this goal, using an energy efficient Medium Access Control (MAC) protocol is key since the radio is a major source of energy consumption in the sensing nodes [3]. The MAC layer coordinates nodes' access to the shared wireless medium. Doing so in an energy efficient way becomes more complicated when nodes of different sampling rates exist in the network and generate different traffic loads.

Based on applications, WMSNs' traffic can be classified into two main categories, multimedia streams (e.g., video streaming) and multimedia data (e.g., snapshot multimedia content). Each of these categories can be further classified, according to the level of Quality of Service (QoS) required by the overlying application, into real-time and delay-tolerant [2]. Multimedia streaming applications put a lot of effort on achieving high bandwidth for a steady flow of data while real-time applications require a delay-bounded delivery of packets. In these cases, energy efficiency is of a lower priority. However, these applications are out of the scope of this paper. Here, we focus on non-streaming and delay-tolerant WMSNs that require relatively lower bandwidth demands than streaming ones [2]. This

[*] Corresponding author. Tel.: +34 603 640 839.
Email addresses: tarek@ac.upc.edu (Tarek AlSkaif), boris.bellalta@upf.edu (Boris Bellalta), guerrero@ac.upc.edu (Manel Guerrero Zapata), josep@ac.upc.edu (Jose M. Barcelo Ordinas)

includes a wide range of environment monitoring and Smart Cities application scenarios, where on the one hand it is essential to keep monitoring the surrounding environment, but on the other hand the phenomenons' observation is delay-tolerant and the generated multimedia traffic is lower -compared to multimedia streams. In this kind of applications MMSs can be deployed to sporadically send still-images or audio files (e.g., images about structural health in a territory, crops status in vineyards, pets and children in a house, sounds and noise in bar zones, among others). This imposes a higher traffic load compared to the typical WSNs where only scalar sensors (SSs) -which sense scalar data and physical attributes (e.g., temperature and humidity readings)- are deployed, and it directly affects the energy efficiency of the MAC layer.

In this paper we study the energy efficiency of the MAC layer in this kind of WMSNs applications by modeling and evaluating the energy consumption of several and different MAC protocols, designed for traditional WSNs, taking into account the existence of MMSs in the network. The paper addresses the spectrum of low data rate applications where the main target is to minimize the energy consumption and increase the lifetime of the sensor network. Therefore, the selected MAC protocols should be those ones which improve the energy efficiency, regardless if they are QoS-aware or if they provide constant bandwidth -as required by streaming applications. To achieve this goal we develop a general sensor network traffic model which allows to integrate different types of sensors with different sampling rates. The model is an extension and a generalization of the model presented in [4] and helps to analyze the effects of various parameters of MMSs -such as the sampling rate, the size of multimedia sample and the density of MMSs- on the traffic each node transmits, receives and overhears. There are previous works on modeling and evaluating the energy consumption of MAC protocols in WSNs like [4, 5]. However, none of those papers models and evaluates the energy consumption of MAC protocols in WMSNs. Moreover, there is a lack of comparisons between the energy consumption of recent MAC protocols and the early designed ones. Therefore, the main goal of this paper is to assess and compare the energy performance of those MAC protocols in low data rate WMSNs, under variable sampling rates and densities of MMSs, in order to find out the suitable MAC protocols for this kind of networks and the WMSNs' scenarios and applications in which each MAC protocol works better.

The rest of the paper is organized as follows: Section 2 provides a brief overview of the related work. In Section 3, the design principles are presented and the multi-class traffic model is derived. The energy consumption of MAC protocols is modeled in Section 4 and Section 5. In Section 6, we conduct a numerical evaluation of the energy performance of MAC protocols under different configurations of WMSNs and in various application scenarios. Finally, the paper is concluded in Section 7.

2. Related Work

The design and implementation of MAC protocols in WSNs have been strongly related to the requirements of applications enabled by sensing nodes. Classical MAC protocols have been originally designed for applications that handle scalar data only. Other MAC protocols have been later developed for more sophisticated applications that usually require a steady flow and/or a real-time delivery of packets. Such applications typically demand high throughput, bounded delay, and high reliability. In this section we will review the two groups of MAC protocols, though later in the paper we will model and evaluate the ones belongs to the first group only, since the set of applications we are considering does not require any streaming support, and using streaming MAC protocols in those applications will increase nodes' energy consumption for an undesired service.

2.1. The main categories of MAC protocols in WSNs

MAC protocols for scalar WSNs have been classified in various categories based on when and how nodes decide to transmit data. These categories are: asynchronous (or random access), synchronous (locally or globally), and hybrid schemes [3, 4, 6]. In terms of energy efficiency, idle listening and collisions are major concerns of MAC protocols in WSNs. Research work have focused on how to improve the performance of MAC protocols in a way the energy wasted in idle listening, collisions, and overhearing is minimized. To reduce idle listening, the duty cycling technique has been widely adopted. With duty cycling, nodes switch periodically between active and sleeping states. Using the asynchronous scheme, each node decides when to wake up autonomously, given the rules defined by the particular MAC protocol, and the duty of the MAC protocol is to establish communication between nodes. Asynchronous MAC protocols for WSNs include: B-MAC [7], X-MAC [8], RI-MAC [9], and PW-MAC [10], among others.

Another category of MAC protocols is the synchronous MAC protocols. This category is further divided into two main branches: locally synchronized and globally synchronized (i.e. frame-slotted) [3]. Locally synchronized MAC protocols (e.g., S-MAC [11] and T-MAC [12]) also adopt the duty cycling mechanism. To save energy, they allow nodes to turn off their radio when no communication occurs during a certain time period. They differ from asynchronous MACs in the sense that each cluster of neighboring nodes are scheduled to wake up at the same time. Frame-slotted MACs (e.g., L-MAC [13] and TreeMAC [14]) divide time into frames and assign time slots to nodes in a way that no two nodes within the two-hop distance are allocated the same time slot. The problem of synchronous MAC protocols is that they require to keep the network synchronized which implies a high control overhead.

2.2. QoS-aware MAC protocols in WSNs

The deployment of resource-constrained sensing nodes in critical environments (e.g., real-time applications) impose additional challenges on the MAC layer in order to assure a certain level of QoS required by the application. For instance, a

MAC protocol has to be flexible and dynamic to changes in the network, minimize the medium access delay by minimizing collisions, and maximize reliability by minimizing traffic losses. There are several examples of MAC protocols in the literature that support QoS metrics such as Q-MAC [15], RL-MAC [16], PQ-MAC [17], CoSenS [18], among others. The QoS-aware MAC protocols for WSNs and WMSNs have been surveyed and classified in [19].

2.3. MAC protocols for streaming WMSNs

Designing a MAC protocol for streaming WMSNs is a complicated task since they require a steady flow of data, in addition, to a delay-bounded delivery of packets, which may be very challenging for any category of MAC protocols mentioned in Section 2.1 (e.g., due to the increasing probability of collisions in asynchronous MACs or the limited slots duration in synchronous MACs). There are several considerations when designing a MAC protocol for video streaming WMSNs which are summarized in [20]. For instance, nodes need to implement intra- and inter-node traffic class differentiation in order to separate traffic according to its classes and serve each class based on its priority. Intra-node traffic class differentiation is achieved by adding queuing management and priority control mechanisms. Inter-node traffic class differentiation requires Contention Window (CW) size control which allows senders to assign a shorter CW to high priority traffic and a larger CW to low priority one. These mechanisms can significantly reduce latency for streaming traffic but at the cost of an increased complexity in the protocol design and low fairness guarantees for nodes with low priority traffic. Saxena [21] is an example of a MAC protocol designed to offer QoS for video streaming WMSNs. The protocol dynamically controls the CW size and duty cycle based on some collected network statistics from the node and the medium such as traffic classes and transmission failures. It shows high adaptive operation to network changes but it causes low-priority traffic to suffer from high latency. In addition, there is no local or global synchronization between nodes which introduces significant idle listening and early sleeping problems [19]. Diff-MAC [22] is another QoS-aware MAC protocol designed for WMSNs with heterogeneous traffic classes by adopting a service differentiation mechanism. In this protocol, long video frames are fragmented into smaller video packets and transmitted as bursts. The CW size and the duty cycle of the node are also adjusted according to the traffic class. The protocol provides fair and fast delivery of data and adapts fast to changing network conditions at the cost of the overhead introduced by service differentiation mechanisms and network monitoring statistics. It also suffers from a lack of sleep-listen synchronization between neighboring nodes [19].

2.4. MAC protocols for low data rate WMSNs

After providing an overview of application-specific MAC protocols, it is clearly observed that existing WMSNs' MAC protocols pay much attention to streaming and real-time applications. However, non-streaming and delay-tolerant traffic class of WMSNs may not require a complex design of MAC protocols like streaming and real-time WMSNs, though they generate higher traffic load than scalar WSNs due to the existence of MMSs. On the other hand, WSNs MAC protocols have been originally designed for scalar sensors with low bandwidth demand and with energy efficiency considerations. Since our focus is on non-streaming and delay-tolerant WMSNs applications, we believe that those WSNs MAC protocols are the best candidates for our applications. Therefore, the main purpose of this paper is to model and evaluate the energy consumption of those MAC protocols in such scenarios. The MAC protocols are selected to be from different categories, such as receiver-initiated and sender-initiated asynchronous MACs, as well as locally and globally synchronized MACs. Then, from each category we choose baseline and recent MAC protocols.

The considered WMSNs include a wide spectrum of applications such as object detection, monitoring and tracking applications (i.e., being widely deployed in Smart Cities). In such applications, a WMSN works typically at very low data rates where collisions are of a little concern [4, 23]. Nevertheless, this could be safeguarded by bounding the maximum traffic flowing through the network (as we will see later in this paper). For instance, in structural health or in crops status in vineyards monitoring applications, MMSs are deployed to send images of buildings/crops. By keeping an archive of images and comparing them with images obtained in different time periods, an improved management/a better productivity could be achieved. However, since the status of these monitored objects is not commonly changing over short periods and does not require a real-time delivery of data, there is no need to sample the environment at high or medium data rates.

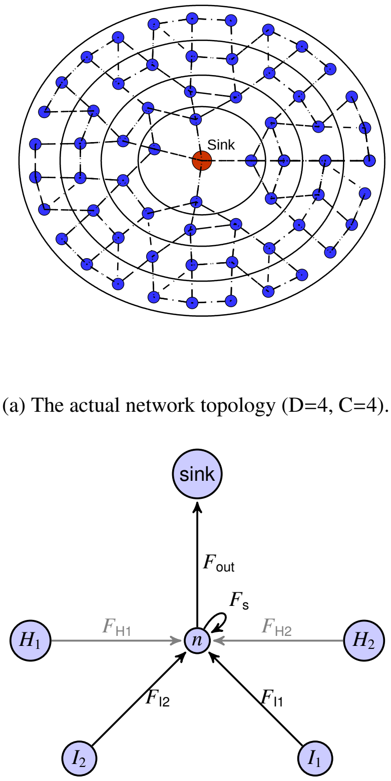
2.5. Platforms for low data rate WMSNs

Several hardware and software platforms have been devised to serve those applications. Cyclops [24] is an imaging platform designed specifically for energy-efficient WMSN applications. It uses a frame differentiating and a background subtraction techniques for detecting moving objects and a low resolution (images of 128x128, 64x64 and 32x32 pixels) to reduce the amount of traffic transmitted. Other platforms that support similar features are Senseye [25] and Firefly [26]. XYZ-ALOHA [27] is another platform that integrates the XYZ networking node with the ALOHA imager. The ALOHA imager outputs metadata (i.e., Address Event Representation) instead of coded images, minimizing the amount of traffic sent towards the sink.

3. System Model and Assumptions

3.1. Design Principles

We focus on a WMSN that consists of a sink, scalar sensors (SSs) and multimedia sensors (MMSs) with a continuous monitoring mode in which nodes take a sample at periodic intervals. The communication pattern is a data gathering tree with traffic flowing hop-by-hop from the leaves (i.e., nodes at different levels) to the root (i.e., the sink) which is placed in the center of the area. Nodes are static and strategically placed in D rings in an increasing number (i.e., rings close to the sink have



(a) The actual network topology (D=4, C=4).

(b) Overview of the traffic model.

Figure 1: Network topology and traffic model.

less nodes than outer rings). The farthest nodes are located in ring d=D and the sink is labeled as d=0. Each node is in the communication range with C neighbors. Routes to the sink are selected according to the shortest path first (SPF) algorithm [28] and they are fairly durable, so that a data gathering tree remains stable during the observation time. An illustrative example of the considered network topology is depicted in Fig. 1(a). The density of MMSs depends on the application. All sensors use the same radio data rate R. Any sensor in the network generates its own traffic (i.e., after taking a sample from the environment) and relays incoming traffic from upper rings. We assume perfect links where both external interference and the effects of fading are negligible since we consider a static WSN channel model [4]. This assumption will allow us to exclusively focus on the characteristics of the MAC protocols, providing a better understanding of the pure energy consumption behavior of each one without external factors. We also assume that the sampling rates of sensors within the WMSN are low enough to consider the collision probability negligible [4, 23], including those with hidden nodes. However, we will include later in the study some load constraints to limit the amount of traffic flowing through the network in order to make collisions negligible.

3.2. *Traffic Model*

For deriving the traffic model, we extend the one proposed by Langendoen [4], which models the traffic flowing through nodes to the sink in a homogeneous sampling rate sensor network, to a model for a multi-class sampling rate sensor network in which we have L classes of nodes, where each class has its own sampling rate. For each node, let F_{s} be the rate at which it samples the environment, F_{I} the rate of incoming traffic it has to forward, and F_{H} the rate of traffic it overhears, which is caused by neighboring nodes. Fig. 1(b) gives an example of the traffic model for a given node n. The overhearing traffic is generated by nodes H_1 and H_2. F_{I1} and F_{I2} are the rates of incoming traffic. F_{out} is the total output traffic rate, which includes the rate of self-generated traffic F_{s}, and the total incoming traffic it has to forward F_{I}.

In a similar way as in [4], N nodes are deployed in the area with a uniform node density. Assuming a unit disk graph communication model, each unit disk contains C+1 nodes on average. Thus, all nodes are in communication range with a fixed number of neighbors C. As mentioned before, the nodes are located in D rings according to their distance to the sink (i.e, in d=0). The first ring contains C nodes, from which we can derive the node density in each ring. The average number of nodes N_{d} in ring d is:

$$N_d = \begin{cases} 1 & d{=}0 \\ Cd^2 - C(d-1)^2 = (2d-1)C & \text{otherwise.} \end{cases} \tag{1}$$

Let us assume a general case where there are L classes of sensors, and the nodes sample the environment at a rate F_{s}^l, according to the class they belong to, where $l \in 1, \ldots, L$. At each ring there is a percentage p_l of nodes of class l, and the average number of nodes in ring d for each class is $N_{\text{d}}^l = p_l(2d-1)C$, while the average number of input links of class l is given by the formula:

$$\begin{aligned} I_{\text{d}}^l &= \frac{N_{\text{d+1}}^l}{N_{\text{d}}} \\ &= p_l \frac{(2d+1)}{(2d-1)}. \end{aligned} \tag{2}$$

We take a node of class l at a ring d that has an incoming traffic of class i, and define $F_{\text{out}}^{d,l,i}$ as the output traffic of class i for this node as follows:

$$F_{\text{out}}^{d,l,i} = \begin{cases} 0 & d = 0, \forall i \\ F_{\text{I}}^{d,i} & 0 < d < D, i \neq l \\ F_{\text{s}}^l + F_{\text{I}}^{d,l} & 0 < d < D, i = l \\ F_{\text{s}}^l & d = D, i = l \\ 0 & d = D, i \neq l, \end{cases} \tag{3}$$

where $F_{\text{I}}^{d,i}$ is the incoming traffic rate of class $i \in 1, \ldots, L$ in ring d. The incoming traffic is on average the same for any node at the same ring, since these nodes have an equal average number of input links of any class, and it is given by the following formula:

$$F_{\text{I}}^{d,i} = \frac{(D^2 - d^2)}{(2d-1)} p_i F_{\text{s}}^i. \tag{4}$$

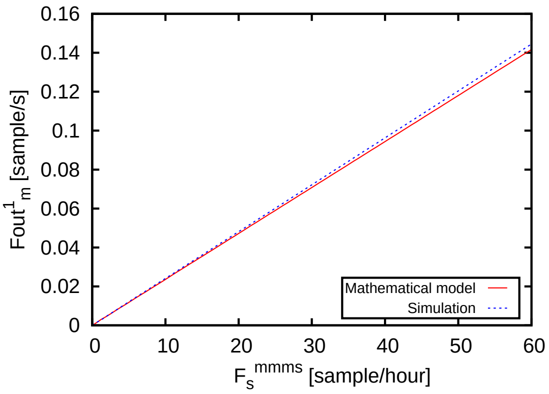


Figure 2: Validation of the mathematical model.

Using formula (2) we can distinguish between the average incoming traffic coming from each class of sensors.

The overhearing traffic for a node in ring d from class i is given by the following formula:

$$F_{\mathrm{H}}^{d,i} = \sum_{l=1}^{L} (N_{\mathrm{d}}^{l} - I_{\mathrm{d}}^{l}) F_{\mathrm{out}}^{d,l,i}. \tag{5}$$

We differentiate between the class of a given node and the class of traffic it forwards using the notations l and i, respectively. The proposed traffic model allows the usage of different sampling rates depending on the class of sensors. To validate this model we simulate a uniformly distributed topology in which we threw 64 sensors randomly around a sink and took a node near the sink (i.e., 1-hop distance) to calculate its average output traffic rate of around 50 runs. The paths are selected according to the SPF algorithm. The sensors are grouped based on their distance to the sink. The average output traffic rate in the random topology and the mathematical model are calculated and compared in Fig. 2. It can be observed that the mathematical model is within 2-4% of the value determined by the random topology.

3.3. Multimedia Sampling Rate

Assume we have two types of sensors: SSs and MMSs. MMS nodes are equipped with cameras and devoted to object detection and object monitoring duties. To do that, MMS nodes periodically take an image at a rate F_s^{mms} and send the image to the sink. The sampling rate, defined as the frequency at which an image is taken, can range from tens of seconds to hours. Every time an image is taken, depending on the image size (e.g., in pixels), and the coding and compression scheme, a MMS will generate data that is larger than a single layer 2 payload. Thus, every multimedia sample (MM) is divided and represented by M payloads, being the size of M dependent on the image taken, and the coding and compressing mechanisms. The size of each multimedia payload depends on the multimedia content and can reach a max value P^m.

As an example, for a 64x64 pixel image, with Red-Green-Blue (RGB) coding (i.e., 24-bit per pixel), an image will have a size of around 100KB. Assuming compression ratios of 90% or less (e.g., after a background subtraction process), the image size can be reduced to near 10KB or less, hence, with a layer 2 payload (P^m), of a size 512B for instance, a MMS will generate around M=20 payloads each of a size P^m=512B. Thus, when accounting for the energy spent in sending and receiving every multimedia sampling, M payloads have to be taken into account. Later in the study we will show how the size of the MM sample (i.e., the value of M) affects the maximum allowed sampling rate of MMSs (F_s^{mms}).

In the case of SS nodes, the sampling rate is also quite low (e.g., one sample per minute) and every sample produces a single packet. The data retrieved by SS nodes is relatively small and could be fit in one single payload P^s. It is clear that the self-generated traffic by MMSs (F_s^{mms}) will be much higher than by SSs (F_s^{ss}), however, we stress that the MMSs' sampling rate (e.g., the number of images taken per second) is low enough to do not cause congestion or queuing delays.

3.4. Sampling Energy Consumption

Since the multimedia applications we are considering are environment monitoring or object detection, we use low cost, low power and low resolution camera sensors like Cyclops [24]. We assume that the amount of power consumed in the subsystems of a MMS node is considerably higher than of a SS. For example a temperature SS consumes $P_{\mathrm{ss}} = 6\ \mu$W for sensing the environment [29], while a MMS that uses a tiny Cyclops camera consumes $P_{\mathrm{mms}} = 42$ mW for capturing an image [30]. We also assume that MMSs do in-node processing and compression of the multimedia content before sending the image to downstream nodes in order to reduce traffic by reducing the size of images and the number of payloads. Let us call e_{s}^{l} the energy spent in capturing and processing a sample from the environment for a node of class l (e.g., $e_{\mathrm{s}}^{\mathrm{mms}}$ if it is a MMS), then the energy spent in sampling the environment is:

$$E_{\mathrm{s}}^{l} = F_{\mathrm{s}}^{l} e_{\mathrm{s}}^{l}. \tag{6}$$

4. Energy Models for Asynchronous MAC Protocols in WMSNs

In this section we model the energy consumption of some baseline and recent asynchronous duty-cycling MAC protocols including both sender-initiated transmission like B-MAC [7] and X-MAC [8], and receiver-initiated transmission like RI-MAC [9] and PW-MAC [10].

4.1. Sender-Initiated MAC Protocols:

4.1.1. B-MAC

Berkeley MAC [7] is an asynchronous MAC protocol for WSNs, in which each node periodically performs a carrier sense to detect the radio channel state during a short period, which is known as *Low Power Listening (LPL)*. If the channel is clear, a sender can hold the channel and send the data, which is preceded by a preamble, to ensure a correct reception by all potential receivers who are duty cycling. Potential receivers stay awake to receive the data when an activity in the channel is detected (i.e, the preamble), see Fig. 3. This reduces the idle-listening overhead without the need for an explicit synchronization between nodes, but it comes at the expense of sending out a long preamble that covers one complete polling interval T_w. The sources of energy consumption in B-MAC are the energy spent in performing a regular carrier sense e_cs, transmitting e_tx, receiving e_rx, overhearing e_ov and the energy spent in taking a sample from the environment e_s. The power drawn in each mode are P_idl, P_tx, P_rx and P_s, respectively and their values are given in Table 1.

The time required to transmit, receive and overhear a packet of class i in B-MAC is:

$$\begin{aligned} T^i_\mathrm{tx} &= T_\mathrm{cs} + T_\mathrm{w} + T^i_\mathrm{msg}, \\ T^i_\mathrm{rx} &= \frac{T_\mathrm{w}}{2} + T^i_\mathrm{msg}, \\ T_\mathrm{ov} &= \frac{T_\mathrm{w}}{2} + T_\mathrm{hdr}, \end{aligned} \tag{7}$$

respectively, where T_cs is the time spent in sensing the channel, T_w is the polling period of a receiver and it represents the length of the preamble, and T^i_msg is the time required for sending one payload of class i. Each payload is preceded by a packet header and followed by an acknowledgement. We account also for the radio switch delay by adding T_SIFS as follows:

$$T^i_\mathrm{msg} = T_\mathrm{hdr} + \frac{P^i}{R} + T_\mathrm{SIFS} + T_\mathrm{ack}. \tag{8}$$

The energy spent in each mode is:

$$\begin{aligned} e^i_\mathrm{tx} &= (T_\mathrm{cs} + T_\mathrm{SIFS})\, P_\mathrm{idl} + \left(T_\mathrm{w} + T_\mathrm{hdr} + \frac{P^i}{R}\right) P_\mathrm{tx} \\ &\quad + T_\mathrm{ack} P_\mathrm{rx}, \end{aligned} \tag{9}$$

$$e^i_\mathrm{rx} = T_\mathrm{SIFS} P_\mathrm{idl} + \left(\frac{T_\mathrm{w}}{2} + T_\mathrm{hdr} + \frac{P^i}{R}\right) P_\mathrm{rx} + T_\mathrm{ack} P_\mathrm{tx}, \tag{10}$$

$$e_\mathrm{ov} = \left(\frac{T_\mathrm{w}}{2} + T_\mathrm{hdr}\right) P_\mathrm{rx}, \tag{11}$$

$$e_\mathrm{cs} = T_\mathrm{cs} P_\mathrm{idl}. \tag{12}$$

We calculate the energy consumption in each mode during a given observation time T_obs. The time in which a node of class l is active in T_obs represents the total transmitting, receiving and overhearing times, and is given by the following:

$$\begin{aligned} T^{d,l}_\mathrm{active} &= T_\mathrm{obs}\Bigg(\left(M^l F^l_\mathrm{s} T^l_\mathrm{tx} + \sum_{i=1}^{L} M^i F^{d,i}_\mathrm{I} T^i_\mathrm{tx}\right) + \left(\sum_{i=1}^{L} M^i F^{d,i}_\mathrm{I} T^i_\mathrm{rx}\right) \\ &\quad + \left(\sum_{i=1}^{L} M^i F^{d,i}_\mathrm{H} T_\mathrm{ov}\right)\Bigg), \end{aligned} \tag{13}$$

and the inactive time is calculated as follows:

$$T^{d,l}_\mathrm{inactive} = T_\mathrm{obs} - T^{d,l}_\mathrm{active}, \tag{14}$$

where M^l is the number of payloads of class l, F^l_s is the rate at which a node of class l samples the environment, $F^{d,i}_\mathrm{I}$ is the incoming traffic rate of any class i in ring d, and $F^{d,i}_\mathrm{H}$ is the overhearing traffic from any class i in ring d.

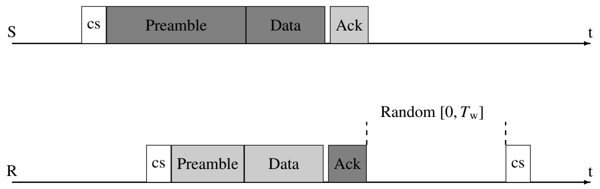


Figure 3: The operation of B-MAC, where a sender S transmits a data packet preceded by a long preamble to ensure a correct reception by a potential receiver R.

Then, the total energy consumed in T_obs in each state is:

$$E^l_\mathrm{s} = \left(F^l_\mathrm{s} e^l_\mathrm{s}\right) T_\mathrm{obs}, \tag{15}$$

$$E^{d,l}_\mathrm{tx} = \left(M^l F^l_\mathrm{s} e^l_\mathrm{tx} + \sum_{i=1}^{L} M^i F^{d,i}_\mathrm{I} e^i_\mathrm{tx}\right) T_\mathrm{obs}, \tag{16}$$

$$E^{d,l}_\mathrm{rx} = \left(\sum_{i=1}^{L} M^i F^{d,i}_\mathrm{I} e^i_\mathrm{rx}\right) T_\mathrm{obs}, \tag{17}$$

$$E^{d,l}_\mathrm{ov} = \left(\sum_{i=1}^{L} M^i F^{d,i}_\mathrm{H} e_\mathrm{ov}\right) T_\mathrm{obs}, \tag{18}$$

$$E_\mathrm{cs} = T_\mathrm{inactive} \frac{T_\mathrm{cs}}{T_\mathrm{w}} e_\mathrm{cs}, \tag{19}$$

$$E^{d,l}_\mathrm{ctrl} = 0. \tag{20}$$

Except where otherwise stated, in the modeling of the next MAC protocols, these equations are computed the same way and they will not be displayed.

Then we compute the total energy consumption as follows:

$$E^{d,l}_{T_\mathrm{obs}} = E^l_\mathrm{s} + E^{d,l}_\mathrm{tx} + E^{d,l}_\mathrm{rx} + E^{d,l}_\mathrm{ov} + E_\mathrm{cs} + E^{d,l}_\mathrm{ctrl}, \tag{21}$$

where $E^{d,l}_\mathrm{ctrl}$ refers to that energy consumed by sending and receiving control packets (e.g., synchronization messages) which is zero in the case of B-MAC.

4.1.2. X-MAC

X-MAC [8] divides the long preamble in B-MAC into a series of short preamble bursts of duration T_sp. Because the destination address is included in the short preambles, non-target receivers can immediately go back to sleep after receiving a short preamble packet, which reduces the energy spent in overhearing. The short preamble bursts are interleaved with short idle times of duration T_ea to allow a receiver to reply with an *early acknowledgment*. Whenever a sender receives an early ACK from the intended receiver, it stops sending the preamble bursts and starts sending the data.

Introducing the early acknowledgement could achieve considerable energy savings by reducing the preamble length to half on average compared to B-MAC, but comes at the price of an increased time for carrier sensing (i.e., $T_\mathrm{cs} + T_\mathrm{ea}$) each time a node wakes up. A node turns off its radio if the medium has

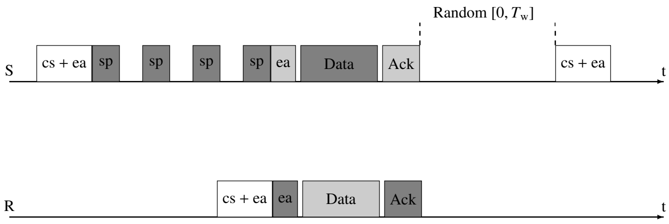


Figure 4: The operation of X-MAC, including the short preamble bursts (sp) and the early acknowledgment (ea).

been idle for a time longer than the gap duration between two short preambles. X-MAC mechanism is illustrated in Fig. 4.

The time required to transmit, receive and overhear a packet of class i in X-MAC is:

$$\begin{aligned} T^i_{\mathrm{tx}} &= T_{\mathrm{cs}} + T_{\mathrm{ea}} + \frac{T_{\mathrm{w}}}{2} + T_{\mathrm{SIFS}} + T^i_{\mathrm{msg}}, \\ T^i_{\mathrm{rx}} &= 1.5(T_{\mathrm{sp}} + T_{\mathrm{ea}}) + T_{\mathrm{SIFS}} + T^i_{\mathrm{msg}}, \\ T_{\mathrm{ov}} &= 1.5(T_{\mathrm{sp}} + T_{\mathrm{ea}}), \end{aligned} \tag{22}$$

respectively, where:

$$T_{\mathrm{w}} = N_{\mathrm{sp}}(T_{\mathrm{cs}} + T_{\mathrm{ea}}), \tag{23}$$

where N_{sp} is the number of short preambles.

The energy spent in each mode is:

$$\begin{aligned} e^i_{\mathrm{tx}} &= (T_{\mathrm{cs}} + T_{\mathrm{ea}} + 2T_{\mathrm{SIFS}})P_{\mathrm{idl}} + \left(\frac{T_{\mathrm{w}}}{2} + T_{\mathrm{hdr}} + \frac{P^i}{R}\right)P_{tx} \\ &\quad + T_{\mathrm{ack}}P_{\mathrm{rx}}, \end{aligned} \tag{24}$$

$$\begin{aligned} e^i_{\mathrm{rx}} &= 2T_{\mathrm{SIFS}}P_{\mathrm{idl}} + \left(1.5(T_{\mathrm{sp}} + T_{\mathrm{ea}}) + T_{\mathrm{hdr}} + \frac{P^i}{R}\right)P_{\mathrm{rx}} \\ &\quad + T_{\mathrm{ack}}P_{\mathrm{tx}}, \end{aligned} \tag{25}$$

$$e_{\mathrm{ov}} = 1.5(T_{\mathrm{sp}} + T_{\mathrm{ea}})P_{\mathrm{rx}}, \tag{26}$$

$$e_{\mathrm{cs}} = (T_{\mathrm{cs}} + T_{\mathrm{ea}})P_{\mathrm{idl}}, \tag{27}$$

$$e_{\mathrm{ctrl}} = 0. \tag{28}$$

Similar to B-MAC, the total energy consumption in X-MAC is:

$$E^{d,l}_{T_{\mathrm{obs}}} = E^l_{\mathrm{s}} + E^{d,l}_{\mathrm{tx}} + E^{d,l}_{\mathrm{rx}} + E^{d,l}_{\mathrm{ov}} + E_{\mathrm{cs}}. \tag{29}$$

4.2. Receiver-Initiated MAC Protocols:

4.2.1. RI-MAC

To improve energy efficiency, receiver-initiated probing has been adopted in some asynchronous MAC protocols. In this type of MAC protocols, a sending node does not start transmitting until the receiver is ready to receive. Receiver-Initiated MAC (RI-MAC) [9] aims at minimizing the time during which a sender and its intended receiver are occupying the wireless medium to find a rendezvous. Each node wakes up periodically and sends a short beacon to notify potential transmitters that it is awake and ready to receive the data. When a node wants to transmit, it samples the channel and remains active (i.e., for an average period $T_{\mathrm{w}}/2$) until receiving a beacon of duration T_{B} from its intended receiver. After receiving the beacon, the transmitter starts sending the data, as shown in Fig. 5.

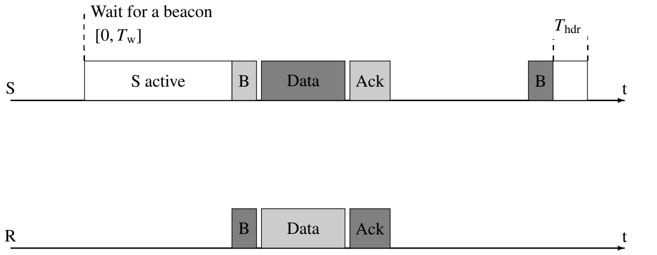


Figure 5: In RI-MAC, a sender S does not occupy the wireless medium until receiving a short beacon (B) from a receiver R meaning that it is awake and ready to receive the data.

The time required to transmit, receive and overhear a packet of class i in RI-MAC is:

$$\begin{aligned} T^i_{\mathrm{tx}} &= \frac{T_{\mathrm{w}}}{2} + T_b + T_{\mathrm{SIFS}} + T^i_{\mathrm{msg}}, \\ T^i_{\mathrm{rx}} &= T_{\mathrm{B}} + T_{\mathrm{SIFS}} + T^i_{\mathrm{msg}}, \\ T_{\mathrm{ov}} &= T_{\mathrm{B}} + T_{\mathrm{SIFS}} + T_{\mathrm{hdr}}, \end{aligned} \tag{30}$$

respectively. Then, after transmitting a beacon, a node expects the incoming packet within a small window T_{hdr}, as shown in Fig. 5. If the node is not the intended receiver it overhears the header only.

The energy spent in each mode is:

$$\begin{aligned} e^i_{\mathrm{tx}} &= \left(\frac{T_{\mathrm{w}}}{2} + 2T_{\mathrm{SIFS}}\right)P_{\mathrm{idl}} + \left(T_{\mathrm{hdr}} + \frac{P^i}{R}\right)P_{\mathrm{tx}} \\ &\quad + (T_{\mathrm{B}} + T_{\mathrm{ack}})P_{\mathrm{rx}}, \end{aligned} \tag{31}$$

$$e^i_{\mathrm{rx}} = 2T_{\mathrm{SIFS}}P_{\mathrm{idl}} + \left(T_{\mathrm{hdr}} + \frac{P^i}{R}\right)P_{\mathrm{rx}} + (T_{\mathrm{B}} + T_{\mathrm{ack}})P_{\mathrm{tx}}, \tag{32}$$

$$e_{\mathrm{ov}} = T_{\mathrm{B}}P_{\mathrm{tx}} + T_{\mathrm{SIFS}}P_{\mathrm{idl}} + T_{\mathrm{hdr}}P_{\mathrm{rx}}, \tag{33}$$

$$e_{\mathrm{B}} = T_{\mathrm{B}}P_{\mathrm{tx}}, \tag{34}$$

$$e_{\mathrm{ctrl}} = 0. \tag{35}$$

The total energy consumption in T_{obs} is:

$$E^{d,l}_{T_{\mathrm{obs}}} = E^l_{\mathrm{s}} + E^{d,l}_{\mathrm{tx}} + E^{d,l}_{\mathrm{rx}} + E^{d,l}_{\mathrm{ov}} + E_{\mathrm{B}}, \tag{36}$$

where the total energy spent in sending out a periodic beacon message in receiver-initiated MAC protocols is calculated in a similar way as the total energy spent in carrier sensing in sender-initiated MAC protocols (see Eq. 19) and it is given by the following formula:

$$E_{\mathrm{B}} = T_{\mathrm{inactive}}\frac{T_{\mathrm{B}}}{T_{\mathrm{w}}}e_{\mathrm{B}}. \tag{37}$$

4.2.2. PW-MAC

Predictive-Wakeup MAC (PW-MAC) [10] is an asynchronous receiver-initiated MAC protocol which reduces the duty cycle at both the receiver and the sender. The goal of PW-MAC is for a sender S to wake up right before its intended

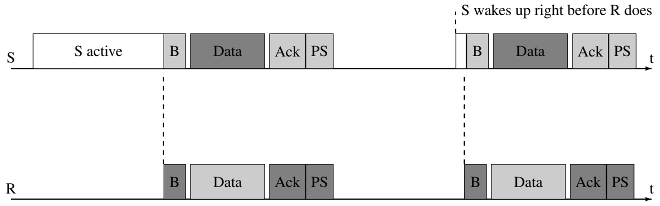


Figure 6: In PW-MAC, the intended receiver's wake-up time is predicted so that the sender S wakes up slightly before the receiver R in order to save the energy spent in idle listening.

receiver R does. As in RI-MAC, each node periodically wakes up and broadcasts a beacon of duration T_{B} to announce that it is awake and ready to receive the data. If S has a packet to send to R, S turns on its radio and waits for a beacon from R. Upon receiving R's beacon, S transmits the data message, setting a special flag in the message header to request R's *Prediction State (PS)*. Then, R sends an ACK followed by a short packet of duration T_{PS} in which it embeds its current time and prediction state. The current time of R is used by S to compute the time difference between S and R's clocks. Thus, using the prediction information, node S can predict future wakeup times of R. The PS of R represents the expected time at which R will wake up next time. In the future, when S has data packets to R, S wakes up for only a short duration T_{ss} right before the predicted wakeup time of R. In contrast to RI-MAC, in which a sender stays awake for on average a half wakeup interval waiting for R, PW-MAC significantly reduces this idle listening time once the prediction state of the receiver is learned by the sender. The mechanism of PW-MAC is illustrated in Fig. 6.

The time required to transmit, receive and overhear a packet of class i in PW-MAC is:

$$\begin{aligned} T^i_{\text{tx}} &= T_{\text{ss}} + T_{\text{B}} + T_{\text{SIFS}} + T^i_{\text{msg}} + T_{\text{PS}}, \\ T^i_{\text{rx}} &= T_{\text{B}} + T_{\text{SIFS}} + T^i_{\text{msg}} + T_{\text{PS}}, \\ T_{\text{ov}} &= T_{\text{B}} + T_{\text{SIFS}} + T_{\text{hdr}}, \end{aligned} \tag{38}$$

respectively, and the energy spent in each mode is:

$$e^i_{\text{tx}} = 2T_{\text{SIFS}}P_{\text{idl}} + \left(T_{\text{hdr}} + \frac{P^i}{R}\right)P_{\text{tx}} + (T_{\text{B}} + T_{\text{ack}} + T_{\text{PS}})P_{\text{rx}}, \tag{39}$$

$$e^i_{\text{rx}} = 2T_{\text{SIFS}}P_{\text{idl}} + \left(T_{\text{hdr}} + \frac{P^i}{R}\right)P_{\text{rx}} + (T_{\text{B}} + T_{\text{ack}} + T_{\text{PS}})P_{\text{tx}}, \tag{40}$$

$$e_{\text{ov}} = T_{\text{SIFS}}P_{\text{idl}} + T_{\text{hdr}}P_{\text{rx}} + T_{\text{B}}P_{\text{tx}}, \tag{41}$$

$$e_{\text{B}} = T_{\text{B}}P_{\text{tx}}, \tag{42}$$

$$e_{\text{ctrl}} = 0. \tag{43}$$

Similar to RI-MAC, there is no explicit channel sensing in PW-MAC. A nodes sends out periodically a beacon message. Moreover, the times spent in sending and receiving the control packet (T_{PS}) are included in the transmission and reception times in Eq. 38.

The total energy consumption in PW-MAC in T_{obs} is:

$$E^{d,l}_{T_{\text{obs}}} = E^l_{\text{s}} + E^{d,l}_{\text{tx}} + E^{d,l}_{\text{rx}} + E^{d,l}_{\text{ov}} + E_{\text{B}}. \tag{44}$$

5. Energy Models for Synchronous MAC Protocols in WMSNs

5.1. Locally Synchronized MAC Protocols

Locally synchronized MAC protocols allow nodes to turn on their radio at synchronized times and turn them off when no communication occurs during some time. A node determines its next wakeup time and broadcasts its schedule before going back to sleep. Although the communication in locally synchronized MAC protocols is grouped at the beginning of each schedule, raising the chances of collisions, they do not face the problem of finding a rendezvous between nodes as in asynchronous MAC protocols.

5.1.1. T-MAC

S-MAC [11] uses a fixed duty cycle which results in an energy waste in idle listening when traffic load fluctuates. It runs at a duty cycle that matches the load of the busiest node in the network. For this reason, S-MAC is not recommended when the traffic load does not remain constant and predictable. Timeout-MAC (T-MAC) [12] is an extension of S-MAC that allows a dynamic adaptation of the duration of the active period (T_{slot}) to the actual load. The active period is dynamically extended or ended according to a certain time-out period $T_{\text{time-out}}$. Time-outs present a simple but effective way to address the idle listening problem when network traffic load varies. T-MAC mechanism is illustrated in Fig. 7.

Nodes in T-MAC wake up periodically. During the active periods, they contend for the channel -if they have packets to send- in a contention window of duration T_{CW}, then they exchange Request-to-Send (RTS) and Clear-to-Send (CTS) packets followed by the actual payload.

Nodes also exchange synchronization messages periodically. At the beginning of each synchronization period, a node sends one synchronization header, and receives synchronization headers from its one-hop neighbors (i.e., each node has C neighbors as mentioned in Section 3) at a rate F_{sync}, which adds additional sources of energy consumption ($e_{\text{tx,sync}}$ and $e_{\text{rx,sync}}$).

The times required for transmitting, receiving, overhearing, and synchronization in T-MAC are:

$$\begin{aligned} T^i_{\text{tx}} &= \frac{T_{\text{CW}}}{2} + T_{\text{RTS}} + T_{\text{SIFS}} + T_{\text{CTS}} + T_{\text{SIFS}} + T^i_{\text{msg}}, \\ T^i_{\text{rx}} &= \frac{T_{\text{CW}}}{2} + T_{\text{RTS}} + T_{\text{SIFS}} + T_{\text{CTS}} + T_{\text{SIFS}} + T^i_{\text{msg}}, \\ T_{\text{ov}} &= \frac{T_{\text{CW}}}{2} + T_{\text{RTS}}, \\ T_{\text{tx,sync}} &= \frac{T_{\text{CW}}}{2} + T_{\text{hdr}}, \\ T_{\text{rx,sync}} &= \frac{T_{\text{CW}}}{2} + T_{\text{hdr}}. \end{aligned} \tag{45}$$

	Parameter	Description	Value
	T_{cs}	Time for carrier sense (ms)	2.5
	R	Data rate (kbyte/s)	31.25
	P^m	Size of a multimedia payload (byte)	512
	P^s	Size of a scalar payload (byte)	32
	MM	Image size (kbyte)	10
	M^m	Number of multimedia Payloads	20
	M^s	Number of scalar Payloads	1
	P_{tx}	Power in transmission mode (mW)	[52.2]
	$P_{\text{rx}} = P_{\text{idl}}$	Power in receiving and idle listening mode (mW)	[56.4]
	P_{mms}	Power for capturing an image (mW)	[42]
	T_{SIFS}	Short inter-frame space (μs)	11
	L_{ack}	Acknowledgment length (byte)	12
	L_{hdr}	Message header length (byte)	12
	T_{w}	Polling period (s)	[0.02, 0.5]
	θ	Frequency tolerance (ppm)	30
X-MAC	L_{sp}	Short Preamble length (byte)	12
	T_{ea}	A gap between short preambles for early ACK (ms)	3.75
RI-MAC	L_{B}	Beacon length (byte)	12
PW-MAC	T_{ss}	A sender S waits a short period before a receiver R wakes up (ms)	5
	L_{B}	Beacon length (byte)	12
	L_{PS}	Prediction State: seed of R + time diff between S and R + last wakeup of R (byte)	2+4+4=10
T-MAC	$L_{\text{RTS}}, L_{\text{CTS}}$	Request-to-Send, Clear-to-Send (byte)	12
	CW	Contention Window	1024
	T_{sync}	Time between synchronization messages (s) ($F_{\text{sync}} = 1/T_{\text{sync}}$)	60
	T_{slot}	Duration of an active period (s)	[0.1, 1]
L-MAC	N_{slots}	Number of slots	32
TreeMAC	N_{slots}	Number of slots	3
	N_{frames}	Number of frames	[12, 20]
	$T_{\text{sync*}}$	Synchronization message interval (s)	5
	T_{sch}	Schedule update interval (s)	8
	T_{bd}	Bandwidth demand update interval (s)	10

Table 1: Parameters of traffic model, MAC protocols and the radio used (CC2420) with the corresponding values.

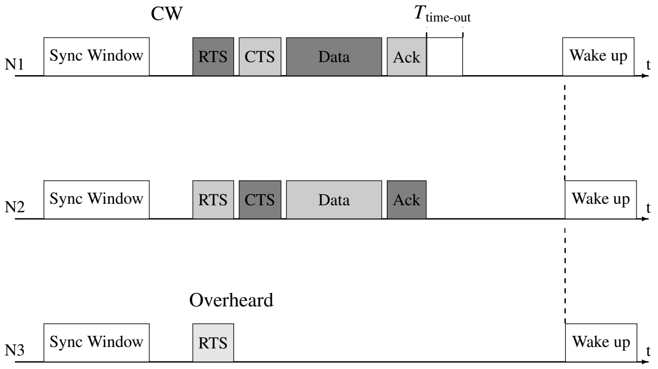


Figure 7: The operation of T-MAC with an adaptive duty cycle using a time-out period of duration $T_{\text{time-out}}$.

Then, after each transmission or reception, a node stays idle for a period T_{idl} until the time-out timer expires (see Fig. 7). It takes into account possible clock drifts from its neighbors as follows:

$$T_{\text{idl}} = T_{\text{guard}} + T_{\text{time-out}}, \tag{46}$$

where:

$$T_{\text{time-out}} = T_{\text{CW}} + T_{\text{RTS}} + T_{\text{SIFS}} + T_{\text{CTS}}, \tag{47}$$

$$T_{\text{guard}} = 4\theta T_{\text{sync}}. \tag{48}$$

The energy spent in each mode is:

$$\begin{aligned} e^i_{\text{tx}} &= \left(\frac{T_{\text{CW}}}{2} + 3T_{\text{SIFS}} + T_{\text{idl}}\right)P_{\text{idl}} + \left(T_{\text{RTS}} + \frac{P^i}{R}\right)P_{\text{tx}} \qquad (49)\\ &+ (T_{\text{CTS}} + T_{\text{ack}})P_{\text{rx}},\\ e^i_{\text{rx}} &= \left(\frac{T_{\text{CW}}}{2} + 3T_{\text{SIFS}} + T_{\text{idl}}\right)P_{\text{idl}} + (T_{\text{CTS}} + T_{\text{ack}})P_{\text{tx}} \qquad (50)\\ &+ \left(T_{\text{RTS}} + \frac{P^i}{R}\right)P_{\text{rx}},\\ e_{\text{ov}} &= \frac{T_{\text{CW}}}{2}P_{\text{idl}} + T_{\text{RTS}}P_{\text{rx}}, \qquad (51)\\ e_{\text{idl}} &= T_{\text{idl}}P_{\text{idl}}, \qquad (52)\\ e_{\text{tx,sync}} &= \frac{T_{\text{CW}}}{2}P_{\text{idl}} + T_{\text{hdr}}P_{\text{tx}}, \qquad (53)\\ e_{\text{rx,sync}} &= C\frac{T_{\text{CW}}}{2}P_{\text{idl}} + CT_{\text{hdr}}P_{\text{rx}}, \qquad (54)\\ e_{\text{cs}} &= 0, \qquad (55) \end{aligned}$$

and the total energy spent in the synchronization and being idle in T_{obs} are (note that the other states are calculated in a way similar to the one above in B-MAC):

$$E_{\text{tx,sync}} = \left(F_{\text{sync}} e_{\text{tx,sync}}\right) T_{\text{obs}}, \tag{56}$$

$$E_{\text{rx,sync}} = \left(F_{\text{sync}} e_{\text{rx,sync}}\right) T_{\text{obs}}, \tag{57}$$

$$E_{\text{idl}} = \left(\frac{T_{\text{obs}}}{T_{\text{slot}}}\right) e_{\text{idl}}, \tag{58}$$

$$E_{\text{ctrl}} = E^{l}_{\text{rx,sync}} + E^{l}_{\text{tx,sync}} + E_{\text{idl}}, \tag{59}$$

where T_{slot} denotes the active period schedule of each node in T-MAC.

The total energy consumption in T_{obs} is:

$$E^{d,l}_{T_{\text{obs}}} = E^{l}_{\text{s}} + E^{d,l}_{\text{tx}} + E^{d,l}_{\text{rx}} + E^{d,l}_{\text{ov}} + E_{\text{ctrl}}. \tag{60}$$

5.2. *Globally Synchronized MAC Protocols*

This class of MAC protocols uses topology information for scheduling the medium access in such a way that no two interfering nodes access the channel at the same time. This is achieved by assigning a unique time slot to each node. Thus, it can deliver a good performance when contention level is high. The time slot duration is predetermined and can hold a maximum amount of bytes. Every node can send a packet in its own slot only. In applications with predictable communication patterns, frame-slotted MAC protocols can achieve considerable energy savings by turning off the radio in slots when no data will be received. For this reason, it is worth to evaluate their energy performance in low data rate WMSNs. In the following subsections we model the energy consumption of two frame-slotted MAC protocols: L-MAC and TreeMAC.

5.2.1. *L-MAC*

Lightweight MAC (L-MAC) [13] features a distributed TDMA scheme which organizes time into frames that are divided into N_{slots} slots (see Fig. 8). Each node can send a packet in its own slot and it performs carrier sensing in the remaining ones in order to check for incoming packets. A node has to wait a number of slots (N_{slots}-1) before being able to send the next packet. In every frame, C neighbors are sending a guarded header to mark their occupancy that is overheard by the given node.

The time required to transmit, receive and overhear a packet of class i in L-MAC is:

$$\begin{aligned} T^{i}_{\text{tx}} &= T_{\text{guard}} + T_{\text{hdr}} + \frac{P^{i}}{R}, \\ T^{i}_{\text{rx}} &= \frac{P^{i}}{R}, \\ T_{\text{ov}} &= C\left(\frac{T_{\text{guard}}}{2} + T_{hdr}\right), \\ T_{\text{cs}} &= (N_{\text{slots}} - 1)T_{\text{cs}}, \end{aligned} \tag{61}$$

respectively, where the guard time is given as follows:

$$T_{\text{guard}} = 4\theta T_{\text{frame}}, \text{ where } T_{\text{frame}} = N_{\text{slots}} T_{\text{slot}}.$$

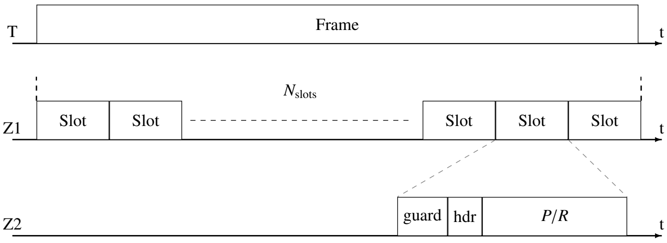


Figure 8: The frame structure in L-MAC.

The energy spent in each mode in a T_{frame} is:

$$e^{i}_{\text{tx}} = T_{\text{guard}} P_{\text{idl}} + \left(T_{\text{hdr}} + \frac{P^{i}}{R}\right) P_{\text{tx}}, \tag{62}$$

$$e^{i}_{\text{rx}} = \frac{P^{i}}{R} P_{\text{rx}}, \tag{63}$$

$$e_{\text{ov}} = C\left(\frac{T_{\text{guard}}}{2} + T_{\text{hdr}}\right) P_{\text{rx}}, \tag{64}$$

$$e_{\text{cs}} = ((N_{\text{slots}} - 1)T_{\text{cs}}) P_{\text{idl}}, \tag{65}$$

$$e_{\text{ctrl}} = 0, \tag{66}$$

and the total energy spent in overhearing and carrier sensing in T_{obs} are (note that the other states are calculated in a similar way to the one in B-MAC):

$$E_{\text{ov}} = \left(\frac{T_{\text{obs}}}{T_{\text{frame}}}\right) e_{\text{ov}}, \tag{67}$$

$$E_{\text{cs}} = \left(\frac{T_{\text{obs}}}{T_{\text{frame}}}\right) e_{\text{cs}}. \tag{68}$$

In a similar way, the total energy consumption in L-MAC in T_{obs} is:

$$E^{d,l}_{T_{\text{obs}}} = E^{l}_{\text{s}} + E^{d,l}_{\text{tx}} + E^{d,l}_{\text{rx}} + E_{\text{ov}} + E_{\text{cs}}. \tag{69}$$

5.2.2. *TreeMAC*

Based on the idea that equal channel access is not fair in the data collection scenario where nodes close to the sink need to forward more data than nodes further away, TreeMAC [14] allows every node to get a number of time slots proportional to its output traffic rate. Such a mechanism is suitable for the network topology mentioned and used in this study. TreeMAC divides each cycle into N_{frames} frames and each frame into three slots (see Fig. 9). By making use of the parent-children relationship, the frame-slot assignment is locally determined and exchanged between parent and children only. A parent determines children frames assignment based on their relative bandwidth demands, and each node calculates the slot assignment based on its hop-count to the sink (i.e., its depth on the tree). By using three slots in each frame, a node can avoid contention with its previous and next hop.

Different from other TDMA-based MAC protocols, the frame-slot assignment in TreeMAC is a two-dimensional

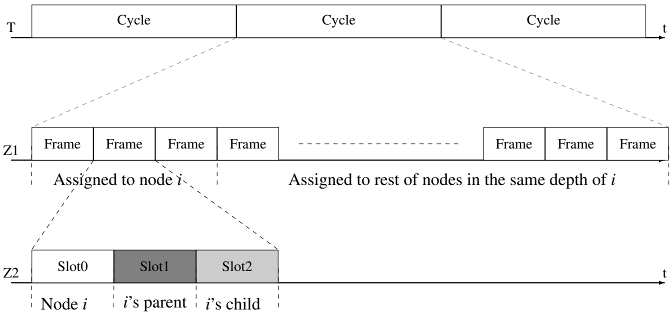


Figure 9: The cycle and frame structures in TreeMAC. Frames are assigned by a parent to its children based on their bandwidth demands, whereas using three slots in each frame is for a node to avoid a contention with its parent and child.

conflict-free sending/receiving and snooping. The frame assignment eliminates the horizontal two-hop interference (i.e., nodes of the same depth on the tree get the same transmission slot but in different frames). The slot assignment eliminates the vertical interference. Given any node, at any time slot, there is at most one active sender in its 1-hop neighborhood (including itself). Each node wakes up in its assigned frames. In its sending slot, it sends the actual payload. In the receiving slot, it performs carrier sensing. TreeMAC requires nodes to update their bandwidth demand T_{bd}, and to send synchronization messages T_{sync*} and schedule updates T_{sch} periodically at different rates (their values are provided in Table 1).

The time required to transmit, receive and overhear a packet of class i in Tree-MAC is:

$$\begin{aligned} T^i_{\text{tx}} &= T_{\text{guard}} + T_{\text{cs}} + T_{\text{hdr}} + \frac{P^i}{R}, \\ T^i_{\text{rx}} &= \frac{P^i}{R}, \\ T_{\text{ov}} &= 2\left(\frac{T_{\text{guard}}}{2} + T_{hdr}\right), \\ T_{\text{cs}} &= T_{\text{cs}}, \end{aligned} \tag{70}$$

respectively. We note that in TreeMAC, a given node overhears only its parent and child in its assigned frames which illustrates why the overhearing time is multiplied by two. In the other frames the node goes back to sleep. It senses the channel in its sending and receiving slots (i.e., when a packet can be received from its child). The guard time in TreeMAC is:

$$T_{\text{guard}} = 4\theta T_{\text{cycle}}.$$

The energy spent in each mode in a T_{frame} is:

$$e^i_{\text{tx}} = (T_{\text{guard}} + T_{\text{cs}})P_{\text{idl}} + \left(T_{\text{hdr}} + \frac{P^i}{R}\right)P_{\text{tx}}, \tag{71}$$

$$e^i_{\text{rx}} = \left(T_{\text{hdr}} + \frac{P^i}{R}\right)P_{\text{rx}}, \tag{72}$$

$$e_{\text{ov}} = 2\left(\frac{T_{\text{guard}}}{2} + T_{\text{hdr}}\right)P_{\text{rx}}, \tag{73}$$

$$e_{\text{cs}} = T_{\text{cs}}P_{\text{idl}}, \tag{74}$$

and the total energy spent in overhearing and carrier sensing in T_{obs} are derived by multiplying the energy spent in each mode, in one frame, by the number of frames assigned to the node in each cycle, and the number of cycles in T_{obs}. It is calculated as follows:

$$E_{\text{ov}} = \left(\frac{N_{\text{frames}}}{N_d}\right)\left(\frac{T_{\text{obs}}}{T_{\text{cycle}}}\right)e_{\text{ov}}, \tag{75}$$

$$E_{\text{cs}} = \left(\frac{N_{\text{frames}}}{N_d}\right)\left(\frac{T_{\text{obs}}}{T_{\text{cycle}}}\right)e_{\text{cs}}, \tag{76}$$

where N_d is the average number of nodes in ring d. Since we are placing nodes strategically in multiple rings, all nodes in the same ring will get an equal number of frames.

The energy spent in synchronization, scheduling, and bandwidth demand updates are:

$$e_{\text{sync*}} = T_{\text{hdr}}P_{\text{tx}} + T_{\text{hdr}}P_{\text{rx}}, \tag{77}$$

$$e_{\text{sch}} = T_{\text{sch}}P_{\text{tx}} + T_{\text{sch}}P_{\text{rx}}, \tag{78}$$

$$e_{\text{bd}} = T_{\text{bd}}P_{\text{tx}} + T_{\text{bd}}P_{\text{rx}}, \tag{79}$$

and the total energy spent in each mode in T_{obs} is:

$$E_{\text{sync*}} = \left(F_{\text{sync*}}e_{\text{sync*}}\right)T_{\text{obs}}, \tag{80}$$

$$E_{\text{sch}} = (F_{\text{sch}}e_{\text{sch}})\,T_{\text{obs}}, \tag{81}$$

$$E_{\text{bd}} = (F_{\text{bd}}e_{\text{bd}})\,T_{\text{obs}}, \tag{82}$$

$$E_{\text{ctrl}} = E_{\text{sync*}} + E_{\text{sch}} + E_{\text{bd}}. \tag{83}$$

The total energy consumption in TreeMAC in T_{obs} is:

$$E^{d,l}_{T_{\text{obs}}} = E^l_{\text{s}} + E^{d,l}_{\text{tx}} + E^{d,l}_{\text{rx}} + E_{\text{ov}} + E_{\text{cs}} + E_{\text{ctrl}}. \tag{84}$$

6. Numerical Evaluation

In this section, we conduct a numerical evaluation of the energy consumption of the MAC protocols using the developed multi-class traffic model presented in Section 3 and the energy models in Sections 4 and 5. First, we start by investigating the traffic load conditions in each MAC protocol, which must be added to the network in order to make collisions negligible. Then, we illustrate how those conditions are tightly related to the sampling rates of nodes, the size of multimedia samples, and some network topology parameters such as the number of rings, the number of nodes in each ring and the density of MMSs. After that, we investigate the energy consumption of the MAC protocols under those traffic load conditions.

The topology considered in the numerical evaluation is a multi-ring topology (D, C), where we have L=2 classes of sensors, MMSs -with density p_m- that sample the environment at a rate F^{mms}_s, and SSs that sample the environment at a rate F^{ss}_s. The size of the captured image depends on the phenomena being monitored. Except where otherwise stated, we assume an image size of 10KB and a multimedia payload of size P=512B which gives us M=20 payloads per image.

6.1. Parameters constraints

In order to make collisions negligible, we present some safeguarding conditions on the amount of traffic flowing through the network against any improper selection of MACs parameters. It is worth noting that each category of MAC protocols has a different traffic boundary condition according to its medium access strategy. However, in all MAC protocols this will be done by adding the condition to the busiest nodes in the network, which have the most packets to send (i.e., nodes close to the sink in ring d=1). The constraints below are derived in a similar way as in [4], and the thresholds are assumed to be the same.

In the case of asynchronous MAC protocols, we derive a general condition which guarantees that the maximum traffic load transmitted by all nodes in d=1, of any class l, to the sink (in d=0) does not exceed 25% of the channel bandwidth. This can be described by the following equation:

$$\sum_{l}^{L} I_0^l F_{\text{out}}^{1,l} M^l T_{\text{tx}} < \frac{1}{4}, \tag{85}$$

where I_0^l is the sink's average number of input links of class l. This condition can be adapted to each asynchronous MAC protocol according to the packet transmission time T_{tx} of each one.

In the case of locally synchronized MAC protocols, such as T-MAC, the total traffic transmitted by all nodes in d=1 during the active period (T_{slot}) should not exceed 25% of the channel bandwidth. This can be described as follows:

$$\sum_{l}^{L} I_0^l F_{\text{out}}^{1,l} M^l T_{\text{slot}} < \frac{1}{4}. \tag{86}$$

In globally synchronized MAC protocols, collisions is avoided since every node has a unique transmission slot. However, we set a bound on the maximum traffic transmitted by bottleneck nodes in d=1 in order to avoid long queuing delays.

In L-MAC we have:

$$\sum_{l}^{L} I_0^l F_{\text{out}}^{1,l} M^l T_{\text{frame}} < \frac{1}{2}. \tag{87}$$

In the case of TreeMAC, the threshold is calculated as follows:

$$\sum_{l}^{L} I_0^l F_{\text{out}}^{1,l} M^l T_{\text{cycle}} < \frac{1}{2}. \tag{88}$$

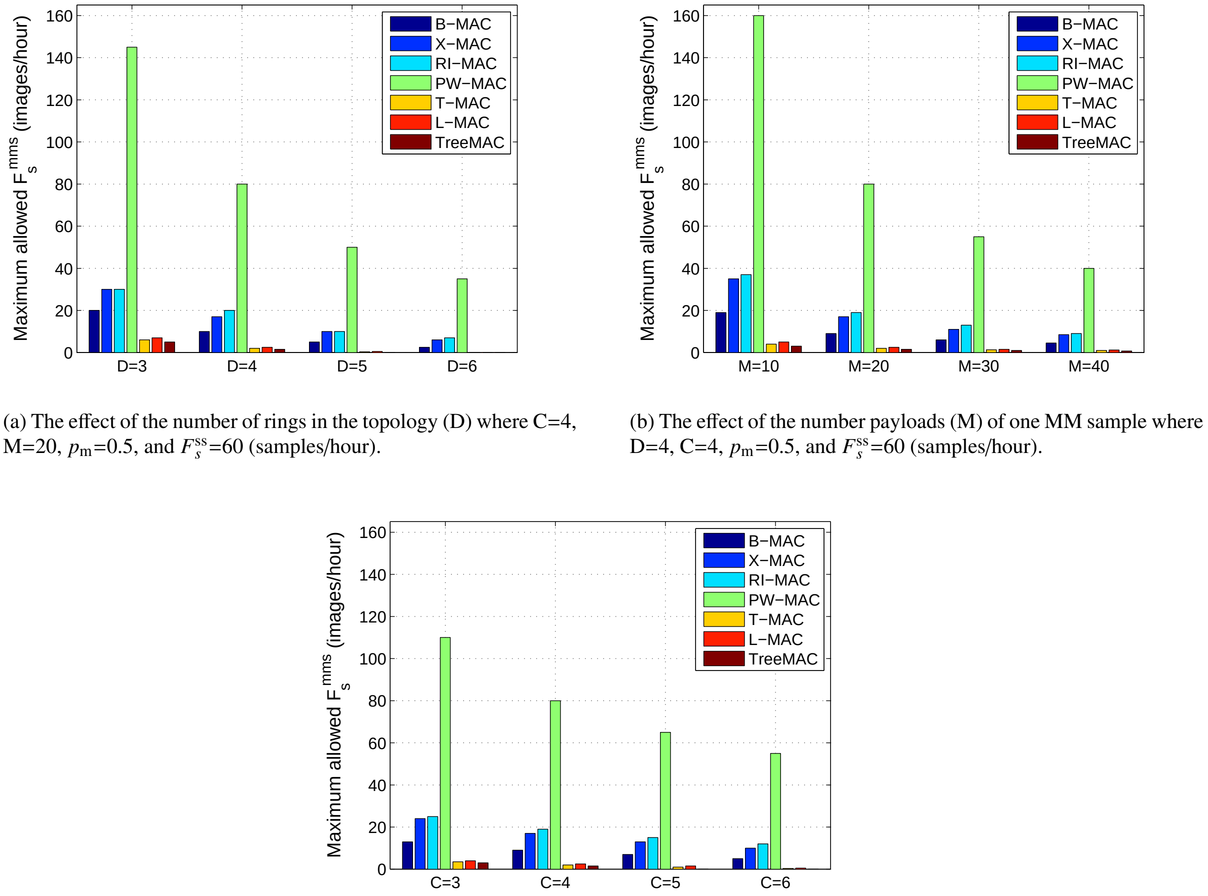
Setting a bound on the amount of traffic flowing through the network implies that the sampling rate of MMSs can not be increased more than a certain value. This also imposes other constraints on some network topology parameters such as the number of rings, the number of nodes in each ring, and the density of MMSs, because the output traffic increases by increasing those parameters. In Fig. 10, we show how the network topology parameters and the size of multimedia sample directly affect the maximum value of MMSs' sampling rate (F_s^{mms}) allowed for each MAC protocol in order to make collisions negligible. This is calculated based on the aggregated output traffic sent by all busy nodes in ring d=1 satisfying the conditions above. In Fig. 10, we assume that SSs sample the environment at a fixed sampling rate F_s^{ss}=60 (samples/hour) and that the density of MMSs is constant p_{m}=50%. For instance, Fig. 10 (a) shows that we can not increase F_s^{mms} in B-MAC more than 20 (samples/hour) when D=3, while it is possible to increase F_s^{mms} in PW-MAC up to 145 (samples/hour) under the same network configurations and size of MM sample. On the other hand, Fig. 10 show that under the same configurations (i.e., F_s^{ss}=60 (samples/hour) and p_{m}=50%) synchronous MAC protocols can not be used in a network with more than D=4 rings or more than C=4 nodes in the first ring, and the maximum allowed F_s^{mms} in the best scenario does not exceed 6, 7 and 5 (samples/hour) for T-MAC, L-MAC and TreeMAC, respectively, when D=3. Hence, it can be inferred from the figure that asynchronous MAC protocols give better flexibility to the range of allowed sampling rates than synchronous MAC protocols for different network configurations. In particular, PW-MAC allows MMSs to sampling the environment at relatively high rates.

6.2. Parameters study

In this section, the energy consumption of the MAC protocols is evaluated. First, we investigate the energy consumption of sender-initiated MAC protocols (B-MAC and X-MAC) and receiver-initiated MAC protocols (RI-MAC and PW-MAC). Then, we analyze the energy consumption of synchronous MAC protocols from the two categories: i) locally synchronized (T-MAC), and ii) globally synchronized (L-MAC and TreeMAC). Finally, we compare the different categories of MAC protocols, and recommend the network settings and MAC parameters suitable for each MAC protocol. The topology considered in this experiment is a multi-ring topology (D=4, C=4), resulting in a network of 64 nodes. Our goal is to assess the energy consumption of the MAC protocols under different values of F_s^{mms}, polling time intervals T_{w} (i.e., in case of asynchronous MACs), and densities of MMSs p_{m}.

We focus our attention on the energy consumption of nodes close to sink (i.e., in ring d=1) since these nodes always have more traffic to send/receive than all other nodes. A node in ring d=1 has to convey its own traffic plus the whole traffic from outer rings. The traffic and radio parameters, as well as the specific parameters for all the MAC protocols are provided in Table 1. The radio parameters are taken from the datasheet of MICAz platform [31] and the Chipcon CC2420 radio [32].

Fig. 11 compares the energy consumption of the selected asynchronous MAC protocols in a WMSN with sampling rates F_s^{ss}=60 (samples/hour) and F_s^{mms} in the interval [1/96,60] (images/hour), and for two different polling period (T_{w}) values: 0.05 and 0.2 seconds. Based on the parameter constraints presented in Section 6.1, the energy consumption of every MAC protocols is only plotted in its allowed interval of F_s^{mms}. From Fig. 11, it can be noticed that in the entire allowed sampling rate



(a) The effect of the number of rings in the topology (D) where C=4, M=20, p_m=0.5, and F_s^{ss}=60 (samples/hour).

(b) The effect of the number payloads (M) of one MM sample where D=4, C=4, p_m=0.5, and F_s^{ss}=60 (samples/hour).

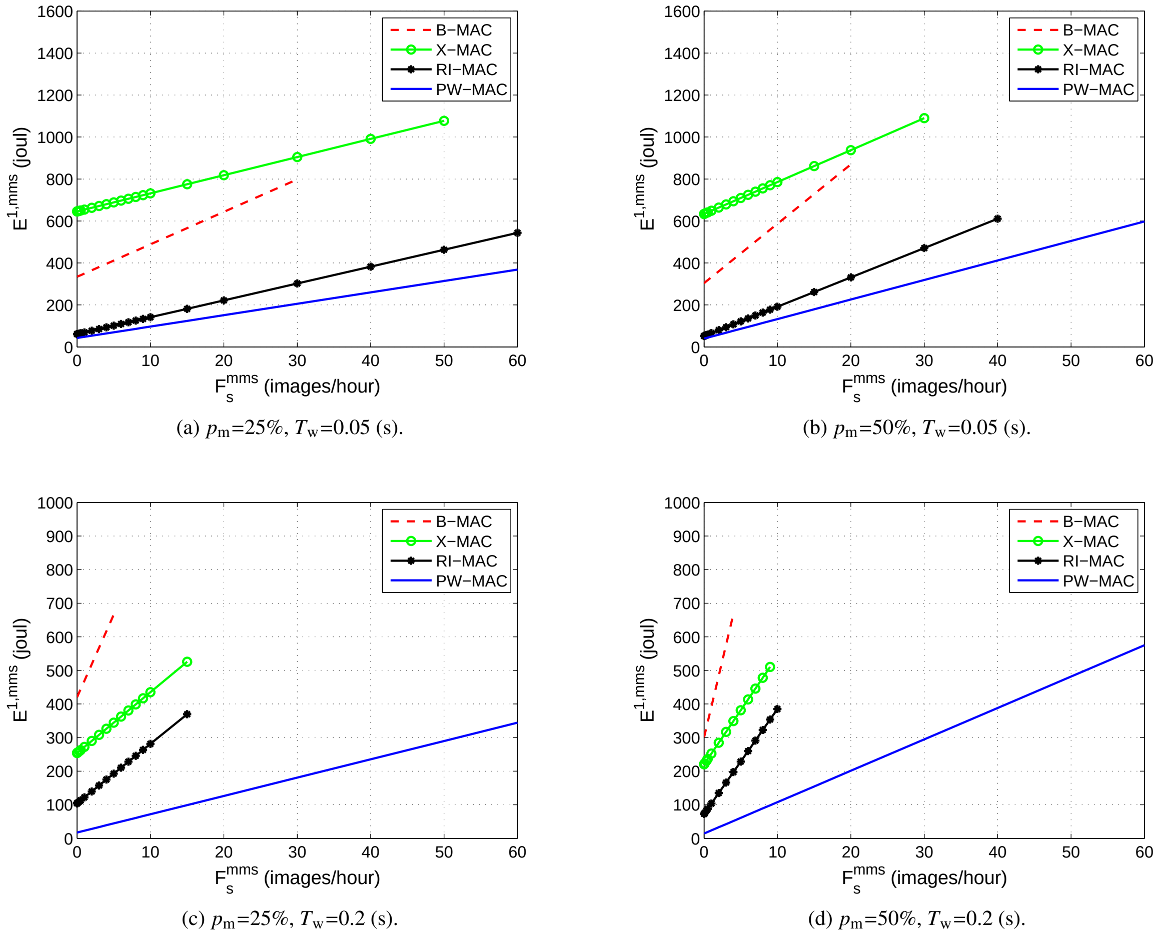
(c) The effect of the number of nodes in the first ring (C) where D=4, M=20, p_m=0.5, and F_s^{ss}=60 (samples/hour).

Figure 10: The effect of network parameters and the size of the MM sample on the maximum allowed sampling rate of MMSs (F_s^{mms}).

interval and for short and long polling periods, the energy consumption of receiver-initiated MAC protocols (i.e., RI-MAC and PW-MAC) is always lower than the sender-initiated ones (i.e., B-MAC, X-MAC). This is because the sender-initiated MAC protocols adopt the duty cycling technique where a node sends a long preamble to ensure communication with its intended receiver. This long preamble is a source of energy consumption in sending, receiving and overhearing (see Section 4). Besides, it results in a longer transmission time (T_{tx}) which limits the maximum allowed sampling rate of MMSs (see Eq. 85). In receiver-initiated MAC protocols, the time during which a sender and its intended receiver are occupying the channel to be able to communicate is reduced, and a sending node does not start transmitting until the receiver is ready to receive. In PW-MAC, a sender wakes up just before its intended receiver which illustrates why PW-MAC consumes the least amount of energy between the asynchronous MACs. This mechanism of PW-MAC also reduces the transmission time of the sender (T_{tx}) and allows for a wider range of F_s^{mms}.

At very low sampling rates, B-MAC achieves lower energy consumption than X-MAC when the polling period T_{w} is short (Fig. 11(a) and Fig. 11(b)). This is because X-MAC has a longer carrier sensing period ($T_{\text{cs}} + T_{\text{ea}}$), and since the polling period is short, nodes have to wake up and perform carrier sensing more often, and as consequence more energy is consumed. However, nodes in B-MAC consume the largest amount of energy as the polling period (T_{w}) gets longer and/or the sampling rate increases. At higher sampling rates, the generated traffic is higher and the carrier sensing is less frequent. In these cases, X-MAC outperforms B-MAC since it uses short preamble bursts which reduces the preamble length to the half on average.

The effect of the density of MMSs p_{m} on the energy consumption of MACs (i.e., p_{m}=0.25 and p_{m}=0.5) is also presented in Fig. 11. In both cases, B-MAC still outperforms X-MAC at low sampling rates when the polling period is short. It can be observed that in the four mentioned scenarios, receiver-initiated MAC protocols have better energy performance and allow for a wider range of sampling rates. In particular, PW-MAC allows MMSs to sample the environment at F_s^{mms} up to 80 (images/hour) under the same network configuration (i.e., C=4, D=4, M=20, and F_s^{ss}=60 (samples/hour)) and when the density of MMSs p_{m} is 50% (see Fig. 10 (a)).



(a) p_{m}=25%, T_{w}=0.05 (s).

(b) p_{m}=50%, T_{w}=0.05 (s).

(c) p_{m}=25%, T_{w}=0.2 (s).

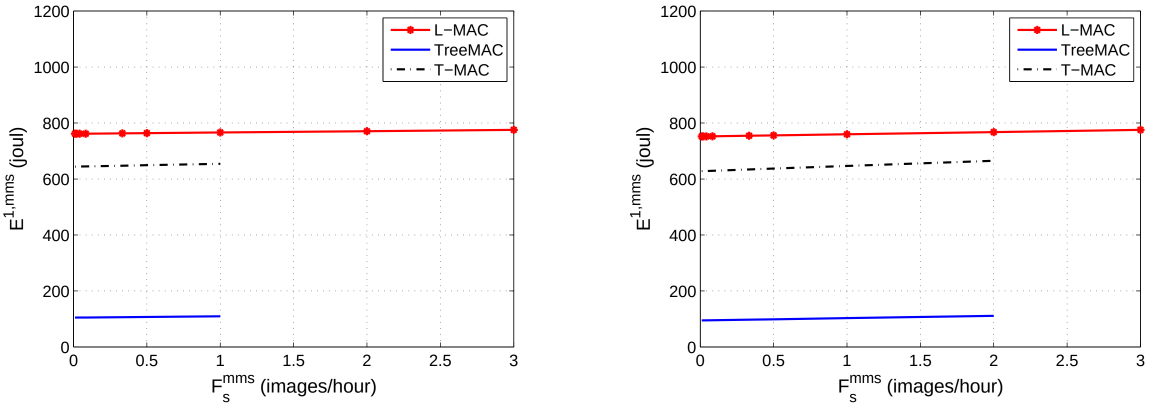
(d) p_{m}=50%, T_{w}=0.2 (s).

Figure 11: The energy consumption of B-MAC, X-MAC, RI-MAC and PW-MAC during $T_{\mathrm{obs}} = 24$ (hour) in a WMSNs of 64 Nodes and for different polling periods and densities of MMSs.

In Fig. 12, we compare the energy consumption of the synchronous MAC protocols modeled in Sections 5. We use the same network configurations as in the previous experiments except that in these protocols it is not allowed to increase F_s^{mms} more than 3 (images/hour), otherwise the traffic load constraints at the bottleneck nodes can not be satisfied (see Section 6.1). This is because the longer duration of T_{slot}, T_{frame}, and T_{cycle}, in T-MAC, L-MAC and TreeMAC, respectively, than T_{x} in asynchronous MAC protocols. From this figure, we notice that for different densities of MMSs, L-MAC and T-MAC consume higher energy than TreeMAC. A node in L-MAC needs to sense the channel in each slot -except the one it owns- during the whole observation time (T_{obs}=24 hours), which is the major source of energy consumption in L-MAC. In T-MAC, a huge amount of energy is spent in the idle mode during T_{obs}. On the contrary, TreeMAC achieves a lower energy consumption since it has a predetermined structure of frames/slots assigned to nodes. Nodes wake up only in their assigned frames without the need of carrier sensing in each frame/slot. Besides, this structure limits overhearing to the assigned frames/slots only, which also helps in reducing the energy consumption. In synchronous MACs there is no need for polling/sensing the channel or sending beacons periodically. However, this comes at the cost of an extra synchronization overhead and a very limited allowed range of sampling rates. Therefore, the usage of these protocols is limited to WMSNs working at very low data rates.

Fig. 12 also shows that for a higher density of MMSs (p_{m}= 50%), the sampling rate of MMSs (F_s^{mms}) can be increased in T-MAC and TreeMAC up to 2 (images/hour). The reason is that the sampling rate of SSs (F_s^{ss}) is constant and set to be 60 (samples/hour). Thus, at very low data rate of MMSs, the output traffic generated from SSs is higher than MMSs. Therefor, when p_{m} is low the total output traffic generated at bottleneck nodes is higher and it decreases as the density of MMSs increases.

Table 2 overviews the scenarios in which each MAC protocol is recommended. It can be concluded that receiver-imitated MAC protocols are suitable for this type of networks, allowing for a wider range of sampling rates, while in synchronous MAC protocols only TreeMAC is recommended and for WMSNs with very low sampling rates.



(a) $p_{\rm m}$=25%.

(b) $p_{\rm m}$=50%.

Figure 12: The energy consumption of L-MAC, TreeMAC and T-MAC during an observation time $T_{\rm obs}$ = 24 (hour) in a WMSNs of 64 Nodes and for different densities of MMSs. At $F_s^{\rm mms}$= 2 (images/hour), when $p_{\rm m}$=25 % we have in total 3520 (packets/hour) which causes Eq. (88) in TreeMAC, for instance, to be higher than 50%, but when $p_{\rm m}$=50 % we have 3200 (packets/hour) and the condition is fulfilled.

Table 2: The recommended MAC protocols for each scenario.

	Asynchronous				Synchronous		
	Receiver-initiated		Sender-initiated				
	B-MAC	X-MAC	RI-MAC	PW-MAC	T-MAC	L-MAC	TreeMAC
Very low sampling rate $F_s^{\rm mms}$	✓		✓	✓			✓
Low sampling rate $F_s^{\rm mms}$		✓	✓	✓			
Low density of MMSs	✓	✓	✓	✓			✓
High density of MMSs		✓	✓	✓			✓
Long polling period				✓	-	-	-
Short polling period	✓		✓	✓	-	-	-

6.3. *Application scenarios*

In this section we use the multi-class traffic model to assess the performance of MAC protocols in different WMSNs application scenarios related to Smart Cities and environment monitoring. We distinguish between two groups of application scenarios: i) indoor scenarios such as smart buildings, houses, and stables, and ii) outdoor scenarios such as urban resilience applications and smart farms/gardens. In each scenario, we integrate two types of sensors (i.e., multimedia and scalar) each with a different sampling rate. The configurations of the selected WMSN in each application are listed in Table 3 and the MAC protocols under these configurations have been verified to satisfy the traffic load conditions in Section 6.1. The energy consumption of MAC protocols in each application is shown in Fig. 13.

6.3.1. *Indoor applications*

In indoor applications, such as smart buildings/houses, various type of sensors and electronic devices are interconnected through a communication network to monitor and control remotely different phenomenons inside the place such as temperature and humidity, lighting, occupancy and movement, kids, plants and pets situation, products and warehouses in shopping centers, among others. In the following subsection, we consider two application scenarios that deploy low data rate WMSNs.

6.3.1.1 Smart building/house

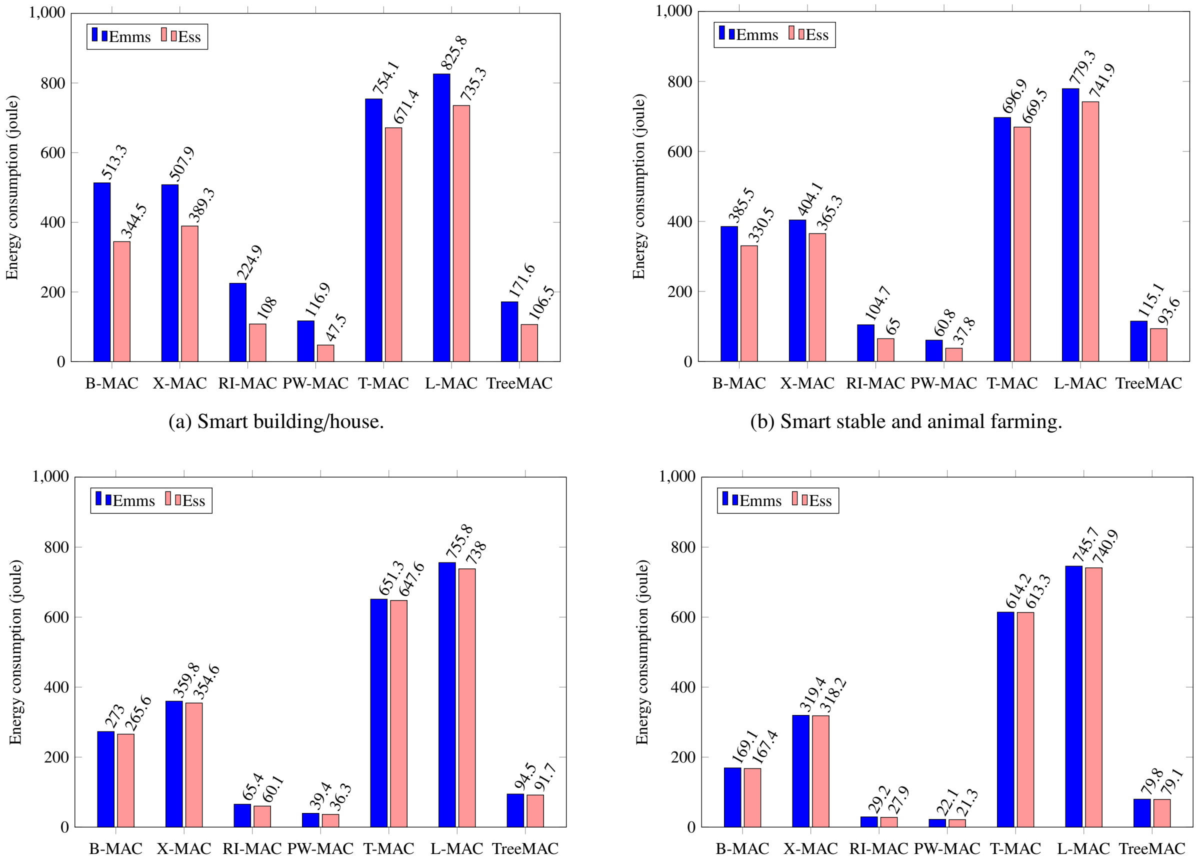
Intelligent buildings, including smart homes and office spaces, have been extensively studied in the literature [33, 34, 35, 36, 37]. All of these projects and studies make extensive use of sensors to monitor objects and spaces inside and around the house/building giving inhabitants the ability to remotely control them. In this scenario, we deploy a WMSN with 16 sensors arranged in D=2 rings and each node with C=4 neighbors.

6.3.1.2 Smart stable and animal farming

We deploy a WMSN of 24 (D=2, C=6) sensors to remotely monitor animals in a stable and in a small animal farm. SSs can monitor the temperature, humidity, door and window open/close status, among others, while MMSs periodically send images about the animals' situation inside and around the stable. In particular, deploying such a WMSN to monitor the animals' situation can help prevent illness and theft, and allows the farmer to remotely keep an eye on the animals during days and nights (e.g., the sensor network deployed for monitoring horses and equine farm management in [38, 39]).

6.3.2. *Outdoor applications*

In this type of applications we consider some applications for low data rate WMSNs where the multimedia and scalar sensors can be deployed together to monitor and control different



(a) Smart building/house.

(b) Smart stable and animal farming.

(c) Urban/territorial resilience.

(d) Smart agriculture.

Figure 13: The energy consumption of MAC protocols in the selected indoor and outdoor application scenarios.

Table 3: The configurations of the WMSN of each scenario (M=20 and T_w= 0.1 (s) in all scenarios).

	Number of sensors	Density of MMSs	Sampling frequency (samples/hour)	
	N	p_m	F_s^{ss}	F_s^{mms}
Smart building/house	16	20%	60	30
Smart stable and animal farming	24	40%	30	15
Urban/territorial resilience	80	60%	4	2
Smart agriculture	150	30%	2	0.5

phenomenons in the city/territory such as structural health (e.g., buildings, bridges and historical monuments), noise and sound monitoring in bar zones and centric areas, rivers and dams situation, ambient control, among others. Outdoor applications also includes smart farms/gardens.

6.3.2.1 Urban/territorial resilience

As we mentioned above, WMSNs can be deployed in urban management systems to monitor and observe the territory, and prevent the disruption of essential city services (e.g., La Garrotxa Urban Resilience project in Catalunya [40]). In this scenario we deploy a WMSN of 80 sensors (D=4, C=5). Since the phenomenas being monitored (e.g., noise and sounds, ambient control, structural health, among others) are non-time critical, we choose low sampling rates for both SSs and MMSs (see Table 3).

6.3.2.2 Smart farm and agriculture

The use of sensor networks in smart agriculture [41] is very promising as multiple environmental parameters can be monitored. This includes a wide range of applications, from crops status and growing conditions analysis to weather observation, such as vineyards, tropical fruits and herbs that are sensitive to cold, where a slight change in climate can affect the final outcome. All of this information can also help to determine the optimum conditions for crops, by keeping an archive of images and comparing them with the figures and images obtained during the best harvests, which leads to better productivity, costs reduction, and improved management (e.g., the Rias Baixas Smart Viticulture project in Galicia [42]). In this scenario we deploy a WMSN of 150 (D=5, C=6) sensors with very low sampling rates.

6.3.3. Discussion

From Fig. 13 we can see that in all the selected scenarios, PW-MAC, RI-MAC and TreeMAC show a low energy consumption performance for both types of sensors. On the contrary, L-MAC and T-MAC consume the highest amount of energy and they are not recommended for this kind of networks and applications. In sender-initiated asynchronous MAC protocols, X-MAC has a better energy performance than B-MAC in smart buildings/house applications since the sampling rates in these applications are comparatively high (F_s^{ss}= 60, F_s^{mms}=30 (samples/hour)). In the mentioned outdoor applications, sensors sample the environment at very low sampling rates which illustrates why B-MAC has a close energy consumption performance to X-MAC.

7. Conclusions

In this paper we derived a multi-class traffic model and used it to analyze the energy consumption of some recent and baseline MAC protocols in low data rate delay-tolerant WMSNs. We modeled the energy consumption of MAC protocols from different categories including asynchronous (sender-initiated and receiver-initiated), and synchronous (locally and globally) MAC protocols. The derived models allow us to compare the performance of MAC protocols as a function of the network topology, the density of multimedia nodes and the sampling rates.

From the numerical analysis, it is noticed that in the asynchronous MAC protocols category, receiver-initiated MAC protocols outperform sender-initiated ones. In particular, PW-MAC shows the lowest energy consumption between the selected asynchronous MAC protocols and it can be used in WMSNs with a wide range of sampling rates. Regarding synchronous MAC protocols, results also show that they are only suitable for WMSNs when the data rates are very low. In that situation, TreeMAC is the one that offers a lower energy consumption.

From the application scenarios we studied, it can be observed that some of the existing MAC protocols in WSNs are suitable for non-streaming non-time critical WMSNs without the need for additional control mechanisms like streaming and QoS-aware MAC protocols. However the selection of the MAC protocol and its parameters strongly depends on the application scenario.

To conclude, this paper offers a mathematical modeling and a numerical evaluation of MAC protocols in WMSNs that we believe it fills a need in the current literature and gives researchers a very clear view of the energy consumption of some recent MAC protocols in WMSNs and Smart Cities application scenarios. Having these models and results may enable future research efforts to improve upon the energy efficiency of the current MAC protocols, and help users to choose the most adequate one for each scenario.

8. Acknowledgements

This work was partially supported by projects TIN2013-47272-C2-2, TEC2012- 32354, SGR-2014-1173 and SGR-2014-881.

9. References

[1] T. C. Minh, B. Bellalta, S. Oechsner, R. Liao, M. Oliver, Managing heterogeneous WSNs in smart cities: Challenges and requirements, arXiv preprint arXiv:1310.6901.

[2] I. F. Akyildiz, T. Melodia, K. R. Chowdhury, A survey on wireless multimedia sensor networks, Computer networks 51 (4) (2007) 921–960.

[3] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, N. Xi, The evolution of MAC protocols in wireless sensor networks: A survey, Communications Surveys & Tutorials, IEEE 15 (1) (2013) 101–120.

[4] K. Langendoen, A. Meier, Analyzing MAC protocols for low data-rate applications, ACM Transactions on Sensor Networks (TOSN) 7 (2) (2010) 19.

[5] C. Cano, B. Bellalta, M. Oliver, Receiver-initiated vs. short-preamble burst MAC approaches for multi-channel wireless sensor networks, in: Information and Communication Technologies, Springer, 2012, pp. 23–32.

[6] C. Cano, B. Bellalta, A. Sfairopoulou, M. Oliver, Low energy operation in WSNs: A survey of preamble sampling MAC protocols, Computer Networks 55 (15) (2011) 3351–3363.

[7] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM, 2004, pp. 95–107.

[8] M. Buettner, G. V. Yee, E. Anderson, R. Han, X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks, in: Proceedings of the 4th international conference on Embedded networked sensor systems, ACM, 2006, pp. 307–320.

[9] Y. Sun, O. Gurewitz, D. B. Johnson, RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks, in: Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM, 2008, pp. 1–14.

[10] L. Tang, Y. Sun, O. Gurewitz, D. B. Johnson, PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 1305–1313.

[11] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, Networking, IEEE/ACM Transactions on 12 (3) (2004) 493–506.

[12] T. Van Dam, K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the 1st international conference on Embedded networked sensor systems, ACM, 2003, pp. 171–180.

[13] L. F. van Hoesel, P. J. Havinga, A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches.

[14] W.-Z. Song, R. Huang, B. Shirazi, R. LaHusen, TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks, Pervasive and Mobile Computing 5 (6) (2009) 750–765.

[15] Y. Liu, I. Elhanany, H. Qi, An energy-efficient QoS-aware media access control protocol for wireless sensor networks, in: Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on, IEEE, 2005, pp. 3–pp.

[16] Z. Liu, I. Elhanany, RL-MAC: A QoS-aware reinforcement learning based MAC protocol for wireless sensor networks, in: Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on, IEEE, 2006, pp. 768–773.

[17] H. Kim, S.-G. Min, Priority-based QoS MAC protocol for wireless sensor networks, in: Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on, 2009, pp. 1–8.

[18] B. Nefzi, Y.-Q. Song, QoS for wireless sensor networks: Enabling service differentiation at the MAC sub-layer using CoSenS, Ad Hoc Networks 10 (4) (2012) 680–695.

[19] M. A. Yigitel, O. D. Incel, C. Ersoy, QoS-aware MAC protocols for wireless sensor networks: A survey, Computer Networks 55 (8) (2011) 1982–2004.

[20] M. Cesana, A. Redondi, N. Tiglao, A. Grilo, J. M. Barcelo-Ordinas, M. Alaei, P. Todorova, Real-time multimedia monitoring in large-scale wireless multimedia sensor networks: Research challenges, in: Next Generation Internet (NGI), 2012 8th EURO-NGI Conference on, IEEE, 2012, pp. 79–86.

[21] N. Saxena, A. Roy, J. Shin, Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks, Computer Networks 52 (13) (2008) 2532–2542.

[22] M. A. Yigitel, O. D. Incel, C. Ersoy, Design and implementation of a QoS-aware MAC protocol for wireless multimedia sensor networks, Computer Communications 34 (16) (2011) 1991–2001.

[23] H. Karl, A. Willig, Protocols and architectures for wireless sensor networks, John Wiley & Sons, 2007.

[24] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, M. Srivastava, Cyclops: In situ image sensing and interpretation in wireless sensor networks, in: Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM, 2005, pp. 192–204.

[25] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu, Senseye: A multi-tier camera sensor network, in: Proceedings of the 13th annual ACM international conference on Multimedia, ACM, 2005, pp. 229–238.

[26] A. Rowe, D. Goel, R. Rajkumar, Firefly mosaic: A vision-enabled wireless sensor networking system, in: Real-time systems symposium, 2007. RTSS 2007. 28th IEEE international, IEEE, 2007, pp. 459–468.

[27] T. Teixeira, E. Culurciello, J. H. Park, D. Lymberopoulos, A. Barton-Sweeney, A. Savvides, Address-event imagers for sensor networks: Evaluation and modeling, in: Proceedings of the 5th international conference on Information processing in sensor networks, ACM, 2006, pp. 458–466.

[28] R. W. Floyd, Algorithm 97: Shortest path, Communications of the ACM 5 (6) (1962) 345.

[29] Texas instruments TMP103 data sheet.
URL http://www.ti.com

[30] M. Alaei, J. M. Barcelo-Ordinas, A collaborative node management scheme for energy-efficient monitoring in wireless multimedia sensor networks, Wireless networks 19 (5) (2013) 639–659.

[31] Crossbow micaz motes.
URL http://xbow.com

[32] Chipcon inc. CC2420 data sheet.
URL http://www.chipcon.com

[33] M. Mozer, R. Dodier, D. Miller, M. Anderson, J. Anderson, D. Bertini, M. Bronder, M. Colagrosso, R. Cruickshank, B. Daugherty, et al., The adaptive house, in: IEE Seminar Digests, Vol. 11059, IET, 2005, pp. v1–39.

[34] Aware home research initiative - Georgia Institute of Technology.
URL http://www.awarehome.gatech.edu/

[35] J. Hsu, P. Mohan, X. Jiang, J. Ortiz, S. Shankar, S. Dawson-Haggerty, D. Culler, Hbci: Human-building-computer interaction, in: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, ACM, 2010, pp. 55–60.

[36] H. Lu, N. Lane, S. Eisenman, A. Campbell, Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones, in: Intl. Workshop on Mobile Devices and Urban Sensing, 2008.

[37] H. Ghayvat, S. Mukhopadhyay, X. Gui, N. Suryadevara, WSN-and IoT-based smart homes and their extension to smart buildings, Sensors 15 (5) (2015) 10350–10379.

[38] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, Wireless sensor networks for habitat monitoring, in: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, ACM, 2002, pp. 88–97.

[39] Monitoring horses and equine facility management.
URL http://www.libelium.com/smart-farming-monitoring-horses-equine-facility-management-waspmote/

[40] Urban resilience in la Garrotxa, Catalunya.
URL http://www.libelium.com/smart-city-urban-resilience-smart-environment/

[41] J. Li, E. Xu, Development on smart agriculture by wireless sensor networks, in: Proceedings of 2014 1st International Conference on Industrial Economics and Industrial Security, Springer, 2015, pp. 41–47.

[42] Smart viticulture in the Rias Baixas, Galicia.
URL http://www.libelium.com/sensors-mag-smart-viticulture-project-in-spain-uses-sensor-devices-to-harvest-healthier-more-abundant-grapes-for-coveted-albarino-wines/