A Low Power Listening with Wake up after Transmissions MAC Protocol for WSNs

Cristina Cano

TESI DOCTORAL UPF / 2010

DIRECTORS DE LA TESI

Dr. Boris Bellalta
Dr. Miquel Oliver

(Departament de Tecnologies de la Informació i les Comunicacions)

UNIVERSITAT
POMPEU FABRA

To my dear Sergio
and my mother.

Acknowledgements

First of all I want to sincerely thank Boris Bellalta. His continuous encouragement, complete dedication and expert guiding have made this work possible. I am glad of having the best advisor I could ever had. I also want to thank Miquel Oliver for giving me his support from the very beginning. Also, specially thanks to my colleagues that have become dear friends: Anna Sfairopoulou, Verónica Moreno, Jaume Barceló, Michail Batikas and Eduard Bonada. Thank you for your help either related to research, technological, teaching or life issues. Your support and companionship have been a huge motivation for me.

I also want to kindly thank Andrés Cisneros, José Luís Sánchez and Víctor Valls. Your incredible dedication has been of great help. I also want to mention my current colleagues at the office: Ruizhi Liao, Trang Cao Minh, Sougata Pal, Manuel Palacín and Albert Domingo, without you all the office will be not so funny.

Additionally, I want to acknowledge the anonymous reviewers of the articles and also the people that kindly accepted to review them. Their valuable comments have substantially improved the quality of the work done.

Finally, I want to particularly thank my mother for being always so proud of me without even caring about a reason. And also my dear Sergio, that has patiently stood with me (that I know it is not an easy task) in the good but specially in the bad moments.

Abstract

In the last few years Wireless Sensor Networks (WSNs) have become an interesting field of research mainly due to the challenges and constraints of their design but also motivated by the broad range of potential applications they can provide. One of the most important constraints is the limited energy resources of the sensor nodes. These limited resources completely influence the design of the protocol stack, and specially the Medium Access Control (MAC) layer, as it is the responsible of controlling the transceiver, that is the most consuming component of a sensor node. The common technique to reduce the energy consumption is to periodically put the transceiver into sleep mode. In sleep mode a sensor node is not able to send or receive data. For this reason it becomes necessary to implement a technique that assures that a node will be awake to receive a message. Among the several MAC protocols designed for WSNs, preamble sampling MAC protocols, that rely on sending a long preamble to overlap with the channel sampling time of the receiver, provide some characteristics that make them specially appealing for WSNs. Specifically, preamble sampling MAC protocols are able to work without synchronism, extremely reduce the energy consumption of sensor nodes and are considerably simple. However, current preamble sampling MAC protocols are not able to cope with increases in the network load that can happen in a WSN, for instance, due to event detection. When the traffic load starts to increase the collision probability becomes noticeable. Moreover, the long preamble transmission makes the collision duration considerably long, specially when there are hidden terminals competing for the channel. In this thesis the aforementioned limitations of preamble sampling are studied and quantified in different scenarios and under a broad range of traffic loads. A new approach, called Low power listening with Wake up after Transmissions MAC (LWT-MAC), has been designed with the goal to overcome preamble sampling limitations while maintaining its reduced energy consumption and simplicity. The new MAC protocol has been evaluated in single hop and multihop WSNs, under different traffic profiles (including event-based messages) and loads. Results have shown that the LWT-MAC protocol is able to significantly improve the performance of WSNs, improving the throughput and reducing the delay but also providing better collective Quality of Service (QoS). Collective QoS is used in WSNs to refer to the QoS observed by the set of messages belonging to a specific event.

Resum

En els darrers anys les xarxes de sensors sense fils han esdevingut una interessant àrea de recerca degut principalment als reptes que presenta el seu disseny però també degut a la gran quantitat d'aplicacions potencials que poden proporcionar. Un dels principals problemes d'aquestes xarxes és la limitació en els recursos energètics dels nodes sensors, cosa que afecta de forma molt marcada al disseny de la pila de protocols. Per aquesta raó, el nivell Medium Access Control (MAC), degut a què és el responsable de controlar la ràdio, el component de major consum energètic d'un node sensor, té un paper molt important. El mecanisme més utilitzat per tal de reduir el consum d'energia és posar la ràdio en mode *sleep* de forma periòdica. En aquest mode un node sensor és incapaç de rebre o enviar paquets, per tant es fa necessària la implementació d'una tècnica per tal d'assegurar que un node es troba despert en el moment d'enviar-li un paquet. D'entre els diferents protocols MAC dissenyats per xarxes de sensors, el protocol MAC *preamble sampling*, que es basa en l'enviament d'un preamble llarg que es solapa amb el temps de mostreig del canal del receptor, proporciona una sèrie de característiques que el fan especialment interessant per a les xarxes de sensors. Aquesta tècnica és capaç de treballar sense necessitat de sincronisme, amb un reduït consum d'energia i de forma molt simple. Per contra, no és capaç de fer front a un augment de la càrrega de tràfic de la xarxa que es pot produir, per exemple, degut a la detecció d'un determinat esdeveniment. Quan la càrrega de tràfic augmenta la probabilitat de col·lisió també s'incrementa. A més, degut a la transmissió del preamble llarg, la duració de les col·lisions és considerablement gran, especialment quan hi han terminals ocults intentant accedir al medi. En aquesta tesi s'estudien les limitacions del protocol MAC *preamble sampling* en diferents escenaris i sota diferents càrregues de tràfic. A més, s'ha dissenyat un nou protocol, anomenat Low power listening with Wake up after Transmissions MAC (LWT-MAC), amb l'objectiu de reduir les limitacions de *preamble sampling* però mantenint el seu baix consum energètic i la seva simplicitat. El rendiment del nou protocol s'ha avaluat en xarxes de sensors d'un sol salt així com en xarxes multi-salt, amb diferents perfils (incloent tràfic basat en esdeveniments) i càrregues de tràfic. Els resultats obtinguts han posat de manifest que el protocol LWT-MAC és capaç de millorar de forma significativa el rendiment de la xarxa, augmentant el throughput, reduint el retard i també proporcionant millor Quality of Service (QoS) col·lectiva. El terme QoS col·lectiva fa referència a la QoS que afecta al conjunt de missatges que pertanyen a un esdeveniment especific.

List of Articles

- C.Cano, B. Bellalta, A. Sfairopoulou, J. Barcelo. *A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs.* IEEE Communication Letters, 13(4):221-223, 2009.
- C.Cano, B. Bellalta, J. Barcelo, A. Sfairopoulou. *A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS.* In Proceedings of the 7th International Conference on Wired / Wireless Internet Communications (WWIC), 2009. *Best paper award*
- C.Cano, B. Bellalta, J. Barcelo, M. Oliver, A. Sfairopoulou. *Analytical Model of the LPL with Wake up after Transmissions MAC protocol for WSNs.* In Procedings of the International Symposium on Wireless Communication Systems (ISWCS 2009). September 2009. *Best paper award*
- C. Cano, B. Bellalta, A. Sfairopoulou, M. Oliver and J. Barceló. *Taking Advantage of Overhearing in Low Power Listening WSNs: A Performance Analysis of the LWT-MAC Protocol.* Accepted for publication in Mobile Networks and Applications (MONET) Journal, Springer. DOI: 10.1007/s11036-010-0280-4
- C. Cano, B. Bellalta, A. Cisneros and M. Oliver. *Quantitative Analysis of the Hidden Terminal Problem in Preamble Sampling WSNs.* Submitted for publication.
- C. Cano, B. Bellalta and M. Oliver. *Wake up after Transmissions and Reduced Channel Contention to Alleviate the Hidden Terminal Problem in Preamble Sampling WSNs.* Submitted for publication.

These publications are also related with contributions to the following research projects:

- Consolider-Ingenio Program CSD2008-00010 (COMONSENS)
- ICT Action IC0906 (WiNeMO)
- PlanNacional I+D TEC2008-0655/TEC (GEPETO)

List of Acronyms

ACK acknowledgement
AC-MAC Adaptive Coordinated MAC
AADCC Asymmetric Additive Duty Cycle Control
AI-LMAC Adaptive, Information-centric and Lightweight MAC
AREA-MAC Asynchronous Real-time Energy-efficient and Adaptive MAC
AS-MAC Adaptive Schedule MAC
AS-MAC' Asynchronous Scheduled MAC
AS-MAC" Asynchronous Sensor MAC
BEAM Burst-aware Energy-efficient Adaptive MAC
BER Bit Error Rate
B-MAC Berkeley MAC
BO backoff
CA Collision Avoidance
CAP Common Active Periods
CB Collective Bandwidth
CCA Clear Channel Assessment
CD Collective Delay
CMAC Convergent MAC
CPL Collective Packet Loss
CR Collective Reliability
CSMA Carrier Sense Multiple Access
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance
CSMA-MPS Minimum Preamble Sampling MAC
CTS Clear To Send
CW Contention Window
DCF Distributed Coordination Function
DDCC Dynamic Duty Cycle Control
DDT Data Distribution Table
DIFS DCF Inter Frame Space
DMAC Data Gathering MAC
DPS-MAC Divided Preamble Sampling MAC
DPS-MAC' Dual Preamble Sampling MAC
DSMAC Dynamic Sensor MAC
DW-LPL Dual Wake-up LPL
EA-ALPL Energy Aware Adaptive LPL

EIFS Extended Inter Frame Space
EMACS Energy Efficient MAC
ENBMAC Enhanced B-MAC
ESRT Event-to-Sink Reliable Transport
FPA Fast Path Algorithm
FSK Frequency-Shift Keying
ID identifier
IRDT Intermittent Receiver-driven Data Transmission
LMAC Lightweight MAC
LPL Low Power Listening
LWT-MAC Low power listening with Wake up after Transmissions MAC
MAC Medium Access Control
MaxMAC Maximally Traffic-Adaptive MAC
MFP-MAC Micro-Frame Preamble MAC
MH-MAC Multimode Hybrid MAC
NAV Network Allocation Vector
PEDAMACS Power Efficient and Delay Aware MAC for Sensor Networks
PER Packet Error Rate
PQ-MAC Priority-based MAC for providing QoS
Q-MAC QoS-aware MAC
QoS Quality of Service
RACK Request ACK
RICER Receiver Initiated CyclEd Receiver
RI-MAC Receiver-Initiated MAC
RL-MAC Reinforcement Learning MAC
RTS Request To Send
SCP-MAC Scheduled Channel Polling MAC
SESP-MAC Signaling-Embedded Short Preamble MAC
SIFS Short Inter Frame Space
SIR Signal-to-Interference Ratio
S-MAC Sensor-MAC
SREQ Send Request
SyncWUF Synchronous Wake Up Frame
TICER Transmitter Initiated CyclEd Receiver
TDMA Time Division Multiple Access
T-MAC Timeout-MAC
TrawMAC Traffic Aware MAC
WiseMAC Wireless Sensor MAC
WLAN Wireless Local Area Network
WSN Wireless Sensor Network
ZeroCal Zero Configuration Algorithm
Z-MAC Zebra MAC

Table of Contents

List of Articles

1 Introduction

Wireless Sensor Networks (WSNs) are a specific type of networks formed by small and low-capable devices able to sense different types of data, process them and communicate wirelessly to one (or more) central unit called sink. There is a large range of potential applications in industrial, military, environmental, health and home automation areas, among others, that can take profit of using WSNs to collect data. However, there are several challenges in designing a WSN mainly due to the high constraints of the sensor nodes but also due to the special characteristics of these networks.

One important characteristic of WSNs is their high application dependence. The different requirements and constraints are directly imposed by the application to be provided. This issue complicates the design of a general protocol stack or standard to be used in different deployments running different applications and leads to the definition of several approaches depending on the application to be supported. What follows is a review of the general characteristics and constraints of these networks but it is important to take into account that they all depend on the application and deployment scenario.

Several WSNs applications require the deployment of dense networks in large, remote and difficult to access areas. For this reason, it is crucial to keep the size and cost of sensor nodes as small as possible. This implies that the energy, computational and memory resources of the sensor nodes are usually limited. Therefore, simplicity and low energy consumption are crucial requirements in designing protocols for these networks. Apart from that, the limited energy resources of sensor nodes and the variable environmental conditions cause high network dynamics that should also be considered when designing mechanisms for WSNs. Another interesting feature is the kind of traffic patterns that are usually found. In these networks it is common that all the sensor nodes periodically send their information to the central unit, causing a communication pattern very different from the traditional point-to-point approach. It can also be needed to report a specific event to the sink, then only those nodes detecting the event will try to send a notification to the central unit. Finally, the sink can also ask the network about a certain information, then the data flows from the sink to all the sensor nodes in a data-centric approach in contrast to the common communication pattern based on destination addresses. The communication patterns should also be considered in different areas of the protocol stack in order to provide a proper solution to the application. There are also challenges in providing scalability in WSNs since deployments can be formed by a large number of sensor nodes. This also causes, in some scenarios, an extra difficulty in order to assign a global identifier to each device.

To summarize, 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 reduced memory, transmitting and computing resources
- Devices with limited energy resources
- High network dynamics
- Periodic, event-based and query-based communication patterns
- The data flows from all or a group of sensors to the central unit and vice versa

Among the different constraints, the limitation on the energy resources is the most important one as it directly affects the network lifetime. In general, in WSNs deployments it is too costly or even impossible to access the sensor nodes in order to replace the batteries. Therefore, it is important to design mechanisms that address the reduction of energy consumption.

Jointly with the transceiver design, the Medium Access Control (MAC) layer is of crucial importance to reduce the energy consumption since it directly controls the transceiver operation that is the most consuming component of a sensor node. The most common technique in WSNs is to put sensor nodes into sleep mode to save energy. In order to ensure that a sensor node is awake to receive the data, several proposals have been defined. The proposed approaches can be divided in three main categories [9]: Time Division Multiple Access (TDMA)-like channel access, protocols with Common Active Periods (CAP) and asynchronous MAC protocols. Among these, asynchronous protocols and specifically Low Power Listening (LPL) protocols, also called preamble sampling, have gained special attention since they can provide considerable low levels of energy consumption when the network load is low, which is the common situation in WSNs. In LPL, nodes sleep and wake up periodically to sample the channel. If the channel is found empty they return to sleep, otherwise they remain awake. A sensor node willing to send data has to, before transmitting the packet, send a long preamble that overlaps with the channel sampling time of the receiver. It can be easily observed that when there is a low load in the network, the protocol considerably reduces the energy consumption because the time needed to sample the channel is extremely short. However, as soon as the network load starts to increase, collisions of long preambles cause a performance degradation that can be intensified by the presence of hidden terminals. Sensor nodes are not able to sense all receptions taking place at the receiver, therefore they can interfere with other transmissions causing collisions at reception. Since the long preamble duration is typically much higher than the data packet transmission, hidden terminal problems can become problematic in preamble sampling WSNs when there are increases in the network load. Increases in the traffic load can occur in a WSN due to event detections: a group of sensor nodes that detect an event will try to inform the sink at the same time. The sink can also create higher network loads by disseminating queries to the network. And, finally, the regions that are closer to the sink are responsible for forwarding the traffic of the entire network, therefore they can also suffer from increased traffic loads. This poor reaction to traffic load increases affects the Quality of Service (QoS) observed by the set of messages belonging to an event (known as collective QoS [18]). Observe that in WSNs it is not important the QoS experienced by an individual message but it is important the QoS of the group of messages related to a specific event.

1.1 Contributions

The main contribution of this thesis is the definition and performance evaluation of the LWT-MAC protocol. It is able to improve the performance under the previously described conditions: *i)* with increases in the network load and *ii)* in presence of hidden terminals. The new MAC protocol, called Low power listening with Wake up after Transmissions MAC (LWT-MAC), has been analyzed under a broad range of traffic loads and patterns in a set of different scenarios. From the performance evaluation the benefits as well as the limitations of the proposed approach have been outlined.

Another important contribution is the performance evaluation of the basic LPL protocol. The performance of the basic LPL mechanism is evaluated with the goal of establishing its limitations with high traffic loads and in presence of hidden terminals. The performance evaluation has been done in different scenarios and traffic conditions, trying to characterize the system behaviour in a broad range of situations.

Moreover, a set of tools have been developed to evaluate the MAC protocols. Analytical models of both LPL and LWT-MAC have been derived.

For LWT-MAC the analytical model is based on single hop networks only, while for LPL it also considers the effects of hidden terminals. The analytical model of the LWT-MAC has been used to optimize the protocol performance by defining a fine tuning of the key parameters of the protocol. Moreover, in order to model more complex scenarios, an extension to the SENSE [19] simulator has been developed to model the LPL and the LWT-MAC protocols. Apart from that, the simulator has been extended with other functionalities like the event-based traffic generator and the simulation of the capture effect of partial overlapping collisions among others. Finally, with the main goal of validating simulation and analytical results a Mica2 testbed has been deployed.

To summarize, the contributions of this thesis are:

- Study of LPL limitations
    - Study of limitations in single hop and multihop networks
    - Performance evaluation with event-based traffic profiles
    - Investigation of hidden terminal effects
- Definition of the LWT-MAC protocol
- Performance evaluation of the LWT-MAC protocol
    - Study of benefits and limitations in single hop and multihop networks
    - Performance evaluation with event-based traffic profiles
    - Investigation of the benefits in presence of hidden terminals
- Performance evaluation tools
    - Analytical models
        - Model of LPL in single hop networks
        - Model of LPL in presence of hidden terminals
        - Model of LWT-MAC in single hop networks
        - Performance optimization of LWT-MAC
    - Simulator extensions
    - WSNs testbed based on Crossbow Mica2 motes

This thesis is presented as a collection of articles, what follows provides the correspondence between the articles, listed in chronological order, and the contribution they address.

1. *A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs*: Definition and initial evaluation of the LWT-MAC protocol in single hop and multihop networks.

2. *A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS*: Study of the performance of the LWT-MAC in presence of event-based traffic. Evaluation of the collective QoS metrics.
3. *Analytical Model of the LPL with Wake up after Transmissions MAC protocol for WSNs*: Analytical model of the LWT-MAC in single hop networks.
4. *Taking Advantage of Overhearing in Low Power Listening WSNs: A Performance Analysis of the LWT-MAC Protocol*: Optimization analysis of the LWT-MAC protocol. Definition of a heuristic configuration and evaluation in single hop and multihop networks.
5. *Quantitative Analysis of the Hidden Terminal Problem in Preamble Sampling WSNs*: Quantification of preamble sampling limitations, both with increases in the network load due to event traffic and in scenarios prone to hidden terminal problems. Definition of an analytical model of single hop and key hidden terminal scenarios.
6. *Wake up after Transmissions and Reduced Channel Contention to Alleviate the Hidden Terminal Problem in Preamble Sampling WSNs*: Quantification in hidden terminal scenarios of the improvement of the LWT-MAC. Each specific functionality of the protocol is evaluated separately in order to quantify its effect in the performance of the network.

1.2 Methodology

An extensive literature review about WSNs, with special interest on the MAC layer, was initially performed in order to review the existing work in the area. It was found that, in presence of event-based traffic, there were no MAC protocol that was able to cope with the traffic increases in a simple and efficient manner. The LPL approach has been identified as the most simple and low consuming MAC protocol for WSNs. That fact aimed at studying the LPL limitations and designing an extension to make LPL capable of reacting to sporadic traffic load increases.

To analyze the LPL limitations and the benefits of the new approach, a performance evaluation method should be defined. Analytical modeling has been used to study the performance in simple scenarios. Analytical models provide a detailed understanding of the system interactions and they can also be used to perform optimization analyses with the goal of tuning the system parameters to improve the network performance. However, analytical modeling has several limitations regarding the complexity of the network and the traffic profiles that can be analyzed. In this specific case, analytical models have been used to study single hop cases and very simple hidden terminal scenarios, thus not enough for evaluating a WSN protocol. In order to evaluate the performance of the network in large multihop scenarios affected by hidden terminal problems and under different kinds of traffic profiles (specially event-based traffic), simulations have been used. The advantage of using simulations is that they allow to evaluate large networks with an extremely lower cost if compared to real deployments. However, a clear drawback is that, similar to analytical modeling, they only provide a representation of reality. Therefore, it is important to develop a simulation model as close to the reality as possible and identify and analyze the discrepancies. The simulator used in this thesis is based on the SENSE simulator [19] but modified to include preamble sampling and the LWT-MAC protocols, apart from other specific functionalities. Although this simulator is not as well-known as other existing simulators, it provides a high degree of flexibility that allows to implement and validate new modules in a simpler manner providing a high level of detail. Finally, a real testbed using Mica2 motes has also been used to evaluate the performance of the network. It allows to obtain the performance in limited scenarios and with very simple traffic profiles but it can be used to validate the results of both simulations and analytical modeling.

The validation of the simulator and the analytical models are crucial in order to ensure that the conclusions drawn from the results will be similar to those obtained in real implementations. The validation of the simulator has been mainly done by checking that each operation is performed as expected in different conditions and scenarios. Apart from that, the results obtained using LPL have been compared with the ones obtained in the Mica2 testbed for a single hop and a simple hidden terminal scenario. This comparison has revealed that although the simulation model defines a set of assumptions that do not match the reality (see Section 4), the results from the simulator and the ones found in the testbed are extremely close. On the other hand, the analytical model validation has been done by comparing its results with the ones obtained by simulation. In this case, once again, both results are close enough to validate the analytical model.

1.3 Structure

This thesis is composed of a collection of articles derived from the research done and a short monograph that provides a general description of the problem addressed, the new MAC protocol proposed and a discussion of the results obtained. Section 2 provides the state-of-the-art of the MAC protocols specifically designed to cope with the special requirements of WSNs. Next, in Section 3, the new proposed MAC protocol is described. After that, in Section 4, an overview of the main performance results of the evaluation of the proposed MAC protocol are presented. Finally, Section 5 summarizes the contributions and outcomes obtained and outlines possible future research issues not covered in this thesis. After the monograph, the list of publications are provided in chronological order.

2 State-of-the-art: MAC Protocols for WSNs

In WSNs, the high energy consumption constraint makes the design of the MAC layer an important issue as it directly controls the transceiver operation, that is the most consuming component in a sensor node. Apart from the energy consumption constraint, the MAC protocol design faces other limitations like the low computational and synchronization capabilities of sensor nodes as well as their low memory capacity [9]. In order to reduce the energy consumption of a sensor node, the different sources of energy waste that should be reduced have been identified [33]:

1. Idle Listening: It 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 is unable to decode any of the packets involved if the received powers are similar. This implies that the transmitters have wasted energy transmitting and that the receiver has spent energy receiving without obtaining any benefit, senders may eventually retry transmission. In presence of hidden terminals this effect is even more important since transmitters cannot sense each other.
3. Overhearing: This source of energy dissipation occurs when a sensor node wastes energy receiving a packet not destined to it.

4. Overhead: Data packets in WSNs are usually small, therefore headers and other kind of overhead (like control messages) imply a high energy waste for WSNs.

MAC protocols designed for Wireless Local Area Networks (WLANs) or mesh networks are not suitable for the special requirements of WSNs due to 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 capable devices with high memory footprints and synchronization capabilities and, most importantly, with unlimited energy resources. Consequently, they do not address the sources of energy waste previously described. The common behaviour of these protocols is to continuously listen to the channel for a possible reception, in WSNs this design will rapidly consume the energy resources of a sensor node due to idle listening. Moreover, as nodes remain all the time listening to the channel, overhearing is common in that kind of protocols. The overhead is also considerable in traditional MAC protocols since the length of the data packets of the applications they are addressed to are significantly higher than the ones found in WSNs. The only source of energy waste that these MAC protocols address is collisions, since they (apart from wasting energy) considerably reduce the performance of the network. Another reason of 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 the energy consumption. Although sporadic increases in the network load can happen due to, for instance, event detections, WSNs usually work at low loads (with delay-tolerant traffic). Therefore, it is not always necessary to provide a high throughput or reduced delay to the application.

In the last years a large number of MAC protocols specially designed for WSNs have been defined. To save energy the most common approach is to put the transceiver into sleep mode since it 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 to ensure that a sensor node will be awake if a node sends something to it should be defined. There are basically three main broad categories to address that issue [9]:

- **Scheduled (TDMA-like)** protocols: in this category sensor nodes are assigned to specific slots in a frame to transmit and/or receive, then

they only wake up in those slots and sleep in the rest. A node has to have information about the time slot in which a neighbour will be awake in order to send data to it. This kind of protocols provide a good performance (controlled delay, high throughput and reduced or zero collision probability) but require a good synchronization among sensor nodes. Moreover, there is always a cost derived from creating and maintaining the slot assignment.
- Protocols with **Common Active Periods**: these protocols organize sensor nodes to wake up and go to sleep at the same time. A sensor node has to wait for the next wake up period to send data. In these protocols, although a certain degree of synchronization is needed, this requirement is not as stringent as in scheduled (TDMA-like) protocols. Protocols with CAP have also a cost associated to create and maintain the schedule (when to sleep and wake up).
- **Asynchronous** MAC protocols: in this kind of protocols no synchronization is required since 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. They can be divided in receiver-initiated and preamble sampling approaches.

The number of existing MAC protocols is extremely large, for this reason only a review of the most relevant MAC protocols of each category is provided. However, more emphasis is put in the asynchronous protocols since they are specially appealing for WSNs due to their simplicity, low energy consumption, the lack of synchronization requirements and their focus on low data traffic profiles, that are common in WSNs.

2.1 Scheduled (TDMA-like) MAC Protocols

This kind of MAC protocols defines when or how a specific node must wake up to transmit or receive. Normally, in WSNs these protocols are TDMA variants and therefore specific slots in a frame are assigned to sensor nodes to send or receive packets. In case of periodic or high volumes of traffic load these protocols provide a good performance, however their design faces several problems [9]. One of the most important concerns is the complexity to create and update the slot assignment, specially in a multihop network. In multihop networks (that is the usual case in WSNs) the goal is to assign slots with the constraint that they cannot be used in a two hop radius with the objective of reducing interference problems. Moreover, in case that the traffic requirements or the topology change the slot assignment should be updated. Apart from that, a precise synchronization is needed. This factor becomes a serious problem in multihop networks formed by low capable devices.

TDMA-like protocols for WSNs can be divided in two broad categories: protocols in which a central device computes the slot assignment and the distributed approaches in which the slot assignment is computed by the sensor nodes.

2.1.1 Centralized Scheduling

In this category of TDMA-like MAC protocols the sink is the responsible for creating and maintaining the schedule. Sensor nodes collect topology information and send it to the sink that creates the slot assignment. One example is the **Power Efficient and Delay Aware MAC for Sensor Networks (PEDAMACS)** [23] protocol designed by Ergen et al. in 2006. PEDAMACS defines four different phases of operation: the topology learning, the topology collection, the schedule and the adjustment phase. In the first phase, the sink floods the network with a *tree construction* packet that allows sensor nodes to obtain topology information. In the next phase, the sink broadcast a *topology collection* packet and after receiving it all sensor nodes reply, using Carrier Sense Multiple Access (CSMA), with their topology information (the sensor nodes they are able to hear). The sink computes the slot assignment and sends this information to the sensor nodes in the schedule phase. After that, sensor nodes wake up in their assigned slots to send or receive packets. Finally, the adjustment phase is used by the sink to collect new topology information from the sensor nodes and correct the slot assignment if necessary. It is assumed that the sink can reach all sensor nodes in one hop, therefore synchronization errors are minimized. Moreover, sensor nodes use different values of transmitting powers: higher values are used for tree construction while smaller values are used for scheduled data transmission. That allows sensor nodes to gather topology information of their neighbours but also of the nodes that can interfere with their transmissions.

2.1.2 Distributed Scheduling

In this category sensor nodes assign themselves a slot to transmit or receive, thus eliminating the cost of sending information to the sink and receiving the slot assignment. One of the most well know approaches is **Lightweight MAC (LMAC)** [61], defined by Van Hoesel et al. In this protocol each sensor node picks a random slot from the free slots (slots not used in a 2-hop distance). Sensor nodes exchange information about the slots they see as free. By doing this, a sensor node can select a slot from the 2-hops away unused slots, i.e., a slot considered free by all of its neighbours. The slot selection starts by a message sent by the sink. After receiving it, sensor nodes select the slot to use for transmitting data on it. Data transmissions are preceded by control messages that serve to synchronize sensor nodes and to inform about the intended destination of the data packet, then non interested nodes can go to sleep. The authors of LMAC defined a similar approach: the **Energy Efficient MAC (EMACS)** [62] protocol. The difference is that in EMACS sensor nodes are allowed to make transmission requests to slot owners in order to transmit on their slots. Each slot is divided in three phases: *Communication Requests* (where other nodes request to transmit), *Traffic Control* and *Data*. A slot owner always transmits a packet in the *Traffic Control* phase that is used for synchronization purposes and to inform about what type of communication will take place in the *Data* section: its own packet or a previously requested transmission. This approach provides more flexibility than LMAC at the cost of the energy consumed in the *Communication Requests* phase. To improve the traffic load adaptability the **Adaptive, Information-centric and Lightweight MAC (AI-LMAC)** [17] protocol is defined. AI-MAC is based on LMAC but it allows a sensor node to select more than one slot depending on its traffic requirements. A parent-child relationship is considered, then a sensor node acting as a parent computes the number of slots a child should use based on the traffic requirements of its children. Each child decides whether to follow the parent's recommendation based on the free slots it sees. To measure the traffic requirements each node maintains a Data Distribution Table (DDT) with statistics on the data generated and forwarded by a sensor node.

2.2 MAC Protocols with Common Active Periods

In MAC protocols with CAP sensor nodes organize themselves to wake up and go to sleep at the same time. When a sensor node wants to send a message it does so in its own active time since its neighbour will also be awake. The most well known approach that defines CAP for communicating in WSNs is the **Sensor-MAC (S-MAC)** [66] protocol, defined by Ye et al. in 2002. In S-MAC neighbouring sensor nodes synchronize themselves to follow the same schedule in the following way: when a sensor node is activated it remains listening for a certain time to receive a synchronization message. A synchronization message includes the next wake up time, relative to the time at which the sender finishes the message transmission. If that message is received the node starts following the same schedule, otherwise it creates its own schedule and follows it. This procedure creates virtual clusters of sensor nodes that wake up and sleep at the same time. To allow communication between clusters, border sensor nodes must follow all the schedules of the virtual clusters they are able to hear. The active period of the schedule is divided in three different parts. The first part is used to transmit synchronization messages, then the part that follows is used by those nodes willing to transmit to send Request To Send (RTS) messages, finally, the last part is used to send the Clear To Send (CTS) messages. Data transmission occurs in the sleep period with the goal that non-interested sensor nodes reduce their overhearing cost. Fig. 1 shows a schematic representation of the S-MAC protocol.

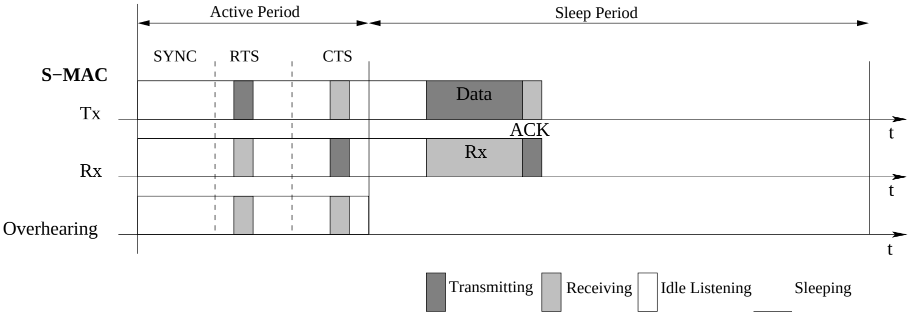


Fig. 1. Schematic representation of the S-MAC protocol

In this category it can also be classified the **IEEE 802.15.4** [2], a standard for small devices with small power resources and low data rate requirements. It defines two different types of access methods: the beacon-enabled and the nonbeacon-enabled modes. In the beacon-enabled mode the channel time is divided into an active part (formed by reserved slots and a contention access) and an inactive part (where the network coordinator can go to sleep). In the nonbeacon-enabled mode the channel access is based on unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) where the coordinator is unable to sleep. Both modes allow end devices to sleep to save energy and wake up periodically to send and/or to poll the coordinator for data. The standard defines the star and peer-to-peer topologies, although it is focused in the star topology leaving some features of the peer-to-peer topology undefined. Peer-to-peer networks based on the IEEE 802.15.4 have several problems to work in medium-to-large WSNs: the beacon-enabled mode suffers from collisions of beacons from different coordinators and the nonbeacon-enabled mode entails high energy consumption requirements.

Basic protocols with CAP are not adaptable to the traffic load since both the active and sleep periods are fixed. If the active period is longer than needed it causes a high idle listening cost, on the contrary, if it is smaller than required, there is an increased channel contention that results in a higher collision probability. Regarding the sleep period, a small value causes an increase in energy consumption while a high value reduces the energy consumption but results in a higher end-to-end delay. Different approaches have been defined to solve, or at least alleviate, that problems.

2.2.1 Adaptive active period

The main goal of **Timeout-MAC (T-MAC)** [60], that was defined by Van Dam et al. in 2003 and is one of the most well-known extensions of S-MAC, is to adapt the active period of the duty cycle to the traffic load. When a sensor node has something to transmit it waits until the beginning of the next active period, contends for the channel and then transmits. Moreover, sensor nodes finish their active period when no packet is received or sent during a time interval called TA. Therefore, if there is no data pending to be transmitted the active period is equal to TA (that accounts for the contention period, the transmission of an RTS and the propagation time). Otherwise, the active period finishes after the transmission of all nodes willing to transmit plus one TA. One of the problems of the T-MAC protocol is the early sleeping problem. It happens when a node that losses the contention wants to transmit after the ongoing transmission to a sensor node that has not overheard it. Observe that then, the receiver will be sleeping. One of the solutions the authors of [60] propose is to provide some time before the start of the data transmission to other nodes willing to transmit so they can send future RTS messages in order to maintain future receivers awake.

2.2.2 Reduction of the sleep delay

Other approaches, instead of adapting the active time to reduce idle listening are focused on reducing the delay caused by the long sleep periods. A common approach is to adapt the sleep period to the traffic load. An example is the **Dynamic Sensor MAC (DSMAC)** [40] protocol. In DSMAC the queuing delay (the time since the packet enters in the queue until it is released from it) is included in each packet header. Based on this information if a sensor node detects that the latency is not tolerable it doubles its duty cycle by reducing the sleep time. Sensor nodes notify the change in the synchronization messages and those that receive this information decide locally whether they should increase the duty cycle. Similarly, in **Adaptive Coordinated MAC (AC-MAC)** [3], sensor nodes adapt their active times per cycle using the number of queued packets as an indicator of the traffic load. The change is announced in RTS/CTS messages. By doing this, nodes receiving that value start following the new duty cycle and include it in their RTS/CTS packets. Apart from that, nodes with higher wake up times per cycle set a smaller Contention Window (CW) to increase their channel access probabilities.

Other kind of solutions aim at reducing the sleep delay by creating extra wake up periods. For instance, the **Adaptive Listening** [67] of S-MAC defines that a sensor node that overhears an RTS or CTS will wake up at the end of the transmission (*adaptive listen time*) just in case it should forward the message as a next hop. Although not all the next-hop nodes are able to overhear a transmission, especially when it happened in an adaptive listen time, some delay benefits can be obtained. Another example is the **Fast Path Algorithm (FPA)** [36] MAC protocol that creates a fast path from a sensor node to the sink. It is assumed that when the route is created, nodes in the path configure themselves to reduce the delay. Based on the hops from the sender, sensor nodes establish the extra wake up times in order to stagger the transmissions. A sensor node 2 hops away sets the extra wake up time at their regular time plus the time needed for carrier sense and data transmission to give time to the 1 hop forwarder to receive the frame. In the same way, a sensor node located at 3 hops sets it twice the value of the last hop. A considerable end-to-end reduction can be obtained, however problems appear when nodes have several fast paths active at the same time.

A slightly different approach is the **Data Gathering MAC (DMAC)** [42] protocol, presented by Lu et al. in 2004, that aims at reducing the sleep delay in a data-gathering tree. It staggers the active times (divided in transmitting and receiving phases) of the sensor nodes in the path to the sink in such a way that the transmission part of a level in the tree overlaps with the listening part of the immediately lower level. Similar to FPA, DMAC schedules extra active periods if more data is pending to be transmitted. The sender with more data to send sets a *more data* flag in the message. Then, a sensor node receiving that message schedules an extra active period at 3 listen plus transmit periods, that waiting time let the other nodes in the network forward the packet to the next hop and minimizes interferences in the multihop path. One of the problems of DMAC is its rigidity: first it is only valid for strictly convergecast communication and second, it assumes quite stable paths since it defines that CAP exist only between parents and children in the tree.

2.3 Asynchronous MAC Protocols

In asynchronous MAC protocols sensor nodes go to sleep and wake up independently of the others. There are two main techniques to send a packet to a receiver that can be sleeping: preamble sampling and receiver-initiated transmissions. In preamble sampling, sensor nodes wake up just to check if the channel is busy, in that case they remain awake. Then, a sensor node willing to transmit it first sends a long preamble to overlap with the channel sampling time of the receiver, and after that it sends the data. On the contrary, in receiver-initiated MAC protocols, sensor nodes wake up periodically and inform (by sending a message) the others. A node with a packet to transmit remains awake until the reception of the receiver's wake up notification and then, it transmits the message. The following sections provide a review of the most important protocols of each category.

2.3.1 Preamble Sampling

In the basic preamble sampling technique sensor nodes sleep and wake up periodically just to sample the channel, if the channel is detected busy they remain awake, otherwise they return to sleep. The time between channel samples (T_{ci}) is fixed and known by all the nodes in the network. In order to ensure 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_{ci}.

El-Hoiydi [21] and Hill et al. [26] presented the preamble sampling technique mostly at the same time in 2002. The former combined it with Aloha and called it preamble sampling while the latter combined it with CSMA and called it **LPL**[1].

Later, Polastre et al. defined **Berkeley MAC (B-MAC)** [52], a preamble sampling MAC protocol with an improved Clear Channel Assessment (CCA). In B-MAC the noise floor is estimated taking samples when it is supposed to be free (for instance just 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) since a valid packet could never have an outlier. This technique reduces the number of false negatives if compared to taking just one sample and comparing it to the noise floor. B-MAC defines also a set of interfaces to control the protocol parameters: backoff (BO), enable/disable CCA, set the value of the duty cycle and enable/disable the acknowledgement (ACK).

Wake up just to sample the channel allows preamble sampling MAC protocols to consume small amounts of energy when the traffic load is considerably small. Compared to protocols with CAP, preamble sampling MAC protocols can reduce the duty cycle maintaining the same delay since the time needed to check the channel activity is significantly smaller than the listening time needed in protocols with CAP. However, basic preamble sampling protocols have several disadvantages due to the transmission of the long preamble:

1. Costly collisions: If there is a collision, the entire long preamble is involved causing a high collision duration (the long preamble can be in the order of thousand milliseconds). Moreover, using Aloha or in presence

---

[1] In this work *preamble sampling* and LPL will be used to refer to the same technique: sending a long preamble before each data transmission

of hidden terminals the vulnerability time is equal to (at least) the long preamble transmission.

2. High overhearing cost: Using the basic preamble technique non-intended (overhearing) nodes receive the entire preamble and the data message since 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.
3. Lack of traffic load adaptability: The duty cycle is fixed for all the sensor nodes independently of their traffic load requirements, thus each packet to be transmitted is affected by the long preamble delay. It has to be considered that variations in the duty cycle of a sensor node should be known by its neighbours in order they can adapt the long preamble accordingly.

There exist several approaches that address some of the disadvantages of the basic preamble sampling technique. These approaches can be divided in 3 different categories: division of the long preamble into a burst of short packets, remember the wake up time of the receiver and duty cycle adaptation.

Division of the Long Preamble into a Burst of Short Packets. This category of preamble sampling MAC protocols divide the long preamble into a series of short packets with the main goal of including some useful information on them. Moreover, this approach is fully compatible with modern packetized radios like the CC2420 in which only fix sized packets can be transmitted.

The most common approach is to include the destination identifier (ID) in the short packets, thus allowing overhearing nodes return to sleep in order to reduce the overhearing cost (see **Enhanced B-MAC (ENBMAC)** [38]). Other works include also some information about when the data transmission will start, by doing this, the destination can go to sleep after receiving the short packet and wake up to receive the data. Examples of this behaviour are the **BMAC+** [6], **SpeckMAC-B** [65], **Divided Preamble Sampling MAC (DPS-MAC)** [37], **Synchronous Wake Up Frame (SyncWUF)** [57] and **Signaling-Embedded Short Preamble MAC (SESP-MAC)** [25] protocols.

Some approaches permit also the transmission of an early ACK that stops the transmission of the burst. In order to allow the transmission of the early ACK a gap should be left clear between short packets, that implies that an extra mechanism should be defined in order to avoid a receiver 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. Some examples are the MAC protocols **Transmitter Initiated CyclEd Receiver (TICER)** [39], **Minimum Preamble Sampling MAC (CSMA-MPS)** [43], the well known **X-MAC** [15], **Adaptive Schedule MAC (AS-MAC)** [13], **MX-MAC** [46], **Patterned Preamble MAC** [30], **Asynchronous Real-time Energy-efficient and Adaptive MAC (AREA-MAC)** [34], **Preamble Sampling with State Information** [55] and **1-hopMAC** [64]. 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 one detects the short packet burst. The first definition of this technique appeared in **Dual Preamble Sampling MAC (DPS-MAC')** [63], and it is also used in **Convergent MAC (CMAC)** [41].

Other approaches, instead of sending a burst of short packets including some useful information, send repetitions of the data message. By doing this, the destination can immediately receive the packet when it wakes up. These approaches are usually combined with the early ACK technique in order to reduce the overhearing cost. Examples of this approach are the already mentioned **MX-MAC** [46] and **Preamble Sampling with State Information** [55] protocols. On the contrary, **SpeckMAC-D** [65] sends repetitions of the data message until it is ensured that the destination has woken up and received the message, the sender does not request any kind of ACK from the recipient.

Other protocols like **Multimode Hybrid MAC (MH-MAC)** [11], **DPS-MAC'** [63], **MIX-MAC** [48] and **Traffic Aware MAC (TrawMAC)** [69] perform different operations depending on the type of packet to be sent. In MH-MAC and DPS-MAC' protocols the early ACK is allowed for unicast messages while the time until the data transmission is included in broadcast messages, allowing receivers to go to sleep until it starts. MIX-MAC selects from a poll of MAC protocols (compatible among them) the best to be used for each packet transmission. For instance, for small packets and 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.

It is worth to mention **Micro-Frame Preamble MAC (MFP-MAC)** [8] that includes in the short packets a hash of the data to allow nodes identify if they have already received it. It is useful to reduce the load of broadcast communication.

More elaborated approaches are the protocols **CMAC** [41], **1hopMAC** [64] and **RA-MAC** [35]. In CMAC and 1hopMAC anycast transmission is used. Then, those nodes with a better cost to the sink reply before using an early ACK. In CMAC the early ACK stops the data transmission while in 1hopMAC is the sender the one that selects the destination based on the 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 are capable of aggregating bursts from other nodes if they overhear their burst transmissions. The destination then creates a schedule for data transmission, alleviating the collision probability.

A completely different technique is the **SEESAW** [14] 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, listening time and number of advertisements of its neighbours.

Remember the Wake up Time of the Receiver. Several MAC protocols make sensor nodes remember the wake up time of the receivers with the goal of reducing the long preamble duration just to account for the possible synchronization error. This technique first appeared in the definition of **Wireless Sensor MAC (WiseMAC)** [22], 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 new data is available for that node. Knowing the channel sampling time allows the sender to transmit a preamble of length $L_{\mathrm{p}} = \min(4\theta t_{\mathrm{sync}}, T_{\mathrm{ci}})$ where θ is the tolerance of the clock,

t_{sync} is the time since the last synchronization happened, and T_{ci} is the sampling period. The computed value of the preamble accounts for the possible clock drifts of the sender and the destination, with a maximum value of the usual long preamble duration set to the listen plus sleep times. Observe that, one disadvantage of these approaches is the incompatibility with broadcast communication, in that case long preambles should be used.

This approach is easily combined with the division of the long preamble into a burst of short packets. Just 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 can include information of when the data transmission will start. Examples of this behaviour are the protocols **CSMA-MPS** [43], **SyncWUF** [57] and **TrawMAC** [69] already seen in the previous section.

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')** [29] tries to address this problem by forcing nodes in a neighbourhood to select a different wake up time. It allows to reduce the contention but there is a cost associated with the advertisement of the schedule. Moreover, broadcast communication is poorly supported (repetitions of the message should be done).

In contrast, other approaches like the **Scheduled Channel Polling MAC (SCP-MAC)** [68] and **BoostMAC** [58] protocols set the wake up time of all neighbouring nodes to the same instant. This solution is compatible with broadcast communication but it suffers from an increased channel contention. The authors of SCP-MAC try to solve that problem defining two contention phases per packet, one for the preamble and the other for the data message. In this way, only winners of the first contention try to send the data message, therefore reducing the collision probability. In BoostMAC, in order to reduce the overhead of communicating the wake up times, sensor nodes predict them by means of machine learning.

Another approach is the **MIX-MAC** [48] protocol, which takes advantage of the information of the wake up times of the receivers in order to synchronize all nodes in a path and staggers their transmissions with the goal of reducing the end to end delay of a burst of packets from a sensor node to the sink.

As a summary a schematic representation of the different preamble sampling extensions is shown in Fig. 2.

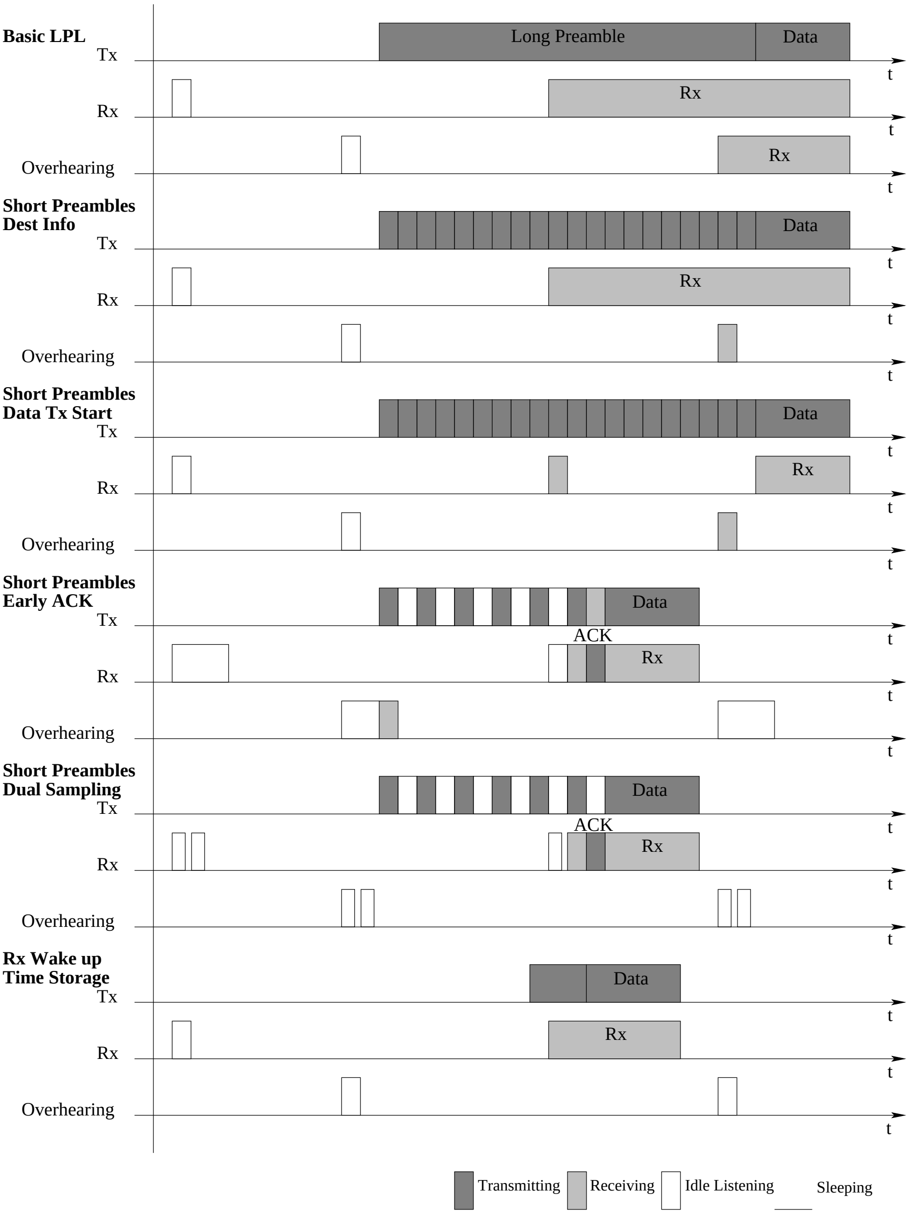


Fig. 2. Schematic representation of preamble sampling and its extensions

Duty Cycle Adaptation. There are different approaches in the literature to adapt the duty cycle and also the preamble duration to achieve a specific goal, normally to handle increases in the traffic load.

Some protocols adapt the duty cycle based on requests (normally embedded in packet transmissions) of the neighborhood. For instance, the **WiseMAC *more* bit** [22] is used by senders to indicate the receiver to stay awake (not follow their duty cycle) when they have more packets to send. An extension to this approach is the **Stay Awake Promise** defined in [27]. It is defined 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. By doing this, other nodes willing to transmit to the same destination can contend for the channel just after the current data transmission. It is useful for handling burst of packets from different nodes, however it is not effective for different sensor nodes that want to send only a message. A similar approach is defined in **Burst-aware Energy-efficient Adaptive MAC (BEAM)** [5]: 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 to reduce the long preamble and thus increasing the throughput. Another example is the **AS-MAC** [13] 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 is 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, that is defined as the time needed to forward the current received packet. Also, in **AREA-MAC** [34], sensor nodes adapt their duty cycle based on specific requests of their neighbours, these requests are performed to handle real-time data in order to minimize the delay from source to sink. Finally, the last example is **Asynchronous Sensor MAC (AS-MAC")** [51] 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 of different nodes, if that is the case **AS-MAC"** defines that the request with smallest sleep interval should be considered.

Another technique is to base the duty cycle adaptation on the observed traffic load. An example is the work presented in [7] that proposes an extension of B-MAC+, and defines that after detecting an increase in the number of incoming packets, a receiver must decrease its duty cycle. After doing that, the sensor node informs its neighbours about that change. The authors define that the notification is included in the short packets of the burst sent before transmitting a packet. A set of pre-defined listening modes associated with an incoming rate must be defined and known by all the sensor nodes. A similar approach is the **BoostMAC** [58] protocol. In this approach sensor nodes increase their sleep times in additive steps (until a certain value) if they find the medium idle when they wake up. Otherwise, if a high traffic is observed, they decrease the sleep time in a multiplicative manner. Instead of announcing the sleep times each node is using, senders decide which preamble length to use based on their own traffic and the cost of sending at different preamble lengths given the probability that the packet must be retransmitted because the receiver is sleeping (this probability is estimated and fine tunned throughout the lifetime of the network). Another example is the **Maximally Traffic-Adaptive MAC (MaxMAC)** [28] protocol that 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 sensor nodes start working on 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 neighbour nodes can adapt the length of the short packet burst accordingly. Two more approaches that fit in this category are the control algorithms to set the sleep schedule presented in [47]: 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 when it successfully sends 5 packets to the destination. On the contrary, the sleep time is decreased in 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 that means that the channel contention is low and nodes can sleep more to save energy. On the contrary, a packet failure can indicate a high contention on the channel, therefore nodes should sleep less. This algorithm can work better than a static sleep time, however it does not take into account the energy consumption or the required data rate. To this aim they define DDCC. The goal of DDCC is to regulate the sleep interval in order to accommodate the number of packets to be sent and minimize 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 of 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)** [45] protocol in which sensor nodes select the sleep interval that provides the required bandwidth and minimizes their energy consumption and the energy consumption of their children. To compute it an energy consumption model is used. It is assumed that sensor nodes will run it 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.

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)** [31] 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** [55] 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 left open how to set the values of the sleep intervals depending on the position in the tree.

A completely different approach is the work presented in [50]. Sensor nodes compute their next wake up time based on the expectation of receiving a new packet with the goal of minimizing the delay or the energy consumption. In order to compute the expectation of receiving a packet the distribution of the packet interarrival times should be known although they present a learning algorithm to approximate it.

2.3.2 Receiver-Initiated

A different approach of asynchronous MAC protocols are the receiver-initiated ones. In this kind of protocols each sensor node sleeps and wakes up independently of the others as in preamble sampling. However, each time that a sensor node wakes up it sends a beacon informing that it is awake and remains listening to the channel during a fixed time. A sensor node willing to send data remains awake until the reception of the receiver's beacon, after that it sends the message (see Fig. 3). The initial definition of this approach for WSNs appeared in the **Receiver Initiated CyclEd Receiver (RICER)** [39] protocol. As well as TICER, it defines a multi-channel operation in which control and data messages are sent in different channels. A recent approach is the single-channel **Receiver-Initiated MAC (RI-MAC)** protocol [59], presented by Sun et al. in 2008. In RI-MAC senders transmit directly (without waiting a random BO) after the reception of the beacon message. In case of collision the receiver broadcasts a new beacon indicating that senders can retry transmission but using a specified random BO. Moreover, in RI-MAC the ACK serves as a new beacon allowing other nodes to send more data to the receiver. A similar approach is the **Intermittent Receiver-driven Data Transmission (IRDT)** protocol [32], in this protocol sensor nodes send also beacons each time they wake up, however, nodes willing to transmit respond to these beacons by sending a request packet, called Send Request (SREQ), that is replied by the receiver with a Request ACK (RACK). After that handshake, the data transmission begins, that is also acknowledged with the transmission of an ACK. In case of SREQ collision, the receiver decreases its sleep time to handle the different transmissions, on the other hand, senders ignore some beacons of the receiver to reduce the channel contention.

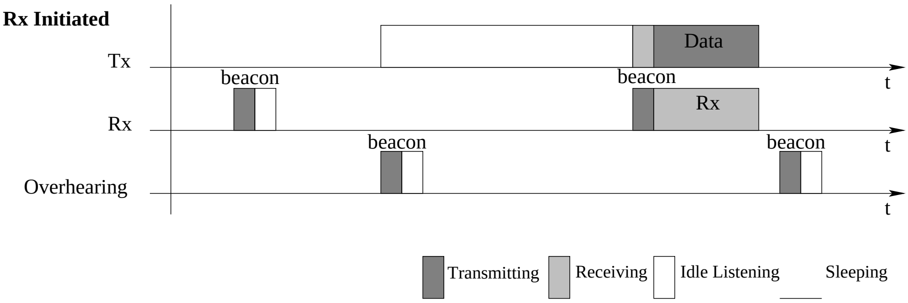


Fig. 3. Schematic representation of a receiver-initiated MAC protocol

The **Dual Wake-up LPL (DW-LPL)** protocol [49] uses the receiver-initiated technique for unicast transmissions while for broadcast transmissions the basic LPL is used. Sensor nodes wake up to sample the channel but also to send beacons. They increase their sleep time if they do not receive responses to their beacons. They can also stop the beacons if they detect that no packets are received during a certain time, the LPL communication is then used as a backup.

Receiver-initiated protocols occupy less channel time for each transmission since the long preamble is replaced by the waiting time of the sender. However, at very low loads their consumption is higher than in LPL approaches as sending beacons is more energy consuming than sampling the channel just to detect if there is a transmission taking place.

2.4 Hybrid MAC Protocols

Other approaches define hybrid mechanisms that switch between different protocol categories depending, in most cases, on the traffic load. Normally, they use a TDMA-like approach at high loads and a more lightweight protocol at low loads.

The most well-known hybrid MAC protocol is **Zebra MAC (Z-MAC)** [53] (presented by Rhee et al. in 2008) that uses CSMA under low contention and TDMA at high contention. In the setup phase each sensor node selects a slot that must be free in a two-hop neighbourhood (in order to avoid hidden terminal problems). Nodes exchange information among them in order to make the slot selection. When there is low contention any node can compete to transmit in any slot while in high contention only the owner can transmit in its slot. Under low contention, owners have priority to access the channel, while non-owners wait a fixed time in order to allow the owner to transmit, if nothing is sent non-owners can compete for channel access. When a sensor node detects high channel contention it informs its neighbours by periodically sending messages in order to enter in the high contention operation. A node decides that there is high contention on the channel based on the number of BO instances generated before transmission (given that if the channel is busy after a BO expiration a new BO is started). Z-MAC allows to increase the throughput of the network when necessary at the cost of creating the schedule and notifying about channel contention. Moreover, it solves the problem of synchronization overhead since it is only necessary under high contention when several packets are transmitted.

Another well known hybrid MAC protocol is **Crankshaft** [24] (by Halkes, 2007). It divides the time into frames and each node selects a slot in which to listen for incoming messages. The slot selection is made based on the MAC address of the node modulo the number of slots per frame. Using this slot selection can result in having multiple slot owners per frame, therefore each sensor node is also allowed to start a transmission on its own slot. Moreover, there are broadcast slots in which all sensor nodes wake up to listen and/or receive. The communication in a slot is contention-based, a node willing to transmit computes a random BO and only if the channel is found empty after the BO countdown it starts transmitting a preamble. After the preamble transmission the winner of the contention transmits the packet.

2.5 Discussion

In order to summarize the previously presented literature review, Table 1 shows the classification of the different MAC protocols reviewed.

From the description of the different categories, it can be observed that there is not a clear solution for WSNs. Each category of MAC protocols has its advantages and disadvantages and the best MAC protocol to use in a specific WSN strictly depends on the application and scenario. TDMA-like protocols are suitable for periodic and high traffic loads since they are able to provide zero collision operation, bounded latency and good throughput. However, the main drawbacks of these protocols are the high cost associated to the schedule creation and maintenance and the requirement of strict synchronization. On the other hand, protocols with CAP are specially appealing for applications with periodic and medium traffic load requirements. As a disadvantage, this kind of protocols have also to create and maintain a schedule for a group of nodes and a certain degree of synchronization is also needed. Finally, preamble sampling protocols are suitable for low traffic demanding applications, since the long preamble limits the throughput that can be achieved. They are, in contrast, very simple and provide very low energy consumption because the time to sample the channel is extremely short.

3 LPL with Wake up after Transmissions MAC Protocol

3.1 Motivation

The main goal of this thesis is to investigate possible solutions to make the MAC protocol capable of reacting to sporadic increases in the network load in a WSN normally working at low loads. Due to the limited resources of sensor nodes, issues like simplicity and low energy consumption are strong requirements regarding the MAC protocol design.

Table 1. Summary of the MAC Protocols Reviewed

Category	Subcategory	List of MAC Protocols
TDMA-like	Centralized	PEDAMACS
	Distributed	LMAC, EMACS, AI-LMAC
CAP	Basic	S-MAC, IEEE 802.15.4
	Adaptive Active Period	T-MAC
	Adaptive Sleep Period	DSMAC, AC-MAC, Adaptive Listening, FPA, DMAC
Asynchronous	Preamble Sampling (Basic)	Preamble Sampling, LPL, B-MAC
	Preamble Sampling (Short Preambles)	ENBMAC, BMAC+, SpeckMAC-B, DPS-MAC, SyncWUF, SESP-MAC TICER, CSMA-MPS, X-MAC, AS-MAC MX-MAC, Patterned Preamble MAC, AREA-MAC, 1-hopMAC, DPS-MAC', CMAC, Preamble Sampling with State Information, SpeckMAC-D, MH-MAC, TrawMAC, MFP-MAC, RA-MAC, SEESAW
	Preamble Sampling (Remember Wake up Time)	WiseMAC, CSMA-MPS, SyncWUF, TrawMAC, AS-MAC', SCP-MAC, BoostMAC, MIX-MAC
	Preamble Sampling (Adaptive Duty Cycle)	WiseMAC more bit, Stay Awake Promise, BEAM, AS-MAC, AREA-MAC, AS-MAC", extension to B-MAC+, BoostMAC, MaxMAC, AADCC, DDCC, ZeroCal, EA-ALPL, Preamble Sampling with State Information
	Rx-Initiated	RICER, RI-MAC, IRDT, DW-LPL
Hybrid		Z-MAC, Crankshaft

Among the different MAC protocols designed for WSNs, preamble sampling provides some interesting features, like the reduced energy consumption and the simplicity, that make it specially appealing to WSNs. Compared with protocols with CAP, preamble sampling provides less complexity and cost since no common schedule should be created, communicated and maintained. Apart from that, preamble sampling consumes less energy than protocols with CAP when the traffic load is low. This is caused by the extremely short channel sampling time that allows a sensor node to return immediately to sleep if the medium is found idle. On the contrary, in protocols with CAP the active period of the duty cycle is commonly large since it should give time for transmitting synchronism messages as well as RTS and CTS packets. If compared to TDMA-like MAC protocols, preamble sampling is much less complex since no schedule should be computed and distributed [33]. Moreover, it provides more flexibility for sensor nodes with different traffic load demands.

However, as explained before, preamble sampling suffers from costly collisions when the network load increases and specially when there are hidden terminals present in the network. The reason of that problem is mainly due to the long preamble that causes long collision durations and a high vulnerability period if there are hidden terminals trying to transmit.

Table 2 shows a summary of the characteristics that each main category of MAC protocols provide. Hybrid MAC protocols are not considered since the diversity in the different kind of approaches makes a general evaluation impossible.

Table 2. Summary of the characteristics of each MAC category. The '++', '+', '-' and '- -' signs denote how well each category is able to provide a specific characteristic compared with the others.

Category	Energy Awareness	Lack of Synchronization Requirements	Simplicity	Traffic Load Adaptability	Hidden Terminal Support
TDMA-like	++	- -	- -	-	++
CAP	+	-	-	-	+
Asynchronous	++	++	++	-	- -

In the literature review about preamble sampling, a set of approaches to solve some of the preamble sampling limitations have been outlined. However, they also have some drawbacks that should be taken into account.

The division of the long preamble into short packets does not provide any benefit regarding traffic load adaptability. If the early ACK is used, a reduction of (on average) half of the preamble can be achieved, thus reducing the duration of a collision. However, this approach does not provide any specific mechanism to adapt the behaviour of the protocol to the traffic load. When hidden terminals are present, if the early ACK is not used, they suffer from the same performance decrease than basic preamble sampling. On the contrary, if the early ACK is allowed, some benefits can be found. If a hidden terminal receives an early ACK of a node not directed to itself it can stop its burst of short preambles allowing the other transmission to be received successfully. However, due to the desynchronization among hidden terminals it can be impossible for a receiver to find a gap to transmit the early ACK, obtaining then, the same performance degradation as the basic preamble sampling technique.

Regarding the protocols that remember the wake up time of the receivers, it can be said that the reduction of size in the preamble is beneficial when the network load increases as collision durations are reduced. However, as the duty cycle is not adapted they cannot react to increases in the network load and therefore, they provide a low throughput and a long sleep delay. Concerning the hidden terminal problem, the performance is quite different depending on whether all nodes wake up at the same time. In those protocols that all nodes wake up at the same time, the hidden terminal problem remains an issue, since all nodes will try transmission a few instants before the common wake up time, thus increasing the collision probability. As hidden terminals cannot sense all the receptions taking place at a sensor node, they can further increase the collision probability in that case. On the contrary, if sensor nodes wake up at different times, the hidden terminal problem is alleviated only if hidden terminals transmit to different receivers. In the case that hidden terminals transmit to a common receiver the problem is comparable to the case in which all nodes wake up at the same time.

Finally, the adaptive duty cycle MAC protocols that have been reviewed aim at adapting to the traffic load either by means of requests of neighbour nodes or estimating it. Those protocols that adapt the duty cycle based on the requests of neighbouring nodes have a clear drawback, that is that they only adapt the duty cycle in cases in which a sensor node has multiple packets to send. They do not address the cases in which several nodes have at least one packet to transmit, that can be the situation when a new event is detected or a query wants to be disseminated. On the other hand, the protocols that estimate the traffic load are usually too complex since they require that sensor nodes perform several operations to compute the duty cycle. Moreover, in some cases, the duty cycle is adapted in steps (for instance, doubling or halving the duty cycle at given time intervals). This behaviour can be quite inefficient when instantaneous increases in the network load happen. What is needed in that cases is a rapid reaction of the protocol to serve all the traffic as quickly as possible. Regarding the hidden terminal problem, adaptive duty cycle MAC protocols can alleviate it until a certain extent since the reduction of duty cycle also reduces the duration of collisions.

Now, in Table, 3 the characteristics of the preamble sampling extensions are summarized. It can be clearly seen that none of the extensions defined is able to cope with hidden terminal problems. Moreover, it can be observed that the adaptive solutions do not maintain the simplicity of the basic approach.

Table 3. Summary of the characteristics of each preamble sampling extension. The '++', '+', '-' and '- -' signs denote how well each category is able to provide a specific characteristic compared with the others.

Category	Energy Awareness	Lack of Synch. Requirements	Simplicity	Traffic Load Adaptability	Hidden Terminal Support
Short Preamble Burst	++	++	++	-	- -
Remember Rx Wake up Time	++	+	+	-	-
Adaptive (requests)	++	++	-	+	-
Adaptive (load estimation)	++	++	- -	++	-

In general, preamble sampling seems a good solution for WSNs that usually work at low loads but that need to react to increases on the traffic load due to event detection, query dissemination or query responses. However, it is still needed to make them react to that increases and address the hidden terminal problem in a simple and low consuming manner.

3.2 LWT-MAC Description

The main goal of the LWT-MAC protocol is to improve the performance of preamble sampling making it react to instantaneous increases in the network load and better managing hidden terminal problems. In designing the new protocol the following requirements have been considered:

- It has to react to instantaneous increases in the network load in a rapid and efficient manner.
- It has to alleviate hidden terminal problems.
- The low energy consumption must be kept as a strong requirement.
- The complexity of the mechanism must be maintained low, comparable to basic preamble sampling.

Inspired by the Adaptive Listening mechanism of S-MAC [67], that wakes up the receiver and transmitter of an ongoing transmission at the end to transmit more data, the preamble sampling technique is modified. It is defined that after a transmission all sensor nodes that have overheard it must wake up. The difference of the proposed approach is that all nodes that wake up are allowed to transmit. By implementing the LWT-MAC protocol, the benefits of LPL are maintained and, at the same time, the network is able to react to instantaneous increases in the network load.

Moreover, in order to alleviate hidden terminal problems, the RTS/CTS access is used in the *scheduled* phase (when all nodes that have overheard a transmission wake up to transmit or receive). This feature allows to cope with hidden terminal problems that become noticeable when the network load increases (i.e., more than one packet need to be transmitted by a set of neighbours in a short time interval). A schematic representation is provided in Fig. 4.

In addition to this behaviour, a non-aggressive retransmission procedure is defined. When a packet transmission fails, instead of immediately retrying transmission, sensor nodes wait a Collision Avoidance (CA) Timer before transmitting. During the CA Timer (set to the duration of a long preamble and packet transmission) sensor nodes keep listening to the channel. The goals of the CA Timer are the following: *i)* allow other transmissions to finish, *ii)* increase the probability of overhearing a packet of an ongoing transmission and get *synchronized* (wake up at the end) to it and *iii)* reduce the consecutive collisions that can happen among hidden terminals. Moreover, in case of a data packet failure in the *unscheduled* phase (when the

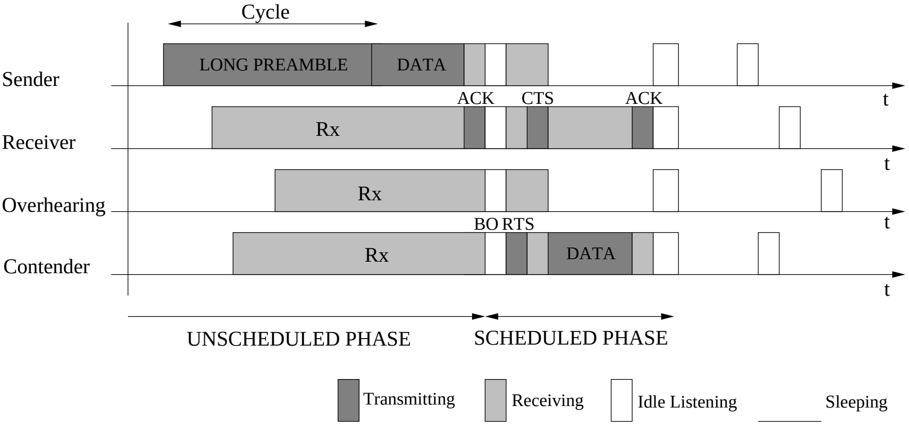


Fig. 4. LWT-MAC Protocol

long preamble is used), an RTS is immediately sent. This mechanism, called *RTS if DATA fails*, aims at increasing the probability of at least receiving the last RTS of a partial overlapping collision among hidden terminals.

Observe that the LWT-MAC protocol maintains a simplicity similar to basic LPL as no synchronization is needed. It also provides a reduced energy consumption when the network load is low. Apart from that, it is able to reduce the sleep delay when several packets need to be transmitted, independently if they are generated at the same or different sensor nodes. And, finally, it takes hidden terminal problems into account and implements specific mechanisms to alleviate them.

The LWT-MAC protocol can notably improve the performance of the network but it also has some limitations that should be identified. The first one is the increase of energy consumption at low loads. Sensor nodes wake up at the end of each transmission to listen to the channel, however when the traffic load is low the probability that a node has something to transmit during that time is also low. As a result there is an increase of energy consumption compared to basic LPL. Nevertheless, it should be considered that although there is and increment in energy consumption, it is usually low as it accounts for the energy waste of being in idle mode only during the maximum BO duration.

Another limitation of the proposed approach is the impossibility of assuring that a neighbour will be awake in the scheduled phase. Waking up all sensor nodes that overheard a transmission does not guarantee that all the neighbours of a specific node were able to overhear the transmission. This affects the delivery rate and forces to enable retransmissions to cope with this problem.

Finally, it has also to be considered that the value of the CW, that is used to compute the random BO, directly influences the capability of LWT-MAC to alleviate the hidden terminal problem. A high value of CW implies a higher probability of correctly receiving an RTS in the scheduled phase even in presence of other hidden terminals trying to transmit. However, increasing the value of CW affects the energy consumption of the sensor nodes, specially when the network load is low.

The different articles included in the publication list provide a general description of the LWT-MAC protocol, being the most complete one the definition that appears in the article ***Taking Advantage of Overhearing in Low Power Listening WSNs: A Performance Analysis of the LWT-MAC Protocol*** that also provides a flowchart that fully describes the protocol behaviour.

To conclude the LWT-MAC description, it is worth to mention that although the protocol is designed to be used with basic preamble sampling, it is also compatible with the division of the long preamble into a burst of short preambles. As already seen in Section 2, these short preambles can include some useful information, like the destination of the packet or the time until the data packet transmission will start, with the goal of reducing the overhearing cost.

4 Performance Evaluation

This section summarizes the results obtained in the study of the LPL limitations and the performance evaluation of the LWT-MAC protocol. In order to obtain the improvement of the LWT-MAC protocol its performance has been compared to the basic LPL technique, that has been used as a reference of LPL performance. However, if compared to the LPL extensions previously described, the LWT-MAC protocol is expected to provide a faster reaction to traffic load increases with a smaller complexity (traffic load is not estimated by sensor nodes) and reduced energy consumption (since no control messages are sent).

Before presenting the performance results, the evaluation scenarios are reviewed discussing the assumptions considered and the parameters used.

4.1 Evaluation Scenarios

The target scenario for the performance evaluation of the LWT-MAC protocol is a random multihop WSN in which sensor nodes periodically send data to the sink and where sporadic increases of the network load occur, for instance due to event occurrences. However, other basic scenarios have also been considered in order to analyze in detail the behaviour of the LWT-MAC protocol. It has also been analyzed a single hop scenario in which all sensor nodes are inside the coverage range of the others. This scenario allows to obtain a reference to the network performance. Additionally, when analyzing the LWT-MAC performance with hidden terminals, two key scenarios are considered: *i)* a set of hidden nodes sending data to a common receiver and *ii)* a grid topology in which the number of neighbours and hidden terminals are controlled. These two last scenarios allow to quantify the hidden terminal problems and how the LWT-MAC can alleviate them.

The development of analytical and simulation models implies the consideration of some key assumptions. An important assumption is the consideration of a noise-free channel. It has been considered that a packet transmission can only fail due to collisions, either complete overlapping due to neighbour nodes simultaneous transmissions or partial collisions due to hidden terminal problems. Transmission errors due to noise will affect the LWT-MAC protocol in the same manner as collisions, making the system return to the unscheduled phase and making sensor nodes wait the CA Timer, therefore decreasing the protocol performance. It has also been considered that the packet size is fixed. This is a reasonable assumption if we take into account the information to be communicated, that is mainly the quantified value of an environmental measurement. Regarding the considered medium access procedure, it has been assumed that sensor nodes follow a similar approach as the IEEE 802.11 Distributed Coordination Function (DCF) MAC protocol [1]. Using this procedure, sensor nodes wait a random BO before transmitting a packet. If the medium is found busy during the BO countdown they store the remaining BO and wait until the transmission finishes. At the end of the ongoing transmission the BO is resumed. During the BO countdown the channel time is assumed to be slotted.

There are also some important issues to be considered in the simulation model of multihop networks. The routing protocol used is based on the shortest path (smallest number of hops) offline computation. The aim of pre-computing the routes is to reduce the impact of the routing protocol related messages in the results. In some multihop simulations it has also been assumed that transmissions to the channel are affected by path loss and slow fading. This consideration aims at modeling the network as close to reality as possible, specially when the capture effect is considered. In some multihop scenarios the reception and carrier sense thresholds have been considered. In this case, if a node receives a packet with a received power below its reception threshold but higher than the carrier sense one it is able to detect the packet but it is not capable of decoding it. Once again, this consideration aims at making the model as realistic as possible.

Finally, regarding the analytical model there are some key assumptions that should be discussed. The analytical model is based on the one described in [10], where an IEEE 802.11 analytical model is presented, therefore the same assumptions are considered. The first and maybe the most important one is the consideration of Poisson traffic. Although this assumption simplifies the analytical modeling it considerably differs from the traffic patterns commonly found in WSNs. Traffic patterns in WSNs can be constant (coming from monitoring applications), event-based (coming from event detections) and query-based (coming from dissemination or responses of queries). For this reason, some multihop scenarios were analyzed with constant and event-based traffic profiles instead of using exponential interarrival packet generation times. The analytical model also considers exponential service time. In this case, the variability of the BO duration, that accounts for other possible transmissions in the neighbourhood, motivates this assumption. Also regarding the analytical model, it has been assumed that the probability that a node transmits on a given slot and the conditional collision probability are independent of the BO stage and constant for all transmission attempts. This consideration is a common assumption in several IEEE 802.11 MAC analytical models (for instance the works in [12, 16, 20, 54]) and, as shown in [44], the latter assumption is a reasonable approximation since the number of collisions is dominated by the collisions that occur at the first BO stage. Moreover, the comparison of the analytical model and simulation results shows considerable close results.

The specific parameters for each scenario are described in each article. The hardware and MAC parameters are mainly based on Mica2 motes and the

TinyOS implementation of S-MAC and B-MAC. Other parameters like the density of nodes in a multihop network, the simulated area and the event coverage range are based on the parameters considered in [56], where a well-known transport protocol for event-based WSNs is presented: the Event-to-Sink Reliable Transport (ESRT) protocol.

4.2 LPL Limitations

In order to study the limitations of the basic preamble sampling technique, performance results in different scenarios along with a comparison with the proposed approach are provided in the articles that compose this thesis. Moreover, a quantitative analysis of the hidden terminal problem is provided in the article ***Quantitative Analysis of the Hidden Terminal Problem in Preamble Sampling WSNs***, that is included in the publication list. In this article a quantification of the impact of hidden terminals is performed. This article analyzes the problem in the key scenarios previously described as well as in a random network with periodic and event-based traffic profiles. The capability of the capture effect to alleviate the problem is also studied in this work. Moreover, an analytical model of preamble sampling both for single hop and networks affected by hidden terminal problems is derived. The analytical framework to model the effects of hidden terminals in preamble sampling networks considers, in contrast to the majority of analytical models in the area, that hidden terminals do not start the BO countdown synchronized. This is an important contribution of this work. One of the main conclusions of this article is that the hidden terminal problem is considerable at medium-to-high traffic loads, its negative effects are clearly observed when the traffic load increases. This result can be observed in Fig. 5, extracted from the article, that shows the throughput per node in the n_h-hidden scenarios in which a set of $n_\mathrm{h} + 1$ hidden terminals transmit to a common receiver. It can be seen that, given that no retransmissions are enabled, the throughput decreases to zero when the traffic load starts to increase. This effect occurs due to the long vulnerability time caused by the transmission of the long preamble that results in a high collision probability and a long collision duration. Moreover, when the traffic load is high, the probability of having consecutive collisions is also considerable. In this figure it can also be observed how the analytical model closely matches the simulation.

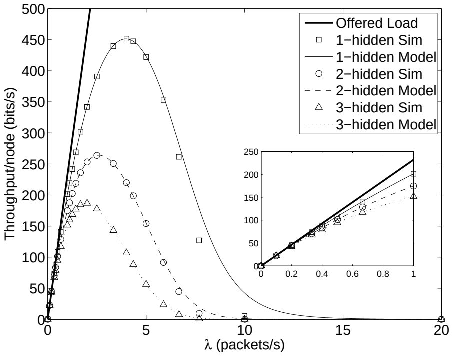


Fig. 5. Throughput per node in the n_{h}-hidden scenarios. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 1$ packets/s)

Another interesting outcome of this work is the inability of the capture effect to considerably alleviate the problem. By considering the capture effect, only a slight performance improvement is found in the different scenarios evaluated.

From the results obtained in this article, some recommendations can be provided. The first one is the consideration of adaptive approaches to reduce the vulnerability time in order to alleviate the effects of hidden terminals. By reducing the long preamble, both the collision probability and the duration of collisions among hidden terminals are decreased. It is also necessary to implement non-aggressive mechanisms to retransmit packets. Common retransmission procedures restart the BO countdown and retry transmission when it expires. In presence of hidden terminals this retransmission procedure can lead to consecutive collisions making the receiver unable to successfully receive any packet.

4.3 LWT-MAC Performance

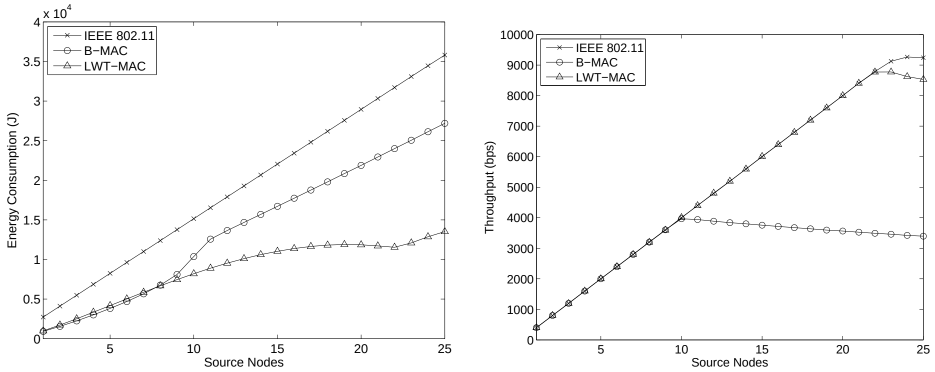
4.3.1 Single Hop Scenario

The first scenario in which the LWT-MAC is evaluated is a single hop network where all sensor nodes are inside the coverage range of the others, i.e., no hidden terminals are present. The main goal of analyzing this scenario is to obtain the resulting network performance due to the wake up after transmissions feature. The expected results are an increase of throughput and a decrease of delay due to the reduction of the duration of the successful transmissions and collisions in the scheduled phase. This reduction will also lead to a decrease of energy consumption at high loads but a slight increase is expected to be found when the network load is low (due to the idle listening after transmission periods).

Observe that this scenario provides the best conditions for the LWT-MAC protocol since without hidden terminals and channel errors all sensor nodes will *synchronize* to ongoing transmissions and therefore, will be able to send their packets after them without making use of the long preamble. The only limitation of this scenario is the occurrence of collisions as it has been designed that collisions move the system to the unscheduled phase in which nodes should use the long preamble before a data transmission. However, after the initial transmission in the unscheduled phase the system will immediately move to the scheduled phase again.

The article ***A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs*** provides the initial evaluation of the protocol in a single hop network. In this evaluation the LWT-MAC protocol is compared by means of simulations to B-MAC and to the IEEE 802.11 MAC. Results show, as expected, that the LWT-MAC protocol is able to substantially increase the throughput and reduce the delay provided by B-MAC at the cost of a slightly increase in the energy consumption at low loads. It can also be observed that the performance of LWT-MAC is close to the one obtained by the IEEE 802.11 MAC protocol but with a considerably lower energy consumption. Fig. 6, extracted from the article, shows the throughput increase and the energy decrease at high loads of the LWT-MAC compared to B-MAC. It can also be observed the slightly increase of energy consumption at low loads.

In addition to the simulation evaluation, an analytical model of the LWT-MAC protocol has been derived. The analytical framework is useful to formally define the behaviour of the protocol and allows to study the interactions among the different elements involved. The article ***Analytical Model of the LPL with Wake up after Transmissions MAC protocol for WSNs*** describes the proposed LWT-MAC analytical model considering a single hop network and validates it comparing the results obtained with simulations. The LWT-MAC analytical model is based on the single hop B-MAC model presented in ***Quantitative Analysis of the Hidden Terminal Problem in Preamble Sampling WSNs***. However, to model the LWT-MAC behaviour it has to be considered that the durations of the suc-

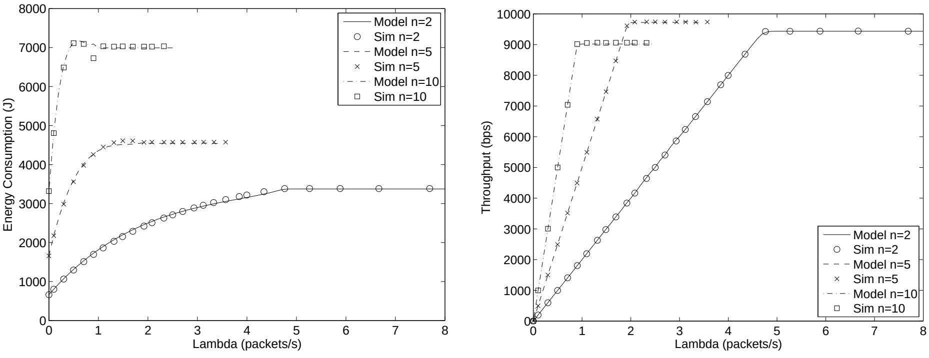


(a) Consumption (b) Throughput

Fig. 6. Single hop scenario. Performance results as the number of source nodes increases.

cessful transmissions and collisions depend on the probability of being in scheduled or unscheduled phases. To this aim, the probability of being in the scheduled phase is defined as the probability that a successful transmission occurs and that a node has still something to transmit. The probability of being in the unscheduled phase is the complementary of the former. A challenging issue of the model is the computation of the time that a node without anything to transmit spends in idle mode at the beginning of the scheduled phase waiting for another node to transmit. Observe that this value is the amount of remaining BO of the node that wins the channel contention. For simplicity, this time has been approximated by the average BO value. This approximation affects the results with moderated traffic load and its effect depends on the number of nodes. With high number of nodes the energy consumption is overestimated (as the probability that a node has a small remaining BO increases). On the contrary, with a small number of nodes, the energy consumed is underestimated (since the probability to transmit a packet that arrives at the queue during the listen after transmission period increases). Fig. 7, also included in the article, shows the validation of the analytical model comparing its results with simulations for different number of sensor nodes and a broad range of traffic loads. It can be seen that the effects of the aforementioned approximation are considerably small.

The derivation of the LWT-MAC analytical model allows to perform optimization analyses with the goal of tuning the key parameters of the protocol.



(a) (b)

Fig. 7. Total energy consumption and throughput with different number of sensor nodes

The article ***Taking Advantage of Overhearing in Low Power Listening WSNs: A Performance Analysis of the LWT-MAC Protocol***, that is an extension of the previous article, describes the optimization analysis performed. In this work a new feature of the protocol has been defined: it has been assumed that sensor nodes decide to wake up after the end of a transmission based on a certain probability. The value of this probability and the sleep time of the duty cycle of LWT-MAC have been considered as the key parameters to tune. The optimization analysis aims at obtaining the optimal value of these parameters with the goal of minimizing the energy consumption but constrained to obtain the same throughput as the IEEE 802.11 MAC protocol. Results show that the optimal value of the probability to wake up after a transmission suddenly increases from 0 to 1 as soon as the traffic load starts to increase. In contrast, the optimal sleep time takes long values for low loads and decreases when the network load starts to increase. However, it is affected by the value of the probability to wake up after a transmission, and shows an inflection point when it changes to 1. These results are used to obtain a heuristic configuration. The heuristic configuration of the key parameters allows to obtain a reasonable good performance for a range of scenarios and traffic conditions without the cost associated to compute the best parameters to use depending on the scenario and traffic load. Obtaining the performance results for different conditions, it has been proposed a fixed configuration of 200 ms for the sleep time and to wake up always after a transmission. By implementing this configuration the protocol reduces its energy consumption at low loads (with the long sleep delay) but maintains its capability to rapidly react to instantaneous increases in the network load. It is important to note that this configuration addresses one of the main limitations of the LWT-MAC protocol, that is its higher energy consumption at low loads due to the idle listening after transmissions periods.

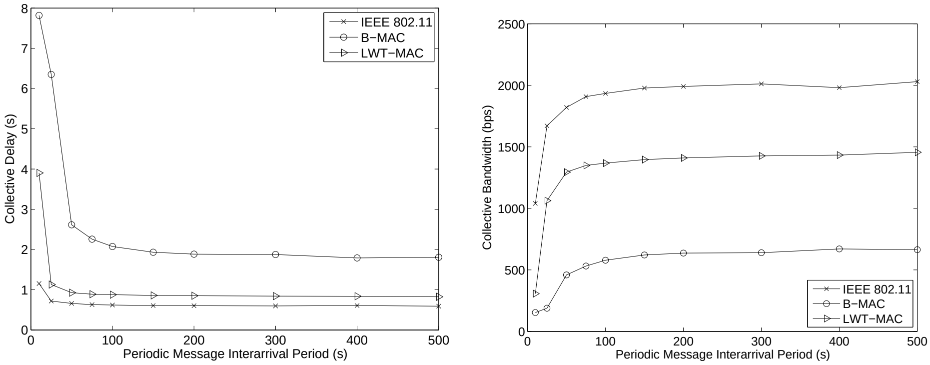
4.3.2 Multi Hop Scenario

To evaluate a MAC protocol for WSNs it is necessary, apart from the evaluation in a single hop network, to consider a multihop network. The performance evaluation of a multihop network allows to analyze the effects of the proposed techniques to mitigate the hidden terminal problems: the use of the RTS/CTS in the scheduled phase, the CA Timer and the *RTS if DATA fails* mechanism. In addition to that, event-based traffic and communication from sensor nodes to the sink are considered in order to evaluate a more realistic scenario.

The initial results in a multihop random network are provided in the article ***A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs***. It is shown that, even without a complete *synchronization* of sensor nodes, i.e., not all nodes are able to overhear all transmissions and then wake up at the end, the performance of the network is substantially improved. The throughput obtained considerably improves the one that results from using B-MAC. The end to end delay is also reduced, the values obtained are similar to the ones provided by the IEEE 802.11 protocol. And, finally, similarly to the single hop case, the energy consumption is slightly higher compared to B-MAC when the network load is low.

An interesting evaluation is to analyze the performance of the network with periodic and event-based traffic profiles. The article ***A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS*** provides an evaluation of the performance, with special interest in collective QoS metrics, of the LWT-MAC in a more realistic scenario. Results show that the proposed approach is able to significantly improve the collective QoS. It is shown how the LWT-MAC is able to: *i)* reduce the delay from event occurrence until its detection at sink, called Collective Delay (CD), *ii)* improve the fraction of total detected events among all events generated, the Collective Reliability (CR), and *iii)* improve the bandwidth observed by the messages related to the same event, known as

Collective Bandwidth (CB). As already seen in the previous scenarios, the energy consumption is notably reduced at high loads but slightly higher than the one obtained by B-MAC when the network load is low. As a summary of the results obtained we refer to Fig. 8, extracted from that article, that shows the CD and CB obtained with B-MAC, LWT-MAC and the 802.11 protocol. It can be observed that the LWT-MAC is able to improve the CD and CB compared to B-MAC, showing a performance close to the 802.11 protocol. In this case it has also been observed that at high traffic loads the CR of B-MAC is less than 50% while in LWT-MAC this value increases to more than 90%.



(a) Collective Delay

(b) Collective Bandwidth

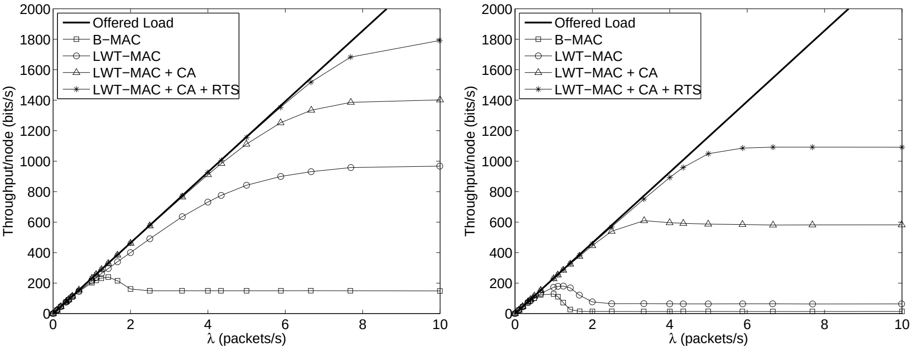
Fig. 8. Collective metrics of the multihop WSNs scenario with periodic and event-based traffic profiles

Another interesting analysis is the evaluation of the heuristic configuration found in ***Taking Advantage of Overhearing in Low Power Listening WSNs: A Performance Analysis of the LWT-MAC Protocol*** in a multihop scenario, also with event-based traffic profiles. As shown in the same article, the results found show that using the heuristic configuration the protocol provides considerable smaller energy consumption than B-MAC (configured with a lower sleep time) at low loads and substantially improves the collective QoS metrics. These results suggest that the increase in the sleep delay of the LWT-MAC is able to overcome one of the main limitations of the LWT-MAC protocol that is its higher energy consumption at low loads, even in a multihop network where not all the sensor nodes are able to sense each other.

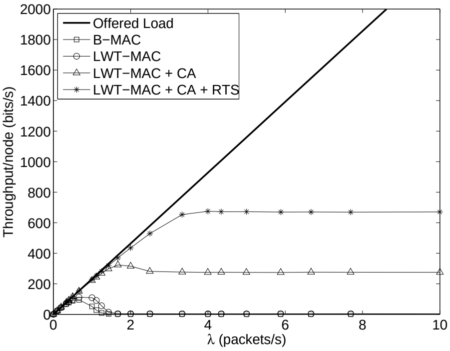
Finally, the last evaluation, aims at identifying the effects of each mechanism designed to cope with hidden terminals. The last article: ***Wake up after Transmissions and Reduced Channel Contention to Alleviate the Hidden Terminal Problem in Preamble Sampling WSNs*** evaluates the performance of the LWT-MAC protocol in the previously described key hidden terminal scenarios. An interesting issue is that different configurations of the LWT-MAC are analyzed: *i)* the basic LWT-MAC with wake up after transmissions, *ii)* the LWT-MAC with the basic wake up after transmissions and the CA Timer feature and *iii)* the LWT-MAC with the basic wake up after transmissions, the CA Timer feature and the *RTS if DATA fails* mechanism. Results, for all the scenarios considered, have shown that the basic LWT-MAC is able to improve the performance in presence of hidden terminals but only until a certain extent. Moreover, the results reveal that the use of the CA Timer allows to considerable improve the performance, even when there are event-based traffic profiles. And, finally, it is shown that the *RTS if DATA fails* mechanism slightly improves the performance of the network compared to the use of the CA Timer alone. An example of the results obtained is shown in Fig. 9 that depicts the throughput per node in the different n_h-hidden scenarios.

In this last article the effects of different CW values in the performance of the network when there are hidden terminals present have also been investigated. It has been found that the CW value should be set to a slight high value to take advantage of the RTS/CTS mechanism in the scheduled phase of LWT-MAC. This setting implies a higher energy consumption at low loads but as soon as the network load starts to increase it provides a considerable reduction of energy waste.

The different interactions of nodes in a multihop scenario and the complexity to model the designed mechanisms to cope with hidden terminals complicate the derivation of an analytical model in that case. For this reason, the evaluation in a multihop scenario has been done by means of simulations only. In this specific case, simulation studies are more efficient than analytical modeling due to the complexity of the latter.



(a) 1-Hidden - Throughput (b) 2-Hidden - Throughput



(c) 3-Hidden - Throughput

Fig. 9. Throughput in the n_{h}-hidden scenario.

5 Concluding Remarks and Future Research Lines

This thesis is focused on the study of the performance of WSNs when there is an increase of network load (due to an event detection or query dissemination and responses) and when there are hidden terminals trying to transmit nearly at the same time. It has been investigated how LPL, one of the most well-known MAC protocols for WSNs, reacts to these situations. Given that LPL uses a long preamble transmission to wake up the receiver it has been found that at medium-to-high loads the performance considerably decreases. At the same time, it has been demonstrated that hidden terminals notably degrade the performance of the system also due to the long preamble transmission that causes a high vulnerability time.

To improve the network performance in these specific cases, a new approach has been defined: the LWT-MAC protocol. The proposed MAC protocol is an extension of LPL that makes overhearing nodes to wake up at the end of a transmission to send or receive data without using the long preamble. Moreover, it implements a set of mechanisms to alleviate hidden terminal problems: *i)* the use of the RTS/CTS mechanism in the *scheduled* phase, *ii)* the CA Timer that reduces channel contention after a collision and *iii)* the *RTS if DATA fails* that sends an RTS immediately after a collision of a packet sent using the long preamble. In this thesis it has been evaluated how the LWT-MAC is able to improve the system performance in single hop, a set of key hidden terminal and multihop scenarios with periodic and event-based traffic profiles. Results have shown that the protocol is capable of increasing the throughput, decreasing the delay and improving collective QoS metrics compared with the basic LPL. One of the main drawbacks of the proposal is its slightly higher energy consumption at low loads caused by idle listening after transmission periods. However, this issue can be overcome by using a longer sleep time, thus making sensor nodes sleep more when the load is low but reacting to sporadic increases in the network load in a rapid manner.

One of the main issues that has been left as a future work is the comparison of the LWT-MAC with the most well-known extensions of LPL, specially those that adapt the duty cycle based on the traffic load. It is expected that the LWT-MAC will be able to react faster to increases in the network load if compared to the approaches summarized in the state-of-the-art. Moreover, it will be interesting to analyze and compare the complexity of the different solutions since it is an important requirement for a MAC protocol addressed to WSNs.

A complete implementation in real motes of the LWT-MAC protocol can also be an interesting future work. Although some tests have already been done with a simple version, a complete implementation and performance evaluation with all the defined features implemented is still required. It is expected that the results from the real testbed follow the same tendencies observed in the simulation and analytical modeling outcomes. However, testbed results can slightly differ from the results presented in this thesis due to channel errors and interferences.

Another interesting issue of study is the combination of the LWT-MAC protocol with a routing protocol specially addressed to WSNs. Cluster-based routing [4] can be a suitable combination for LWT-MAC, specially those protocols that do not require synchronization. Sensor nodes belonging to a cluster can send their information using LWT-MAC and therefore improve the collective QoS in case of event occurrence. Another interesting combination is the use of the LWT-MAC with data-centric based protocols [4] in which queries about specific data are disseminated and sensor nodes notify the sink in case the requested information is detected. In that case, the LWT-MAC will help in disseminating the query in a more efficient manner and also in sending the information of event detection to the sink since, in both cases an increase of network load occurs.

Finally, the adaptation of MAC protocols to work with multichannel operation is also another interesting area of research. If the penetration of sensor nodes in our daily lives reaches the expected value, the scarce frequency will be a serious problem [70]. For that reason, it is important to devise cognitive radio mechanisms for spectrum agile solutions, letting sensor nodes decide the best frequency band to use at each specific moment. This feature affects the MAC layer operation since it is necessary to ensure that a sensor node is awake to transmit to it but, in that case, it is also needed to know at which frequency it is listening.

References

1. IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11*, 1999 Edition (Revised 2007).
2. IEEE Std 802.15.4. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for low-rate wireless personal area networks. *IEEE Std 802.15.4*, 2003 Edition (Revised 2006).
3. J. Ai, J. Kong, and D. Turgut. An Adaptive Coordinated Medium Access Control for Wireless Sensor Networks. In *Proceedings of the International Symposium on Computers and Communications*, volume 1, pages 214–219, 2004.
4. K. Akkaya and M. Younis. A Survey on Routing Protocols for Wireless Sensor Networks. *Ad Hoc Networks*, 3(3):325–349, 2005.
5. M. Anwander, G. Wagenknecht, T. Braun, and K. Dolfus. BEAM: A Burst-Aware Energy-Efficient Adaptive MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 7th International Conference on Networked Sensing Systems*, 2010.
6. M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio. Increasing the Efficiency of Preamble Sampling Protocols for Wireless Sensor Networks. In *Proceedings of the First Mobile Computing and Wireless Communication International Conference (MCWC 06)*, pages 117–122, 2006.
7. M. Avvenuti and A. Vecchio. Adaptability in the B-MAC+ Protocol. In *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 946–951, 2008.

8. A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications (ICC 06)*, pages 3365–3370, 2006.
9. A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. MAC Essentials for Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 12(2):222–248, 2010.
10. B. Bellalta, M. Oliver, M. Meo, and M. Guerrero. A Simple Model of the IEEE 802.11 MAC Protocol with Heterogeneous Traffic Flows. In *Proceedings of the IEEE Eurocon*, November 2005.
11. L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto. A Wireless Sensor MAC Protocol for Bursty Data Traffic. In *Proceedings of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2007.
12. G. Bianchi. Performance Analysis of the IEEE 802. 11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.
13. L. Bing, Z. Lin, and Z. Huimin. An Adaptive Schedule Medium Access Control for Wireless Sensor Networks. In *Proceedings of the Sixth International Conference on Networking (ICN'07)*, pages 12–12, 2007.
14. R. Braynard, A. Silberstein, and C. Ellis. Extending Network Lifetime Using an Automatically Tuned Energy-aware MAC Protocol. In *Proceedings of the Third European Wireless Sensor Networks Workshop (EWSN 2006)*, pages 244–259, 2006.
15. M. Buettner, G.V. Yee, E. Anderson, and 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 (Sensys '06)*, pages 307–320, 2006.
16. M.M. Carvalho and J.J. Garcia-Luna-Aceves. Delay Analysis of IEEE 802.11 in Single-hop Networks. In *Proceedings of the 11th IEEE International Conference on Network Protocols*, pages 146–155, 2003.
17. S. Chatterjea, L. Van Hoesel, and P. Havinga. AI-LMAC: An Adaptive, Information-Centric and Lightweight MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference.*, pages 381–388, 2004.
18. D. Chen and P.K. Varshney. QoS Support in Wireless Sensor Networks: A Survey. In *Proceedings of the International Conference on Wireless Networks (ICWN'04)*, pages 227–233.
19. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
20. X.J. Dong and P. Varaiya. Saturation Throughput Analysis of IEEE 802.11 Wireless LANs for a Lossy Channel. *IEEE Communications Letters*, 9(2):100–102, 2005.
21. A. El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad hoc Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications. (ICC 02)*, volume 5, pages 3418–3423, 2002.
22. A. El-Hoiydi and J.D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. *Algorithmic Aspects of Wireless Sensor Networks*, pages 18–31, 2004.
23. S.C. Ergen and P. Varaiya. PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks. *IEEE Transactions on Mobile Computing*, 5(7):920–930, 2006.
24. G. Halkes and K. Langendoen. Crankshaft: An Energy-efficient MAC Protocol for Dense Wireless Sensor Networks. In *Proceedings of the 4th European Conference on Wireless Sensor Networks*, pages 228–244, 2007.

25. K. Han, S. Lim, S. Lee, J.W. Lee, and S. An. Signaling-Embedded Short Preamble MAC for Multihop Wireless Sensor Networks. In *Proceedings of the International Conference on Information Networking. Towards Ubiquitous Networking and Services*, pages 1–10. Lecture Notes in Computer Science, 2008.
26. J. Hill and D. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, pages 12–24, 2002.
27. P. Hurni and T. Braun. Increasing Throughput for WiseMAC. In *Proceedings of the Fifth Annual Conference on Wireless on Demand Network Systems and Services*, 2008.
28. P. Hurni and T. Braun. Maxmac: A Maximally Traffic-adaptive MAC Protocol for Wireless Sensor Networks. In *Proceedings ot the 7th European Conference, EWSN 2010.*, pages 289–305, 2010.
29. B. Jang, J.B. Lim, and M.L. Sichitiu. AS-MAC: An Asynchronous Scheduled MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. (MASS 2008)*, pages 434–441, 2008.
30. I. Joe and H. Ryu. A Patterned Preamble MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 16th International Conference on Computer Communications and Networks*, pages 1285–1290, 2007.
31. R. Jurdak, P. Baldi, and C. Videira Lopes. Energy-aware Adaptive Low Power Listening for Sensor Networks. In *Proceedings of the International Workshop for Networked Sensing Systems (INSS)*, 2005.
32. D. Kominami, M. Sugano, M. Murata, T. Hatauchi, and Y. Fukuyama. Performance Evaluation of Intermittent Receiver-driven Data Transmission on Wireless Sensor Networks. In *Proceedings of the 6th International Symposium on Wireless Communication Systems*, pages 141–145, 2009.
33. K. Kredo II and P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, 51(4):961–994, 2007.
34. P. Kumar, M. Gunes, Q. Mushtaq, and B. Blywis. A Real-time and Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the IFIP International Conference on Wireless and Optical Communications Networks, 2009. WOCN '09.*, 2009.
35. D. Lee and K. Chung. RA-MAC: An Energy Efficient and Low Latency MAC Protocol Using RTS Aggregation for Wireless Sensor Networks. In *Proceedings of the International Conference on Advanced Technologies for Communications*, pages 150–153, 2008.
36. Y. Li, W. Ye, and J. Heidemann. Energy and Latency Control in Low Duty Cycle MAC Protocols. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2005.
37. S. Lim, Y. Ji, J. Cho, and S. An. An Ultra Low Power Medium Access Control Protocol with the Divided Preamble Sampling. In *Proceedings of the Third International Symposium on Ubiquitous Computing Systems*, pages 210 – 224, 2006.
38. S. Lim, S. Kim, J. Cho, and S. An. Medium Access Control with an Energy-efficient Algorithm for Wireless Sensor Networks. In *Proceedings of the Personal Wireless Communications*, pages 334 – 343, 2006.
39. E.A. Lin, J.M. Rabaey, and A. Wolisz. Power-efficient Rendez-vous Schemes for Dense Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications (ICC 04)*, pages 3769–3776, 2004.

40. P. Lin, C. Qiao, and X. Wang. Medium Access Control with a Dynamic Duty Cycle for Sensor Networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'04)*, pages 1534–1539, 2004.
41. S. Liu, K.W. Fan, and P. Sinha. CMAC: An Energy-efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 5(4):1–34, 2009.
42. G. Lu, B. Krishnamachari, and C.S. Raghavendra. An Adaptive Energy-efficient and Low-latency MAC for Data Gathering in Wireless Sensor Networks. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium.*, pages 224–231, 2004.
43. S. Mahlknecht and M. Bock. CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks. In *Proceedings of the IEEE International Workshop on Factory Communication Systems*, pages 73–80, 2004.
44. D. Malone, I. Dangerfield, and D. Leith. Verification of Common 802.11 MAC Model Assumptions. *Passive and Active Network Measurement*, pages 63–72, 2007.
45. A. Meier, M. Woehrle, M. Zimmerling, and L. Thiele. ZeroCal: Automatic MAC Protocol Calibration. *Distributed Computing in Sensor Systems*, 1:1–14, 2010.
46. C.J. Merlin and W.B. Heinzelman. Network-aware Adaptation of MAC Scheduling for Wireless Sensor Networks. In *Proceedings of the Conference on Distributed Computing in Sensor Systems (Poster Session)*, pages 24–28, 2007.
47. C.J. Merlin and W.B. Heinzelman. Duty Cycle Control for Low-power-listening MAC Protocols. *IEEE Transactions on Mobile Computing*, pages 497–502, 2010.
48. C.J. Merlin and W.B. Heinzelman. Schedule Adaptation of Low-Power-Listening Protocols for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 9(5):672–685, 2010.
49. J. Na, S. Lim, and C.K. Kim. Dual Wake-up Low Power Listening for Duty Cycled Wireless Sensor Networks. *EURASIP Journal on Wireless Communications and Networking*, pages 1–12, 2008.
50. X. Ning and C.G. Cassandras. Dynamic Sleep Time Control in Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 6(3):1–37, 2010.
51. J. On, J. Kim, J. Lee, Y. Kim, and H. Chong. A MAC Protocol with Adaptive Preloads Considering Low Duty-cycle in WSNs. In *Proceedings of the 3rd International Conference on Mobile Ad-hoc and Sensor Networks*, pages 269–280, 2007.
52. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
53. I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, 2008.
54. J.W. Robinson and T.S. Randhawa. Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 22(5):917–928, 2004.
55. E.R. Sanchez, C. Chaudet, and M. Rebaudengo. Improving Preamble Sampling Performance in Wireless Sensor Networks with State Information. In *Proceedings of the Sixth International Conference on Wireless On-Demand Network Systems and Services*, 2009.
56. Y. Sankarasubramaniam, O.B. Akan, and I.F. Akyildiz. ESRT: Event-to-sink Reliable Transport in Wireless Sensor Networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad hoc Networking & Computing*, pages 177–188. ACM, 2003.

57. X. Shi and G. Stromberg. SyncWUF: An Ultra Low-Power MAC Protocol for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 6(1):115–125, 2007.
58. K. Stone and M. Colagrosso. Efficient Duty Cycling Through Prediction and Sampling in Wireless Sensor Networks. *Wireless Communications and Mobile Computing. Special Issue: Cognitive Radio, Software Defined Radio And Adaptive Wireless Systems.*, pages 1087–1102, 2007.
59. Y. Sun, O. Gurewitz, and 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 (Sensys '08)*, pages 1–14, 2008.
60. T. van Dam and K. Langendoen. An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys '03)*, page 171, 2003.
61. L. Van Hoesel and P. Havinga. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks. In *Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS)*, 2004.
62. L. Van Hoesel and P. Havinga. Poster Abstract: A TDMA-based MAC Protocol for WSNs. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 303–304, 2004.
63. H. Wang, X. Zhang, F. Naït-Abdesselam, and A. Khokhar. DPS-MAC: An Asynchronous MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 14th International Conference on High Performance Computing*, pages 393–404, 2007.
64. T. Watteyne, D. Barthel, M. Dohler, and I. Augé-Blum. A Cross-Layered Communication Architecture for WSNs based on Virtual Coordinates. *Technical Report.*
65. K.J. Wong and D.K. Arvind. SpeckMAC: Low-power Decentralised MAC Protocols for Low Data Rate Transmissions in Specknets. In *Proceedings of the 2nd International Workshop on Multi-hop Ad hoc Networks: From Theory to Reality*, pages 71–78, 2006.
66. W. Ye, J. Heidemann, and D. Estrin. An Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1567–1576, 2002.
67. W. Ye, J. Heidemann, and D. Estrin. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.
68. W. Ye, F. Silva, and J. Heidemann. Ultra-low Duty Cycle MAC with Scheduled Channel Polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, page 334, 2006.
69. X. Zhang, J. Ansari, and P. Mähönen. Traffic Aware Medium Access Control Protocol for Wireless Sensor Networks. In *Proceedings of the 7th ACM International Symposium on Mobility Management and Wireless Access - MobiWAC '09*, page 140, 2009.
70. G. Zhou, J.A. Stankovic, and S. Son. Crowded Spectrum in Wireless Sensor Networks. *IEEE EmNets*, 2006.

A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs

C. Cano, B. Bellalta, A. Sfairopoulou and J. Barceló

Universitat Pompeu Fabra

Abstract. This work presents a new MAC protocol for WSNs that combines an unscheduled channel access, based on low-power listening, with an opportunistic scheduled wake up after transmissions mechanism. The proposed MAC provides a good trade-off between energy consumption, complexity and performance in terms of throughput and delay. It achieves the benefits of scheduled MAC protocols without the cost of maintaining and sharing the schedule, as well as the low consumption and simple operation of unscheduled MAC protocols.

1 Introduction

Wireless Sensor Networks (WSNs) are formed by small and low cost devices that sense data from the environment and communicate it to a central device [1]. The most important constraint in WSNs is energy consumption as devices are usually battery-powered and the deployments are large and not easily accessible. The Medium Access Control (MAC) layer design is one of the most critical issues as it directly controls the transceiver operation, which is the most consuming component in a sensor node if compared to the processor or the sensor unit [4]. Traditional wired and wireless MAC protocols are not suitable for WSNs due to their high energy consumption. Their operation is based on continuously listen to the channel to potentially receive a message, which results in two main energy waste sources: idle listening (listening to the channel while nothing is transmitted) and overhearing (listening a message destined to another node). Additionally, in WSNs, also due to the energy constraint, it is even more important to limit the number of collisions and overheads than in wired and wireless traditional networks.

The most common approach to reduce the energy consumption in WSNs is to periodically turn the transceiver to sleep mode (duty cycle operation).

Two different categories can be identified depending on how the protocol decides when to sleep and wake up: scheduled and unscheduled MAC protocols. *Scheduled* MAC protocols organize nearby nodes to sleep and wake up at the same time. Then, a sender knows when to send a packet to a neighbour as it knows when it will be awake. Conversely, in *unscheduled* MAC protocols each node goes to sleep and wakes up independently of the other nodes. Mechanisms like the long preamble transmission are used to ensure the correct reception of messages.

Scheduled and unscheduled protocols have different advantages and disadvantages. Scheduled protocols limit the idle listening and overhearing at the cost of an increased complexity to maintain and share the schedule. On the other hand, unscheduled protocols are simpler (the processing and memory resources can be smaller) but they suffer from higher idle listening and overhearing.

In this work, a traffic-aware self-adaptive MAC protocol that combines the benefits of both approaches is presented. In low load conditions the unscheduled access is used while, as the load increases, a scheduled access is adopted, without schedule creation and sharing requirements. The unscheduled access is based on the Low Power Listening (LPL) Berkeley MAC (B-MAC) [6] protocol and a variation of the Sensor-MAC (S-MAC) [7] protocol is used in the scheduled access. With low traffic load, B-MAC shows better performance as the energy consumption and delay are reduced. However, as traffic grows, the continuous collision of preambles, specially in a multihop network, degrades significantly its performance. On the other hand, S-MAC with adaptive duty cycle can provide low delay, low energy consumption and a better hidden terminal management in high load conditions. The hybrid MAC protocol presented here, called Low power listening with Wake up after Transmissions MAC (LWT-MAC), combines the benefits of both approaches and can increase the efficiency in low and high traffic conditions. Results show that the throughput and delay are improved in single and multihop scenarios maintaining the low energy consumption.

The adaptability of the protocol is specially appealing for event-based WSNs. These networks usually work at low loads but it is a challenge to make them react to instantaneous increases of the network load. For periodic data, the protocol behaves as LPL, with small delay and energy consumption while at events occurrence, the delay is reduced and the hidden terminal problems are alleviated.

2 LPL MAC Protocol with Scheduled Wake Up after Transmissions

In B-MAC each sensor node goes to sleep and wakes up periodically and independently of the others. When a node wants to communicate with a neighbour, it first sends a long preamble that lasts an entire duty cycle (listen plus sleep time) and, just after that, it sends the message. The receiver detects the long preamble when it wakes up and remains awake in order to receive the message. As all the neighbouring nodes overhear the long preamble and the data message, or Request To Send (RTS)/Clear To Send (CTS) in case they are used, as it is an optional feature, they can know when the transmission ends.

The proposed protocol extends the B-MAC protocol taking advantage of the local synchronization of all nodes that overheard a transmission. The scheduled wake up after transmissions makes overhearing nodes to wake up simultaneously at the end of the ongoing transmission in order to send or receive packets without requiring the transmission of the long preamble. It is similar to the adaptive listening of S-MAC with the difference that here any node can send a packet, not only the recipient of the last transmission. In the scheduled phase, RTS/CTS messages are mandatory. They are used to allow overhearing nodes to go to sleep until the end of the current transmission by setting the Network Allocation Vector (NAV) timer to its duration. It is assumed that all nodes wake up immediately after a transmission, although a sleep period can be added to save more energy following the S-MAC design [7]. To prevent collisions, nodes wait a random backoff (BO) before sending an RTS packet (notice that then, the listen time should increase). In case nodes wake up after a transmission and nothing is received, they go to sleep during a random time (with maximum value equal to the sleep time) moving towards the unscheduled phase. A diagram of the process is depicted in Fig. 1. It is interesting to see that at low load conditions the protocol behaves similar to B-MAC while, as the load increases, neighbouring nodes wake up at the same time, allowing them to send frames without requiring the use of the long preamble, in a similar way to S-MAC. If the transmission of a packet fails, the protocol begins the retransmission procedure that differs depending on which phase it occurs:

Retransmission procedure in the unscheduled access. If a packet is not acknowledged during the unscheduled access, i.e., the acknowledgement (ACK) or the CTS - if the optional RTS/CTS mechanism is used - are not

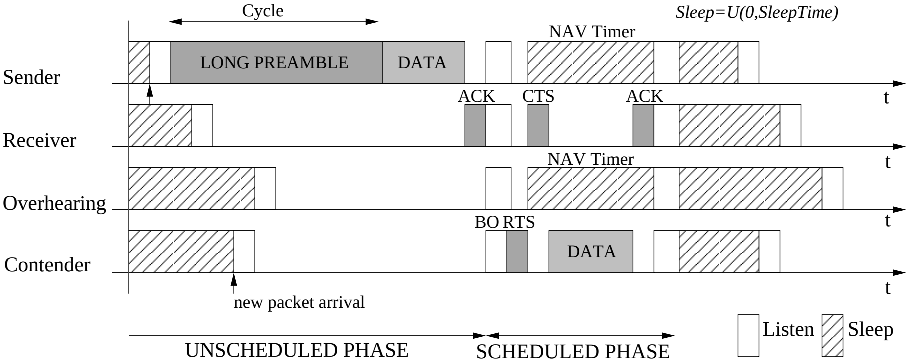


Fig. 1. LPL with scheduled wake up after transmissions

received, which can be caused by a collision between preambles, the sender will retry the transmission by sending an RTS after waiting a random BO. If, after that, the CTS is not received (two consecutive packet transmission failures), it is assumed that the intended recipient is either in another transmission or waiting for a transmission to finish. To alleviate more possible consecutive collisions, a Collision Avoidance (CA) timer is defined. During the CA timer (that is set to a value equal to a preamble and frame transmission) the node keeps listening to the channel as it will likely receive a message of the ongoing transmission. If that is the case, it can sleep until the transmission finishes and retry transmission using the scheduled method. Otherwise, if the CA timer expires, the node retries transmission using the long preamble again.

Retransmission procedure in the scheduled access. If a transmission fails during the scheduled access it can be either because of the CTS or the ACK are not received. In case the RTS fails (CTS not received), it is assumed that the recipient has probably not overheard the past transmission and, therefore, it is sleeping (notice that the long preamble has not been sent before). In this case, the node will retry to send the message by sending the long preamble first in order to wake up the receiver. If the packet transmission is not acknowledged the sender waits a CA timer as previously described in the unscheduled access.

3 Results

The SENSE simulator [3] has been used for performance evaluation purposes. It has been selected due to its extensibility to implement the new components required, such as the LPL extension of the physical layer and the evaluated MAC protocols: IEEE 802.11, B-MAC and the proposed LWT-MAC[1]. Additionally, the Floyd algorithm has been included to compute the shortest path between any pair of nodes in a multihop network.

Two different scenarios have been evaluated: *i)* a single hop scenario in which all nodes are inside the coverage range of each other and *ii)* a multihop scenario with 100 nodes randomly placed in a 100×100 m^2 area, where the radio range of each node is 43 m. All simulations have a duration of $100,000$ s. Table 1 shows the parameters used in the simulation, most of them have been extracted from the ns-2 implementation of S-MAC [5] and from the B-MAC specification [6]. Other considerations used in the simulations are: *i)* the RTS/CTS mechanism has been used in all MAC protocols for a fair comparison (for B-MAC and LWT-MAC the RTS is immediately sent after the long preamble) and *ii)* sensor nodes generate new packets following a Poisson distribution which are directed to the sink.

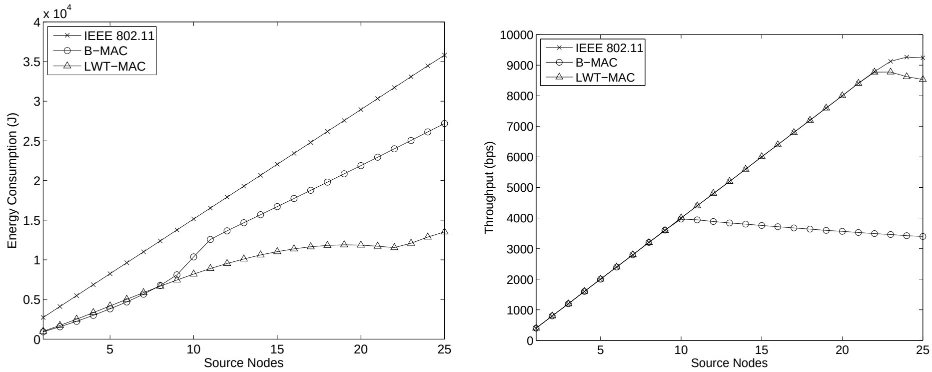
Table 1. Simulation Parameters

Parameter	Value	Parameter	Value
Data Rate	20 kbps	Listen/Sleep Time	24.5/75.5 ms
Slot	1 ms	Data Packet Size	100 bytes
DIFS	10 ms	Control Packet Size	8 bytes
SIFS	5 ms	Tx Consumption	24.75 mW
Retry Limit	5	Rx/Idle Consumption	13.5 mW
Queue Length	10 pkts	Sleep Consumption	15 μW
CW$_{\min}$ (B-MAC)	64	CW$_{\min}$ (802.11)	32
CW$_{\max}$ (B-MAC)	64	CW$_{\max}$ (802.11)	1024

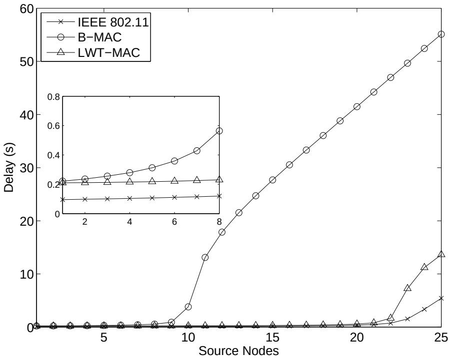
Fig. 2 shows the energy consumption, throughput and delay for the single hop scenario with different number of source nodes. All sensors generate packets with a rate $\lambda = 0.5$ packets/s. In throughput (Fig. 2(b)) and delay (Fig. 2(c)), LWT-MAC outperforms B-MAC, even showing a significantly

---

[1] Performance results of the original S-MAC has not been included in the comparison as it was shown that it provides worse results in throughput, energy consumption and delay compared to B-MAC [6]



(a) Consumption (b) Throughput

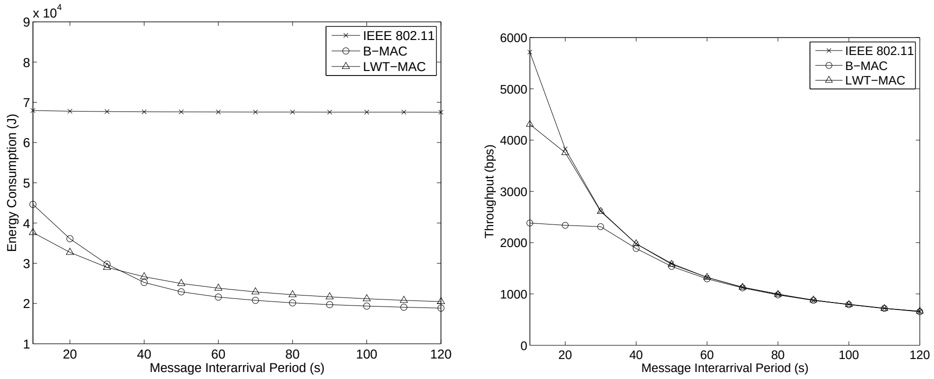


(c) Delay

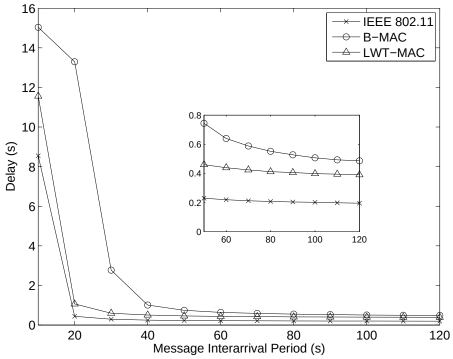
Fig. 2. Single hop scenario. Performance results as the number of source nodes increases.

lower energy consumption when the number of nodes increases (Fig. 2(a)) due to the wake up after transmissions mechanism, as it reduces the total preambles sent. Regarding the IEEE 802.11, LWT-MAC achieves very similar results in terms of throughput and delay with a significantly lower energy consumption.

In Fig. 3 the results of the multihop random scenario with different interarrival times are depicted. Fig. 3(a) shows that LWT-MAC consumes slightly more energy than B-MAC except when the traffic load is high. As all nodes are not perfectly synchronized in this scenario, the improvement of LWT-MAC in energy consumption appears because in B-MAC nodes are most of the time awake listening to preambles, which collide due to hidden terminal problems. LWT-MAC provides better performance in terms of delay and



(a) Consumption (b) Throughput



(c) Delay

Fig. 3. Multihop scenario. Performance results with different message interarrival times.

throughput (Fig. 3(b) and Fig. 3(c)) mainly due to the scheduled access and the CA timer. Specifically, the CA timer reduces the persistence of the protocol, increasing the performance with high traffic load and alleviating hidden terminal problems.

Results show that the LWT-MAC protocol provides a high performance while maintaining a low energy consumption for the considered scenarios. Although it cannot provide a perfect synchronization among all nodes in the network (only overhearing nodes get synchronized), the results prove that the performance in multihop scenarios where hidden nodes exist is improved. Providing a local (instead of global) synchronization eliminates the cost of creating and sharing the schedules making the protocol simpler and more efficient.

4 Conclusion

A new MAC protocol for WSNs that combines both an unscheduled channel access and a scheduled wake up after transmissions has been presented. By coordinating neighbouring nodes after transmissions, the resulting protocol can be more efficient in high load conditions and reducing hidden terminal problems. Results show a performance improvement, delivering higher throughput and smaller delay while maintaining the low energy consumption offered by LPL in low traffic conditions.

The traffic-aware adaptability and the simplicity of the protocol make it suitable for event-based WSNs that, at the same time, combine periodic readings and event messages, as it can react to sporadic changes in the traffic load, reducing the delay and maintaining a low energy consumption.

Finally, it is worth to mention that the scheduled wake up after transmissions can be combined with any B-MAC variant like the X-MAC [2] protocol in order to obtain a better energy efficiency.

References

1. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.
2. M. Buettner, G.V. Yee, E. Anderson, and 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 (Sensys '06)*, pages 307–320, 2006.
3. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
4. K. Kredo II and P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, 51(4):961–994, 2007.
5. NS2. Network Simulator 2. *http://www.isi.edu/nsnam/ns/*.
6. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
7. W. Ye, J. Heidemann, and D. Estrin. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.

A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS

C.Cano, B.Bellalta, J. Barceló and A.Sfairopoulou

Universitat Pompeu Fabra

Abstract. WSNs usually combine periodic readings with messages generated by unexpected events. When an event is detected by a group of sensors, several notification messages are sent simultaneously to the sink, resulting in sporadic increases of the network load. Additionally, these messages sometimes require a lower latency and higher reliability as they can be associated to emergency situations. Current MAC protocols for WSNs are not able to rapidly react to these sporadic changes on the traffic load, mainly due to the duty cycle operation, adopted to save energy in the sensor nodes, resulting in message losses or high delays that compromise the event detection at sink. In this work, two main contributions are provided: first, the collective QoS definitions are applied to measure event detection capabilities and second, a novel traffic-aware Low Power Listening MAC to improve the network response to sporadic changes in the traffic load is presented. Results show that the collective QoS in terms of collective throughput, latency and reliability is improved maintaining a low energy consumption at each individual sensor node.

1 Introduction

In Wireless Sensor Networks (WSNs) a group of nodes communicate environmental data to a central device, called sink [3]. Sensor nodes are formed by the sensor unit (that detects the data) and the communication module (that sends the data wirelessly). Usually, WSNs deployments are large and not easily accessible (for instance, a forest deployment). These two main characteristics impose several constraints on the sensor nodes: devices need to be small and inexpensive implying limited power sources, memory and processing resources. Thus, the most critical concern is the energy consumption, which must be minimized to achieve the expected network lifetime. Among all the tasks of a sensor node, communicating the data is the most consuming one due to the transceiver power consumption. Apart from the transceiver design, the Medium Access Control (MAC) protocol is the key factor influencing the transceiver operation [6]. This is the reason of the large amount of MAC protocols specially designed for WSNs. The common approach to reduce the energy consumption is to periodically put the transceiver into sleep mode, working in a low duty cycle operation, listening to the channel only a small percentage of time. Thus, this solution is able to increase the battery duration of the sensor nodes but degrades the network performance.

Usually, WSNs applications do not require Quality of Service (QoS) in terms of strict guarantees for the network delay and/or packet losses, specially in those WSNs dedicated to a simple monitoring task of a stable system. However, in an event-based WSN the messages related to the events occurrence require QoS guarantees in order to increase the probability that the event is properly detected at sink. Notice that, in this case, the QoS is related to the events (event detection delay, event detection reliability, etc.) and not to individual messages. To this aim, collective QoS is defined as the QoS in terms of throughput, latency, reliability and packet loss of the set of messages related to a certain event. Event-based common applications include target tracking, emergency detection and disaster relief among others.

There are several proposed MAC protocols which try to provide traditional QoS as well as low energy consumption. However, to the best of our knowledge, there is not any MAC protocol designed to provide collective QoS in WSNs as a primary objective. We will show that the protocol presented here improves the defined collective QoS metrics maintaining the energy consumption as a strong constraint, as it is designed to only react to the sporadic increases of traffic load due to the messages caused by an event detection.

The rest of the paper is organized as follows: the definitions of the collective QoS metrics are provided in Section 2, while in Section 3 an overview of the different MAC protocols for WSNs is presented, including those that take QoS into account. In Section 4, the protocol to increase the collective QoS in event-based WSNs is described and its results are discussed in Section 5. Finally, some concluding remarks are given.

2 QoS in WSNs: A Collective Approach

Due to the high energy constraint in WSNs, QoS (understood as in traditional communication networks) has not been given enough attention, as in most cases, the techniques used to reduce the energy consumption conflict with those used to guarantee the traditional QoS, such as the duty cycle operation at the MAC layer. Moreover, like other issues in WSN, QoS needs are highly application dependent. For instance, periodic readings usually do not need a strict grade of QoS since this kind of applications are normally non sensitive to high delays and can tolerate a certain percentage of packet loss. However, event-based applications need that the event-related messages arrive at sink with a certain grade of QoS. Notice that the QoS requirements differ from the traditional end-to-end ones where the metrics are measured packet by packet. The QoS should now be measured in a data-centric way, which is known as collective QoS. Collective QoS is defined as the QoS (delay, bandwidth, packet loss, etc.) of the set of packets related to a specific event [4]; i.e., it is not important the delay of the individual messages but it is crucial the latency from the event generation until the event detection. Here, we extend the collective QoS metrics to: Collective Delay (CD), Collective Bandwidth (CB), Collective Packet Loss (CPL) and Collective Reliability (CR).

CD refers to the time span between the event occurrence and the event detection at sink, assuming that the sink needs to receive Nr packets referring to an event to ensure the event occurrence. In this work, it is considered that the sink detects the event when it receives Nr packets related to it, independently of the time span. Therefore, the CD of event i is defined as follows [7]:

$$\mathrm{CD}_i = \mathrm{Tr}_i(\mathrm{Nr}) - \min\left(\mathrm{Td}_i(j)\right),\ j \,\epsilon\, \mathbf{SE}_i \tag{1}$$

where $\mathrm{Tr}_i(\mathrm{Nr})$ is the instant at which the Nr^{th} message of event i reaches the sink, $(\mathrm{Td}_i(j))$ is the moment at which the j sensor detects event i and $\mathbf{SE}_i$ is the group of sensors that detect the i^{th} event.

Similarly, the CB can be defined as the bandwidth required to detect an event:

$$\mathrm{CB}_i = \frac{\mathrm{Nr} \cdot L_{\mathrm{data}}}{\mathrm{CD}_i} \tag{2}$$

where L_{data} is the packet length.

The number of packets lost during the information delivery period, CPL, can be defined as follows:

$$\mathrm{CPL}_i = \mathrm{Nl}_i(t_{\mathrm{d}}),\ t_{\mathrm{d}} = [\min(\mathrm{Td}_i(j)),\ \mathrm{Tr}_i(\mathrm{Nr})],\ j \in \mathbf{SE}_i \tag{3}$$

where Nl_i is the number of packets lost of event i in the time interval t_{d} (from event detection by the sensor node to event detection at sink).

The number of packets lost can be linked to the CR, which is the fraction of correctly detected events among all events generated (G):

$$\mathrm{CR} = \frac{1}{G} \cdot \sum_{k=1}^{G} I(k),\ I(k) = \begin{cases} 1 & \text{if } k : (|\mathbf{SE}_i| - \mathrm{TPL}_i \geq \mathrm{Nr}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where TPL_i is the total number of packets lost related to event i and $|.|$ denotes the number of elements (cardinality) of the set.

3 MAC Protocols for WSNs

In the last decades several advances related to Wireless Networks have been made. In this sense, it is important to point out the great success of the IEEE 802.11 [1] standard that defines the physical and medium access control layers of a wireless node. The channel sharing method is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) technique in which nodes sense the channel and wait a random time before transmitting, reducing the number of collisions. Unfortunately, these protocols are not suitable for WSNs, mainly due to the limited energy resources available in sensor devices. Energy consumption of a WSN occurs in three different domains: sensing, data processing and communicating. Among these, radio communication is the major consumer of energy. In traditional wired and wireless MAC protocols nodes are always listening to the channel in order to receive possible transmissions, but in WSNs this feature will rapidly deplete the battery of sensor nodes. Apart from that, control messages, carrier sensing and acknowledgements, common in traditional MAC protocols, become noticeable overhead if compared to the small data payloads found in most

WSNs. To reduce the energy consumption, these sources of energy waste should be reduced [6]:

- Collisions: When a collision occurs none of the packets involved can be correctly received. Therefore, retransmissions, that cause an extra energy consumption, are needed.
- Idle Listening: Listening to the medium when there is nothing to receive. This effect has been identified as the major energy waste in WSNs [13].
- Overhearing: To receive a message from the wireless channel destined to another recipient.
- Overhead: Control messages and extra information in the data packet.

3.1 Common Energy-aware MAC Protocols

In order to reduce the energy consumption in WSNs the most common approach is to implement a low duty cycle MAC protocol that combines listen with sleep intervals. These protocols can be classified into two main categories: scheduled protocols (in which some kind of organization between nodes is made in order to decide when to sleep) and unscheduled (where each node independently selects its own schedule)[1].

3.1.1 Scheduled Protocols

Scheduled MAC protocols [6] reduce the energy waste by coordinating the sensor nodes with a common schedule (when to listen and sleep). In this kind of protocols nodes know when their neighbours will be awake to receive messages. Having an organization to access the channel limits the idle listening periods and the overhearing. The drawback of these mechanisms is the cost to create and maintain the schedule. Apart from that, synchronization becomes a problem because periodic beacons should be used or a higher precision oscillator has to be included in the sensor thereby increasing its cost.

The most representative and probably the most studied MAC protocol for WSNs is Sensor-MAC (S-MAC) [13]. In S-MAC, neighbouring nodes are

---

[1] Here the terms *scheduled* and *unscheduled* refer to the organization of the listen and sleep intervals not in the access to the channel.

synchronized together to the same schedule in order to reduce control overhead. The network is then divided into virtual clusters of nodes synchronized together. Nodes broadcast synchronism packets to allow neighbours to learn their schedules. Therefore, all nodes can communicate with all their neighbours although they have different schedules. This technique allows to reduce the energy waste due to idle listening. To avoid collisions, nodes use physical and virtual carrier sense and the Request To Send (RTS)/Clear To Send (CTS) mechanism.

In this category it can also be classified the IEEE 802.15.4 [2], a standard for small devices with small power resources and low data rates requirements. It defines two different types of access methods: the beacon-enabled and the nonbeacon-enabled modes. In the beacon-enabled mode the channel time is divided into an active part (formed by reserved slots and a contention access) and an inactive part (where the network coordinator can go to sleep). In the nonbeacon-enabled mode the channel access is based on an unslotted CSMA/CA where the coordinator is unable to sleep. Both modes allow end devices to sleep to save energy and wake up periodically to send and/or to poll the coordinator for data. The standard defines the star and peer-to-peer topologies, although it is focused in the star topology leaving some features of the peer-to-peer topology undefined. Peer-to-peer networks based on the IEEE 802.15.4 have several problems to work in medium-to-large WSNs, the beacon-enabled mode suffers from collisions of beacons from different coordinators and the nonbeacon-enabled mode entails high energy consumption requirements.

3.1.2 Unscheduled Protocols

On the other hand, unscheduled or random MAC protocols [6] have the advantage of their simplicity. Since there are no schedules to be maintained or shared, the node consumes fewer processing resources, it requires a smaller memory and the number of messages that have to be transmitted is reduced. Moreover, they are more flexible to support different types of traffic loads. However, there exist idle listening and overhearing.

The most representative protocol is Berkeley MAC (B-MAC) [12], in which each node selects a sleep schedule independently of the neighbourhood. Each time a node wants to send a packet, it first sends a preamble long enough to be listened by the intended recipient. To guarantee that the preamble and the listening period of the recipient overlap, the length of the preamble has to be equal to the duty cycle. If a node detects activity on the channel while it is listening, it will remain awake to receive the packet. Messages can be immediately transmitted if the carrier sense determines that the medium is idle. However, the long preamble transmission increases the energy spent in overhead, overhearing and the latency to send a packet.

3.1.3 Hybrid Protocols

As far as the authors know there is not any hybrid protocol that combines scheduled and unscheduled accesses. In this work a traffic-aware self-adaptive MAC protocol that combines the benefits of both approaches is presented. In low load conditions the unscheduled access is used while, as the load increases, a scheduled access after transmissions is adopted without requiring additional signalling for schedule creation and maintenance.

3.2 QoS-Aware MAC Protocols for WSNs

There are not any MAC protocol that considers collective QoS metrics. Those which are focused on QoS are based on end-to-end traditional QoS metrics instead. For instance, Priority-based MAC for providing QoS (PQ-MAC) [11] is a slotted protocol that defines different types of priority to access the channel. It consumes less energy than S-MAC but it suffers from a costly setup phase to assign a slot to each node. QoS-aware MAC (Q-MAC) [8] defines multiple queues on each sensor node to provide different service levels and uses different values of Contention Window (CW) depending on the packet criticality among other metrics. Finally, Reinforcement Learning MAC (RL-MAC) [9] adapts the duty cycle of S-MAC based on the inferred state of the other nodes and defines three different service levels with different CW.

4 LWT-MAC: LPL with scheduled Wake up after Transmissions

To improve the collective QoS, the MAC protocol should react to the sporadic increases in the network load due to an event detection without compromising the energy consumption. The protocol presented here combines an unscheduled access (used under low load) and a scheduled access (used at high load). The unscheduled access is based on the Low Power Listening (LPL) B-MAC [12] protocol and a variation of S-MAC [13] is used in the scheduled access. With low traffic load, B-MAC shows better performance as the energy consumption and delay are reduced. However, as traffic grows, the continuous collision of preambles, specially in a multihop network, significantly degrades its performance. On the other hand, S-MAC with adaptive duty cycle can provide low delay, low energy consumption and a better hidden terminal management in high load conditions.

The Low power listening with Wake up after Transmissions MAC (LWT-MAC) extends the normal operation of B-MAC by taking advantage of the local synchronization of all nodes that overhear a transmission. The scheduled wake up after transmissions wakes up overhearing nodes simultaneously at the end of the ongoing transmission in order to send or receive packets without requiring the transmission of the long preamble (see Fig. 1). In this scheduled phase, RTS/CTS messages are mandatory; they are used to allow overhearing nodes to go to sleep until the end of the current transmission by setting the Network Allocation Vector (NAV) timer to its duration. It is assumed that all nodes wake up immediately after a transmission, although a sleep period can be added to save more energy following the S-MAC design [13]. To prevent collisions, nodes wait a random backoff (BO) before sending an RTS packet, then, the listen time after transmissions should be equal to the maximum BO. In the case that nodes wake up after a transmission and nothing is received, they go to sleep during a random time (with maximum value equal to the sleep time) moving towards the unscheduled phase.

If the transmission of a packet fails, the protocol begins the retransmission procedure that differs depending on which phase it occurs:

4.1 Retransmission procedure in the unscheduled access

If a transmission failure occurs during the unscheduled access, the acknowledgement (ACK) is not received, the sender retransmits the packet by sending an RTS after waiting a random BO. As the transmission failure can be caused by collisions of preambles, the retransmission increases the probability to receive the RTS correctly. After that, in case the CTS is not received (there are two consecutive transmission failures), it is assumed that the intended recipient is either involved in another transmission or waiting

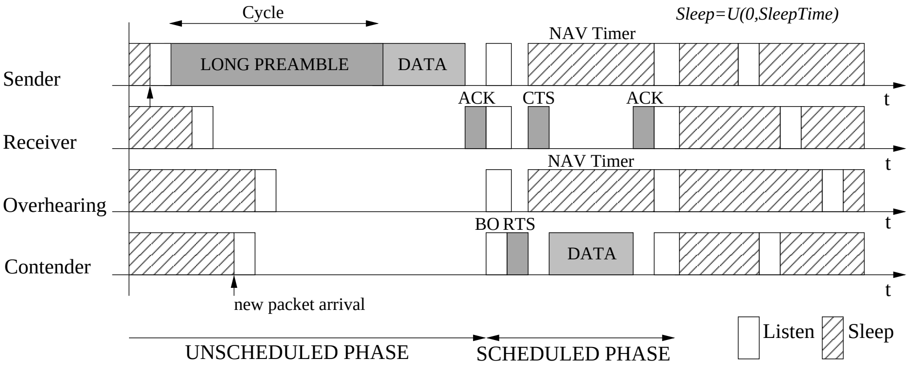


Fig. 1. LPL with scheduled wake up after transmissions

for a transmission to finish. In order to alleviate consecutive collisions, the sender does not immediately retransmit the message. Instead, it waits a Collision Avoidance (CA) timer (set to the duration of a preamble and a frame transmission). During the CA timer the node keeps listening to the channel, that provides the opportunity to get synchronized with the current ongoing transmission (if any) after overhearing a related message. If that is the case, it can sleep until the transmission finishes and retry transmission using the scheduled method (see Fig. 2). Otherwise, if the CA timer expires, the node retries transmission using the long preamble again.

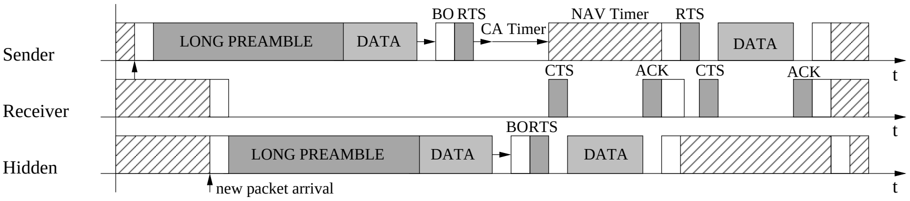


Fig. 2. Retransmission procedure in the unscheduled access

4.2 Retransmission procedure in the scheduled access

If a transmission fails during the scheduled access it can be either because the CTS or the ACK are not received. In case the RTS fails (CTS not received), it is assumed that the recipient has not overheard the past transmission and, therefore, it is sleeping (notice that the long preamble has not been previously sent). In this case, the node will retry to send the message by sending the long preamble first in order to wake up the receiver (see Fig. 3). If the packet transmission is not acknowledged the sender waits a CA timer as previously described in the unscheduled access.

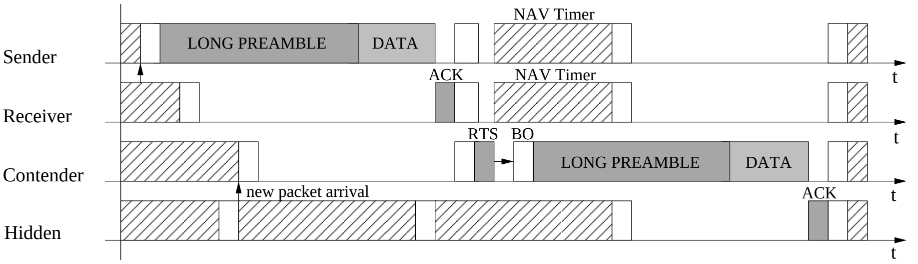


Fig. 3. Retransmission procedure in the scheduled access

Note that at instantaneous increases of the network load (for instance at events occurrence) the protocol reduces the time to send a packet and provides a better hidden terminal management if compared to B-MAC. This allows to increase the collective QoS of the event-based messages.

5 Results

To evaluate the performance of the presented protocol, the SENSE [5] simulator has been used. The LPL extension of the physical layer has been implemented as well as the compared MAC protocols: IEEE 802.11 (used as a reference only), B-MAC and the proposed LWT-MAC[2]. Additionally, to simulate a multihop network avoiding the effects of the routing protocol[3] the Floyd algorithm has been used to compute the shortest path between any pair of nodes. Finally, an event generator connected to the application

---

[2] Performance results of the original S-MAC have not been included in the comparison as it provides worse results in throughput, energy consumption and delay compared to B-MAC [12].

[3] Static routes are created at the beginning of the simulation avoiding route management traffic to influence the results.

layer of each node has been implemented. It generates random events and notifies those sensor nodes that are inside the coverage radius of the event in order to send event-based messages to the sink.

5.1 Scenario

The considered scenario (see Fig. 4) is a multihop event-based WSN with 100 nodes randomly placed in a 100×100 m^2 area. The radio range of each node is 43 m. All simulations have a duration of $500,000$ s. For a fair comparison, the RTS/CTS mechanism has been used in all MAC protocols. In the case of B-MAC and LWT-MAC, the RTS is immediately sent after the long preamble. Each sensor node generates two kind of traffic profiles: *i)* periodic messages generated following a Poisson distribution and *ii)* event-based messages that are generated if the sensor node is inside the coverage radius of the event. The offered load coming from the periodic data is changed while the number of events generated and their positions are maintained for all the simulations. Event positions are selected randomly inside the area and they have a constant coverage radius of 30 m. The time between events follows an exponential distribution with mean 600 s and Nr (the number of messages that are required to detect an event at sink) has been set to 5. Table 1 shows the parameters used in the simulation, most of them have been extracted from the ns-2 implementation of S-MAC [10] and from the B-MAC specification [12].

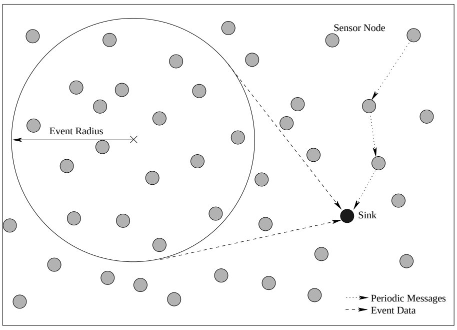


Fig. 4. Considered WSN scenario

Table 1. Simulation Parameters

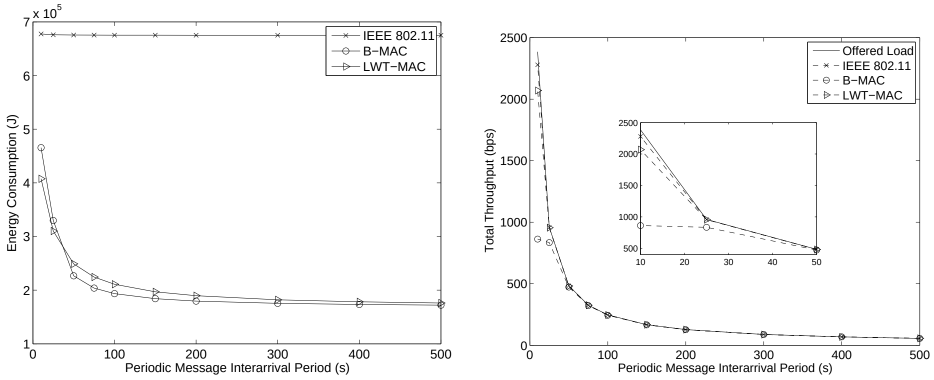
Parameter	Value	Parameter	Value
Data Rate	20 kbps	Listen/Sleep Time	24.5/75.5 ms
Slot	1 ms	Data Packet Size (L_{data})	30 bytes
DIFS	10 ms	Control Packet Size	8 bytes
SIFS	5 ms	Tx Consumption	24.75 mW
Retry Limit	5	Rx/Idle Consumption	13.5 mW
Queue Length	10 pkts	Sleep Consumption	15 μW
CW_{min} (B-MAC)	64	CW_{min} (802.11)	32
CW_{max} (B-MAC)	64	CW_{max} (802.11)	1024

5.2 Performance Results

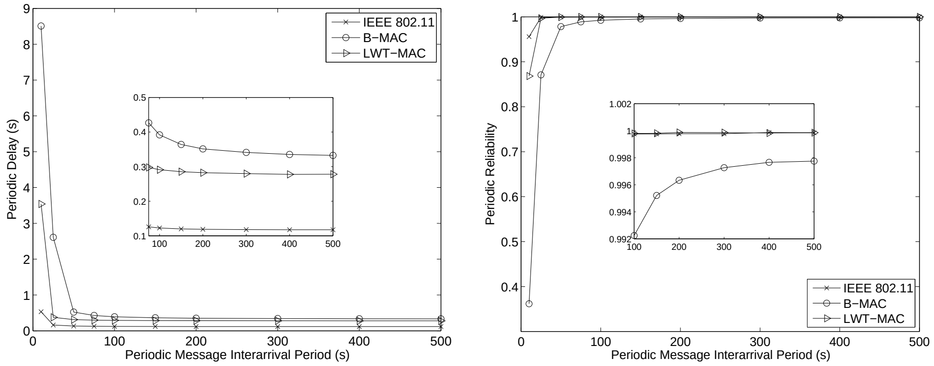
Fig. 5 and Fig. 6 show the performance results with different packet interarrival values for the periodic data. It can be seen that the energy consumption (Fig. 5(a)) of the LWT-MAC is slightly higher if compared to the B-MAC protocol except for high traffic loads. This is caused by the listen after transmissions mechanism, which at low traffic loads results in unnecessary listening times after transmissions as the probability that a neighbour node (including the node that has transmitted) has a packet ready to be transmitted is low. However, when traffic load increases, this probability is higher, and the listen after transmissions mechanism moves the network to the scheduled access, allowing to send messages without sending the long preamble and reducing the number of collisions. If compared to the IEEE 802.11 the energy reduction is noticeable.

Regarding the traditional metrics, throughput (Fig. 5(b)), delay (Fig. 5(c) and Fig. 5(e)) and reliability (Fig. 5(d) and Fig. 5(f)) for periodic and individual event-based messages, the results show that the LWT-MAC protocol provides a performance similar to the IEEE 802.11 protocol, delivering smaller delay (more noticeable in event-based messages) and higher throughput and reliability than B-MAC.

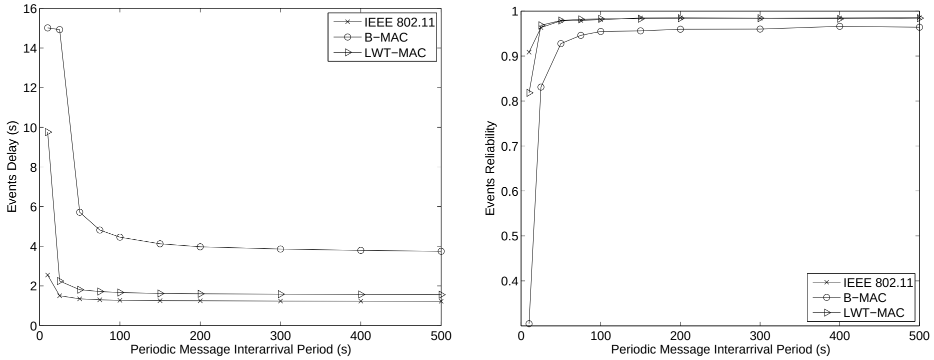
Fig. 6(a) and Fig. 6(b) as well as Table 2 show the CD, CB and CR metrics measured from the event-based messages of those events correctly detected. In Fig. 6(a) it can be seen how the LWT-MAC improves the CD if compared to the B-MAC protocol, showing a performance close to the IEEE 802.11. In Fig. 6(b), where the CB is depicted, LWT-MAC also shows a better performance compared to B-MAC. Regarding the CR (Table 2) a significant improvement is found with the LWT-MAC protocol, increasing



(a) Energy Consumption (b) Total Throughput



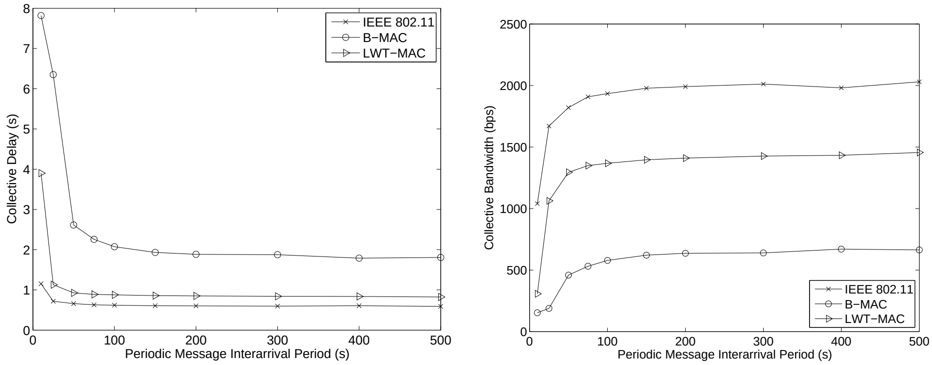
(c) Delay of Periodic Messages (d) Reliability of Periodic Messages



(e) Delay of Event-based Messages (f) Reliability of Event-based Messages

Fig. 5. Performance results of the multihop WSNs scenario with periodic and event-based traffic profiles

the percentage of events detected from less than 50% of the B-MAC to more than 90% when the traffic load is high (10s periodic interarrival time), for other loads the reliability is almost 100% in all the cases.



(a) Collective Delay (b) Collective Bandwidth

Fig. 6. Collective metrics of the multihop WSNs scenario with periodic and event-based traffic profiles

Table 2. Collective Reliability for 10 s and 25 s Periodic Interarrival Time

MAC Protocol	10 s	25 s
IEEE 802.11	0.998	0.999
B-MAC	0.490	0.997
LWT-MAC	0.986	0.999

6 Concluding Remarks

In this work a self-adaptive hybrid MAC protocol that combines unscheduled and scheduled accesses is presented. At low traffic loads the unscheduled access maintains a low energy consumption and when the load increases (for instance due to an event occurrence) the scheduled access provides small delay and higher throughput and reliability.

The wake up after transmissions allows to synchronize neighbouring nodes without the cost of creating, sharing and maintaining the schedule, apart from requiring lower synchronization capabilities if compared to the existing scheduled MAC protocols.

Results show that the LPL MAC protocol with scheduled wake up after transmissions is specially appealing for event-based WSNs as it can fit the requirements of periodic readings and react to sporadic changes in the network load. Moreover, it significantly improves the collective QoS in terms of delay, bandwidth and reliability while keeping a low energy consumption.

References

1. IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11*, 1999 Edition (Revised 2007).
2. IEEE Std 802.15.4. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for low-rate wireless personal area networks. *IEEE Std 802.15.4*, 2003 Edition (Revised 2006).
3. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.
4. D. Chen and P.K. Varshney. QoS Support in Wireless Sensor Networks: A Survey. In *Proceedings of the International Conference on Wireless Networks (ICWN'04)*, pages 227–233.
5. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
6. K. Kredo II and P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, 51(4):961–994, 2007.
7. X.Z. Lin, J.J. Zhou, and C.D. Mu. Collective Real-Time QoS in Wireless Sensor Networks. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'06)*, pages 1–4, 2006.
8. Y. Liu, I. Elhanany, and H. Qi. An Energy-efficient QoS-aware Media Access Control Protocol for Wireless Sensor Networks. In *Proceedings of the International Conference on Mobile Adhoc and Sensor Systems Conference*, pages 189–191, 2005.
9. Z. Liu and I. Elhanany. RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC'06)*, pages 768–773, 2006.
10. NS2. Network Simulator 2. *http://www.isi.edu/nsnam/ns/*.
11. K.J. Paek, J. Kim, U.S. Song, and C.S. Hwang. Priority-Based Medium Access Control Protocol for Providing QoS in Wireless Sensor Networks. *IEICE Transactions on Information and Systems*, 90(9):1448, 2007.
12. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
13. W. Ye, J. Heidemann, and D. Estrin. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.

Analytical Model of the LPL with Wake up after Transmissions MAC protocol for WSNs

C. Cano, B. Bellalta, J. Barceló, M. Oliver and A. Sfairopoulou

Universitat Pompeu Fabra

Abstract. The LWT-MAC protocol extends the normal operation of B-MAC by taking advantage of the local synchronization that can be achieved after each packet transmission. Sensor nodes wake up at the end of each successful transmission to send or receive messages eliminating the need for the long preamble transmission. Current LPL analytical models are extremely simple as they do not consider the energy waste due to collisions. In this work, a detailed analytical model of the LWT-MAC protocol, which considers the energy waste of collisions and overhearing for both saturated and unsaturated conditions, is presented. Results show a reduction of the energy consumption as the traffic increases and an improvement, in terms of throughput and delay, compared to B-MAC. Additionally, a performance evaluation is presented in which some key network parameters (check interval, traffic load and the number of sensor nodes) are modified in order to analyze the behavior of the studied LWT-MAC protocol.

1 Introduction

Wireless Sensor Networks (WSNs) consist of small devices that sense environmental data and send it to a central device. Normally, sensor nodes are low-cost devices with reduced processing, memory and battery resources deployed in remote and large areas. The battery replacement in these networks is, therefore, too costly or even impossible making energy consumption the most important constraint. For this reason, the Medium Access Control (MAC) protocol is crucial as it directly influences the transceiver operation that is the most consuming component of a sensor node. The common approach to reduce energy consumption in WSNs is to periodically put the transceiver into sleep mode, working in a low duty cycle operation instead of continuously listen the channel as in traditional wired and wireless networks without power constraints.

A well-known MAC protocol is Berkeley MAC (B-MAC) [6] in which each node periodically and independently of the others samples the radio channel to detect activity, what is known as Low Power Listening (LPL) operation. Then, when a node wants to send a message, it first sends a preamble long enough to overlap with the listen time (active part of the duty cycle) of the receiver. Note that, at very low loads, the energy consumption of the sensor nodes is extremely reduced. However, as the load increases (for instance, due to events occurrence) the collisions of preambles become a significant energy waste, even more important in large scale WSNs with hidden terminal problems.

The Low power listening with Wake up after Transmissions MAC (LWT-MAC) MAC protocol [3] was designed to maintain a low energy consumption at low loads while at the same time being able to react to instantaneous increases of the network load. A local synchronization after transmissions is adopted that ensures that all nodes that have overheard the last transmission will be awake to receive a new message, hence without requiring the long preamble transmission.

Analytical models of WSNs MAC protocols allow to derive performance optimizations of the different parameters involved: duty cycle, Contention Window (CW) or packet size among others, depending on different network scenarios. However, existing analytical models of WSNs MAC protocols (for instance, the ones used in [6] and [7]) are extremely simple as they only consider the time to transmit a packet without taking into account the time and energy wasted in collisions. Usually, WSNs work under low load conditions but it is important to consider the case of high traffic, specially in event-based WSNs. More detailed analytical models of scheduled MAC protocols such as the Sensor-MAC (S-MAC) [5] and nanoMAC [8] have been done, however, the LPL has not been studied so exhaustively.

In this work, the LWT-MAC protocol is studied from an analytical point of view. The presented LPL analytical model considers the energy waste due to collisions and overhearing in a single-hop network under saturated and unsaturated conditions. It models the behavior of the LWT-MAC protocol, although it can be easily adapted to model other LPL MAC protocols such as B-MAC.

The rest of the paper is organized as follows: Section 2 describes the LWT-MAC protocol, then in Section 3 the analytical model is presented. Results are discussed in Section 4 and finally some conclusions are given in Section 5.

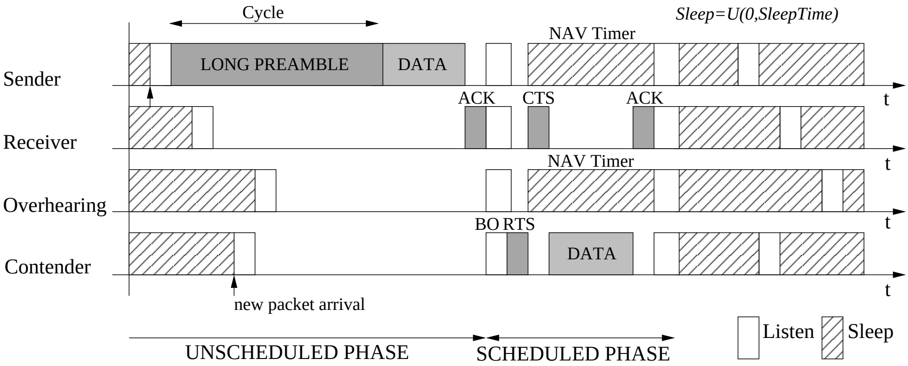


Fig. 1. LWT-MAC access mechanism

2 The LWT-MAC protocol

The LWT-MAC extends the operation of B-MAC by taking advantage of the local synchronization of all nodes that overhear a transmission [3]. The LWT-MAC defines that all overhearing nodes wake up simultaneously at the end of each successful transmission in order to send or receive packets. Since all nodes that have overheard the last transmission will be awake the long preamble is no longer necessary (Fig. 1). In this scheduled phase it is mandatory to compute a random backoff (BO) before attempting transmission and the use of the four-way-handshake. Notice that, the use of the random BO forces all nodes to listen after each transmission at least the value of the CW. If no message is sent during the listen after transmissions period, nodes go to sleep for a random time (with a maximum value equal to the sleep time of the duty cycle) moving towards the unscheduled phase.

Note that at instantaneous increases of the network load (for instance at events occurrence) the protocol reduces the time to send a packet improving the network performance. The behavior of the protocol with periodic and event-based traffic profiles and its ability to increase the collective Quality of Service (QoS) was studied in [2].

For illustration purposes the energy consumption and throughput of LWT-MAC compared with the results obtained using IEEE 802.11 and B-MAC in a 10-node single hop network are shown in Fig. 2 and 3 (the parameters used are depicted in Table 1). Observe that the throughput is significantly improved if compared to B-MAC at the cost of a slightly higher energy consumption at low loads, where idle listening after transmission periods occur. However, the energy consumption is reduced at high loads due to the suppression of the long preamble transmission.

Table 1. Default Parameters

Parameter	Value	Parameter	Value
r (data rate)	20 kbps	$T_{\text{listen}}/T_{\text{sleep}}$	24.5/75.5 ms
σ (empty slot)	1 ms	L_{data} (packet size)	1000 bits
DIFS	10 ms	$L_{\text{rts}}, L_{\text{cts}}, L_{\text{ack}}$	64 bits
SIFS	5 ms	E_{tx}	24.75 mW
CW	64	$E_{\text{rx}}, E_{\text{idle}}$	13.5 mW
K (queue size)	100 pkts	E_{sleep}	15 μW
R (retry limit)	7	T (time)	$1 \cdot 10^5$ s

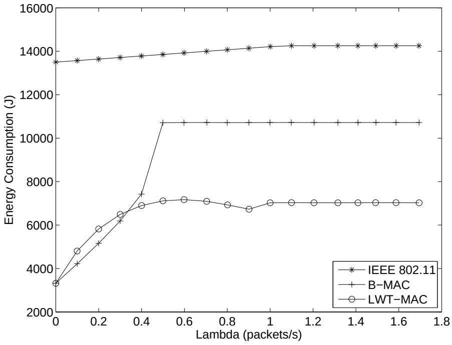


Fig. 2. Energy Consumption of IEEE 802.11, B-MAC and LWT-MAC

3 Analytical Model

3.1 Network Performance Metrics

The LWT-MAC analytical model is based on the one described in [1] where an IEEE 802.11 analytical model to compute traditional metrics such as throughput, delay and queue occupation is presented. The analytical model assumes ideal channel conditions (no channel errors or hidden terminal problems) and that each node computes a random BO (between 0 and CW)

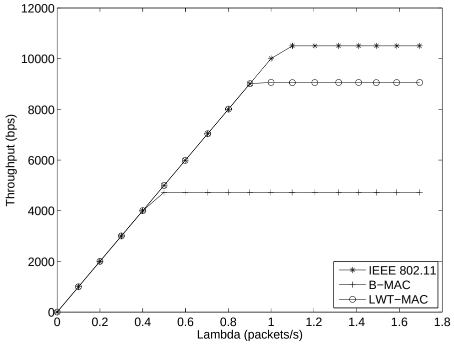


Fig. 3. Throughput of IEEE 802.11, B-MAC and LWT-MAC

before each transmission attempt[1]. For simplicity reasons the sensor nodes are considered to be homogeneous (equal traffic profiles and capabilities).

A sensor node is modeled as a single queue of length K packets. Each node generates packets following a Poisson distribution with rate λ packets/s and average packet length L_{data} bits. From these assumptions the following metrics can be computed:

$$A = \lambda X, \qquad \rho = A(1 - P_{\mathrm{b}}), \qquad P_{\mathrm{b}} = \frac{(1-A)A^K}{1 - A^{K+1}} \tag{1}$$

where A is the offered load, ρ is the queue utilization, P_{b} denotes the blocking probability and X refers to the service time (the time since the packet arrives at the head of the queue until it is released from it, assuming that it follows an exponential distribution).

The service time can be obtained as:

$$X = (M-1)\,(B\alpha + T_{\mathrm{c}}) + B\alpha + T_{\mathrm{s}} \tag{2}$$

where M is the average number of required transmission attempts for a packet, B is defined as the average number of slots selected before each transmission attempt, α is the average slot duration and T_{c} and T_{s} are the durations of a collision and a successful transmission respectively.

---

[1] The effect of the Collision Avoidance (CA) Timer described in [3] has not been studied, it is considered that collisions move the system to the unscheduled mode.

The average number of attempts per packet successfully transmitted or discarded due to maximum retry limit (R) reached is computed as:

$$M = \frac{1 - p^{R+1}}{1 - p} \tag{3}$$

where p is the conditional collision probability (assumed to be constant for all transmission attempts):

$$p = 1 - (1 - \tau)^{n-1} \tag{4}$$

Being n the total number of nodes in the network and τ the steady state probability that a node transmits in a random slot given that it has a packet ready to be transmitted:

$$\tau = \frac{\rho}{B + 1} \tag{5}$$

B is obtained as shown in Eq. 6.

$$B = \frac{\mathrm{CW} - 1}{2} \tag{6}$$

The average slot duration (α), is calculated considering the duration of the slot depending on the channel state (Eq. 7). As the channel is assumed to be error-free, it can only be in *empty*, $successful$ or *collision* states with their corresponding probabilities p_e, p_s, p_c.

$$\alpha = p_\mathrm{e}\sigma + p_\mathrm{s}(T_\mathrm{s} + \sigma) + p_\mathrm{c}(T_\mathrm{c} + \sigma) \tag{7}$$

where σ is the empty slot duration.

The channel state probabilities are related to the stationary probability that the rest of the nodes (except the one that is in BO) try to transmit in a given random slot. These are given by:

$$\begin{aligned} p_\mathrm{e} &= (1 - \tau)^{n-1} \\ p_\mathrm{s} &= (n - 1)\tau(1 - \tau)^{n-2} \\ p_\mathrm{c} &= 1 - p_\mathrm{e} - p_\mathrm{s} \end{aligned} \tag{8}$$

Finally, the throughput per node can be computed as:

$$S = \rho \frac{L_{\text{data}}}{X} \tag{9}$$

The channel occupation durations (Eq. 10 and 11) can be computed considering the length of the messages involved and the time intervals between them. Note that in the unscheduled mode the preamble is sent before each packet transmission attempt. The analytical model presented in this work only considers the transmission of packets preceded by Request To Send (RTS)/Clear To Send (CTS) messages, although the model is easily adaptable to also consider the basic access method.

$$T_{\text{c}} = \text{DIFS} + \frac{(L_{\text{p}} \cdot p_{\text{unsch}}) + L_{\text{rts}}}{r} + \text{EIFS} \tag{10}$$

$$T_{\text{s}} = \text{DIFS} + \frac{(L_{\text{p}} \cdot p_{\text{unsch}}) + L_{\text{rts}} + L_{\text{cts}} + L_{\text{data}} + L_{\text{ack}}}{r} + 3\text{SIFS} \tag{11}$$

where $L_{\text{p}}, L_{\text{rts}}, L_{\text{cts}}$ and L_{ack} are the lengths of the preamble, RTS, CTS and acknowledgement (ACK) messages respectively, while r refers to the transmission rate. The probability of transmitting a packet in scheduled mode (p_{sch}) is the probability that after a successful transmission, any other node has still data to be transmitted. On the other hand, the probability to transmit a packet in the unscheduled mode (p_{unsch}) is obtained as the complementary of the former:

$$p_{\text{sch}} = \frac{p_{\text{ss}}}{1 - p_{\text{es}}}(1 - (1 - \rho)^n), \qquad p_{\text{unsch}} = 1 - p_{\text{sch}} \tag{12}$$

where p_{ss} and p_{es} are the probabilities of successful and empty slots from the network point of view:

$$p_{\text{es}} = (1 - \tau)^n \qquad p_{\text{ss}} = n\tau(1 - \tau)^{n-1} \tag{13}$$

The analytical model is solved using a fixed point approximation. With those metrics the energy consumption of a sensor node during a certain amount of time can be computed.

3.2 Energy Consumption

The total energy consumption of a sensor node can be divided in four parts: *i)* the energy spent to transmit and *ii)* receive messages, *iii)* the energy wasted in overhearing, and *iv)* the energy spent in duty cycle (sleeping and waking up in inactive periods):

$$e = e_{\text{tx}} + e_{\text{rx}} + e_{\text{ov}} + e_{\text{dc}} \tag{14}$$

Let N_{s} be the total number of messages a node successfully sends during a time T, it can be derived using the throughput S and the packet length L_{data} as $N_{\text{s}} = T\,(S/L_{\text{data}})$.

The energy spent to transmit N_{s} messages is computed taking into account the energy needed to successfully transmit a packet ($e_{\text{s,tx}}$) and the energy spent in collisions for each unsuccessful attempt ($e_{\text{c,tx}}$):

$$e_{\text{tx}} = N_{\text{s}}\,(e_{\text{s,tx}} + (M-1)e_{\text{c,tx}}) \tag{15}$$

The values of $e_{\text{s,tx}}$ and $e_{\text{c,tx}}$ can be obtained considering the energy spent to transmit and receive the messages involved and the empty time intervals (Eq. 16 and 17). Moreover, the empty slots of the BO procedure must be considered (note that the busy slots of the BO countdown are part of the receiving or overhearing energy consumptions).

$$\begin{aligned} e_{\text{c,tx}} = {} & E_{\text{idle}}\left(\text{DIFS} + B\sigma p_{\text{e}} + 2\text{SIFS} + \frac{L_{\text{cts}}}{r}\right) \\ & + E_{\text{tx}}\frac{(L_{\text{p}} \cdot p_{\text{unsch}}) + L_{\text{rts}}}{r} \end{aligned} \tag{16}$$

$$\begin{aligned} e_{\text{s,tx}} = {} & E_{\text{idle}}(\text{DIFS} + B\sigma p_{\text{e}} + 3\text{SIFS} + T_{\text{e}}) \\ & + E_{\text{tx}}\frac{(L_{\text{p}} \cdot p_{\text{unsch}}) + L_{\text{rts}} + L_{\text{data}}}{r} \\ & + E_{\text{rx}}\frac{L_{\text{cts}} + L_{\text{ack}}}{r} \end{aligned} \tag{17}$$

where $E_{\text{tx}}, E_{\text{rx}}$ and E_{idle} denote the energy consumptions of being in transmission, reception and idle modes.

Notice that, after a successful transmission, all nodes in the network keep listening to the channel in case another node has something to transmit. If there are no more data to send, this time remains empty and each node stays in the idle mode during a T_{listen} (added to avoid synchronization problems) plus the CW, this is:

$$T_{\mathrm{e}} = (1-\rho)^n (T_{\mathrm{listen}} + \sigma \mathrm{CW}) \tag{18}$$

For simplicity reasons, it has been considered that each node receives N_{s} packet destined to it. Therefore, the total energy consumption to receive those messages is computed as:

$$e_{\mathrm{rx}} = N_{\mathrm{s}} \cdot e_{\mathrm{s,rx}} \tag{19}$$

The colliding packets have not been considered here. It is assumed that the unsuccessful transmissions of a node collide with those that are destined to it and that the probability to collide with more than one packet can be neglected.

The energy consumption to receive a packet is:

$$\begin{aligned} e_{\mathrm{s,rx}} = p_{\mathrm{sch}} \cdot e_{\mathrm{b}} + p_{\mathrm{unsch}} \cdot e_{\mathrm{p}} + E_{\mathrm{idle}}(3\mathrm{SIFS} + T_{\mathrm{e}}) \\ + E_{\mathrm{rx}} \left(\frac{L_{\mathrm{rts}} + L_{\mathrm{data}}}{r} \right) + E_{\mathrm{tx}} \left(\frac{L_{\mathrm{cts}} + L_{\mathrm{ack}}}{r} \right) \end{aligned} \tag{20}$$

where e_{b} is the energy of a busy listening after transmissions period, i.e., the idle time interval before sending a packet in the scheduled mode. Observe that, those nodes without data to send must also wait the remaining BO of the other nodes.

This time has been approximated by $B\sigma$ as shown in Eq. 21. This approximation does not affect when the traffic load is low (the scheduled probability is small) or high (the probability that a node has no data to transmit is negligible). However, it affects the results with moderate traffic load and its effect depends on the number of nodes. With high n the energy consumption is overestimated (as the probability that a node has a small remaining BO increases) while with n small the energy consumed is underestimated (since the probability to transmit a packet that arrives at the queue during the listen after transmission period increases).

$$e_{\mathrm{b}} = E_{\mathrm{idle}}(\mathrm{DIFS} + B\sigma(1-\rho)) \tag{21}$$

The parameter e_{p} refers to the energy spent receiving the long preamble. If a node has something to transmit it will be listening to the channel, therefore it will receive the entire long preamble of any other transmission in the medium. Otherwise, the node will be in duty cycle mode and on average it will wake up in the middle of the other's long preamble transmissions plus its own T_{listen}:

$$\begin{aligned} e_{\mathrm{p}} = \rho & \left(E_{\mathrm{idle}}\mathrm{DIFS} + E_{\mathrm{rx}}\frac{L_{\mathrm{p}}}{r} \right) \\ & + (1-\rho)\left(\frac{(E_{\mathrm{idle}}T_{\mathrm{listen}} + E_{\mathrm{rx}}\frac{L_{\mathrm{p}}}{r})}{2} \right) \end{aligned} \tag{22}$$

Similarly, the energy consumption due to overhearing is computed as:

$$e_{\mathrm{ov}} = N_{\mathrm{s}}(n-2)\left(e_{\mathrm{s,ov}} + \frac{M-1}{2}e_{\mathrm{c,ov}} \right) \tag{23}$$

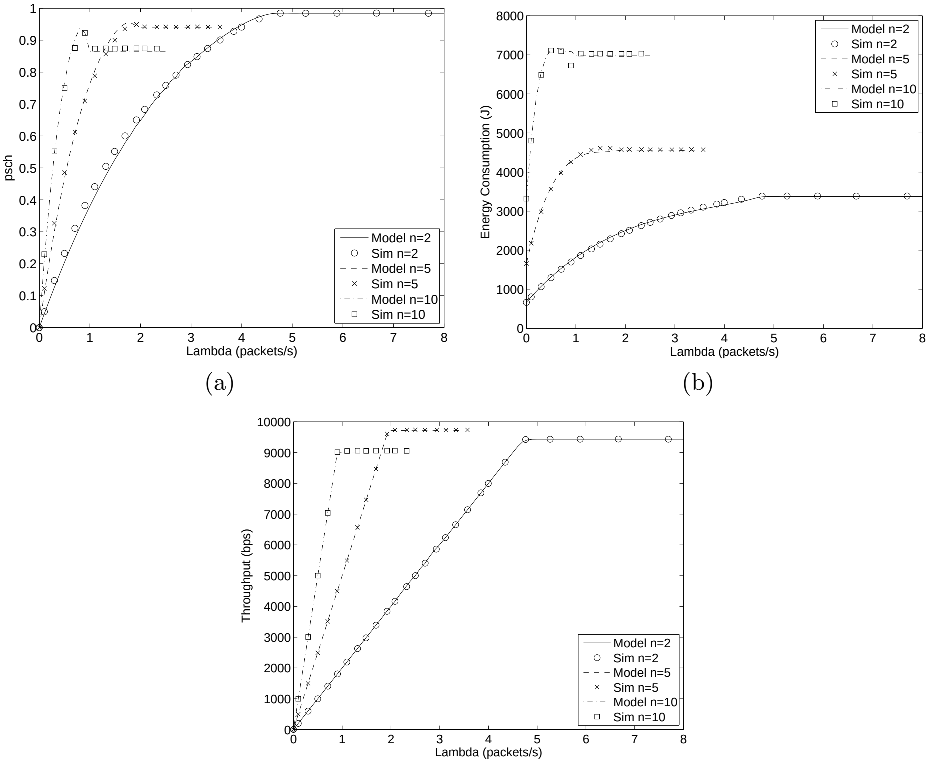
where $e_{\mathrm{s,ov}}$ and $e_{\mathrm{c,ov}}$ are the energy consumption to overhear a successful transmission or a collision respectively:

$$\begin{aligned} e_{\mathrm{c,ov}} = & \, p_{\mathrm{sch}} \cdot e_{\mathrm{b}} + p_{\mathrm{unsch}} \cdot e_{\mathrm{p}} + E_{\mathrm{rx}}\frac{L_{\mathrm{rts}}}{r} \\ & + E_{\mathrm{idle}}\left(2\mathrm{SIFS} + \frac{L_{\mathrm{cts}}}{r} \right) \end{aligned} \tag{24}$$

$$\begin{aligned} e_{\mathrm{s,ov}} = & \, p_{\mathrm{sch}} \cdot e_{\mathrm{b}} + p_{\mathrm{unsch}} \cdot e_{\mathrm{p}} + E_{\mathrm{idle}}T_{\mathrm{e}} + E_{\mathrm{rx}}\frac{L_{\mathrm{rts}}}{r} \\ & + E_{\mathrm{sleep}}\left(\frac{L_{\mathrm{cts}} + L_{\mathrm{data}} + L_{\mathrm{ack}}}{r} + 3\mathrm{SIFS} \right) \end{aligned} \tag{25}$$

where E_{sleep} denotes the energy of being in sleep mode.

Finally, the rest of time (T_{inactive}), that can be obtained using the equations above computing the time instead of the energy, and the total time T, each



(a)

(b)

(c)

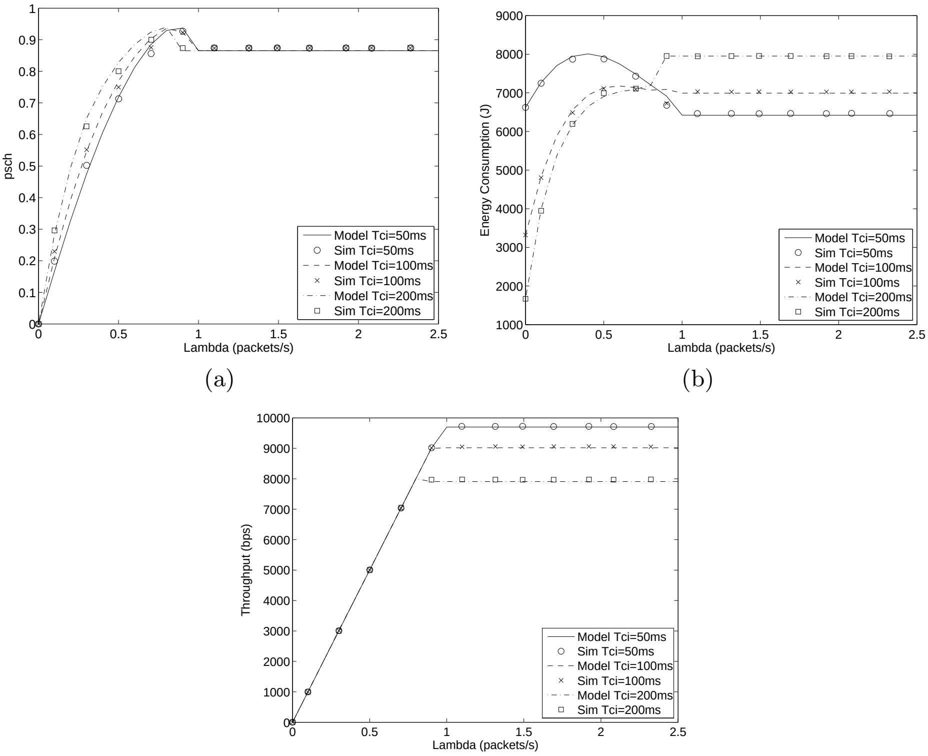
Fig. 4. Scheduled probability, total energy consumption and throughput with different number of sensor nodes

node performs a low duty cycle operation, listening and sleeping according to the duty cycle:

$$e_{\mathrm{dc}} = T_{\mathrm{inactive}} \left(E_{\mathrm{idle}} \frac{T_{\mathrm{listen}}}{T_{\mathrm{ci}}} + E_{\mathrm{sleep}} \frac{T_{\mathrm{sleep}}}{T_{\mathrm{ci}}} \right) \tag{26}$$

where T_{ci} is the check interval:

$$T_{\mathrm{ci}} = T_{\mathrm{listen}} + T_{\mathrm{sleep}} \tag{27}$$



(a)

(b)

(c)

Fig. 5. Scheduled probability, total energy consumption and throughput with different values of the check interval

4 Performance Evaluation

The SENSE simulator [4] has been used to compare the results obtained with the analytical model. The scenario consists of n nodes randomly placed in an area smaller than the maximum coverage range, thus assuring a full connectivity among them. The default parameters used for the evaluation are shown in Table 1 (see Section 2).

Fig. 4 shows the analytical and simulation results of LWT-MAC with different number of sensor nodes. Observe how the scheduled probability (Fig. 4(a)) increases with the traffic load until a certain point at which the number of collisions (that move the system to the unscheduled mode) is noticeable. With a higher number of nodes, the scheduled probability increases faster (since the traffic load is higher), however it also becomes constant earlier as the number of collisions is also higher. The total energy consumption curve (Fig. 4(b)) is strongly related to the scheduled probability. Note how the model underestimates the energy consumption approaching saturation for 2 and 5 nodes and overestimates it with 10 nodes. The reason for this discrepancy is the aforementioned approximation on the value of e_{b}. It is also interesting to observe the throughput (Fig. 4(c)), that increases with the number of nodes, however the saturation throughput decreases with 10 nodes as the probability of collision becomes higher.

Fig. 5 shows the results with 10 nodes and different check intervals (the T_{listen} value is maintained while T_{sleep} changes to fit the given check interval). The scheduled probability (Fig. 5(a)) remains equal for all the check intervals at high loads, however at lower loads it increases faster with higher values of the check interval. This difference occurs because with longer check intervals the probability that a node has received something to transmit during an ongoing transmission increases. In the energy consumption (Fig. 5(b)) it can be seen that at low loads the use of short time intervals notably penalizes the energy consumption as nodes wake up unnecessarily more often. However, as the load increases, to have longer check intervals implies higher energy consumption caused by the collisions that move the network to the unscheduled mode (moreover, after a collision nodes wait awake listening the entire preamble transmission). The same is valid for the throughput (Fig. 5(c)), which is lower when longer preambles are used.

5 Conclusion

In this work an analytical model of the LWT-MAC, an LPL MAC protocol for WSNs that takes advantage of the overhearing, is presented. The analytical model computes the network performance metrics and the energy consumption taking into account collisions in both saturated and unsaturated conditions. Results show that the behavior of the protocol strongly depends on the probability of being in scheduled mode. Varying different network and LPL parameters the performance in terms of energy and throughput of the LWT-MAC changes considerably. Thus, an interesting next step is to perform optimization studies to adapt the parameters to each network scenario.

References

1. B. Bellalta, M. Oliver, M. Meo, and M. Guerrero. A Simple Model of the IEEE 802.11 MAC Protocol with Heterogeneous Traffic Flows. In *Proceedings of the IEEE Eurocon*, November 2005.
2. C. Cano, B. Bellalta, J. Barceló, and A. Sfairopoulou. A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS. In *Proceedings of the 7th International Conference on Wired / Wireless Internet Communications (WWIC)*, 2009.
3. C. Cano, B. Bellalta, A. Sfairopoulou, and J. Barceló. A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs. *IEEE Communication Letters*, 13(4):221–223, 2009.
4. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
5. J. Haapola. Multihop Medium Access Control for WSNs: An Energy Analysis Model. *EURASIP Journal on Wireless Communications and Networking*, 2005(4):523–540, 2005.
6. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
7. W. Ye, J. Heidemann, and D. Estrin. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.
8. Y. Zhang, C. He, and L. Jiang. Performance Analysis of S-MAC Protocol under Unsaturated Conditions. *IEEE Communications Letters*, 12(3):210–212, 2008.

Taking Advantage of Overhearing in Low Power Listening WSNs: A Performance Analysis of the LWT-MAC Protocol

C. Cano, B. Bellalta, A. Sfairopoulou, M. Oliver and J. Barceló

Universitat Pompeu Fabra

Abstract. LWT-MAC is a new Low Power Listening MAC protocol for WSNs designed to rapidly react to instantaneous increases of the network load. It takes advantage of overhearing by waking up all nodes at the end of a transmission to send or receive packets without needing to transmit the long preamble before. In this work, detailed analytical models of the LWT-MAC and B-MAC protocols, for both saturated and unsaturated conditions, are presented. Moreover, the key LWT-MAC parameters are optimized in order to minimize the energy consumption, constrained to obtain the same throughput as the IEEE 802.11 (CSMA/CA) MAC protocol. From the behavior of the optimal LWT-MAC parameters, a heuristic configuration is proposed. Finally, the LWT-MAC is compared to B-MAC, in both single and multi-hop scenarios, showing improvements in energy consumption, throughput and delay.

1 Introduction

Wireless Sensor Networks (WSNs) consist of small devices that sense environmental data and send it to a central collector. Normally, sensor nodes have reduced processing, memory and battery resources and are deployed in remote and large areas. The battery replacement in these networks is, therefore, too costly or even impossible, making energy consumption the most important constraint. For this reason, the Medium Access Control (MAC) protocol is crucial as it directly influences the transceiver operation that is the most consuming component of a sensor node. The common approach to reduce energy consumption in WSNs is to periodically put the transceiver into sleep mode, working in a low duty cycle operation instead of continuously listening to the channel as in traditional wired and wireless networks without power constraints, like in IEEE 802.11 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [1].

A well-known MAC protocol for WSNs is Berkeley MAC (B-MAC) [17] in which each node periodically and independently of the others samples the radio channel to detect activity, what is known as Low Power Listening (LPL) operation. Then, when a node wants to send a message, it first sends a preamble long enough to overlap with the listening time (active part of the duty cycle) of the receiver. Using B-MAC the energy consumption of the sensor nodes is extremely reduced at very low loads. However, as the load increases, for instance, due to events occurrence, the collisions of preambles become a significant energy waste, even more important in large scale WSNs with hidden terminal problems.

The Low power listening with Wake up after Transmissions MAC (LWT-MAC) protocol (presented by the authors in [7]) was designed to maintain a low energy consumption at low loads while, at the same time, being able to react to instantaneous increases of the network load. A local synchronization after transmissions, that ensures that all nodes that have overheard the last transmission will be awake to receive a new message, is adopted, hence without requiring the long preamble transmission.

Analytical models of WSNs MAC protocols allow to derive performance optimizations of the different parameters involved: duty cycle, Contention Window (CW) or packet size among others, depending on different network scenarios. However, existing analytical models of WSNs MAC protocols (for instance, the ones used in [17] and [20]) are extremely simple as they only consider the time to transmit a packet without analyzing the time and energy wasted in collisions. Although these simple analytical models are valid for low traffic loads they become inaccurate when the traffic load increases. More detailed analytical models of scheduled MAC protocols such as the Sensor-MAC (S-MAC) [11] and nanoMAC [22] have been presented, however, the LPL operation has not been studied so exhaustively.

In this work, the LWT-MAC protocol is studied from an analytical point of view. The presented LWT-MAC analytical model considers the energy waste due to collisions and overhearing in a single-hop network under saturated and unsaturated conditions. The analytical model presented is adapted to model the B-MAC protocol and it can also be extended to model other LPL MAC protocols (like the X-MAC protocol [4]). Using the analytical model, the LWT-MAC key network parameters are identified and optimized to minimize the energy consumption but maintaining a good performance in terms of delay and throughput. From the optimization process, a heuristic configuration is derived. The LWT-MAC with the heuristic configuration is compared with B-MAC in a single-hop network and in a multi-hop scenario with periodic and event-based traffic profiles.

The rest of the paper is organized as follows: Section 2 provides a comparison of the proposed approach with similar existing mechanisms, then, in Section 3 the LWT-MAC is described. The LWT-MAC analytical model and the adaptation to model the B-MAC protocol, as well as their validation are presented in Section 4. The optimization analysis of the LWT-MAC protocol is performed in Section 5 while the performance results are discussed in Section 6. Finally, some concluding remarks are given in Section 7.

2 Related Work

LPL MAC protocols as B-MAC [17] perform well when the traffic load of the network is low, however as the traffic load increases the continuous collisions of preambles considerably decreases the network performance. The LWT-MAC protocol aims at improving the performance by waking up neighboring nodes at the end of a transmission in order to send or receive packets [7].

Some similar mechanisms have already been defined in order to address the same problem. For instance, the multi-hop streaming capability defined in the Scheduled Channel Polling MAC (SCP-MAC) protocol [21] that increases the duty cycle upon a message reception in order to reduce the end-to-end delay. Another example is described in [10] where the sender of a message activates a *send more* bit in the header of the packet indicating that there is more data pending to be transmitted. The destination of the message stays awake at the end of the transmission to receive the packet, thus eliminating the need of the long preamble transmission. In scheduled MAC protocols the same idea can be followed: in S-MAC [20] and Timeout-MAC (T-MAC) [18] each node that overhears an Request To Send (RTS) or Clear To Send (CTS) stays awake just in case it should forward the message.

All those mechanisms differ from the approach presented in this work in the sense that in LWT-MAC *every* node that has overheard a transmission is allowed to transmit at the end, not only the sender or the recipient of the last transmission. This approach improves the performance when the traffic load of the network increases. The load of the network can increase, for instance, due to an event detection, in this situation a set of nearby nodes try to transmit a packet to inform the sink about the event occurrence.

Other approaches like the X-MAC protocol [4] aim at estimating the traffic load of the network and then adapt the duty cycle accordingly. Those protocols provide better performance but they suffer from an increased complexity. The LWT-MAC protocol, in contrast, adapts to the traffic load in a simple manner, only by waking up all neighboring nodes after a transmission.

3 Taking Advantage of Overhearing: The LWT-MAC Protocol

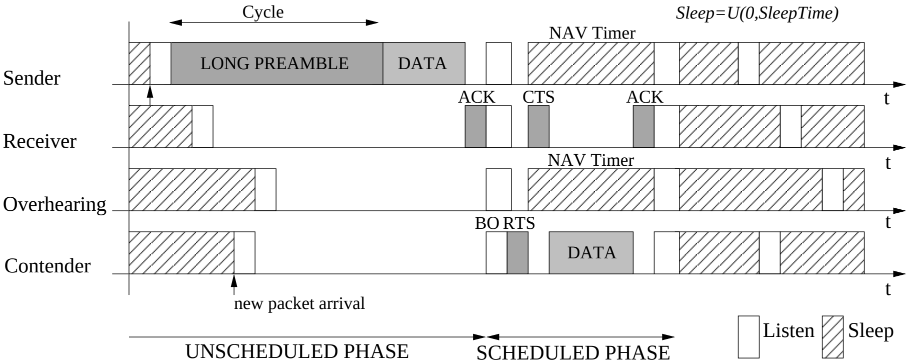
The LWT-MAC extends the operation of B-MAC by taking advantage of the local synchronization of all nodes that overhear a transmission [7]. The LWT-MAC defines that all overhearing nodes wake up simultaneously at the end of each successful transmission in order to send or receive packets. Since all nodes that have overheard the last transmission will be awake, the long preamble is no longer necessary (the medium access mechanism is depicted in Fig. 1(a)).

In this scheduled phase it is mandatory to compute a random backoff (BO) before attempting transmission, however in the unscheduled phase it is optional. This forces all nodes to listen after each transmission at least the value of the CW. During the random BO of the scheduled phase nodes should keep listening to the channel in case any other node starts a transmission.

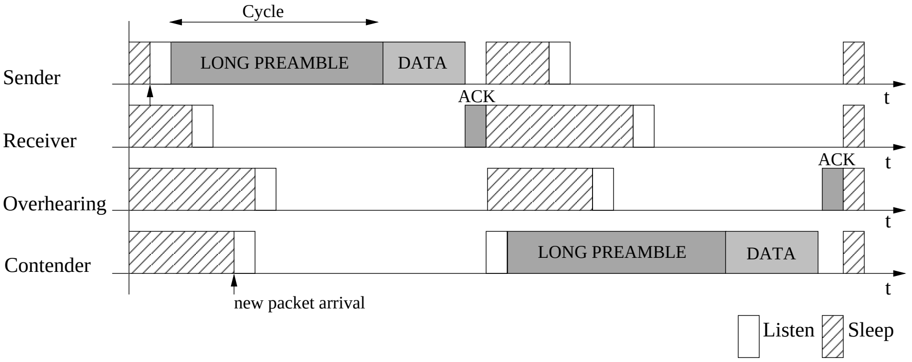
The use of the RTS/CTS transaction before the packet transmission in the scheduled phase is recommended. The four-way handshake will help in alleviating hidden terminal problems and it is useful to let overhearing nodes sleep during the entire packet transmission. The duration of the transmission should be included in RTS, CTS and Data messages and will allow the overhearing nodes to set a Network Allocation Vector (NAV) timer (as defined in [1]) and sleep until the transmission finishes.

If no message is sent during the listening after transmissions period, nodes go to sleep for a random time (with a maximum value equal to the sleep time of the duty cycle) moving towards the unscheduled phase.

Observe that, compared to B-MAC (Fig. 1(b)), there is a reduction of the energy waste, both in transmission and reception, due to the suppression of continuous long preamble transmissions. Moreover, by suppressing long preambles the delay and the channel occupation durations are reduced as well. Additionally, at instantaneous increases of the network load the protocol reduces the time required to send a packet improving the network performance. However, it should be noted that the performance benefits of LWT-MAC mainly depend on the probability that the next transmission receiver has overheard the last transmission, that in turn depends on the topology and on how the routes to the sink are selected.



(a) LWT-MAC Protocol



(b) B-MAC Protocol

Fig. 1. Comparison of LWT-MAC and B-MAC Medium Access Mechanisms

If a packet transmission fails, a retransmission procedure is initiated. The details of this procedure are different in the scheduled and unscheduled modes of operation.

3.1 Retransmission procedure in the unscheduled access

If a transmission failure occurs during the unscheduled access, the acknowledgement (ACK) is not received, the sender retransmits the packet by sending an RTS after waiting a random BO. As the transmission failure can be caused by collisions of preambles, the retransmission increases the probability to receive the RTS correctly. After that, in case the CTS is not received (there are two consecutive transmission failures), it is assumed that the intended recipient is either involved in another transmission or waiting for a transmission to finish. In order to alleviate consecutive collisions, the sender does not immediately retransmit the message. Instead, it waits a Collision Avoidance (CA) timer (set to the duration of a preamble and a frame transmission). During the CA timer the node keeps listening to the channel, that provides the opportunity to get synchronized with the current ongoing transmission (if any) after overhearing a message involved. If that is the case, it can sleep until the transmission finishes and retry transmission using the scheduled method. Otherwise, if the CA timer expires, the node retries transmission using the long preamble again. The CA timer has been added as an optional mechanism to reduce hidden terminal problems, however, it can be deactivated when needed, for instance in single-hop networks in which all nodes are inside the coverage range of the others.

The RTS/CTS mechanism can also be activated in the unscheduled mode [17], i.e., an RTS is sent after the transmission of the long preamble. In this case, if a transmission failure occurs (CTS not received) the sender assumes that the recipient is involved in another transmission and waits the CA timer as previously described. Note that, after a collision among hidden terminal nodes, the receiver will hopefully receive at least one of the RTS sent. By activating the RTS/CTS in the unscheduled mode the use of the NAV timer will also allow to let overhearing nodes to sleep as previously explained.

3.2 Retransmission procedure in the scheduled access

If a transmission fails during the scheduled access it can be either because the CTS or the ACK are not received. In case the RTS fails (CTS not received), it is assumed that the recipient has not overheard the past transmission and, therefore, it is sleeping (notice that the long preamble has not been previously sent). In this case, the node will retry to send the message by sending the long preamble first in order to wake up the receiver. If the packet transmission is not acknowledged the sender waits a CA timer as already explained in the unscheduled access.

The detailed specification of the protocol behavior is depicted in Fig. 2.

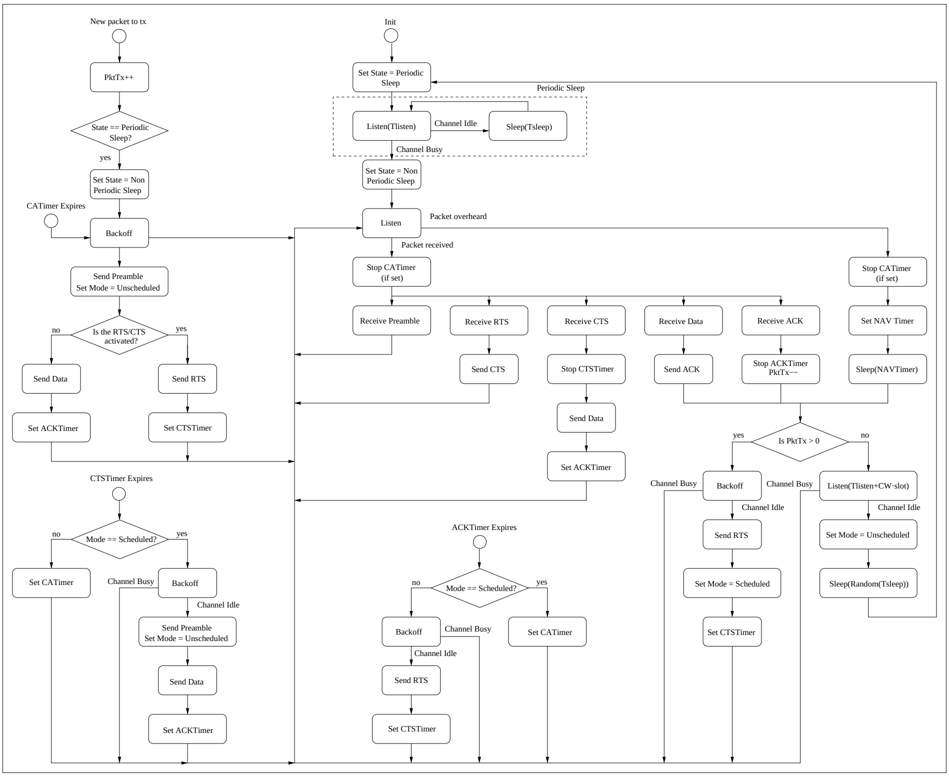


Fig. 2. Flowchart of the LWT-MAC protocol. The *Mode* value is initialized to *Unscheduled* and *PktTx* is initialized to zero

3.3 Performance Benefits

For illustration purposes the energy consumption and throughput of LWT-MAC compared with the results obtained using IEEE 802.11 [1] and B-MAC in a 10-node single hop network are shown in Fig. 3 and 4 (the parameters used are depicted in Table 1). Observe that the throughput is significantly improved if compared to B-MAC at the cost of a slightly higher energy consumption at low loads, where idle listening after transmission periods occur. However, the energy consumption is reduced at high loads due to the suppression of the long preamble transmission.

Table 1. Default Parameters

Parameter	Value	Parameter	Value
r (data rate)	20 kbps	$T_{\text{listen}}/T_{\text{sleep}}$	24.5/75.5 ms
σ (empty slot)	1 ms	L_{data} (packet size)	1000 bits
DIFS	10 ms	$L_{\text{rts}}, L_{\text{cts}}, L_{\text{ack}}$	64 bits
SIFS	5 ms	E_{tx}	24.75 mW
CW	64	$E_{\text{rx}}, E_{\text{idle}}$	13.5 mW
K (queue size)	100 pkts	E_{sleep}	15 μW
R (retry limit)	7	T (time)	$1 \cdot 10^5$ s

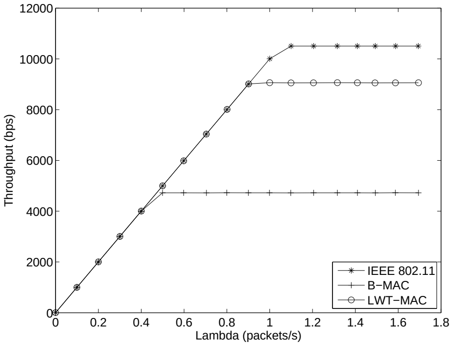


Fig. 3. Throughput of IEEE 802.11, B-MAC and LWT-MAC in a 10-node single hop network

3.4 Addressing Collective Quality of Service (QoS)

The LWT-MAC wake up after transmissions capability makes it a good candidate to be used in event-based WSNs where instantaneous increases of the traffic load occur due to event detection at nearby sensor nodes. The QoS observed by event-based messages is crucial in order to assure the correct and fast event detection at sink. However, this QoS differs from the traditional definition in which the QoS measurement is made packet by

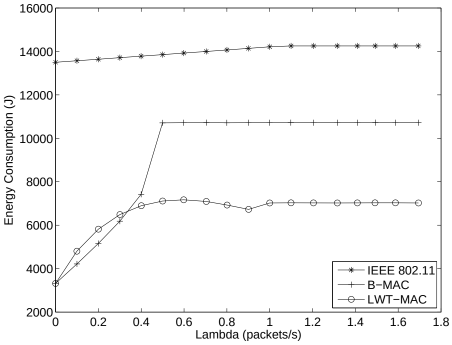


Fig. 4. Energy Consumption of IEEE 802.11, B-MAC and LWT-MAC in a 10-node single hop network

packet. In event-based WSNs the QoS should refer to the group of messages related to an event, it is known as collective QoS. Collective QoS is defined as the QoS (delay, bandwidth, packet loss, etc.) of the set of packets related to a specific event [8]; i.e., the delay of the individual messages is not crucial but the latency from the event generation until the event detection at sink is critical. A MAC protocol that efficiently reacts to event-based traffic will increase the collective QoS of the messages involved.

The LWT-MAC protocol is designed in order to improve the collective QoS of event messages while maintaining a low energy consumption. As far as the authors know, there is not any other MAC protocol designed keeping in mind collective QoS metrics. Those which are focused on QoS are based on end-to-end traditional QoS metrics instead [5,13–16,19].

The behavior of the protocol with periodic and event-based traffic profiles and its ability to increase the collective QoS was studied in [6].

4 LWT-MAC Analytical Model

In this section an analytical model of the LWT-MAC protocol in single-hop WSNs is described. The analytical model assumes ideal channel conditions (no channel errors or hidden terminal problems) and that each node computes a random BO (between 0 and CW) before each transmission attempt[1].

---

[1] The effect of the CA timer previously described has not been studied, it is considered that collisions move the system to the unscheduled mode.

For simplicity reasons the sensor nodes are considered to be homogeneous (equal traffic profiles and capabilities). Table 2 provides the description of some relevant variables used.

The extension of the analytical model to a multi-hop network is a challenging task since collisions can happen due to hidden terminal problems [2]. Moreover, in WSNs, where hidden terminals can wake up at any moment during an ongoing transmission, the multi-hop analysis becomes even more difficult than in traditional wireless networks where nodes are always listening to the channel. Therefore, the multi-hop analysis is left for future study.

4.1 Network Performance Metrics

The LWT-MAC analytical model is based on the one described in [3] where an IEEE 802.11 (CSMA/CA) analytical model to compute traditional metrics such as throughput, delay and queue occupation is presented.

Table 2. Notation

Notation	Description
n	Number of nodes
L_{data}	Packet Size (bits)
λ	Rate of packet generation (packets/s)
K	Queue Size (packets)
$L_{\mathrm{rts,cts,ack}}$	RTS,CTS, ACK Packet Size (bits)
L_{p}	Preamble Size (bits)
r	Transmission Rate (bits/s)
T	Time (s)
S	Throughput (bits/s)
M	Average number of transmission attempts per packet
B	Average number of slots in BO
p_{sch}	Scheduled probability
p_{w}	Probability to wake up after a transmission
ρ	Queue utilization
τ	Transmission probability in a given slot
CW	Contention Window of the random BO
σ	Empty slot duration (s)
ζ	Refers to the metrics: $S, M, B, p_{\mathrm{sch}}, p_{\mathrm{e}}, \rho$
T_{sleep}	Sleep Time of the duty cycle (s)
T_{listen}	Listen Time of the duty cycle (s)

A sensor node is modeled as a single queue of length K packets. Each node generates packets following a Poisson distribution with rate λ packets/s and average packet length L_{data} bits. From these assumptions the following metrics can be computed:

$$A = \lambda X, \qquad \rho = A(1 - P_{\mathrm{b}}), \qquad P_{\mathrm{b}} = \frac{(1-A)A^K}{1-A^{K+1}} \tag{1}$$

where A is the offered load, ρ is the queue utilization, P_{b} denotes the blocking probability and X refers to the service time (the time since the packet arrives at the head of the queue until it is released from it, assuming that it follows an exponential distribution).

The service time can be calculated as:

$$X = (M-1)\left(B\alpha + T_{\mathrm{c}}\right) + B\alpha + T_{\mathrm{s}} \tag{2}$$

where M is the average number of required transmission attempts per packet, B is defined as the average number of slots selected before each transmission attempt, α is the average slot duration and T_{c} and T_{s} are the durations of a collision and a successful transmission respectively.

The average number of attempts per packet successfully transmitted or discarded (M) due to maximum retry limit (R) reached is computed as:

$$M = \frac{1-p^{R+1}}{1-p} \tag{3}$$

where p is the conditional collision probability (assumed to be constant for all transmission attempts):

$$p = 1 - (1-\tau)^{n-1} \tag{4}$$

Being n the total number of nodes in the network and τ the steady state probability that a node transmits in a random slot given that it has a packet ready to be transmitted:

$$\tau = \frac{\rho}{B+1} \tag{5}$$

Assuming that the BO is uniformly distributed in the range [0-CW], B can be obtained as shown in Eq. 6.

$$B = \frac{\mathrm{CW} - 1}{2} \tag{6}$$

The average slot duration (α), is calculated considering the duration of the slot depending on the channel state (Eq. 7). As the channel is assumed to be error-free, it can only be in *empty*, $successful$ or *collision* states with their corresponding probabilities p_{e}, p_{s}, p_{c}.

$$\alpha = p_{\mathrm{e}}\sigma + p_{\mathrm{s}}(T_{\mathrm{s}} + \sigma) + p_{\mathrm{c}}(T_{\mathrm{c}} + \sigma) \tag{7}$$

where σ is the empty slot duration.

The channel state probabilities are related to the stationary probability that the rest of the nodes (except the one that is in BO) try to transmit in a given random slot. These are given by:

$$\begin{aligned} p_{\mathrm{e}} &= (1-\tau)^{n-1} \\ p_{\mathrm{s}} &= (n-1)\tau(1-\tau)^{n-2} \\ p_{\mathrm{c}} &= 1 - p_{\mathrm{e}} - p_{\mathrm{s}} \end{aligned} \tag{8}$$

Finally, the throughput per node can be computed as:

$$S = \rho\frac{L_{\mathrm{data}}}{X}(1 - p_{\mathrm{d}}) \tag{9}$$

where p_{d} is the probability to discard a packet due to maximum retry limit reached:

$$p_{\mathrm{d}} = p^{R+1} \tag{10}$$

The channel occupation durations (Eqs. 11 and 12) can be computed considering the length of the messages involved and the time intervals between them. Note that in the unscheduled mode the preamble is sent before each packet transmission attempt. The analytical model presented in this work only considers the transmission of packets preceded by RTS/CTS messages, although the model is easily adaptable to also consider the basic access method. Observe also that it has been considered the use of the DIFS, SIFS and EIFS time periods as in the IEEE 802.11 (CSMA/CA) [1].

$$T_{\mathrm{c}} = \mathrm{DIFS} + \frac{(L_{\mathrm{p}} \cdot p_{\mathrm{unsch}}) + L_{\mathrm{rts}}}{r} + \mathrm{EIFS} \tag{11}$$

$$T_{\mathrm{s}} = \mathrm{DIFS} + \frac{(L_{\mathrm{p}} \cdot p_{\mathrm{unsch}}) + L_{\mathrm{rts}} + L_{\mathrm{cts}} + L_{\mathrm{data}} + L_{\mathrm{ack}}}{r} + 3\mathrm{SIFS} \tag{12}$$

where $L_{\mathrm{p}}, L_{\mathrm{rts}}, L_{\mathrm{cts}}$ and L_{ack} are the lengths of the preamble, RTS, CTS and ACK messages respectively, while r refers to the transmission rate. The probability of transmitting a packet in scheduled mode (p_{sch}) is the probability that after a successful transmission, any other node has still data to be transmitted. On the other hand, the probability to transmit a packet in the unscheduled mode (p_{unsch}) is obtained as the complementary of the former:

$$p_{\mathrm{sch}} = \frac{p_{\mathrm{ss}}}{1 - p_{\mathrm{es}}}(1 - (1 - \rho)^n), \qquad p_{\mathrm{unsch}} = 1 - p_{\mathrm{sch}} \tag{13}$$

where p_{ss} and p_{es} are the probabilities of successful and empty slots from the network point of view:

$$p_{\mathrm{es}} = (1 - \tau)^n, \qquad p_{\mathrm{ss}} = n\tau(1 - \tau)^{n-1} \tag{14}$$

The analytical model is solved using a fixed point approximation. With the metrics obtained the energy consumption of a sensor node during a certain amount of time can be computed.

4.2 Energy Consumption

The total energy consumption of a sensor node can be divided in four parts: *i)* the energy spent to transmit and *ii)* receive messages, *iii)* the energy wasted in overhearing, and *iv)* the energy spent in duty cycle (sleeping and waking up in inactive periods):

$$e = e_{\mathrm{tx}} + e_{\mathrm{rx}} + e_{\mathrm{ov}} + e_{\mathrm{dc}} \tag{15}$$

Let N_{s} be the total number of messages a node successfully sends during a time T, it can be derived using the throughput S and the packet length L_{data} as $N_{\mathrm{s}} = T\,(S/L_{\mathrm{data}})$.

The energy spent to transmit N_{s} messages is computed taking into account the energy needed to successfully transmit a packet ($e_{\mathrm{s,tx}}$) and the energy spent in collisions for each unsuccessful attempt ($e_{\mathrm{c,tx}}$):

$$e_{\mathrm{tx}} = N_{\mathrm{s}}\,(e_{\mathrm{s,tx}} + (M-1)e_{\mathrm{c,tx}}) \tag{16}$$

The values of $e_{\mathrm{s,tx}}$ and $e_{\mathrm{c,tx}}$ can be obtained considering the energy spent to transmit and receive the messages involved and the empty time intervals (Eqs. 17 and 18). Moreover, the empty slots of the BO procedure must be considered (note that the busy slots of the BO countdown are part of the receiving or overhearing energy consumptions).

$$\begin{aligned} e_{\mathrm{c,tx}} = {} & E_{\mathrm{idle}}\left(\mathrm{DIFS} + B\sigma p_{\mathrm{e}} + 2\mathrm{SIFS} + \frac{L_{\mathrm{cts}}}{r}\right) \\ & + E_{\mathrm{tx}}\frac{(L_{\mathrm{p}} \cdot p_{\mathrm{unsch}}) + L_{\mathrm{rts}}}{r} \end{aligned} \tag{17}$$

$$\begin{aligned} e_{\mathrm{s,tx}} = {} & E_{\mathrm{idle}}(\mathrm{DIFS} + B\sigma p_{\mathrm{e}} + 3\mathrm{SIFS} + T_{\mathrm{e}}) \\ & + E_{\mathrm{tx}}\frac{(L_{\mathrm{p}} \cdot p_{\mathrm{unsch}}) + L_{\mathrm{rts}} + L_{\mathrm{data}}}{r} \\ & + E_{\mathrm{rx}}\frac{L_{\mathrm{cts}} + L_{\mathrm{ack}}}{r} \end{aligned} \tag{18}$$

where $E_{\mathrm{tx}}, E_{\mathrm{rx}}$ and E_{idle} denote the energy consumptions of being in transmission, reception and idle modes.

Notice that, after a successful transmission, all nodes in the network keep listening to the channel in case another node has something to transmit. If there are no more data to send, this time remains empty and each node stays in the idle mode during a T_{listen} (added to avoid synchronization problems) plus the CW, this is:

$$T_{\mathrm{e}} = (1-\rho)^n (T_{\mathrm{listen}} + \sigma \mathrm{CW}) \tag{19}$$

For simplicity reasons, it has been considered that each node receives N_{s} packet destined to it. Therefore, the total energy consumption to receive those messages is computed as:

$$e_{\mathrm{rx}} = N_{\mathrm{s}} \cdot e_{\mathrm{s,rx}} \tag{20}$$

The colliding packets have not been considered here. It is assumed that the unsuccessful transmissions of a node collide with those that are destined to it and that the probability to collide with more than one packet can be neglected.

The energy consumption to receive a packet is:

$$\begin{aligned} e_{\mathrm{s,rx}} = {} & p_{\mathrm{sch}} \cdot e_{\mathrm{b}} + p_{\mathrm{unsch}} \cdot e_{\mathrm{p}} + E_{\mathrm{idle}}(3\mathrm{SIFS} + T_{\mathrm{e}}) \\ & + E_{\mathrm{rx}} \left(\frac{L_{\mathrm{rts}} + L_{\mathrm{data}}}{r} \right) + E_{\mathrm{tx}} \left(\frac{L_{\mathrm{cts}} + L_{\mathrm{ack}}}{r} \right) \end{aligned} \tag{21}$$

where e_{b} is the energy of a busy listening after transmissions period, i.e., the idle time interval before sending a packet in the scheduled mode. Observe that, those nodes without data to send must also wait the remaining BO of the other nodes.

This time has been approximated by $B\sigma$ as shown in Eq. 22. This approximation does not affect when the traffic load is low (the scheduled probability is small) or high (the probability that a node has no data to transmit is negligible). However, it affects the results with moderate traffic load and its effect depends on the number of nodes. When n is high the energy consumption is overestimated (as the probability that a node has a small remaining BO increases) while with small n the energy consumed is underestimated (since the probability to transmit a packet that arrives at the queue during the listen after transmission period increases). Then:

$$e_{\mathrm{b}} \approx E_{\mathrm{idle}}(\mathrm{DIFS} + B\sigma(1-\rho)) \tag{22}$$

The parameter e_{p} refers to the energy spent receiving the long preamble. If a node has something to transmit it will be listening to the channel, therefore it will receive the entire long preamble of any other transmission in the medium. Otherwise, the node will be in duty cycle mode and on average it will wake up in the middle of the other's long preamble transmissions plus its own T_{listen}:

$$\begin{aligned} e_{\text{p}} &= \rho \left(E_{\text{idle}}\text{DIFS} + E_{\text{rx}}\frac{L_{\text{p}}}{r} \right) \\ &+ (1-\rho)\left(\frac{E_{\text{idle}}T_{\text{listen}} + E_{\text{rx}}\frac{L_{\text{p}}}{r}}{2} \right) \end{aligned} \tag{23}$$

Similarly, the energy consumption due to overhearing is computed as:

$$e_{\text{ov}} = N_{\text{s}}(n-2)\left(e_{\text{s,ov}} + \frac{M-1}{2}e_{\text{c,ov}} \right) \tag{24}$$

where $e_{\text{s,ov}}$ and $e_{\text{c,ov}}$ are the energy consumption to overhear a successful transmission or a collision respectively:

$$\begin{aligned} e_{\text{c,ov}} &= p_{\text{sch}} \cdot e_{\text{b}} + p_{\text{unsch}} \cdot e_{\text{p}} + E_{\text{rx}}\frac{L_{\text{rts}}}{r} \\ &+ E_{\text{idle}}\left(2\text{SIFS} + \frac{L_{\text{cts}}}{r} \right) \end{aligned} \tag{25}$$

$$\begin{aligned} e_{\text{s,ov}} &= p_{\text{sch}} \cdot e_{\text{b}} + p_{\text{unsch}} \cdot e_{\text{p}} + E_{\text{idle}}T_{\text{e}} + E_{\text{rx}}\frac{L_{\text{rts}}}{r} \\ &+ E_{\text{sleep}}\left(\frac{L_{\text{cts}} + L_{\text{data}} + L_{\text{ack}}}{r} + 3\text{SIFS} \right) \end{aligned} \tag{26}$$

where E_{sleep} denotes the energy of being in sleep mode.

Finally, the rest of time (T_{inactive}), that can be obtained using the equations above computing the time instead of the energy, and the total time T, each node performs a low duty cycle operation, listening and sleeping according to the duty cycle:

$$e_{\text{dc}} = T_{\text{inactive}}\left(E_{\text{idle}}\frac{T_{\text{listen}}}{T_{\text{ci}}} + E_{\text{sleep}}\frac{T_{\text{sleep}}}{T_{\text{ci}}} \right) \tag{27}$$

where T_{ci} is the check interval:

$$T_{\mathrm{ci}} = T_{\mathrm{listen}} + T_{\mathrm{sleep}} \tag{28}$$

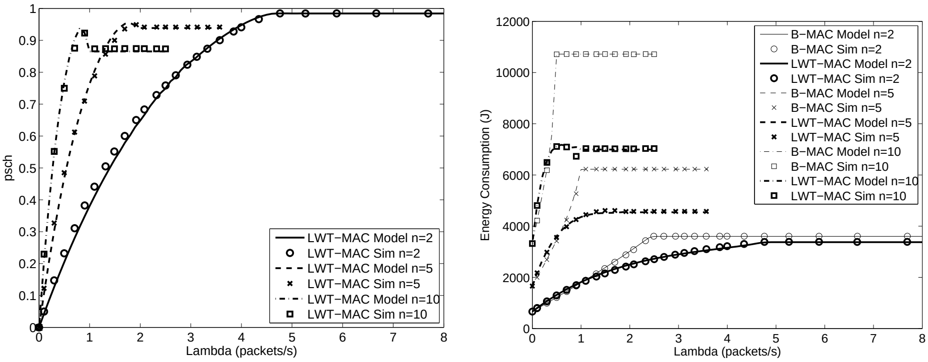
4.3 B-MAC Analytical Model

The previous analytical model can be adapted to model the behaviour of the B-MAC protocol. To achieve that aim two modifications are needed: *i)* the probability of being in scheduled mode (p_{sch}) has to be fixed to 0, meaning that the protocol always works in the unscheduled mode and *ii)* the empty listen time after a successful transmission has not to be considered ($T_{\mathrm{e}} = 0$) when computing the energy consumption of a sensor node since the listen after transmissions capability does not apply in B-MAC.

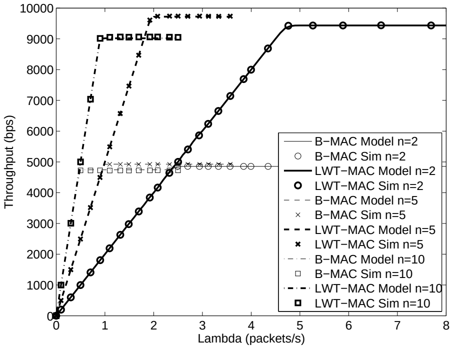
4.4 Analytical Model Validation

The SENSE simulator [9] has been used to validate the results obtained with the analytical model. The scenario consists of n nodes randomly placed in an area smaller than the maximum coverage range, thus, assuming ideal channel conditions, a full connectivity among them is assured. The default parameters used for the evaluation are shown in Table 1 (see Section 3).

Fig. 5 shows the analytical and simulation results of B-MAC and LWT-MAC with different number of sensor nodes. Observe how the scheduled probability of LWT-MAC (Fig. 5(a)) increases with the traffic load until a certain point at which the number of collisions (that move the system to the unscheduled mode) is noticeable. With a higher number of nodes, the scheduled probability increases faster (since the traffic load is higher), however it also becomes constant sooner as the collision probability is also higher. The total energy consumption of LWT-MAC (Fig. 5(b)) is strongly related to the scheduled probability. Note how the model underestimates the energy consumption approaching saturation for 2 and 5 nodes and overestimates it with 10 nodes. The reason for this discrepancy is the aforementioned approximation on the value of e_{b}. In contrast, for B-MAC, since it always works in unscheduled mode, its accuracy is not affected by the e_{b} approximation. It is also interesting to observe the throughput (Fig. 5(c)) that slightly increases, in both protocols, with the number of nodes, however the



(a) Scheduled Probability (b) Energy Consumption

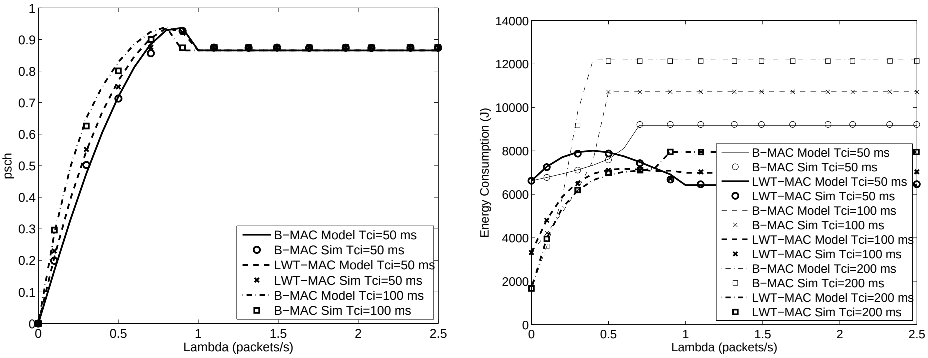


(c) Throughput

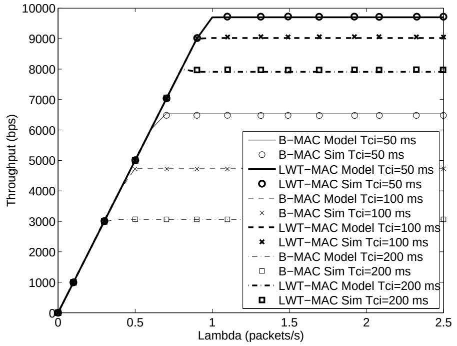
Fig. 5. Scheduled probability, total energy consumption and throughput with different number of sensor nodes

saturation throughput decreases with 10 nodes as the collision probability increases.

Fig. 6 shows the results with 10 nodes and different check intervals (the T_{listen} value is maintained while T_{sleep} changes to fit the given check interval). The scheduled probability of LWT-MAC (Fig. 6(a)) remains equal for all the check intervals at high loads, however at lower loads it increases faster with higher values of the check interval. This difference occurs because with longer check intervals the probability that a node has received something to transmit during an ongoing transmission increases. In the energy consumption (Fig. 6(b)) it can be seen, for both protocols, that at low loads the use of short time intervals notably penalizes the energy consumption as nodes wake up unnecessarily more often. However, as the load increases, to have longer



(a) Scheduled Probability (b) Energy Consumption



(c) Throughput

Fig. 6. Scheduled probability, total energy consumption and throughput with different values of the check interval

check intervals in LWT-MAC implies higher energy consumption caused by the collisions that move the network to the unscheduled mode (moreover, after a collision nodes wait awake listening the entire preamble transmission). However, the obtained energy consumption values at high loads are substantially lower than the ones obtained using B-MAC. In the throughput (Fig. 6(c)), it can be observed that lower saturation values are obtained when longer preambles are used, in both cases: using B-MAC and LWT-MAC. In this case, once again, it can be seen how the LWT-MAC improves the B-MAC performance increasing notably the saturation throughput.

5 Optimization Analysis

In this section, a performance optimization is done in order to derive the best parameter configuration of the LWT-MAC protocol for a single-hop scenario. A first goal is to identify how the different parameters that define the LWT-MAC operation affect its performance and how they can be tuned to minimize the energy consumption without significantly reducing the throughput and delay.

5.1 Key Parameters Optimization

One of the parameters of crucial importance is the sleep time of the duty cycle (T_{sleep}) as it directly affects the performance of the network. High values of the T_{sleep} allow to decrease the energy consumption but at the cost of reducing the throughput and increasing the delay. Additionally, it is possible to define a probability of waking up after successful transmissions (p_{w}), based on which sensor nodes decide to wake up at the end of a successful transmission or go to sleep remaining in the unscheduled access. Observe that, an additional mechanism should be implemented to inform nodes to wake up at the end of each transmission based on this probability, otherwise receivers can be sleeping when nodes send packets without using the long preamble. The sender can, for instance, notify sensor nodes to wake up at the end of the transmission in the data or RTS messages. At low traffic loads, setting that probability to small values allows to reduce the energy consumption as fewer idle listening after transmission periods will occur, however, as load increases, higher values of p_{w} reduce the energy waste since the number of long preamble transmissions are reduced. Note that by defining this probability Eqs. 13, 18, 21 and 26 should be rewritten as shown in Eqs. 29, 30, 31 and 32 respectively:

$$p_{\text{sch}} = p_{\text{w}} \frac{p_{\text{ss}}}{1 - p_{\text{es}}} (1 - (1 - \rho)^n), \qquad p_{\text{unsch}} = 1 - p_{\text{sch}} \tag{29}$$

$$\begin{aligned} e_{\text{s,tx}} = {} & E_{\text{idle}}(\text{DIFS} + B\sigma p_{\text{e}} + 3\text{SIFS} + p_{\text{w}} T_{\text{e}}) \\ & + E_{\text{tx}} \frac{(L_{\text{p}} \cdot p_{\text{unsch}}) + L_{\text{rts}} + L_{\text{data}}}{r} \\ & + E_{\text{rx}} \frac{L_{\text{cts}} + L_{\text{ack}}}{r} \end{aligned} \tag{30}$$

$$\begin{aligned} e_{\mathrm{s,rx}} = {} & p_{\mathrm{sch}} \cdot e_{\mathrm{b}} + p_{\mathrm{unsch}} \cdot e_{\mathrm{p}} + E_{\mathrm{idle}}(3\mathrm{SIFS} + p_{\mathrm{w}} T_{\mathrm{e}}) \\ & + E_{\mathrm{rx}} \left(\frac{L_{\mathrm{rts}} + L_{\mathrm{data}}}{r} \right) + E_{\mathrm{tx}} \left(\frac{L_{\mathrm{cts}} + L_{\mathrm{ack}}}{r} \right) \end{aligned} \tag{31}$$

$$\begin{aligned} e_{\mathrm{s,ov}} = {} & p_{\mathrm{sch}} \cdot e_{\mathrm{b}} + p_{\mathrm{unsch}} \cdot e_{\mathrm{p}} + E_{\mathrm{idle}} p_{\mathrm{w}} T_{\mathrm{e}} + E_{\mathrm{rx}} \frac{L_{\mathrm{rts}}}{r} \\ & + E_{\mathrm{sleep}} \left(\frac{L_{\mathrm{cts}} + L_{\mathrm{data}} + L_{\mathrm{ack}}}{r} + 3\mathrm{SIFS} \right) \end{aligned} \tag{32}$$

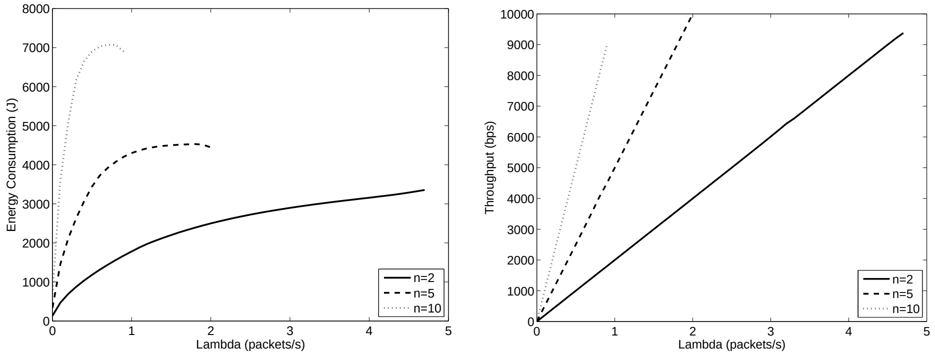
Other parameters such as the CW, used to compute the random BO, also affect the performance, but compared to the T_{sleep} and p_{w} their influence is limited. Parameters like the packet length or the traffic load are considered fixed.

5.2 Optimization Function

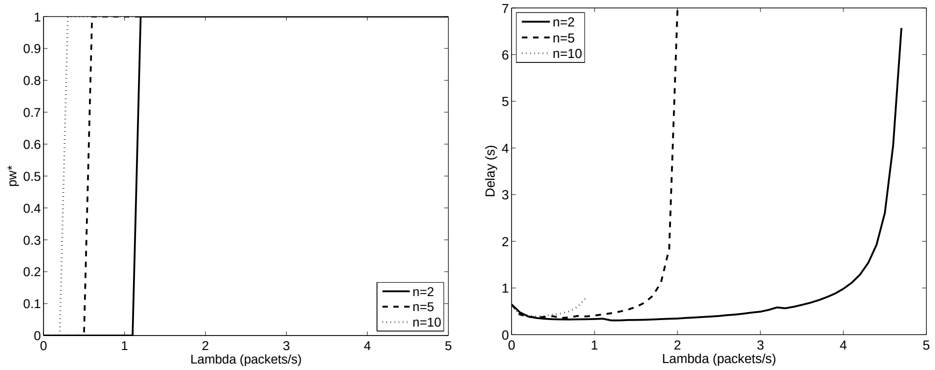
The main goal of the optimization process is to minimize the energy consumption (e) but constrained to achieve the same throughput as the IEEE 802.11 (CSMA/CA) MAC protocol. The IEEE 802.11 has been chosen as a reference as it does not implement the duty cycle operation, resulting in an upper bound in terms of performance (given that the other common parameters, such as the CW and the RTS/CTS option are equally configured in both approaches). The optimization analysis considers the throughput as the only constraint since the average packet transmission delay will necessarily increase to allocate space for the long preamble transmission, which is the price that the LWT-MAC pays to obtain a lower energy consumption than the IEEE 802.11 without reducing its throughput. The optimization function is shown in Eq. 33.

$$[T^*_{\mathrm{sleep}}, p^*_{\mathrm{w}}] = \underset{T_{\mathrm{sleep}} \in [0,0.5], p_{\mathrm{w}} \in [0,1], S \geq S_{802.11}}{\arg\min} e(n, L_{\mathrm{data}}, \zeta, T, T_{\mathrm{sleep}}, p_{\mathrm{w}}) \tag{33}$$

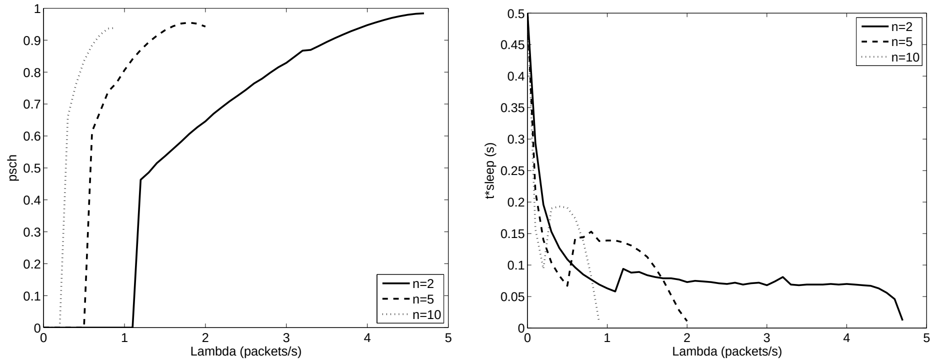
Observe that, the metrics referred by ζ (see Table 2) are a function of n, λ and L_{data} and can be obtained using the previously described analytical model.



(a) Energy Consumption (b) Throughput



(c) p_{w}^{*} (d) Delay



(e) Scheduled Probability (f) T_{sleep}^{*}

Fig. 7. Performance Metrics with Optimum T_{sleep}^{*} and p_{w}^{*} in a Single-Hop Network

5.3 Optimization Results

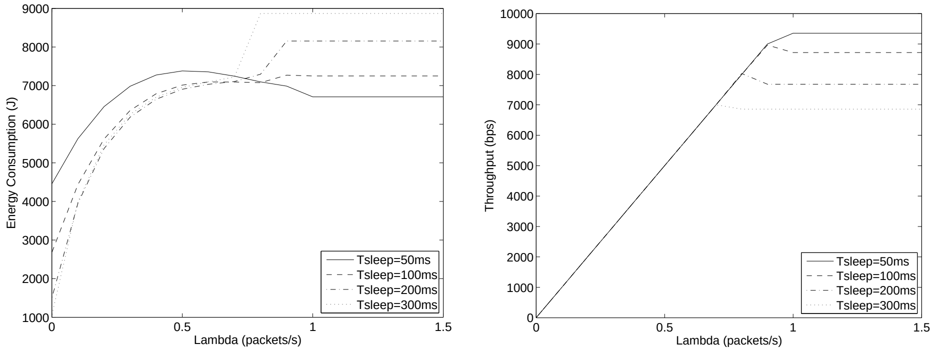
By applying Eq. 33 the optimal sleep time (T_{sleep}^{*}) and wake up probability (p_{w}^{*}) can be obtained for a given scenario. The optimal values are those that minimize the energy consumption and achieve the same throughput as the IEEE 802.11. To see how the optimal parameters change for different traffic loads (A) and number of nodes (n), a single-hop network with the parameters shown in Table 1 (see Section 3) has been evaluated.

Results are shown in Fig. 7. Note that, once the throughput of IEEE 802.11 cannot be achieved the metrics are not longer depicted. The minimum energy consumption (Fig. 7(b)) for the achieved throughput is depicted in Fig. 7(a). Observe, in Fig. 7(c), how the optimal probability to wake up after a successful transmission (p^*_w) increases rapidly from 0 to 1 when the traffic load starts to be noticeable. In contrast, the T^*_sleep (Fig. 7(f)) takes high values at low loads, meaning that, for the load requirements, sensor nodes can be sleeping during a longer time in each duty cycle. As the load increases, the value of T^*_sleep decreases, however it shows an inflection point and begins to increase. This effect is caused by the listening after transmissions probability that suddenly increases to 1 reducing the offered load of the network as it reduces the time needed to send a message. After this behavior, the T^*_sleep continues decreasing in order to maintain the throughput but also to reduce the duration of collisions. Moreover, as it has been considered that after a collision nodes also keep listening to the channel, thus receiving the entire next long preamble transmission, the consumption with high traffic loads increases with the value of the sleep time (that makes the long preamble to increase). The values T^*_sleep and p^*_w directly influence the delay and the scheduled probability. The delay, depicted in Fig. 7(d), shows a small increase at very low loads caused by the long sleep time and remains more or less constant until the queues become saturated. Regarding the scheduled probability (Fig. 7(e)), it is directly affected by p^*_w that bounds its value at low loads.

From these results, it can be concluded that, if the load can be estimated, the best configuration is: *a)* at low loads set the T_sleep to long values and p_w equal 0 and *b)* at high loads decrease the T_sleep value and set p_w equal 1.

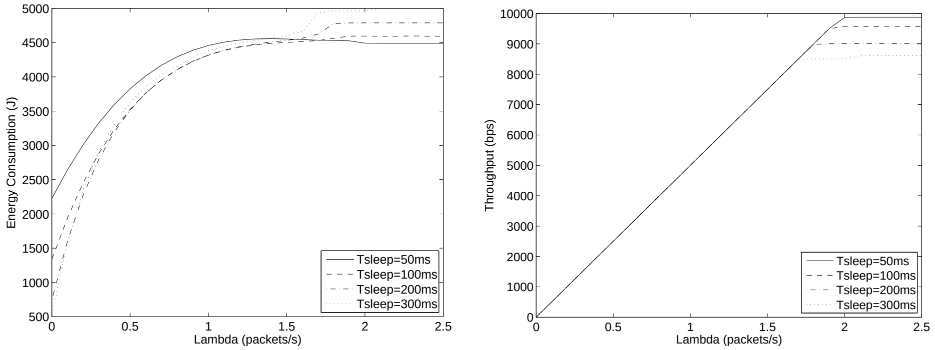
5.4 Heuristic Configuration of the LWT-MAC Parameters

The estimation of the traffic load is a difficult task, even more in event-based WSNs where the traffic profiles differ from the traditional ones, showing sporadic and instantaneous increases of the traffic load due, for instance, to events occurrence. Moreover, sensor nodes are devices with limited capabilities in terms of processing and memory resources making the load



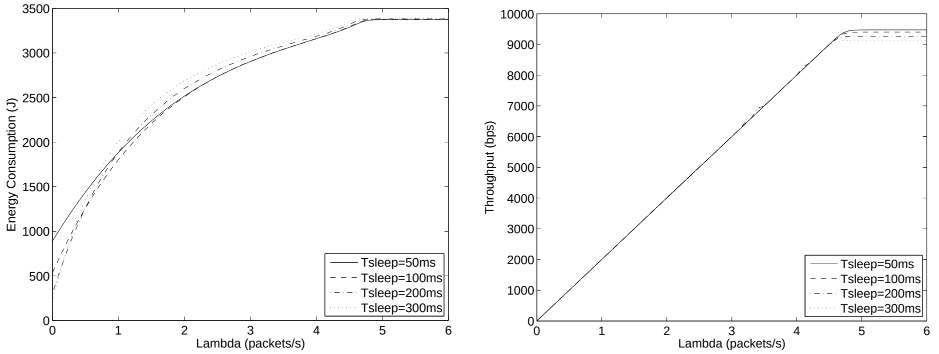
(a) Energy Consumption $n = 10$

(b) Throughput $n = 10$



(c) Energy Consumption $n = 5$

(d) Throughput $n = 5$



(e) Energy Consumption $n = 2$

(f) Throughput $n = 2$

Fig. 8. Performance Metrics for different T_{sleep} and $p_{\mathrm{w}} = 1$ in a Single-Hop Network

estimation an arduous task. However, if the load can be estimated in a fast and reliable way, the best option will be to use the LWT-MAC with the optimal values for T_{sleep} and p_{w}. Otherwise, from the observations made in the optimization analysis performed in the previous section, a heuristic parameter configuration to provide low energy consumption, high throughput and small delay through the entire load range can be made.

The major disadvantage of LWT-MAC is that it consumes more energy than B-MAC at low loads due to idle listening after transmission periods, however a moderately long value of T_{sleep} can help to decrease the energy consumption at low loads. To benefit from the advantages of LWT-MAC, if the T_{sleep} is fixed to a long value, the p_{w} probability should always be set to one. With this configuration, the LWT-MAC will consume less energy at low loads and it will maintain its capability to react to instantaneous increases of the network load.

Fig. 8 shows for different number of sensor nodes the value of the energy consumption and throughput with different values of T_{sleep} and varying the traffic load. It can be seen that for $T_{\mathrm{sleep}} = 50$ ms the energy consumption at low loads is significantly higher. However, it decreases substantially with $T_{\mathrm{sleep}} = 100$ ms and, still a bit more, with $T_{\mathrm{sleep}} = 200$ ms. However, with $T_{\mathrm{sleep}} = 300$ ms the reduction of energy consumption at low loads is extremely small, compared to the obtained with 200 ms, and at the cost of a lower throughput. Therefore, a T_{sleep} around 200 ms provides a considerably reduction of the energy consumption at low loads maintaining an acceptable value for the throughput. However, once the network saturates the energy consumption increases with higher values of T_{sleep} since it is considered that all nodes keep listening to the channel after a collision, thus receiving the entire long preamble of the retransmissions. Nevertheless, in the normal operation, the network is expected to work from low to moderate load conditions, not in saturation.

Thus, the suggested heuristic configuration is: $T_{\mathrm{sleep}} = 200$ ms and $p_{\mathrm{w}} = 1$.

6 Performance Evaluation

In this section, the heuristic configuration of $T_{\mathrm{sleep}} = 200$ ms and $p_{\mathrm{w}} = 1$ is evaluated and compared to B-MAC with $T_{\mathrm{sleep}} = 75$ ms in a single-hop and a multi-hop network. The results of LWT-MAC with $T_{\mathrm{sleep}} = 75$ ms and $p_{\mathrm{w}} = 1$ have also been included to keep them as a reference. For a fair comparison the RTS/CTS procedure is used in both B-MAC and LWT-MAC (for the scheduled and unscheduled accesses), therefore an RTS is sent immediately after the long preamble transmission. After overhearing an RTS or CTS message, nodes go to sleep for the ongoing transmission duration in both B-MAC and LWT-MAC. In the multi-hop case event-based traffic profiles will also be considered.

The SENSE simulator [9], as explained before, has been used to obtain the results. In this case the simulator has also been extended with the B-MAC protocol and the event-based traffic profile. The channel has been considered error-free.

6.1 Single-Hop Network

The single-hop scenario consists of n nodes randomly placed in an area smaller than the maximum coverage range. All sensor nodes generate messages following a Poisson distribution and send them to the sink (Fig. 9). The default parameters used for the evaluation are shown in Table 1 (see Section 3). In this scenario the CA timer is deactivated.

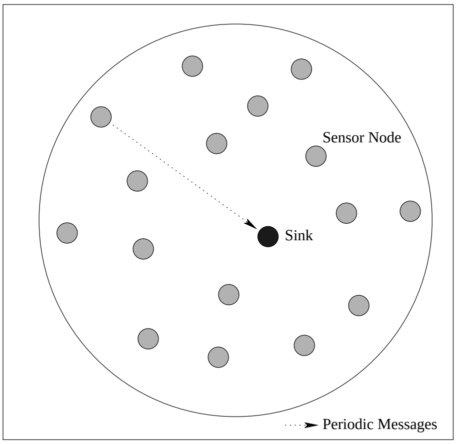
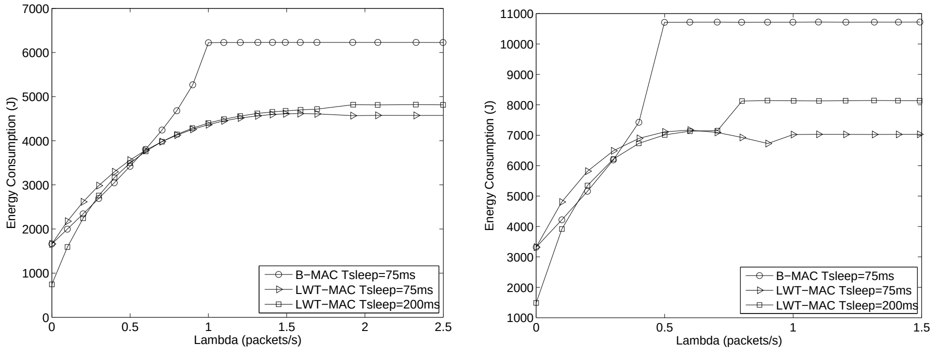
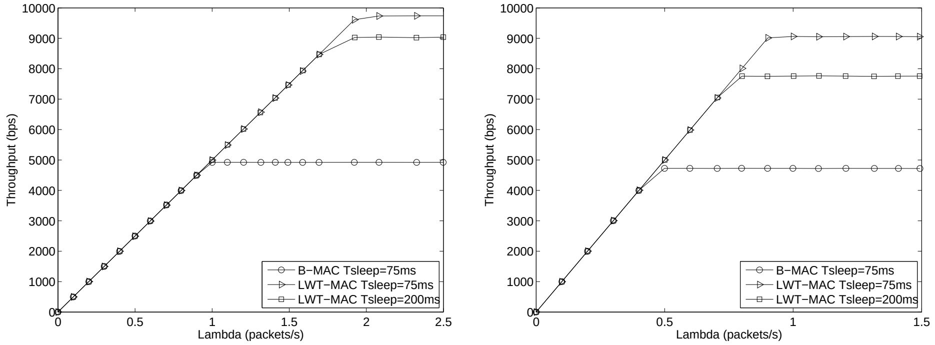


Fig. 9. Single-Hop Scenario

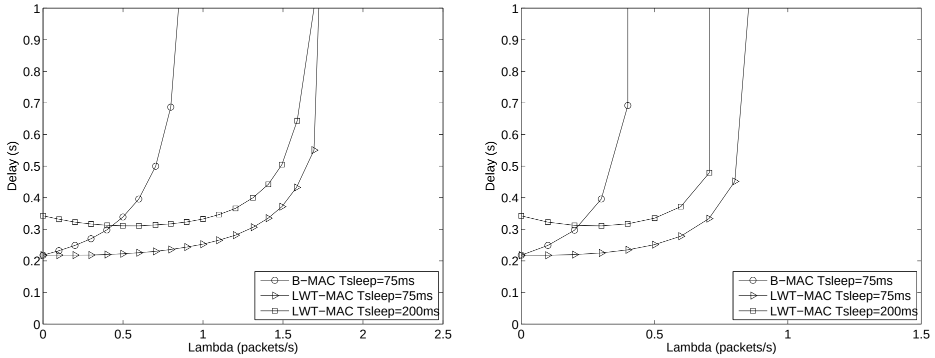
Results (Fig. 10) show that the energy consumption of LWT-MAC with the heuristic parameter configuration is similar to the one obtained by B-MAC at low loads as can be observed in Fig. 10(a) and 10(b). However, at high loads the suppression of the long preamble reduces the energy consumption of the LWT-MAC protocol. The suppression of the long preamble transmission is the cause of the higher throughput of the LWT-MAC compared to



(a) Energy Consumption $n = 5$ (b) Energy Consumption $n = 10$



(c) Throughput $n = 5$ (d) Throughput $n = 10$



(e) Delay $n = 5$ (f) Delay $n = 10$

Fig. 10. Performance Metrics with Recommended Heuristic T_{sleep} and p_{w} in a Single-Hop Network with 5 and 10 Sensor Nodes

the B-MAC as depicted in Fig. 10(c) and 10(d). However, the increase of the T_{sleep} to maintain the energy consumption makes the delay to slightly increase at low loads as can be seen in Fig. 10(e) and 10(f).

Compared to the LWT-MAC with $T_{\text{sleep}} = 75$ ms, the heuristic configuration provides slightly worse results in terms of throughput and delay but considerably reduces the energy consumption at low loads.

6.2 Multi-Hop Network

The multi-hop scenario consists of a multi-hop event-based WSN with 100 nodes randomly placed in a 100×100 m^2 area. The radio range of each node is 43 m and the Floyd algorithm has been used to compute the shortest path between any pair of nodes. Each sensor node generates two kinds of traffic profiles: *i)* messages generated following a Poisson distribution and *ii)* event-based messages (see Fig. 11). A random event generator selects randomly the event position and notifies the sensor nodes that are inside the coverage radius of the event in order to send event-based messages to the sink. Events have a constant coverage radius of 30 m, the time between events follows an exponential distribution with mean 600 s and the number of event messages needed at sink to reliably detect an event has been set to 5. In this scenario the CA timer is activated. The parameters used are shown in Table 3.

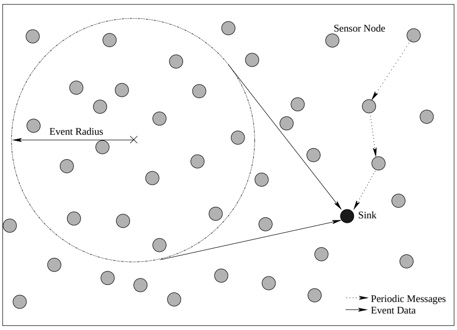
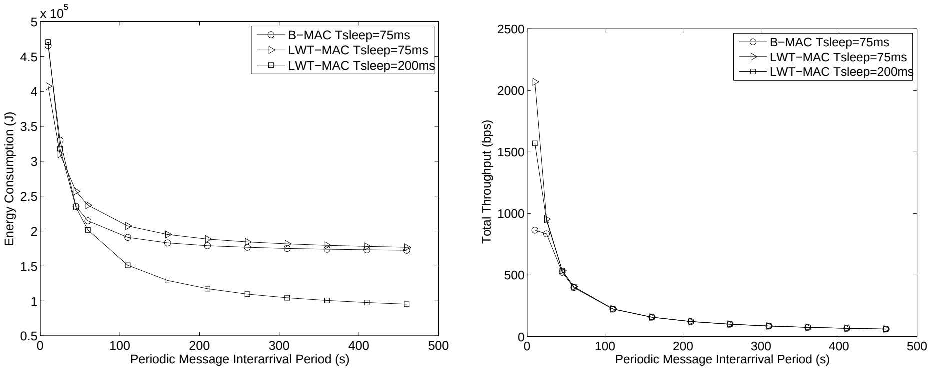
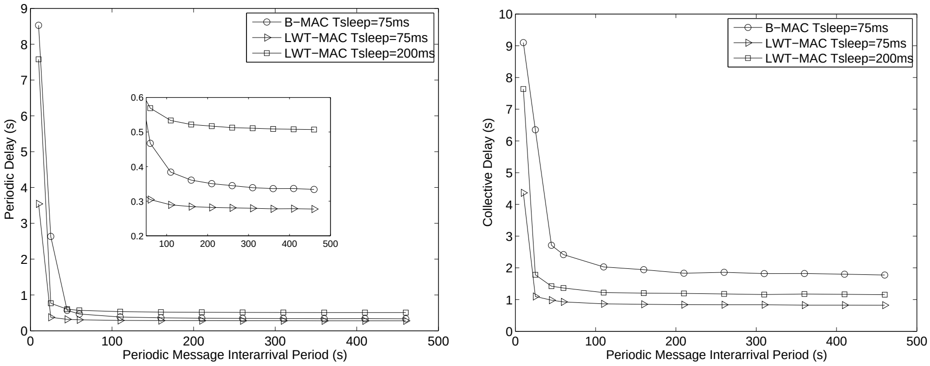


Fig. 11. Multi-Hop Scenario with Periodic and Event-based Traffic Profiles

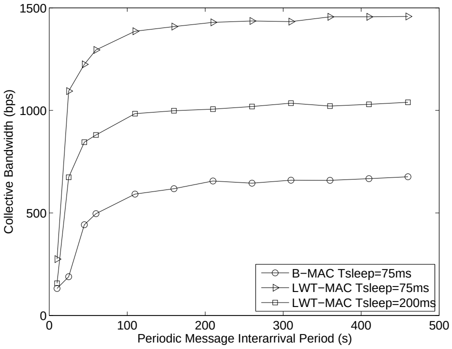
The results obtained are shown in Fig. 12. It is observed that at low loads the energy consumption of the LWT-MAC with the heuristic configuration is considerably lower than using the B-MAC (Fig. 12(a)). This effect appears due to the longer T_{sleep} but also due to the CA timer that avoids continuous collisions of preambles and better manages hidden terminal problems.



(a) Energy Consumption (b) Throughput



(c) Delay of periodic messages (d) Collective Delay



(e) Collective Bandwidth

Fig. 12. Performance Metrics with Recommended Heuristic T_{sleep} and p_{w} in a Multi-Hop Network with Periodic and Event-based Traffic Profiles

The LWT-MAC provides also better results in throughput, depicted in Fig.

Table 3. Default Parameters

Parameter	Value	Parameter	Value
r (data rate)	20 kbps	T_{listen}	24.5 ms
σ (empty slot)	1 ms	L_{data} (packet size)	240 bits
DIFS	10 ms	$L_{\text{rts}}, L_{\text{cts}}, L_{\text{ack}}$	64 bits
SIFS	5 ms	E_{tx}	24.75 mW
CW	64	$E_{\text{rx}}, E_{\text{idle}}$	13.5 mW
K (queue size)	10 pkts	E_{sleep}	15 μW
R (retry limit)	5	T (time)	$5 \cdot 10^5$ s

12(b), with a higher value of the saturation throughput and delay (Fig. 12(c)).

The collective QoS metrics are shown in Fig. 12(d), 12(e) and Table 4. The LWT-MAC with heuristic configuration achieves lower collective delay (Fig. 12(d)), defined as the time span between the event occurrence and the event detection at sink [12]. It also provides better collective bandwidth for the event-based messages as shown in Fig. 12(e) and better collective reliability (see Table 4) for a message interarrival period equals 10 and 25 s. For other loads the reliability is almost 100% in all the cases. Collective bandwidth refers to the bandwidth required to detect an event while collective reliability is the fraction of correctly detected events among all events generated [6].

Table 4. Collective Reliability for Periodic Interarrival Time equals 10 and 25 s

MAC Protocol	10 s	25 s
B-MAC $T_{\text{sleep}} = 75$ ms	0.509	0.995
LWT-MAC $T_{\text{sleep}} = 75$ ms	0.992	0.999
LWT-MAC $T_{\text{sleep}} = 200$ ms	0.891	0.999

Observe that, the LWT-MAC with $T_{\text{sleep}} = 75$ ms provides, as seen in the single-hop scenario, better results in throughput and delay but also in the collective metrics. However, the heuristic configuration allows to noticeably decrease the energy waste at low loads by obtaining better collective and individual metrics than B-MAC.

7 Concluding Remarks

In this work an analysis of the LWT-MAC protocol has been performed. A LWT-MAC analytical model that computes the network performance metrics and the energy consumption taking into account collisions in both saturated and unsaturated conditions has been presented. Moreover, the optimal configuration for the probability to wake up after a successful transmission (p_{w}) and the sleep time of the duty cycle (T_{sleep}) have been obtained depending on the number of nodes and the load of the network in a single-hop scenario. From the optimization results, a heuristic configuration for the T_{sleep} and the p_{w} is suggested. The use of the proposed heuristic parameter configuration avoids the complexity of other mechanisms that, for example, adapt the parameters based on the traffic load estimation, which can be unfeasible in WSNs, and provides a near-optimal performance in a wide range of situations. Additionally, the LWT-MAC with the heuristic parameter configuration has been compared with B-MAC in a single-hop network as well as in a multi-hop scenario with event-based traffic. Results show that the energy consumption of the sensor nodes is maintained similar or lower to the consumed by B-MAC and that the other performance metrics, specially those regarding to collective QoS, are substantially improved. Although the heuristic parameter configuration has been obtained for a single-hop scenario, it has been shown that it is also valid in a multi-hop network.

References

1. IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11*, 1999 Edition (Revised 2007).
2. F. Alizadeh-Shabdiz and S. Subramaniam. Analytical Models for Single-hop and Multi-hop Ad hoc Networks. *Mobile Networks and Applications*, 11(1):75–90, 2006.
3. B. Bellalta, M. Oliver, M. Meo, and M. Guerrero. A Simple Model of the IEEE 802.11 MAC Protocol with Heterogeneous Traffic Flows. In *Proceedings of the IEEE Eurocon*, November 2005.
4. M. Buettner, G.V. Yee, E. Anderson, and 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 (Sensys '06)*, pages 307–320, 2006.
5. M. Caccamo, L.Y. Zhang, L. Sha, and G. Buttazzo. An Implicit Prioritized Access Protocol for Wireless Sensor Networks. In *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS'02)*, 2002.
6. C. Cano, B. Bellalta, J. Barceló, and A. Sfairopoulou. A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS. In *Proceedings*

of the 7th International Conference on Wired / Wireless Internet Communications (WWIC), 2009.

7. C. Cano, B. Bellalta, A. Sfairopoulou, and J. Barceló. A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs. *IEEE Communication Letters*, 13(4):221–223, 2009.
8. D. Chen and P.K. Varshney. QoS Support in Wireless Sensor Networks: A Survey. In *Proceedings of the International Conference on Wireless Networks (ICWN'04)*, pages 227–233.
9. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
10. A. El-Hoiydi, J.D. Decotignie, and J. Hernandez. Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. In *Proceedings of the Fifth European Wireless Conference*, pages 563–569, 2004.
11. J. Haapola. Multihop Medium Access Control for WSNs: An Energy Analysis Model. *EURASIP Journal on Wireless Communications and Networking*, 2005(4):523–540, 2005.
12. X.Z. Lin, J.J. Zhou, and C.D. Mu. Collective Real-Time QoS in Wireless Sensor Networks. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'06)*, pages 1–4, 2006.
13. Y. Liu, I. Elhanany, and H. Qi. An Energy-efficient QoS-aware Media Access Control Protocol for Wireless Sensor Networks. In *Proceedings of the International Conference on Mobile Adhoc and Sensor Systems Conference*, pages 189–191, 2005.
14. Z. Liu and I. Elhanany. RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC'06)*, pages 768–773, 2006.
15. G. Lu, B. Krishnamachari, and CS Raghavendra. An Adaptive Energy-efficient and Low-latency MAC for Data Gathering in Wireless Sensor Networks. In *Proceedings of the Parallel and Distributed Processing Symposium*, 2004.
16. K.J. Paek, J. Kim, U.S. Song, and C.S. Hwang. Priority-Based Medium Access Control Protocol for Providing QoS in Wireless Sensor Networks. *IEICE Transactions on Information and Systems*, 90(9):1448, 2007.
17. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
18. T. van Dam and K. Langendoen. An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys '03)*, page 171, 2003.
19. T. Watteyne, I. Augé-Blum, and S. Ubéda. Dual-mode Real-time MAC Protocol for Wireless Sensor Networks: A Validation/Simulation Approach. In *Proceedings of the 1st International Conference on Integrated Internet Ad hoc and Sensor Networks*, 2006.
20. W. Ye, J. Heidemann, and D. Estrin. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.
21. W. Ye, F. Silva, and J. Heidemann. Ultra-low Duty Cycle MAC with Scheduled Channel Polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, page 334, 2006.
22. Y. Zhang, C. He, and L. Jiang. Performance Analysis of S-MAC Protocol under Unsaturated Conditions. *IEEE Communications Letters*, 12(3):210–212, 2008.

Quantitative Analysis of the Hidden Terminal Problem in Preamble Sampling WSNs

C. Cano, B. Bellalta, A. Cisneros and M. Oliver

Universitat Pompeu Fabra

Abstract. Collisions in preamble sampling WSNs cause a high waste of resources due to the mandatory transmission of a long preamble before every message. Moreover, when hidden terminals are present, collisions are even more noticeable since the vulnerability time is proportional to the long preamble duration. This effect considerably reduces the network performance as both the number of collisions and the resources spent in them are significant. The effects of hidden nodes in preamble sampling WSNs, in different case scenarios and under variable offered loads, are analyzed in this work. Results show that the impact of hidden terminals in the network performance is non-negligible, specially at medium to high loads. These results should be considered by MAC protocol designers in order to devise mechanisms to mitigate the hidden terminal problem.

1 Introduction

During the last few years several new Medium Access Control (MAC) protocols have been designed to cope with the energy consumption constraint of Wireless Sensor Networks (WSNs). These approaches were mainly motivated by the fact that the MAC layer controls the transceiver, that is the most energy consuming component of a sensor node. The common approach is to perform a low duty cycle operation in which sensor nodes sleep during most of the time and wake up only to send or receive data. These approaches can be divided in two major categories [19]: *i)* scheduled, where some kind of organization among sensor nodes is performed to decide when to sleep and wake up and *ii)* unscheduled, where sensor nodes decide their duty cycle independently of the other nodes in the network.

Unscheduled MAC protocols are specially well suited for WSNs as precise synchronization is not needed, they are simpler and in most cases consume less energy than scheduled approaches. Among these, Low Power Listening (LPL), also known as preamble sampling, protocols have shown very good results in terms of energy consumption as they are capable of working at considerably low duty cycles ($\sim 1\%$). In this kind of protocols, sensor nodes wake up during a short period of time just to sample the medium: if the medium is empty they return to sleep but remain awake to receive the data if the medium is found busy. In the basic preamble sampling technique [12, 14, 21] a sensor node willing to transmit before sending the data it transmits a long preamble that overlaps with the listening time of the receiver, thus assuring it will be awake to receive the packet. Observe that at low loads the energy consumption is extremely reduced but, as soon as the load increases, collisions become very costly as the entire long preamble is involved.

The hidden terminal problem [25] is a well-known effect that occurs in wireless networks. The most simple case happens when two or more nodes, that are unable to sense each other, transmit simultaneously to a common receiver, causing a collision. In traditional Wireless Local Area Networks (WLANs), the most common way to alleviate this problem is the use of the Request To Send (RTS)/Clear To Send (CTS) mechanism [17], that is in fact used in IEEE 802.11 [1] networks as an optional feature. Using this method, a sender first transmits an RTS message that is replied with a CTS by the destination if it is ready to receive the data. As the length of the RTS is considerably small (if compared to data messages common in WLANs), the vulnerability time is reduced, thus mitigating the effects of the hidden terminals. On the contrary, in preamble sampling WSNs, the vulnerability time is high due to the transmission of the long preamble, thus the hidden terminal is considerably more problematic than in WLANs. Note that if the RTS/CTS mechanism is used, the long preamble should be transmitted at least before the RTS message, but also before the CTS to guarantee the correct data transmission (otherwise, hidden terminals can be sleeping). This approach maintains the high vulnerability time but it also results in too much overhead for the typically small data messages found in this kind of networks.

Collisions from hidden nodes can be mitigated due to the capture effect, that is the ability of certain radios to correctly receive a message in spite of overlapping collisions (given that the difference of received signal strengths among them is sufficient). This effect, studied for the case of wireless sensor transceivers in [23, 28] and also considered in this work, can help mitigating the hidden terminal problem, however, it is not capable of completely solving it as collisions among packets with similar received signal strengths can still happen.

Although the literature regarding WSNs is extensive, a detailed quantitative analysis of the hidden terminal problem in preamble sampling WSNs is still needed. It is usually considered that at the low loads that WSNs work, this problem is irrelevant. However, although some WSNs work at low traffic loads during most of the time, increases of the network load can suddenly happen caused by query dissemination or due to event detection at nearby sensor nodes. Moreover, the convergecast communication typical in WSNs shows higher loads in those nodes closer to the sink.

This work provides a clear understanding of the hidden terminal problem in preamble sampling WSNs in different scenarios and under a broad range of offered loads. The methodology of this study is based on both analytical modeling and simulations. A preamble sampling analytical model that captures the impact of hidden terminals in a set of key scenarios has been developed. Simulations complement the analytical outcomes in these key scenarios but also allow to evaluate the more complex scenarios studied in this work.

The rest of the paper is organized as follows: Section 2 reviews the related work of the hidden terminal problem in WSNs, then, in Section 3, the system considerations and assumptions are detailed. The analytical framework is presented in Section 4. After that, the quantitative analysis is performed in Section 5. Finally, Section 6 discusses the effects of hidden terminals in improved preamble sampling techniques and Section 7 provides some concluding remarks.

2 Related Work

The hidden terminal problem [25] has been extensively studied, specially in IEEE 802.11 [1] networks. However, although hidden terminal effects are more problematic in preamble sampling WSNs, there have not been many efforts to study and quantify it. This problem has been considered, as well as some ideas to alleviate it, in the definition of some new preamble sampling MAC protocols. However, as far as the authors know, this is the first approach to quantify the problem in different LPL WSNs scenarios and under a broad range of offered loads.

The authors of [20] propose, in order to mitigate the hidden terminal problem, to increase the sensitivity of the sensor nodes. This technique, although effective to some extent, affects the performance of the network by reducing the throughput that can be achieved, apart from increasing the energy consumption of the sensor nodes (since higher sensitivity implies higher current draw).

Other works, such as [29], where a tree-like topology is considered, recommend to delay the transmissions each time a node overhears a message from its parent, assuming that its grandparent will forward the message after that. However, hidden terminal problems can also happen between independent generated messages, as will be seen throughout this paper.

Another proposed solution is to tune the backoff (BO), increasing either the slot duration [4] or the Contention Window (CW) [24]. These techniques are effective for non-preamble sampling WSNs where message transmissions are comparable to BO durations, however the BO values that could alleviate collisions among hidden terminals in preamble sampling WSNs are so high that will extremely reduce the performance of the network.

Finally, there are solutions that group into clusters non-hidden nodes and use different listening times for each group, such as the work presented in [18]. However, they are addressed to scheduled approaches where synchronization is required.

3 System Considerations

A preamble sampling WSNs has been considered. Therefore, sensor nodes sleep and wake up following the duty cycle. Periodically, sensor nodes wake up and sample the channel, if the medium is found idle, they go to sleep. Otherwise, if the medium is found busy, they remain active to receive the message. When a sensor node wants to send a message, it wakes up immediately and performs the medium access procedure. A long preamble that overlaps with the listening time of the receiver is transmitted before every message, thus assuring it will be awake during the data transmission. The preamble sampling technique considered in this work does not consider the adaptation of the duty cycle (and therefore of long preamble duration) to the traffic load. However, a discussion of how the hidden terminal problem can affect the performance in improved preamble sampling techniques is provided in Section 6.

Before accessing the channel, sensor nodes compute a random BO. Assuming that the medium is slotted, the BO time is decremented in one unit each slot time that the medium is detected free. Once the BO expires, the node transmits the packet. If the medium is sensed busy during the BO countdown, the BO is frozen and it is restarted when the channel is detected free again. This medium access procedure is a Carrier Sense Multiple Access (CSMA) approach and it is the same as the one defined in the medium access control of IEEE 802.11 [1] networks.

It has been assumed that sensor nodes generate packets following a Poisson distribution with a constant packet length. In order to obtain a clear understanding of the problem only local communication is considered. This behaviour eliminates the variable traffic loads in the network and therefore the differences in the hidden terminal problem among different regions.

It has been considered that transmissions to the wireless channel suffer from path loss (due to the distance between transmitter and receiver) and slow fading. The shadow fading for each communication pair is considered symmetric and time invariant. Thus, the power at which frames sent by A are received at B is the same than from B to A, assuming that A and B use the same transmitting power.

In order to eliminate interfering factors in the analysis the wireless channel is assumed to be noise-free, therefore only the interferences produced by collisions cause transmission errors. If a collision occurs among packets of *similar* received signal strengths, all packets involved are lost. However, after a sensor node starts receiving a packet (or long preamble), overlapping collisions without corrupting the initial reception are possible (capture effect), given that the difference of signal strengths is high *enough*[1]. Considering a non-coherent Frequency-Shift Keying (FSK) modulation and Manchester encoding (as used in Mica2 motes [10]), the Packet Error Rate (PER) is calculated as shown in Eq. 1 [31].

$$\mathrm{PER} = 1 - (1 - \mathrm{BER})^{2L} \tag{1}$$

where L is the packet length (including headers) in bits and BER is the Bit Error Rate, approximated as $\mathrm{BER} = \exp(-\mathrm{SIR}/2)/2$, in absence of noise. The Signal-to-Interference Ratio (SIR) can be calculated as the relation of received and interfering powers $P_\mathrm{r}/\sum P_\mathrm{int}$. Note that the length of the long

---

[1] In order to capture a packet it should be received before the interfering signals. This feature tries to model the fact that many radios cannot resynchronize to a stronger signal if they have already received a weaker one and it is a common assumption in many wireless simulators [11].

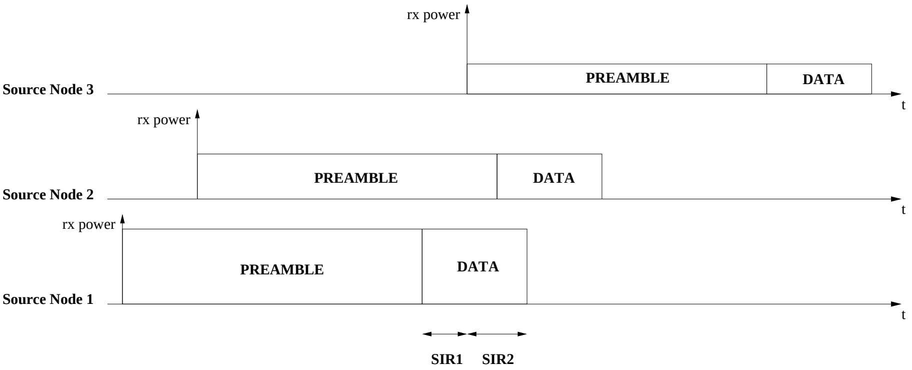


Fig. 1. Example of total and partial overlapping collisions. In the example, Source Node 1 arrives at the receiver with the highest power. Its packet error rate is computed taking into account SIR1, that only affects the first part of the message and includes the interference from Source Node 2, and SIR2, that affects the last part and includes the sum of interferences from Source Nodes 2 and 3.

preamble is not considered as it does not provide any useful information and can, therefore, be corrupted.

Observe that the calculation above is only valid for complete interfering collisions (i.e., collisions that start during the preamble reception), however, in a hidden terminal scenario partial collisions should also be considered. In that case, the PER is calculated as:

$$\text{PER} = 1 - \prod_{k=1}^{n_{\text{int}}} (1 - \text{BER}_k)^{2L_k} \tag{2}$$

where n_{int} is the number of different interference regions of a packet, BER_k is the bit error rate of each region and depends on the number of interferences that affect it, and L_k refers to the number of bits that form each region. See Fig. 1 where an example of $n_{\text{int}} = 2$ is shown.

Moreover, reception and carrier sense thresholds are considered. It is assumed that packets with received power over the reception threshold (sensitivity) can be correctly received. However, if the received power is below the reception threshold but higher than the carrier sense threshold, the packet cannot be understood but sensor nodes are able to detect that the medium is busy. Otherwise, if the received power is lower than the carrier sense threshold sensor nodes cannot detect that there is a transmission taking place.

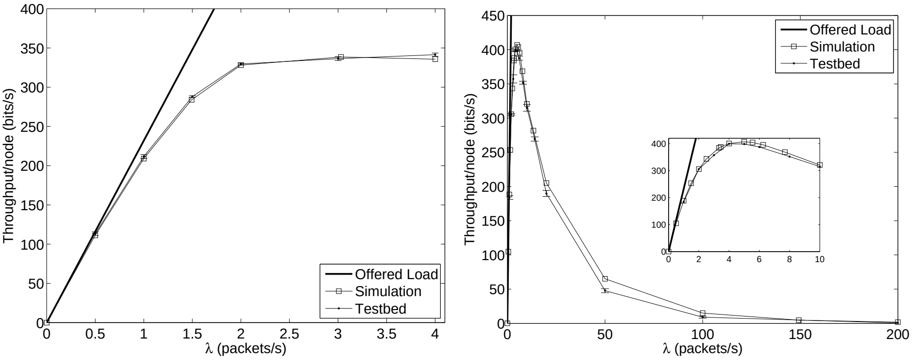
A preamble sampling simulator that maps the above considerations and assumptions has been used for the evaluation. The simulator framework is based on the SENSE simulator [9], however it has been extended in order to include the preamble sampling MAC protocol, the shadowing propagation model and an improved capture effect feature (able to compute the SIRs under multiple overlapping collisions).

The simulator has been validated comparing its results with a Mica2 testbed. Fig. 2 shows simulation and testbed (average and 95% confidence intervals) results in a 5-node network and in a typical hidden terminal scenario in which 2 nodes hidden from each other transmit to a common receiver. In order to assure that both transmitters in the hidden terminal case cannot sense each other, it is not enough to check that they are unable to successfully receive packets as the carrier sense mechanism can still detect that there is a transmission in the medium. Thus, with the goal to guarantee that the hidden terminal problem is taking place the carrier sense mechanism has been deactivated, therefore they always found the medium idle even if the other node is transmitting. In these scenarios the acknowledgement (ACK) and retransmissions have been disabled and the queue length at each sensor node is equal to zero. The rest of parameters (based on Mica2 motes [10]) are shown in Tables 1 and 2.

This section has presented the default system considerations and assumptions of this work. Most of them are defined in order to avoid factors that can interfere the quantification of the problem. However, the effect of the hidden terminal problem in a more realistic scenario considering periodic and event-based traffic profiles, convergecast communication and retransmissions enabled is also studied using simulations in Section 5.5.

4 Analytical Model

The development of analytical models helps in providing a better understanding of the network functionality and allows to study its performance in a broad range of scenarios with a lower computational effort if compared to simulations. Analytical models are also useful to perform optimization analysis that can provide the best operational parameters for a given network scenario. Therefore, to understand how WSNs operate and provide



(a) Throughput per node in a single hop 5-node network

(b) Throughput per node of two hidden terminals transmitting to a common receiver. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 10$ packets/s)

Fig. 2. Comparison of simulation and Mica2 mote testbed results.

Table 1. Hardware Parameters for Mica2 mote with CC1000 transceiver at 868 MHz.

Parameter	Value
Data Rate	19.2 kbps
Tx Power	0 dBm
Sensitivity	−99 dBm
Supply Voltage	3 V
Current Consumption in Tx Mode	24.5 mA
Current Consumption in Rx/Idle	17.6 mA
Current Consumption in Sleep Mode	15 μA

Table 2. MAC Parameters.

Parameter	Value
Data Packet Size (L_{data})	29 bytes
Headers (L_{header})	11 bytes
Total Packet Size (L)	40 bytes
Listen/Sleep Time	2.45/97.55 ms
Contention Window (CW)	32
Slot Time (σ)	417 μs

further insights on the hidden terminal effects on the WSNs performance, an analytical model is presented. It captures the different traditional performance metrics (delay, throughput, etc.) and the energy consumption of the sensor nodes.

Each sensor node is modeled as a single $M/M/1/K$ queue, assuming that: *i)* packets are generated following a Poisson distribution with rate λ packets/s, and *ii)* the service time (X), the time since a packet of L bits arrives at the head of the queue until it is released from it, follows an exponential distribution. From the $M/M/1/K$ assumption, the offered load (A), the queue utilization (ρ) and the blocking probability (P_{b}) can be computed as shown in Eq. 3.

$$A = \lambda X, \qquad \rho = A(1 - P_{\mathrm{b}}), \qquad P_{\mathrm{b}} = \frac{(1-A)A^K}{1 - A^{K+1}} \tag{3}$$

Considering that lost packets are not retransmitted, the service time can be calculated considering the average number of slots selected before each transmission attempt (B), the average slot duration (α) and the duration of a transmission (T_{tx}) as shown in Eq. 4

$$X = B\alpha + T_{\mathrm{tx}} \tag{4}$$

Assuming that the ACK and RTS/CTS mechanisms are deactivated[2], the channel occupation duration of the node of interest transmission can be expressed as:

$$T_{\mathrm{tx}} = \frac{L_{\mathrm{p}} + L}{r} \tag{5}$$

where L_{p} is the length of the preamble and r refers to the transmission rate.

Assuming that the BO is uniformly distributed in the range [0-CW], B can be obtained as shown in Eq. 6.

$$B = \frac{\mathrm{CW} - 1}{2} \tag{6}$$

The average slot duration (α) is calculated considering the duration of the slot depending on the channel state, given that the node of interest is in BO (Eq. 7). Assuming that the channel is noise-free, it can only be in *empty*, *successful* or *collision* states with their corresponding probabilities p_{e}, p_{s} and p_{c}.

---

[2] Observe that although the model presented does not consider the ACK and RTS/CTS, it is easily adaptable to include these extensions.

$$\alpha = p_{\mathrm{e}}\sigma + p_{\mathrm{s}}(T_{\mathrm{s}} + \sigma) + p_{\mathrm{c}}(T_{\mathrm{c}} + \sigma) \tag{7}$$

where σ is the empty slot duration. The channel state probabilities as well as the channel occupation durations of other's successful or collision transmissions (T_{s} and T_{c} respectively) are scenario-dependent. These metrics will be computed in Sections 4.1 and 4.2 for the cases where all nodes are in coverage range and in the presence of hidden terminals.

Thus, considering that the ACK mechanism is deactivated, the average throughput per node can be easily obtained as:

$$S = \rho \frac{L_{\mathrm{data}}}{X}(1 - p) \tag{8}$$

where p is the conditional collision probability. Its mathematical expression depends on each specific scenario and it will also be formulated in Sections 4.1 and 4.2 depending if all nodes are inside the coverage range or not.

Finally, the per-packet delay can be calculated as shown in Eq. 9.

$$D = \frac{\mathrm{EQ}}{\lambda(1 - P_{\mathrm{b}})} \tag{9}$$

where EQ refers to the average number of packets in the queue:

$$\mathrm{EQ} = \begin{cases} \dfrac{A}{1-A} - \dfrac{(K+2)A^{K+2}}{(1-A)^{(K+2)}} & \text{if } A < 1 \\[2ex] \dfrac{K+1}{2} & \text{if } A = 1 \end{cases} \tag{10}$$

The analytical model is solved using a fixed point approximation and the metrics obtained are used to compute the energy consumption, that is presented in Section 4.3.

4.1 Case 1: All Nodes in Coverage Range

When all nodes are in coverage range, i.e., all sensor nodes (n) can sense each other, the conditional collision probability (assumed to be constant for all transmission attempts) can be computed as shown in Eq. 11. Given that the node of interest transmits, there will be a collision any time that any of the other $n-1$ nodes in the network transmit in the same slot.

$$p = 1 - (1 - \tau)^{n-1} \tag{11}$$

Being τ the steady state probability that a node transmits in a random slot given that it has a packet ready to be transmitted:

$$\tau = \frac{\rho}{B+1} \tag{12}$$

The channel state probabilities are related to the stationary probability that the rest of nodes (except the one that is in BO) try to transmit in a given random slot, see Eq. 13. If none of the remaining nodes $(n-1)$ transmit, the slot remains empty, if only one transmits, it results in a successful transmission, and finally, if more than one transmit, the channel state results in a collision.

$$\begin{aligned} p_{\mathrm{e}} &= (1-\tau)^{n-1} \\ p_{\mathrm{s}} &= (n-1)\tau(1-\tau)^{n-2} \\ p_{\mathrm{c}} &= 1 - p_{\mathrm{e}} - p_{\mathrm{s}} \end{aligned} \tag{13}$$

Finally, and assuming that the ACK and RTS/CTS mechanisms are deactivated, the channel occupation durations can be computed as shown in Eq. 14.

$$T_{\mathrm{s}} = T_{\mathrm{c}} = T_{\mathrm{tx}} = \frac{L_{\mathrm{p}} + L}{r} \tag{14}$$

4.2 Case 2: Presence of Hidden Terminals

Several multihop analytical models for wireless networks, with more emphasis on the IEEE 802.11 protocol, exist in the literature, being one of the most complete the one presented by Alizadeh et al. in [2]. However, as the vast majority, they assume that hidden terminals enter in the contention phase synchronized. This assumption allows to model the conditional collision probability with the hidden terminals as $p = 1 - (1-\tau)^{\upsilon}$, where $\upsilon = n_{\mathrm{h}} T_{\mathrm{v}}/\sigma$, being n_{h} the number of hidden terminals and T_{v} the vulnerability time. The same approach is also followed in the only, as far as the authors know, multihop analytical model of CSMA with preamble sampling [27].

Tsertou et al. first claimed and studied the inaccuracy of this assumption [26]. They show that the model becomes problematic when the vulnerability time is higher than the average BO duration due to the time difference of the BO phase among hidden terminals. Observe that this is the case of preamble sampling WSNs in which the vulnerability time includes the long preamble transmission that is usually large (in the range of thousands of ms). They also provide a different approach to model the effect of hidden terminals, however, although the model is quite complex, only saturation conditions and a network formed by just two transmitting and hidden nodes are considered.

The analytical model of hidden terminals presented in this work follows the same approach presented in Section 4.1 but the conditional collision probability is computed without assuming synchronization among hidden nodes. This has been done by defining three main assumptions: *i)* all source nodes in the network are hidden terminals of each other, *ii)* hidden terminals do not have transmitting neighbours and *iii)* the maximum BO is smaller than the data packet transmission time.

Observe that assumptions *i)* and *ii)* are the same than the ones in [26] and hold for WSNs if the load of the network is low as the effect of hidden terminals dominates among the effects of neighbouring nodes. The extension of the analytical model to a general network is a challenge due to the aforementioned desynchronization among hidden terminals. As will be explained later, the approach presented here assumes that there will be a collision always that a hidden node has something to transmit. Note that this assumption does not hold in a general network, where a sensor node has transmitting neighbours and can, therefore, reduce its transmission attempts due to communication in the neighbourhood. Moreover, in this case, the average time spent in BO is highly difficult to compute because of the possible continuous overlapping collisions that also happen due to the desynchronization effect.

Considering that the source nodes do not have transmitting neighbours and that they are hidden terminals among them, the conditional collision probability can be computed as shown in Eq. 15. Assuming that the maximum

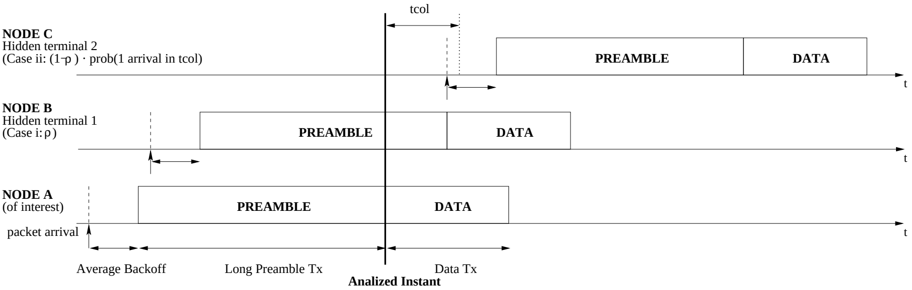


Fig. 3. Representation of the conditional collision probability in the presence of hidden terminals. Observe that the packet transmission from node A will collide if any other hidden terminal has a packet to transmit (or it is already transmitting it) in the analyzed instant (node B) or it has the queue empty and generates a packet during t_{col} (node C), where t_{col} is the data packet transmission time minus the average BO.

BO is smaller than the data packet transmission time, the probability that a packet transmission fails is: *i)* the probability that any other hidden terminal (n_{h}) has something to transmit (or it is, in fact, transmitting something) when the data packet transmission starts or *ii)* it generates a packet in the time interval t_{col} after the start of the data packet transmission, where $t_{\mathrm{col}} = \frac{L}{r} - B\sigma$. Refer to Fig. 3 where a schematic representation is provided.

$$p = 1 - \left((1-\rho)e^{-\lambda t_{\mathrm{col}}}\right)^{n_{\mathrm{h}}} \tag{15}$$

In this specific case, since the transmitting nodes do not overhear any other transmission, the channel states probabilities of the source nodes are simply:

$$\begin{aligned} p_{\mathrm{e}} &= 1 \\ p_{\mathrm{s}} &= 0 \\ p_{\mathrm{c}} &= 0 \end{aligned} \tag{16}$$

Finally, the successful transmission duration is $T_{\mathrm{tx}} = (L_{\mathrm{p}} + L)/r$, as shown in the previous case. The rest of metrics are obtained using the equations in Section 4. Note that, without loss of generality, these metrics are computed for the transmitting nodes only, not the receiver.

4.3 Energy Consumption

The total energy consumption of a sensor node can be divided in four parts: *i)* the energy spent to transmit and *ii)* receive messages, *iii)* the energy wasted in overhearing, and *iv)* the energy spent in duty cycle (sleeping and waking up in inactive periods):

$$e = e_{\mathrm{tx}} + e_{\mathrm{rx}} + e_{\mathrm{ov}} + e_{\mathrm{dc}} \tag{17}$$

Let N be the total number of messages a node sends during a time of observation T, it can be approximated as $N \simeq T\lambda(1 - P_{\mathrm{b}})$ when K is small. Thus, the total energy spent to transmit N messages is computed following Eq. 18, taking into account the energy needed to transmit the messages and the empty slots of the BO countdown (note that the busy periods will be included in the receiving and overhearing energy consumptions).

$$e_{\mathrm{tx}} = N \left(E_{\mathrm{idle}} \left(p_{\mathrm{e}}\, B\, \sigma \right) + E_{\mathrm{tx}} \left(\frac{L_{\mathrm{p}} + L}{r} \right) \right) \tag{18}$$

where E_{idle} and E_{tx} are the energy consumptions of being in idle and transmission modes respectively. Observe that Eq. 18 accounts for that messages that are successfully sent, discarded due to maximum retry limit and those that collide.

For *Case 1* where all nodes are in coverage range, the energy consumptions receiving and in overhearing are shown in Eqs. 19 and 21. However, for *Case 2*, where source nodes do not have transmitting neighbours, the energy consumptions receiving and in overhearing of the source nodes are zero ($e_{\mathrm{rx}} = e_{\mathrm{ov}} = 0$).

When all nodes are in coverage range it has been considered for simplicity that each node receives N_{s} packets destined to it, where N_{s} is the total number of messages a node successfully sends during a time T, i.e, $N_{\mathrm{s}} = T\,(S/L_{\mathrm{data}})$. Therefore, the total energy consumption to receive those messages is computed as:

$$e_{\mathrm{rx}} = N_{\mathrm{s}} \left(e_{\mathrm{p}} + E_{\mathrm{rx}} \left(\frac{L}{r} \right) \right) \tag{19}$$

where E_{rx} is the energy consumption of being in reception mode and the parameter e_{p} refers to the energy spent receiving the long preamble. If a node has something to transmit it will be listening to the channel, receiving the entire long preamble of any other transmission in the medium. Otherwise, the node will be in duty cycle mode and, on average, it will wake up in the middle of the other's long preamble transmissions:

$$e_{\text{p}} = E_{\text{rx}} \frac{L_{\text{p}}}{r} \left(\rho + \frac{1}{2}(1-\rho) \right) \tag{20}$$

Note that the colliding packets have not been considered in Eq. 19. It is assumed that the unsuccessful transmissions of a node collide with those that are destined to it and that the probability to collide with more than one packet can be neglected.

Similarly, the energy consumption due to overhearing is computed as:

$$e_{\text{rx}} = (n-2) \left(N_{\text{s}} + \frac{(N - N_{\text{s}})}{2} \right) \left(e_{\text{p}} + E_{\text{rx}} \left(\frac{L}{r} \right) \right) \tag{21}$$

Finally, the rest of time (T_{inactive}), that can be obtained using the equations above computing the time instead of the energy, and the total time T, each node performs a low duty cycle operation, listening and sleeping according to the duty cycle:

$$e_{\text{dc}} = T_{\text{inactive}} \left(E_{\text{idle}} \left(\frac{T_{\text{listen}}}{T_{\text{ci}}} \right) + E_{\text{sleep}} \left(\frac{T_{\text{sleep}}}{T_{\text{ci}}} \right) \right) \tag{22}$$

where E_{sleep} is the energy consumption in sleep mode and T_{ci} is the check interval:

$$T_{\text{ci}} = T_{\text{listen}} + T_{\text{sleep}} \tag{23}$$

The validation of the analytical model can be found in Sections 5.1 and 5.2 where the results from the analytical framework are compared to simulations in a single hop network and in the presence of hidden terminals without transmitting neighbours.

5 Quantitative Analysis

This section presents the quantitative results of the hidden terminal problem in different case scenarios. Three reference scenarios are considered: a single hop network (for reference purposes), the n_{h}-hidden (where a set of transmitting hidden terminals send data to a receiver), a grid and a network formed by nodes located randomly in the area. The random scenario is evaluated also under more realistic assumptions like event-based traffic profiles, convergecast communication and retransmissions enabled.

The single hop, the n_{h}-hidden and grid scenarios allow to carefully evaluate the problem. For this reason, it is important to fix the specific topologies, with this goal the carrier sense threshold is considered equal to the reception threshold and only the path loss model has been used. The path loss exponent is considered to be equal 4 while the log-normal standard deviation of the shadowing propagation model (used in the random scenario) has been set to 7 dB [3]. The carrier sense threshold (used also in the random scenario) has been set 12 dB higher than the sensitivity.

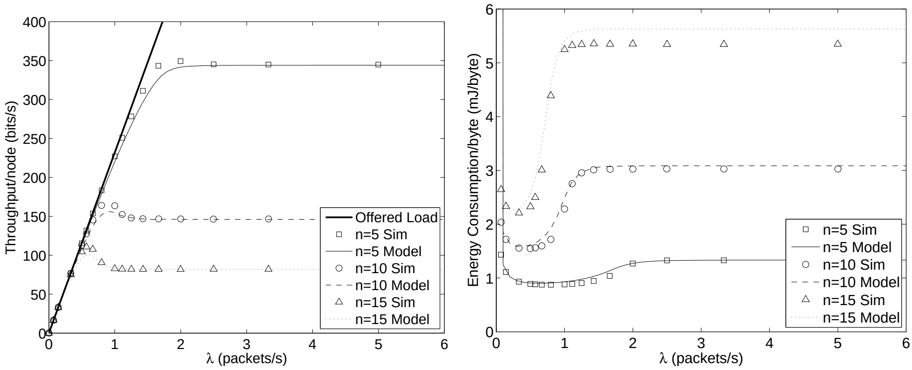
Table 3 summarizes the considerations of the three *basic* scenarios (single hop, n_{h}-hidden and grid), the random scenario and the more realistic random scenario while Tables 1 and 2 in Section 3 show the hardware and MAC parameters considered. The queue length (K) has been set to 10 packets.

Table 3. Summary of considerations for each scenario. The abbreviations PL and SW refer to Path Loss and Shadowing respectively.

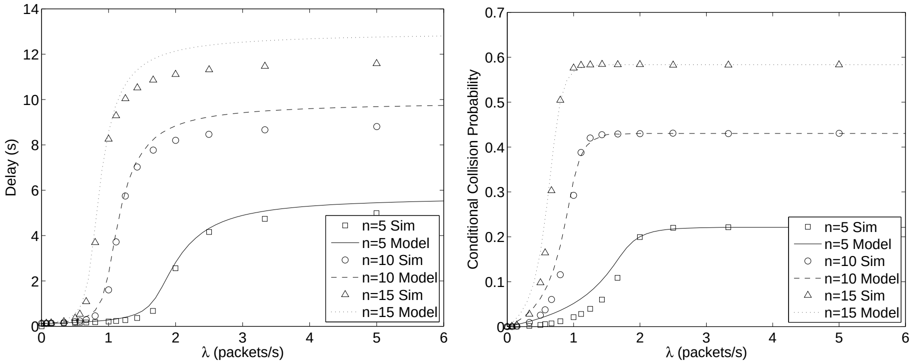
Parameter	Basic	Random	*Realistic* Random
Traffic Profile	Poisson	Poisson	Periodic and Events
Communication	Local	Local	Convergecast
Propagation Model	PL	PL+SW	PL+SW
Carrier Sense	= Sensitivity	< Sensitivity	< Sensitivity
Retx and ACKs	Disabled	Disabled	Enabled

5.1 Single Hop Scenario

The first scenario is based on a single hop network formed by n nodes transmitting data to a receiver. The set of nodes are located in the area assuring that all of them are inside the coverage range of the others. This



(a) Total throughput per node (b) Energy consumption/byte received



(c) Delay (d) Conditional collision probability

Fig. 4. Performance results in the single hop scenario.

scenario will be used as a reference of the WSNs performance when no hidden terminals are present. As all nodes are in coverage range no capture effect is possible. As stated in the simulator description, no capture effect is considered if collisions occur at the same instant (one propagation time after the initial reception is considered also the same instant).

The performance results are depicted in Fig. 4. It shows the results obtained using the simulator and the analytical model presented in Sections 3 and 4.1 respectively. The network saturation point is clearly seen in the figures, note that it appears at lower loads as the number of nodes increases. Observe, in Fig. 4(a), that shows the achieved throughput as well as the offered load, how the network is able to support without losses a load of more than 1 packet/s with $n = 5$ but less than 0.5 packets/s with $n = 15$. The rest of metrics, energy consumption per byte successfully sent (Fig. 4(b)), delay (Fig. 4(c))

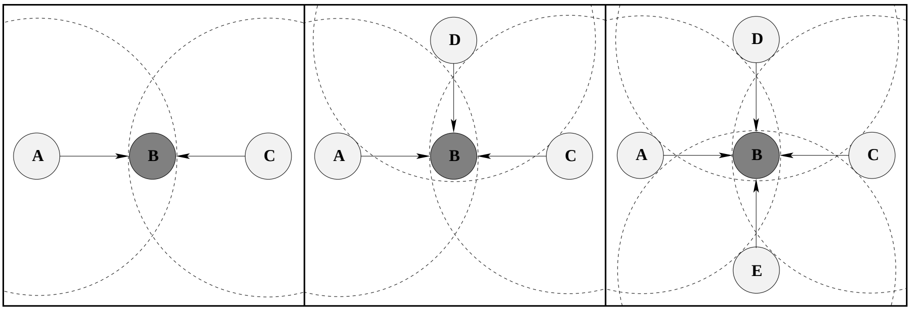


Fig. 5. 1, 2 and 3-hidden scenarios.

and conditional collision probability (Fig. 4(d)) show a similar tendency with the one observed in the throughput. Note also that the energy per byte clearly depends on the number of active nodes, more transmitting nodes implies a higher number of collisions and also an increased overhearing cost. Apart from the quantitative results, it is worth noting how the simulation and the analytical model closely match.

5.2 n_{h}-Hidden Scenario

This scenario is formed by a receiver, located in the middle, and a variable number (from 2 to 4) of source nodes. All source nodes are hidden from each other and they do not have any transmitting neighbour (see Fig. 5). The term n_{h}-hidden refers to the case where there are $n_{\mathrm{h}} + 1$ source nodes, therefore from each node point of view the number of hidden terminals is n_{h}. As the distance from the different source nodes to the receiver is the same and the path loss model is used, no capture effect is possible (i.e., all packets that suffer a collision are lost).

These scenarios are helpful in order to quantify the effect of hidden terminals when they do not reduce their transmission attempts because of communication in the neighbourhood. These scenarios can also be seen as a simplification of a random scenario at very low load conditions, when the effect of hidden terminals dominates among the effect caused by neighbours. Observe that this scenario obeys the assumptions stated in Section 4.2 where the analytical model in presence of hidden terminals is described.

Results are shown in Fig. 6. Note that although the number of hidden terminals is low, the network starts showing losses at an offered load equal to 0.4 packets/s per node, see Fig. 6(a). In the same figure it can be observed that when the collision probability is higher than 50% (Fig. 6(d)), the throughput shows an inflection point and starts to decrease with the offered load until it reaches zero, meaning that the successive overlapping collisions prevent the successful reception of any packet at high loads. Regarding the energy consumption per byte (Fig. 6(b)), it decreases with the number of hidden terminals only due to the small number of collisions since the effect of overhearing is not present. Observe also that as there is no overhearing, the delay, shown in Fig. 6(c), is the same for the different configurations. The conditional collision probability (Fig. 6(d)), as well as the other metrics, show the same tendency found in the throughput. Notice also in this scenario, the close match between the analytical model and the simulations.

As a conclusion of the results of this scenario, it can be said that the effect of hidden terminals is non-negligible even at the lower loads considered, but when the load is considerable, the network is unable to provide any service at all.

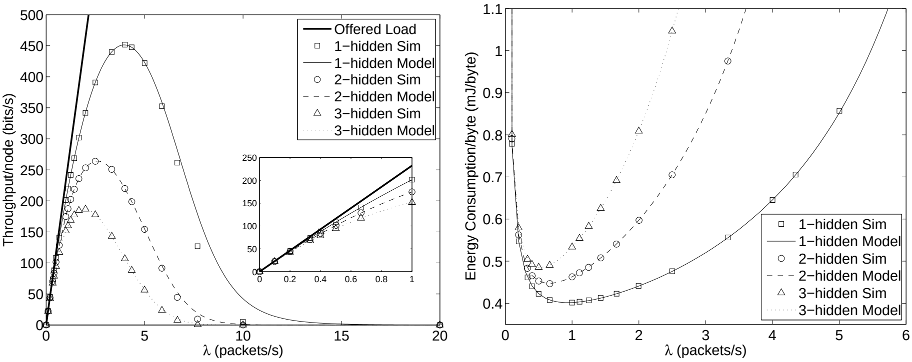
5.2.1 n_{h}*-Hidden vs Single Hop*

An interesting result is to compare the performance obtained in the n_{h}-hidden scenarios with the single hop ones (all nodes inside the coverage range of each other) when the number of sources nodes is the same. This gives an idea of the performance decrease of having a hidden terminal instead of a neighbour.

Fig. 7 shows the throughput decrease comparing the one obtained in the single hop scenario for $2, 3$ and 4 source nodes with the obtained in the different n_{h}-hidden scenarios. It can be observed that, even for the lower loads, the reduction is quite important. For instance, at 1 packet/s, the decrease is more than 10, 20 and 30% with 1, 2 and 3 hidden terminals respectively. At higher loads, the decrease is equal to 100% since the throughput in the n_{h}-hidden scenarios tends to zero.

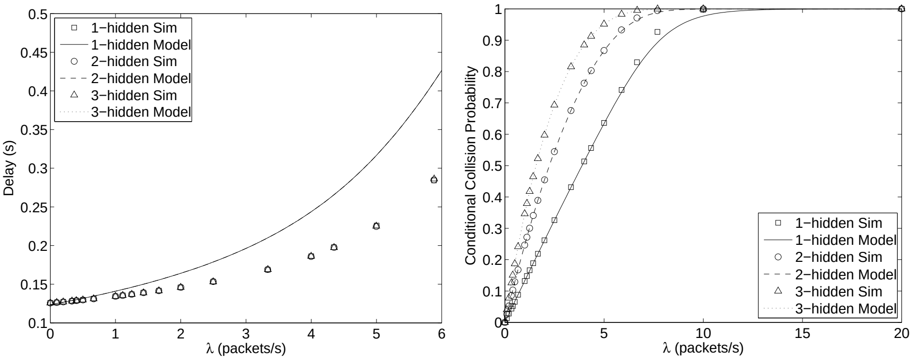
5.2.2 n_{h}*-Hidden with Capture*

In order to evaluate how the capture effect can mitigate the hidden terminal problem, source node A of the scenario shown in Fig. 5 has been deliberately moved in order to be closer to the receiver. The differences among the



(a) Total throughput per node. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 1$ packets/s).

(b) Energy consumption/byte received



(c) Delay

(d) Conditional collision probability

Fig. 6. Performance results in the n_h-hidden scenarios.

received signal strengths are such that the packet error rate of a packet coming from node A that suffers a collision affecting the entire message (even if all of the n_h hidden nodes collide) is negligible.

The increase in the aggregated throughput of the n_h-hidden scenarios with capture is shown in Fig. 8. In this specific case, in which only the packets coming from node A can be captured, the gain is not very high (less than 10%) for offered loads lower than 1 packet/s due the low collision probability which is shown in Fig. 6(d). However, as soon as the load increases, the gain of the capture becomes noticeable since without capture no packet can be correctly received.

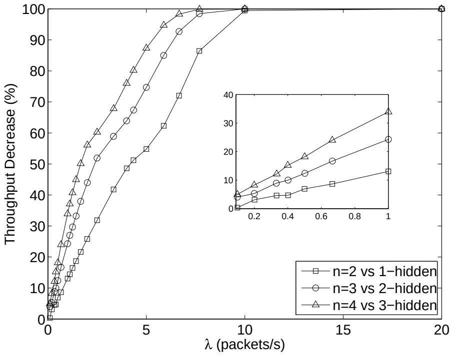


Fig. 7. Decrease in the aggregated throughput when all nodes are inside the coverage range compared to the $n_{\rm h}$-hidden scenarios. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 1$ packets/s).

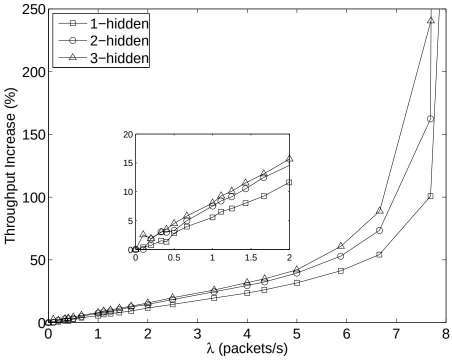


Fig. 8. Improvement in the aggregated throughput with capture effect in the n-hidden scenarios when one of the source nodes is closer to the sink. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 2$ packets/s).

5.3 Grid Scenario

In this scenario nodes are placed equidistant to each other forming a grid as shown in Fig. 9. The node in the middle of the grid does not transmit while the others transmit to a randomly selected neighbour. To avoid border effects a 9×9 grid has been considered but only the metrics measured at the node in the middle of the grid and at one of its neighbours are considered.

The goal of this scenario is to compare the 3-hidden scenario to a case where the interferences produced by hidden terminals are decreased due to communication in their neighbourhood. Observe that each neighbour of the node

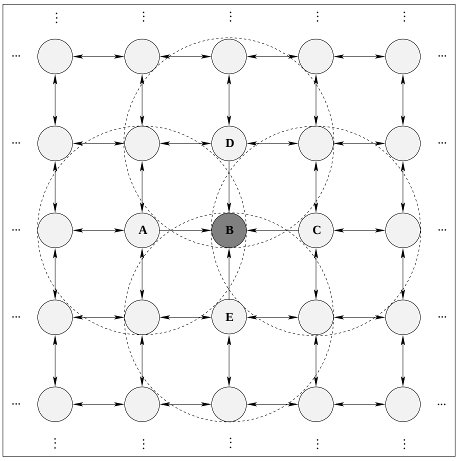
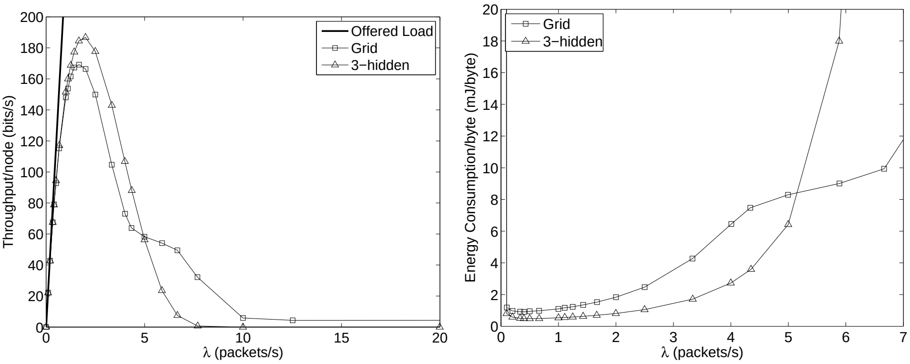


Fig. 9. Grid scenario.

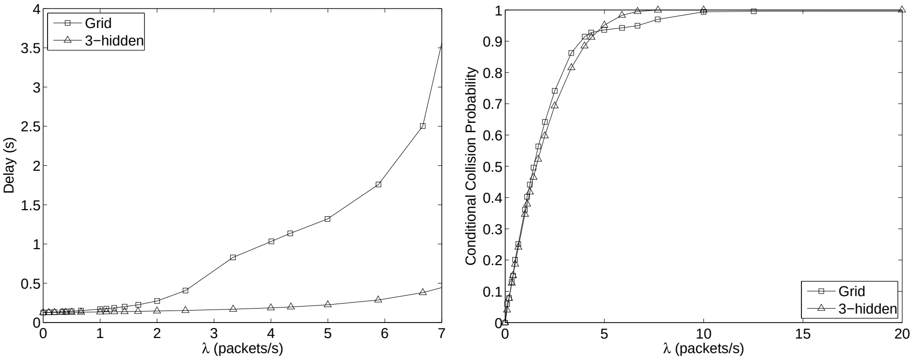
in the middle (hidden terminals among them) has 3 transmitting neighbours. Thus, these hidden nodes will reduce their transmission attempts as they have to contend with their neighbours to access the channel, increasing the BO duration accounting for the possible successful transmissions or collisions between their neighbours.

In this scenario, due to the equidistant distances among neighbours and the propagation model used, the capture effect is also not possible.

The results, compared to the ones obtained in the 3-hidden scenario, are depicted in Fig. 10. Related to the throughput (Fig. 10(a)), it can be seen that for loads lower than 4 packets/s, the results of the grid follow the same tendency than the results of the 3-hidden scenario. However, the throughput achieved is smaller due to the increased BO duration since source nodes should wait until transmissions in their neighbourhood finish. For loads higher than 4 packets/s a higher throughput, if compared to the 3-hidden case, can be observed. The reason of this improvement is a reduction of the probability that a hidden terminal transmits during a given packet transmission caused by communication in its neighbourhood (it finds the medium busy). As previously seen in the other scenarios, the results of energy consumption, delay and conditional collision probability, depicted in Figs. 10(b-d) respectively, show a clear similarity with the tendency observed in the throughput.



(a) Total throughput per node (b) Energy consumption/byte received



(c) Delay (d) Conditional collision probability

Fig. 10. Performance results in the grid scenario.

As a concluding remark, similar to the 3-hidden scenario, losses are noticeable even at the lower loads considered. However, a slight improvement is found at high loads due to the reduced hidden terminal transmission attempts.

5.4 Random Scenario

In this scenario a variable number of sensor nodes (100, 200 and 300) are randomly placed in a 500×500 m^2 area, resulting in three different densities ($\gamma_{100} = 4 \cdot 10^{-4}$, $\gamma_{200} = 8 \cdot 10^{-4}$ and $\gamma_{300} = 12 \cdot 10^{-4}$ nodes/m^2 respectively). In order to avoid routing effects, each node sends data to a randomly selected neighbour, see Fig. 11. Only metrics computed at inner nodes are analyzed, with the goal of avoiding border effects.

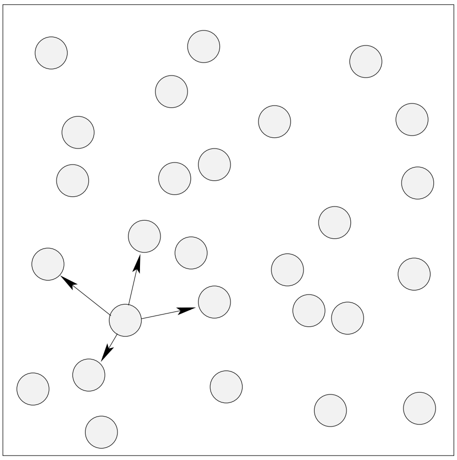


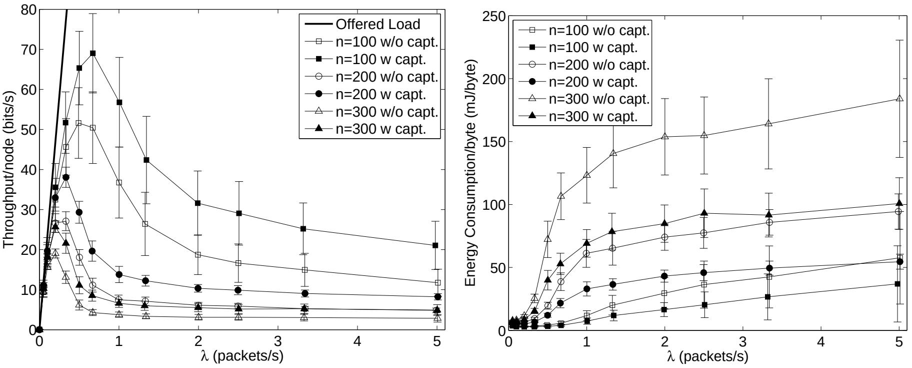
Fig. 11. Random scenario.

The aim of this scenario is to analyze how the performance of the network changes by increasing the average number of hidden terminals. However, as the number of hidden terminals per communication pair is modified by changing the density, the total neighbours per node also vary. This effect complicates the comparison among the different configurations as changes in the number of hidden terminals also imply a variation in the number of neighbouring nodes.

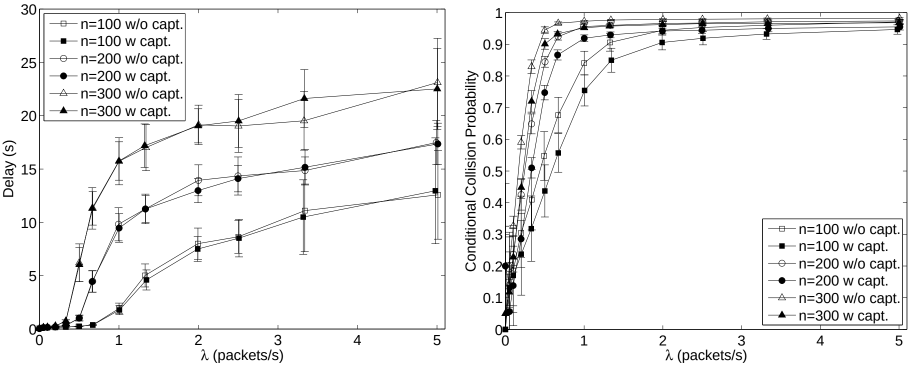
The number of neighbours per node and hidden terminals per communication pair can be approximated by considering the path loss propagation model (although the shadowing model is used in this scenario). The average neighbours (those nodes that can sense the target node transmissions, i.e., in the carrier sense range) per node is obtained as: $n = \gamma \pi r'^2 - 1$, where $r' = 98$ m given the transmission power, propagation model and carrier sense threshold considered. To compute the average number of hidden nodes, the exclusive area of two circles of the same radius (r') separated a distance d is:

$$A_\mathrm{e} = \pi r'^2 - \left(2r'^2 \cos^{-1}\left(\frac{d}{2r'}\right) - \frac{1}{2}d\sqrt{4r'^2 - d^2}\right) \qquad (24)$$

The distance between the two centers that makes the inner area equal to the outer one is $d = r'/\sqrt{2}$, therefore the average number of hidden terminals



(a) Total throughput per node (b) Energy consumption/byte received



(c) Delay (d) Conditional collision probability

Fig. 12. Performance results in the random scenario with (w capt.) and without (w/o capt.) capture effect.

(average number of nodes in the exclusive area) can be computed as $n_e = \gamma A_e$.

Solving for the different densities considered, we have that there are on average 11 neighbours and 5 hidden terminals when there are 100 nodes in the network, 23 and 10 when there are 200 and 35 and 15 when 300 nodes are considered.

Fig. 12 shows the results (average value and 95% confidence intervals of 10 independent simulations with nodes placed in different positions) with the capture effect activated (w capt.) and deactivated (w/o capt.). It can be observed that the throughput achieved (Fig. 12(a)) is significantly smaller than the offered load for the entire range of loads considered. Even for the lower loads (see Fig. 13) the percentage of lost throughput is significant. For instance, with 100 nodes without the capture effect enabled, this difference is approximately 30% for only 0.2 packets/s offered load and more than 80% for 1 packet/s (a bit less if the capture effect is activated). Observe, once again, the similarities of the tendency among the different metrics (energy per byte, delay and conditional collision probability, Figs. 12(b-d)) and the throughput. It is important to note the considerable improvement obtained in the different metrics, except the delay (as retransmissions are not considered), when the capture effect is activated, however, the low performance is maintained.

The configuration with 100 source nodes can be compared to the single hop case with $n = 10$ (Fig. 4) as the number of neighbours is approximately the same. However, in the random case the average number of hidden terminals per communication pair is 5 while in the single hop case there are no hidden terminals present. Observe that the maximum achievable throughput differs in 80 bits/s (approximately a 53% decrease) considering the capture effect case. Moreover, losses appear at offered loads lower than 0.1 packets/s while in the single hop case losses start at 1 packet/s. Therefore, having 5 hidden terminals (on average) affects the performance of the network noticeably, even for the lower offered loads.

As concluding remarks it should be pointed out that, as already seen in the previous scenarios, the effects of hidden terminals are important at high loads but they are also considerable at lower loads. Moreover, it has been shown that the capture effect provides considerable improvements when there are hidden terminals present although the global performance of the network is still significantly low.

5.5 Random Scenario with Event-based and Periodic Traffic Profiles

While the rest of the scenarios evaluated in this work provide a simple framework to quantify the problem, the goal of this scenario is to evaluate the hidden terminal effect in a more realistic network. The scenario of interest is based on the previous random scenario but with the following extensions[3]:

---

[3] Other configuration settings like the propagation model and carrier sense are the same as the ones considered in the random scenario of Section 5.4.

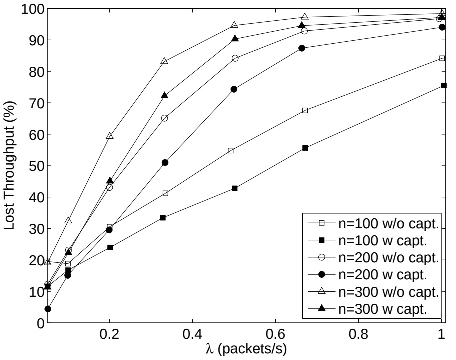


Fig. 13. Percentage of lost throughput per node for less than 1 packet/s in the random scenario.

- Two traffic profiles are considered: periodic low rate messages and event-based data coming from event detection.
- The ACK mechanism and retransmissions are enabled.
- Convergecast communication is considered: sensor nodes send all their data to the sink.

It has been considered that sensor nodes send messages periodically to the sink, this traffic profile is based on a typical monitoring application. At each sensor node the time between periodic message generation is constant and it has been set equal to 1 hour[4]. Moreover, if sensor nodes detect an event they notify the sink immediately by sending an event-based message. The event radius is considered equal to 150 m, therefore nodes with a distance to the event center smaller than this value will detect the event and will generate a message. Communication from the sensor nodes to the sink is based on the shortest path. Routes to the sink are precomputed in order to avoid routing messages to interfere with the simulation. The capture effect has been enabled, however if a packet transmission fails at the MAC layer (the ACK is not received) sensor nodes wait a random BO and retry transmission given that the maximum retry limit (set to 5) has not been reached. The ACK length has been set equal to 5 bytes and it has been considered that a receiver waits 11σ to respond with an ACK after a data packet reception.

A 200-node network in which sensor nodes continuously send periodic messages is simulated. After 5, 400 s of the simulation start an event is generated

---

[4] The first generation instant is different for all sensor nodes.

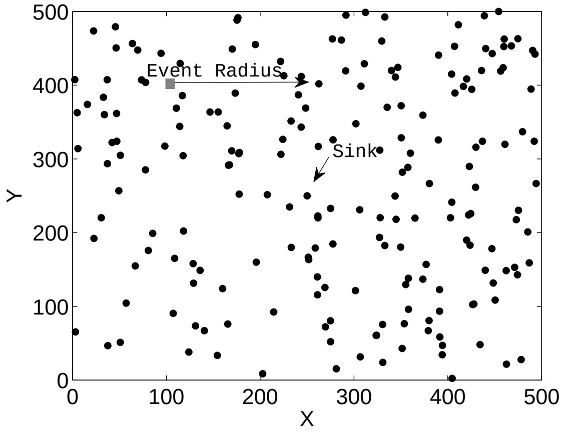
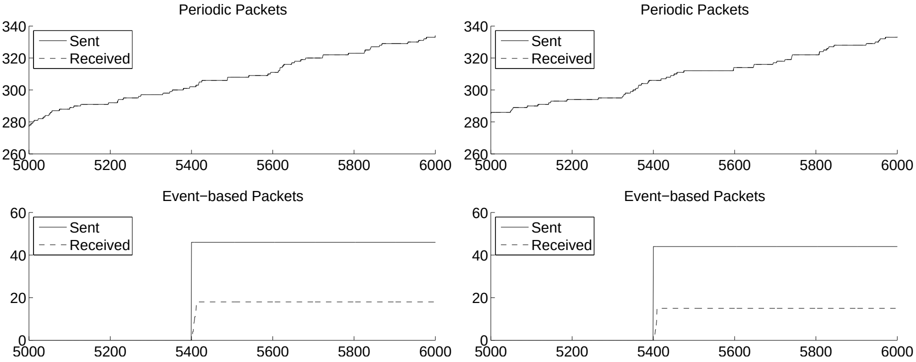


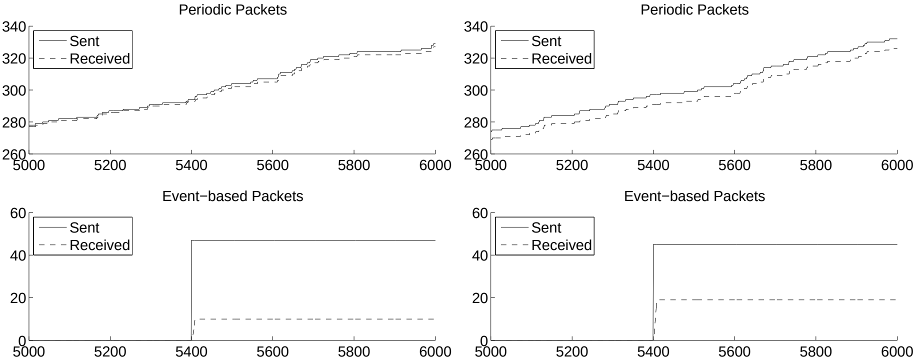
Fig. 14. Random scenario with event generation.

in a specific position (see Fig. 14). In that moment, sensor nodes inside the radius of the event try to send a message to the sink (placed in the middle of the topology). Results of 4 independent simulations (with different placement of sensor nodes) are shown in Fig. 15. It shows the number of received and sent packets of each traffic profile. It can be observed that the periodic messages are nearly all received (only *'Simulation 3'* and *'Simulation 4'* show 2 and 4 periodic packets lost respectively) while the event-based messages suffer from a higher loss rate. Although not all the messages must be received at sink to reliably detect an event [8], the reliability obtained is considerably low (a $32,3\%$ on average). This low reliability in the event-based messages is caused by the increment of contention among those sensor nodes detecting the event, both, nodes in coverage range and hidden terminals.

The results of this scenario reflect for a more realistic case the conclusions obtained by the quantification analysis of the more controlled scenarios. An increase of the network load causes a performance reduction due to higher contention among neighbour sensor nodes but also due to the high vulnerability time in collisions among hidden terminals.



(a) Simulation 1 (b) Simulation 2



(c) Simulation 3 (d) Simulation 4

Fig. 15. Performance results (number of packets sent and received over time) in the random scenario with periodic and event-based traffic profiles.

6 Effects of Hidden Terminals in Improved Preamble Sampling Techniques

From the first definition of preamble sampling [12, 14], several extensions to improve its performance have been proposed. However, for simplification purposes, only the basic preamble sampling technique with CSMA medium access has been studied in this work. Thus, the goal of this section is to discuss how the hidden terminal problem can affect those improved preamble sampling techniques present in the literature. The main preamble sampling improvements can be divided into three broad categories: *i)* the division of the long preamble into series of small short packets, *ii)* the reduction of the long preamble by remembering the receiver wake up time and *iii)* the adaptation of the duty cycle (and thus, the preamble) to the traffic load.

The division into small and short packets of the long preamble (for instance, the approaches presented in [5, 6, 20]) allows a reduction of overhearing by including the destination identifier (ID) of the data packet, therefore those nodes that do not need to receive the message can go to sleep. Moreover, some techniques leave idle spaces between the short packets to allow the transmission of an early ACK that stops the transmission of the preamble. Observe that although considerable benefits can be obtained in terms of energy consumption, the hidden terminal problem affects in a similar way as if the long preamble is transmitted. Some benefits can be obtained with the transmission of the early ACK since it reduces the size of the preamble and therefore the vulnerability time, however, collisions can still happen during the transmission of (on average) half of the preamble and the data packet as interfering nodes can be sleeping during the transmission of the early ACK.

The protocols that remember the wake up time of the receiver and send only the necessary preamble to cope with the clock drifts (some of the most well-known approaches are [13, 15, 30]) allow a notable reduction of the duration of the long preamble and thus, the vulnerability time. For these cases in which hidden terminals transmit to receivers that have different wake up times, the hidden terminals effects are, therefore, decreased if compared to the transmission of the entire long preamble. However, note that as all nodes (hidden terminals or not) that want to transmit to a common receiver will try transmission a few instants before it wakes up, the probability of collision increases. Moreover, if all nodes wake up at approximately the same time, as defined in [30], the probability of correctly receiving a packet in presence of hidden terminals decreases considerably.

The preamble sampling protocols that can, to some extent, lighten the effects of hidden terminals are those that are capable of adapting their behaviour to the traffic load (for example, different approaches of traffic load adaptability are the works presented in [6, 7, 16, 22]). Some of them adapt the duty cycle, and therefore, the duration of the long preamble to the traffic load while others define hybrid protocols in which as soon as the load increases synchronous approaches are adopted. However, how to rapid and efficiently estimate the load of the network is a challenge in common low-capable sensor nodes, even more difficult when event-based traffic profiles are present as sporadic increases of the load can happen due to event detection at nearby sensor nodes.

7 Conclusions

In this work, the effects of hidden terminals in preamble sampling WSNs have been studied. The performance of the network has been evaluated by means of a preamble sampling simulator and an analytical approach. The analytical framework presented in this work is the first approach to model the hidden terminal effects in preamble sampling WSNs considering the desynchronization among them and it has shown a good accuracy compared to simulation results.

Results for different kinds of scenarios and traffic profiles have shown that the hidden terminal problem is of high importance at medium to high loads, that can be caused for instance by event detection. It has also been illustrated that the capture effect cannot substantially improve the overall performance of the network.

These results should be considered by designers of MAC protocols in order to develop mechanisms to lighten those effects, specially if increments of the network load are expected in specific moments (for instance due to event detection or query dissemination) or regions (those nodes closer to sink transmit more data). From the results obtained, two main recommendations to MAC designers can be provided. The first one is the consideration of adaptive approaches capables of reducing the vulnerability time (for instance decreasing the long preamble transmission duration) as soon as the network load increases, thus mitigating the negative effects of hidden terminals from medium to high loads. Another recommendation that can be derived is the avoidance of too aggressive retransmissions. Although the effect of retransmissions is not deeply studied in this work, it can be devised that some retransmission techniques could lead to continuous and overlapping collisions due to the desynchronization among hidden terminals, making the network unusable.

References

1. IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11*, 1999 Edition (Revised 2007).

2. F. Alizadeh-Shabdiz and S. Subramaniam. Analytical Models for Single-hop and Multi-hop Ad hoc Networks. *Mobile Networks and Applications*, 11(1):75–90, 2006.
3. J.B. Andersen, T.S. Rappaport, and S. Yoshida. Propagation Measurements and Models for Wireless Communications Channels. *IEEE Communications Magazine*, 33(1):42–49, 1995.
4. A. Bachir, D. Barthel, M. Heusse, and A. Duda. Hidden Nodes Avoidance in Wireless Sensor Networks. In *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*, 2005.
5. A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications (ICC 06)*, pages 3365–3370, 2006.
6. M. Buettner, G.V. Yee, E. Anderson, and 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 (Sensys '06)*, pages 307–320, 2006.
7. C. Cano, B. Bellalta, A. Sfairopoulou, and J. Barceló. A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs. *IEEE Communication Letters*, 13(4):221–223, 2009.
8. D. Chen and P.K. Varshney. QoS Support in Wireless Sensor Networks: A Survey. In *Proceedings of the International Conference on Wireless Networks (ICWN'04)*, pages 227–233.
9. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
10. Chipcon. CC100. Single Chip Very Low Power RF Transceiver. http://focus.ti.com/lit/ds/symlink/cc1000.pdf.
11. M. Durvy, O. Dousse, P. Thiran, and L. EPFL. Modeling the 802.11 Protocol Under Different Capture and Sensing Capabilities. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM 07)*, pages 2356–2360, 2007.
12. A. El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad hoc Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications. (ICC 02)*, volume 5, pages 3418–3423, 2002.
13. A. El-Hoiydi and J.D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. *Algorithmic Aspects of Wireless Sensor Networks*, pages 18–31, 2004.
14. J. Hill and D. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, pages 12–24, 2002.
15. B. Jang, J.B. Lim, and M.L. Sichitiu. AS-MAC: An Asynchronous Scheduled MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. (MASS 2008)*, pages 434–441, 2008.
16. R. Jurdak, P. Baldi, and C. Videira Lopes. Adaptive Low Power Listening for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 6(8):988–1004, 2007.
17. P. Karn. MACA-A New Channel Access Method for Packet Radio. In *Proceedings of the ARRL/CRRL Amateur Radio Computer Networking Conference*, 1990.
18. A. Koubaa, R. Severino, M. Alves, and E. Tovar. H-NAMe: Specifying, Implementing and Testing a Hidden-Node Avoidance Mechanism for Wireless Sensor Networks. *IPP-HURRAY Technical Report, HURRAYTR-071113*, 2008.
19. K. Kredo II and P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, 51(4):961–994, 2007.

20. S. Mahlknecht and M. Bock. CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks. In *Proceedings of the IEEE International Workshop on Factory Communication Systems*, pages 73–80, 2004.
21. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
22. I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, 2008.
23. D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (Sensys '06)*, pages 237–250, 2006.
24. Y. Sun, O. Gurewitz, and 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 (Sensys '08)*, pages 1–14, 2008.
25. F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-access and the Busy-tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.
26. A. Tsertou and D.I. Laurenson. Revisiting the Hidden Terminal Problem in a CSMA/CA Wireless Network. *IEEE Transactions on Mobile Computing*, 7(7):817–831, 2008.
27. Y. Wang, T. Li, and M. Xiong. Performance Analysis of CSMA with Preamble Sampling for Multi-hop Wireless Sensor Networks. In *Proceedings of the International Conference on ITS Telecommunications Proceedings*, pages 921–924, 2006.
28. K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the Capture Effect for Collision Detection and Recovery. In *Proceedings of the IEEE Workshop on Embedded Networked Sensors (EmNetS '05)*, pages 45–52, 2005.
29. A. Woo and D.E. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proceedings of the International Conference on Mobile Computing and Networking*, page 235, 2001.
30. W. Ye, F. Silva, and J. Heidemann. Ultra-low Duty Cycle MAC with Scheduled Channel Polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, page 334, 2006.
31. M. Zuniga and B. Krishnamachari. Analyzing the Transitional Region in Low Power Wireless Links. In *Proceedings of the IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 04)*, pages 517–526, 2004.

Wake up after Transmissions and Reduced Channel Contention to Alleviate the Hidden Terminal Problem in Preamble Sampling WSNs

C. Cano, B. Bellalta, M. Oliver

Universitat Pompeu Fabra

Abstract. The hidden terminal problem is a well-known problem in wireless networks that degrades the network performance due to collisions between nodes unable to sense each other. Preamble sampling WSNs are more sensitive to that problem than wireless local area networks due to the high vulnerability time caused by the long preamble transmission. This is specially important when the traffic load is high, for instance due to event detection and query dissemination or in regions closer to the sink. Typical mechanisms to reduce the negative effects of hidden terminals, like the use of RTS/CTS messages, cannot alleviate the problem in preamble sampling WSNs as the long preamble is still needed, thus not reducing the high vulnerability time. To alleviate the hidden terminal problem in this kind of networks the LWT-MAC, a new preamble sampling MAC protocol, is presented. The LWT-MAC protocol adapts to the sporadic increases in the network load by waking up sensor nodes at the end of a transmission. At the same time, it reduces the channel contention by implementing a non-aggressive retransmission procedure. In this work, the ability of the LWT-MAC protocol to reduce hidden terminal problems is evaluated and compared to the basic preamble sampling technique. Results show that the LWT-MAC protocol is able to significantly lighten the hidden terminal problem, thus improving the overall system performance.

1 Introduction

During the last few years several new Medium Access Control (MAC) protocols have been designed to cope with the energy consumption constraint of Wireless Sensor Networks (WSNs). These approaches were mainly motivated by the fact that the MAC layer controls the transceiver, which is the most energy consuming component of a sensor node. The common approach is to perform a low duty cycle operation in which sensor nodes sleep during most of the time and wake up only to send or receive data. These approaches can be divided in two major categories [15]: *i)* scheduled, where some kind of organization to decide when to sleep and wake up is performed among sensor nodes and *ii)* unscheduled, where sensor nodes decide their duty cycle independently of the other nodes in the network.

Unscheduled (or random) MAC protocols are specially well suited for WSNs as precise synchronization is not needed, they are simpler and in most cases consume less energy than scheduled approaches [15]. Among these, preamble sampling, or Low Power Listening (LPL), protocols [9, 11, 17], have shown very good results in terms of energy consumption as they are capable of working at considerably low duty cycles ($\sim 1\%$). In this kind of protocols, sensor nodes wake up during a short period of time just to sample the medium: if the medium is empty they return to sleep, otherwise they remain awake to receive the data. A sensor node, before sending a message, transmits a long preamble that overlaps with the listening time of the receiver, thus assuring it will be awake to receive the packet. At low loads the energy consumption is extremely reduced but as soon as the load increases (for instance due to event detection or query dissemination), collisions become very costly as the entire long preamble is involved. Moreover, collisions among hidden terminals become more problematic than in Wireless Local Area Networks (WLANs) as the vulnerability time (defined as the period of time in which a collision can happen if any other node transmits) includes the long preamble transmission that is, usually, in the order of hundreds of milliseconds. The Request To Send (RTS)/Clear To Send (CTS) mechanism, that is able to considerably reduce the hidden terminal problem in IEEE 802.11-based wireless networks [1], is not able to solve the problem in preamble sampling WSNs as the long preamble should be transmitted at least before the RTS, and hence the long vulnerability time remains approximately equal.

The Low power listening with Wake up after Transmissions MAC (LWT-MAC) protocol [6] is an extension of the preamble sampling technique that is able to:

1. Adapt to sporadic increases in the network load: The LWT-MAC protocol wakes up overhearing nodes at the end of a transmission to send or receive data. When all nodes are awake (scheduled phase) there is no need to send the long preamble before the data transmission.
2. Alleviate hidden terminal problems: The LWT-MAC approach uses RTS/CTS messages in the scheduled phase and waits a Collision Avoidance (CA) Timer listening to the channel before trying to retransmit.

The LWT-MAC protocol was presented in [6], where an initial performance evaluation in single hop and multihop scenarios were performed. After that, the behavior of the protocol with periodic and event-based traffic profiles and its ability to increase the collective Quality of Service (QoS), that is the QoS observed by the set of packets belonging to a specific event, was studied in [5]. In this work, the ability of this protocol to reduce hidden terminal problems is evaluated. Its results are compared with the ones obtained using the basic preamble sampling technique in different scenarios and under a broad range of traffic loads.

The rest of the paper is organized as follows: Section 2 describes the hidden terminal problem in preamble sampling WSNs and an evaluation of the different solutions to reduce it. After that, Section 3 describes the functionality of the LWT-MAC protocol. The scenario considerations are described in Section 4 while in Section 5 the performance evaluation results are discussed. Finally, some conclusions are provided.

2 The Hidden Terminal Problem in Preamble Sampling WSNs

The hidden terminal problem [20] is a well-known problem that occurs in wireless networks. It happens when two or more nodes, that are unable to sense each other, transmit to a common receiver, causing a collision. In WLANs, the most common way to alleviate this problem is the use of the RTS/CTS mechanism [14]. For instance, it is used in IEEE 802.11 networks as an optional feature. Using this method, a sender first transmits an RTS message that is replied with a CTS by the destination if it is ready to receive the data. As the length of the RTS is considerably small (if compared to data messages common in WLANs), the vulnerability time is reduced, thus mitigating the effects of the hidden terminals. Moreover, the CTS message informs hidden terminals about the transmission that is going to take place.

On the contrary, in preamble sampling WSNs, the vulnerability time is high due to the transmission of the long preamble, thus the hidden terminal is considerably more problematic than in WLANs. Note that if the RTS/CTS mechanism is used, the long preamble should be transmitted at least before the RTS message (to wake up the receiver), but also before the CTS to guarantee the correct data transmission (otherwise, hidden terminals can be sleeping when the CTS message is transmitted). This approach results in a large overhead if compared to the typically small data messages found in this kind of networks.

The hidden terminal problem becomes even more problematic when the network load starts to increase. Although WSNs usually work at very low loads, instantaneous increases in the network load, that can be caused by event detection or query dissemination, can happen. Therefore, it is important to consider also medium to high traffic loads.

2.1 The Hidden Terminal Problem in Improved Preamble Sampling Techniques

In this work the LWT-MAC protocol performance is, for simplicity, compared to the basic preamble sampling technique. Although several extensions of preamble sampling MAC protocols exist, they are also prone to the hidden terminal effects. The main preamble sampling improvements can be divided into three broad categories: *i)* the division of the long preamble into a series of small short packets, *ii)* the reduction of the long preamble by remembering the receiver wake up time and *iii)* the adaptation of the duty cycle (and thus, the preamble) to the traffic load.

The division of the long preamble into small and short packets (for instance, the approaches presented in [3, 4, 16]) allows a reduction of overhearing by including the destination identifier (ID) of the data packet, therefore those nodes that do not need to receive the message can go to sleep. However, it does not provide any benefit regarding the hidden terminal problem as the vulnerability time is maintained. Only those approaches that stop the set of short packets by allowing the transmission of an early acknowledgement (ACK) can slightly alleviate those problems as the vulnerability time is halved on average. However, in that case, due to the desynchronization among hidden terminals, it can be impossible for the receiver to find a gap to send the early ACK, thus behaving as the non early ACK technique.

The protocols that remember the wake up time of the receiver and send only the necessary preamble to cope with the clock drifts (some of the most well-known approaches are [10, 12, 22]) allow a notable reduction of the duration of the long preamble and thus, the vulnerability time. Therefore, for these cases in which hidden terminals transmit to receivers that have different wake up times, the hidden terminal effects are decreased if compared to the transmission of the entire long preamble. However, all nodes (hidden terminals or not) that want to transmit to a common receiver will try transmission a few instants before it wakes up. Thus, the probability of collision increases. Moreover, if all nodes wake up at approximately the same time, as defined in [22], the probability of correctly receiving a packet in presence of hidden terminals considerably decreases.

The preamble sampling protocols that, to some extent, can lighten the effects of hidden terminals are those that are capable of adapting their behaviour to the traffic load (for example, different approaches of traffic load adaptability are the works presented in [4, 13, 18]). Some of them adapt the duty cycle, and therefore, the duration of the long preamble to the traffic load. However, how to rapidly and efficiently estimate the load of the network is a challenge in common low-capable sensor nodes, even more difficult when event-based traffic profiles are present as sporadic increases in the load can happen due to event detection at nearby sensor nodes.

2.2 Mechanisms to Alleviate the Hidden Terminal Problem in Preamble Sampling

To the best of the authors' knowledge there is no approach that efficiently mitigates the hidden terminal problem in preamble sampling WSNs.

The authors of [16] propose, in order to mitigate the hidden terminal problem, to increase the sensitivity of the sensor nodes. This technique is able to partially solve the problem at the expense of reducing the throughput and increasing the node energy consumption since higher sensitivity implies higher current draw [8].

Other works, such as [21], where a tree-like topology is considered, recommend to delay the transmissions each time a node overhears a message from its parent, assuming that its grandparent will forward the message after that. However, the solution is limited to tree topologies and it does not consider that hidden terminal problems may also happen between independently generated messages.

Another proposed solution is to tune the backoff (BO), increasing either the slot duration [2] or the Contention Window (CW) [19]. These techniques are effective for non-preamble sampling WSNs where message transmissions are comparable to BO durations. However, the BO values that could alleviate collisions among hidden terminals in preamble sampling WSNs are so high that they would drastically reduce the performance of the network.

3 Low Power Listening with Wake Up after Transmissions: The LWT-MAC Protocol

The LWT-MAC protocol [6] is a simple low power listening MAC protocol designed with the goal to make preamble sampling MAC protocols able to react to instantaneous increases in the network load, specially due to event detection. To cope with increases in the network load, the LWT-MAC protocol synchronizes neighbouring sensor nodes to eliminate the long preamble transmission. It also reduces the channel contention in case of a packet failure with the goal to alleviate hidden terminal problems.

3.1 Basic Functionality

The functionality of the protocol is based on the basic preamble sampling technique, the Berkeley MAC (B-MAC) protocol [17]. A sensor node willing to transmit first sends a long preamble to overlap with the listening time of the receiver and after that, it sends the data packet. However, in the LWT-MAC protocol, all the nodes that have overheard a transmission wake up at the end to send or receive packets. As all the sensor nodes that have overheard a transmission will be awake there is no need to send the long preamble. If all nodes wake up and there is no more data to send they return to sleep after the maximum time that can be spent in BO (σCW, where σ is the duration of an empty slot). Otherwise, sensor nodes wait a random BO and start transmission. In this kind of *scheduled* phase the RTS/CTS mechanism is used in order to cope with hidden terminal problems (see Fig. 1).

3.2 Retransmissions

The retransmission procedures are depicted in Figs. 2 and 3 depending on whether sensor nodes are in the unscheduled or scheduled phase.

- Unscheduled Phase: As can be seen in Fig. 2 if a packet fails in the unscheduled phase a retransmission using an RTS is tried. This feature aims at increasing the probability of receiving at least the last RTS of a collision among hidden terminals. This mechanism (called *RTS if DATA fails*) is evaluated in this work comparing its results with the basic access mechanism.

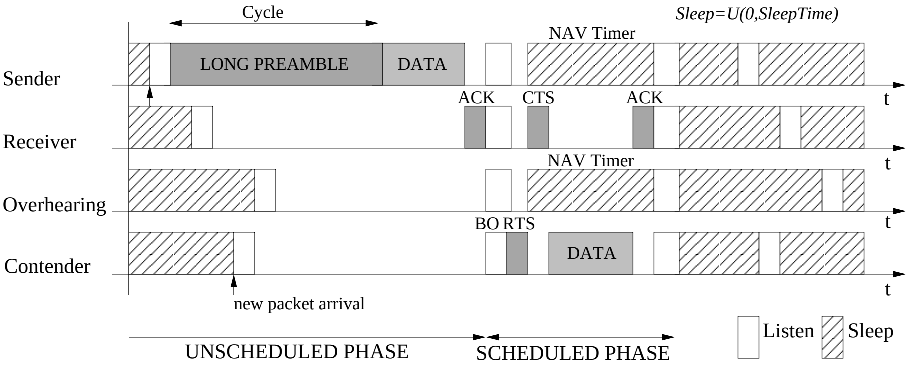


Fig. 1. LWT-MAC channel access

- Scheduled Phase: In case that an RTS or CTS transmission failure occurs at the scheduled phase, the sensor nodes must return to the unscheduled phase and retry transmission by sending the long preamble first as the recipient can be sleeping if it has not overheard the last transmission (see Fig. 3).
- The CA Timer: Note also that instead of retransmitting a frame immediately in the unscheduled phase, nodes wait a CA Timer (set to the entire long preamble and packet transmission) listening to the channel. The goal of the CA Timer is to let other transmissions finish and also increase the chances to receive a packet of a current ongoing transmission and get synchronized to it. For instance, if a transmission was initiated at a hidden terminal, nodes willing to retransmit can receive CTS or ACK messages from the receiver, then they will be able to wake up at the end of the ongoing transmission. The CA Timer is also used in the scheduled phase if the ACK of the data packet is not received.

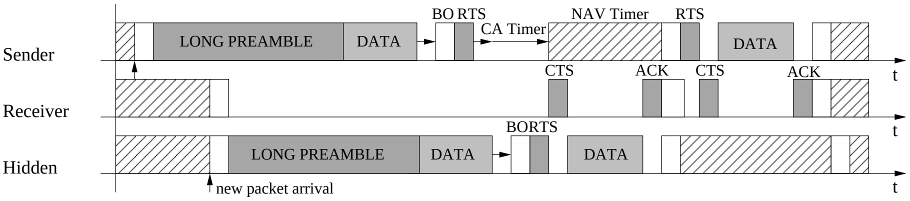


Fig. 2. Retransmission in the unscheduled phase

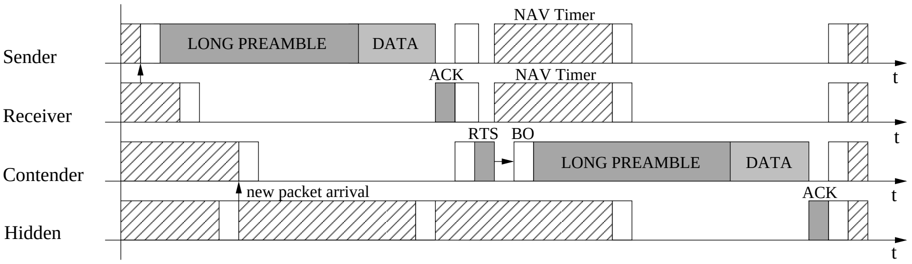


Fig. 3. Retransmission due to *desynchronization* among transmitter and receiver in the scheduled phase

4 Scenarios

With the goal to evaluate the performance of the LWT-MAC protocol in presence of hidden terminals four different scenarios have been considered:

1. **Single Hop**: The first scenario is a single hop network in which all nodes are in coverage range with the rest (Fig. 4). This configuration allows to obtain a reference to the performance of the network when no hidden terminals are present.
2. n_h**-Hidden**: The second scenario is formed by a set of $(n_\mathrm{h} + 1)$ transmitting nodes that are hidden from each other and that transmit to a common neighbour (see Fig. 5). This scenario makes it possible to evaluate the performance of the network when it is only affected by transmissions of hidden terminals.
3. **Grid**: The third scenario is a grid topology, as shown in Fig. 6, in which all nodes transmit to a randomly selected neighbour. Observe that it is similar to the 3-hidden case but in this case each transmitting node has 3 neighbours that also try to transmit. The goal of this scenario is to analyze the hidden terminal effects when their channel access probabilities are affected by transmissions in the neighbourhood.
4. **Random**: The last scenario is formed by a set of nodes randomly placed in the area that transmit to the sink (see Fig. 7). The goal of this case is to analyze the performance of the protocol in a more realistic scenario.

In the basic scenarios (single hop, n_h-hidden and grid) it has been assumed that sensor nodes generate packets following a Poisson distribution with a constant packet length. In contrast, in the random scenario both periodic and event-based traffic profiles are considered, constant packet length is also considered for the packets associated to that traffic profiles.

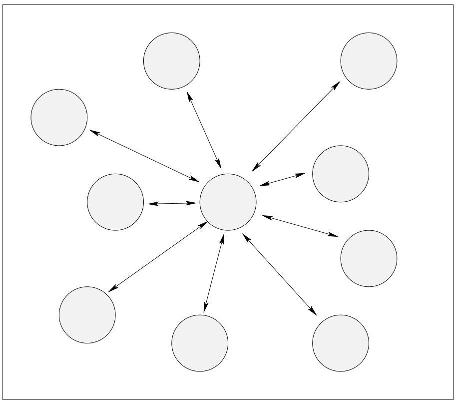


Fig. 4. Single hop scenario

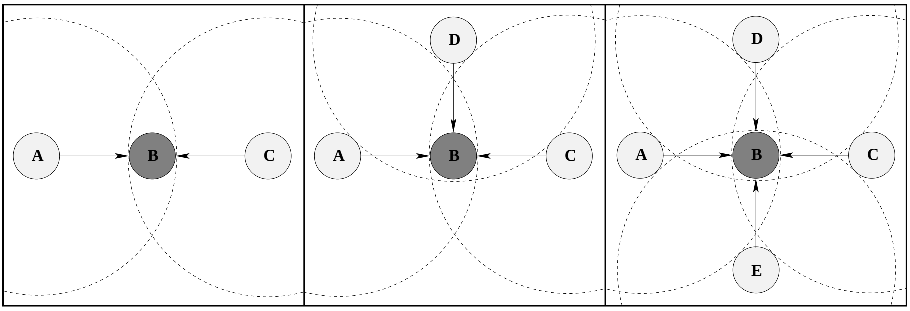


Fig. 5. 1, 2 and 3-hidden scenarios.

4.1 Evaluated Protocols

To evaluate the performance of the proposed approach, the LWT-MAC has been compared, for simplicity, with the basic preamble sampling technique (the **B-MAC** protocol). Moreover, with the goal to deeply analyze how the different mechanisms of the LWT-MAC protocol affect the performance

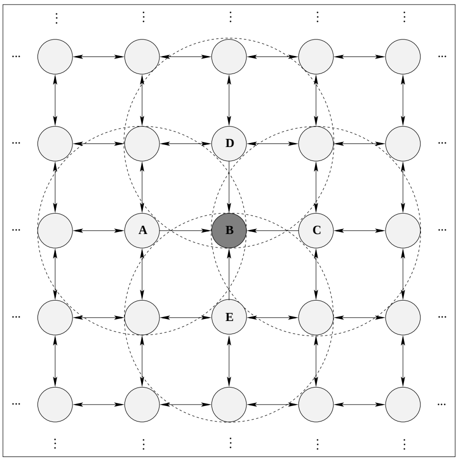


Fig. 6. Grid scenario.

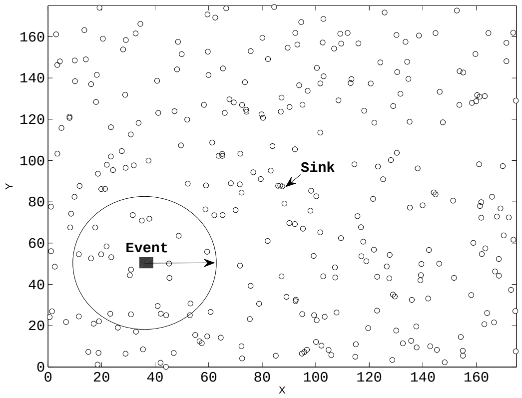


Fig. 7. Random scenario.

in presence of hidden terminals, the results have been obtained with three different configurations:

1. Basic **LWT-MAC**: sensor nodes wake up at the end of a transmission to send or receive packets using the RTS/CTS mechanism.
2. **LWT-MAC+CA**: sensor nodes wake up at the end of a transmission to send or receive packets using the RTS/CTS mechanism and use the CA Timer as specified in Section 3.2.
3. **LWT-MAC+CA+RTS**: sensor nodes wake up at the end of a transmission to send or receive packets using the RTS/CTS mechanism and use the CA Timer and the *RTS if DATA fails* as specified in Section 3.2.

4.2 Assumptions and Parameters

For the performance evaluation some assumptions have been made. It has been considered, for both B-MAC and LWT-MAC, that before accessing the channel to send a long preamble or an RTS, sensor nodes compute a random BO between 0 and CW. Assuming that the medium is slotted, the BO time is decremented by one unit each slot time that the medium is detected free. Once the BO expires, the node transmits the packet. If the medium is sensed busy during the BO countdown, the BO is frozen and it is resumed when the channel is detected free again. This medium access procedure is a Carrier Sense Multiple Access (CSMA) approach and it is the same as the one defined in the medium access control of IEEE 802.11 [1] networks.

Table 1. Hardware Parameters, most of them are obtained from the Mica2 mote with CC1000 transceiver at 868 MHz specification [8].

Parameter	Value
Data Rate	19.2 kbps
Tx Power	0 dBm
Sensitivity	−99 dBm
Carrier Sense Threshold	−111 dBm
Supply Voltage	3 V
Current Consumption in Tx Mode	24.5 mA
Current Consumption in Rx/Idle	17.6 mA
Current Consumption in Sleep Mode	15 μA

Table 2. MAC Parameters.

Parameter	Value
Queue Length	10 packets
Data Packet Size	29 bytes
Control Packet Size	5 bytes
Headers	11 bytes
Listen/Sleep Time	2.45/97.55 ms
Slot Time (σ)	417 μs
Retry Limit	5

In the basic scenarios (single hop, n_{h}-hidden and grid) the main goal is to analyze the problem controlling the different factors that can affect the results. For this reason, it has been considered that transmissions to the wireless channel suffer from path loss only and that the reception threshold is equal to the carrier sense threshold. Moreover, the communication of these scenarios is local, i.e., no multihop communication is allowed.

However, in the random scenario where a more realistic view is required, it has been considered that transmissions to the wireless channel suffer from path loss and shadowing. Apart from that, it has been assumed that packets with received power over the reception threshold (sensitivity) can be correctly received, while receptions with power below the reception threshold but higher than the carrier sense threshold can be detected but not decoded. Finally, the communication in this scenario flows from sensor nodes to the sink. This multihop communication is based on the shortest path, the metric considered is the number of hops to the sink. The routes to the sink are precomputed in order to avoid routing messages to interfere with the simulation.

For all the scenarios analyzed the wireless channel is assumed to be noise-free. Moreover, no capture effect is considered, i.e., if a collision occurs all packets involved are lost.

A preamble sampling simulator that maps the above considerations and assumptions has been used for the evaluation. The simulator framework is based on the SENSE simulator [7], but it has been extended in order to include the B-MAC preamble sampling channel access and the LWT-MAC protocol. Hardware and MAC parameters are shown in Tables 1 and 2.

4.3 The Value of the Contention Window

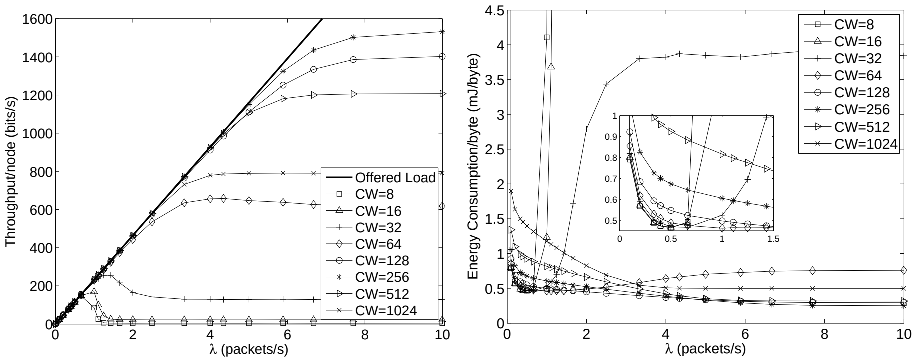
The value of the CW is of crucial importance in the LWT-MAC protocol when there are hidden terminals present since it directly influences the collision probability among RTS messages in the scheduled phase. A small CW value increases the RTS collision probability and therefore decreases the probability of successfully transmitting a packet in the scheduled phase, reducing the performance of the network. In contrast, a high CW value improves the probability of correctly receiving an RTS even in presence of hidden terminals, however, the energy consumption increases due to a higher listening time before transmitting.

The value of the CW is analyzed in the n_{h}-hidden scenarios with the goal to obtain a reference when only hidden terminal problems are affecting the overall network performance.The results of the LWT-MAC protocol with the CA Timer considered and the *RTS if DATA fails* mechanism deactivated are shown in Fig. 8. The throughput per node and energy consumption per byte successfully delivered for different CW values are depicted for each scenario. Note that the higher the CW value the higher the saturation throughput until a certain point at which it starts to decrease, this value is 512 for the 1 and 2-hidden scenarios (Figs. 8(a) and 8(c)) and 1024 for the 3-hidden one (Fig. 8(e)). Regarding the energy consumption per byte successfully received (Figs. 8(b), 8(d) and 8(e)), it can be observed that higher CW values imply higher energy consumption at low loads but a reduction at high loads. Using a high CW value results in higher energy consumption at low loads as nodes spent more time in BO before to transmit. However, at high loads, using a higher CW reduces the collision probability and also increases the probability of successfully transmitting in the scheduled phase.

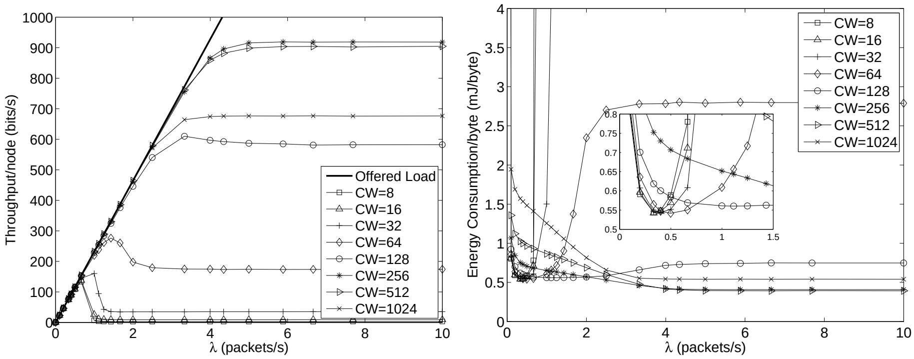
It can be observed that a CW value equal to 128 in these scenarios provides a good trade-off, specially at low loads, between performance and cost. This value will be the one used for the performance evaluation presented in the next section.

5 Performance Results

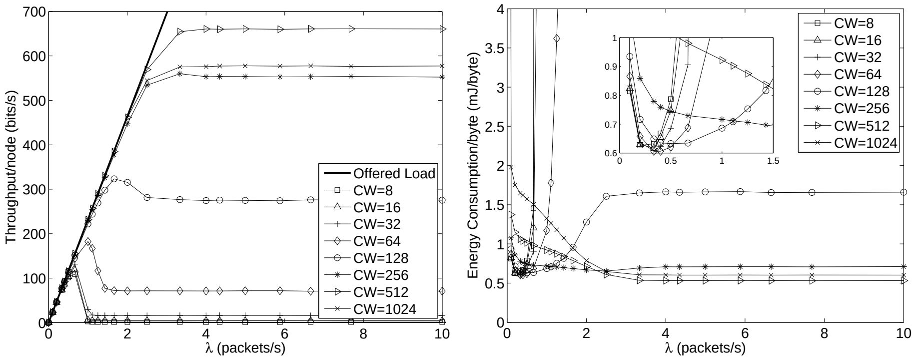
In this section the results comparing the performance of the B-MAC and the LWT-MAC protocols in the different scenarios considered are presented.



(a) 1-Hidden - Throughput (b) 1-Hidden - Consumption



(c) 2-Hidden - Throughput (d) 2-Hidden - Consumption



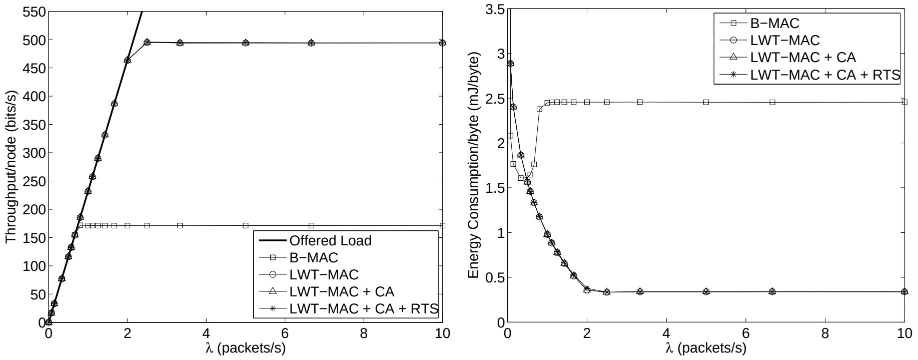
(e) 3-Hidden - Throughput (f) 3-Hidden - Consumption

Fig. 8. Performance results of the LWT-MAC protocol with variable CW in the n_h-hidden scenario.

5.1 Single Hop Scenario

Fig. 9 depicts the throughput and energy consumption per byte successfully sent in the single hop scenario. Observe that, the LWT-MAC protocol provides a throughput gain of more than 200% compared to the B-MAC protocol. It also allows to considerably reduce the energy consumption at high loads (approximately a reduction of 80%). At low loads a slightly higher value is obtained caused by the idle listening after transmission periods, that happens when all nodes listen after a successful transmission but no packet is received. This effect happens at low loads as the probability that any other node wants to transmit a packet is low.

Observe also that the results with the CA Timer and the *RTS if DATA fails* mechanism enabled are the same as the ones obtained with the basic LWT-MAC. When the traffic load is high the sensor nodes spent most of the time in the scheduled phase and they only return to the unscheduled case when there is a collision. In the scheduled phase the *RTS if DATA fails* mechanism is not used and collisions affecting the data packet are not possible (since all nodes are in coverage range and overhear the RTS/CTS exchange). Thus, the CA Timer is never activated in this phase in a single hop scenario. Therefore, the probability of using the CA Timer and the *RTS if DATA fails* mechanism (i.e., suffering a collision in the unscheduled phase) is small.



(a) Throughput

(b) Consumption

Fig. 9. Performance results in the single hop scenario.

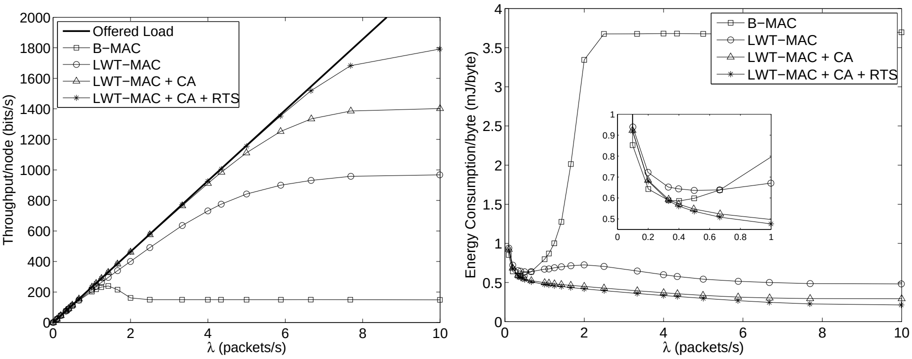
5.2 n_h-Hidden Scenarios

Fig. 10 shows the results of the n_h-hidden scenarios. It can be observed that the saturation throughput (Figs. 10(a), 10(c), 10(e)) obtained with the basic LWT-MAC protocol is noticeably higher compared to B-MAC in the 1-hidden scenario but it is similar in the 2 and 3-hidden cases since the wake up after transmissions capability alone is not able to schedule the different hidden terminals in that cases. In contrast, the LWT-MAC protocol with the CA Timer feature enabled provides a considerably higher saturation throughput compared to B-MAC in the three scenarios. Observe also that the highest throughput is obtained with the LWT-MAC with CA Timer and with the *RTS if DATA fails* option enabled.

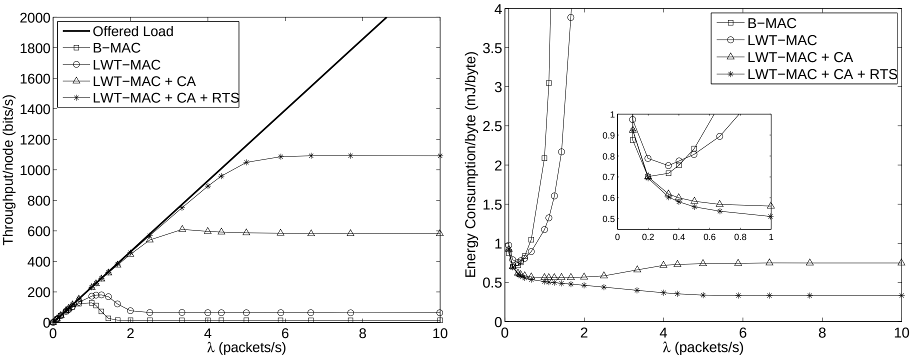
Regarding the energy consumption per byte that is correctly received at the recipient (Figs. 10(b), 10(d), 10(f)), it can be observed that it is reduced at high loads with the LWT-MAC with the CA Timer (with and without the *RTS if DATA fails* option). This is because nodes spent a considerably high part of the time in the scheduled phase. The *RTS if DATA fails* mechanism provides a higher reduction since instead of retransmitting using a long preamble a short RTS is used. Without the CA Timer feature enabled the reduction is notable for the 1-hidden scenario only, in the other scenarios the basic LWT-MAC protocol does not achieve a notable reduction in energy consumption compared to B-MAC. In contrast, at low loads, the energy consumption of LWT-MAC (in the three cases) is slightly higher compared to the one obtained by B-MAC. The reason of this increment is the aforementioned idle listening after transmission times.

5.3 Grid Scenario

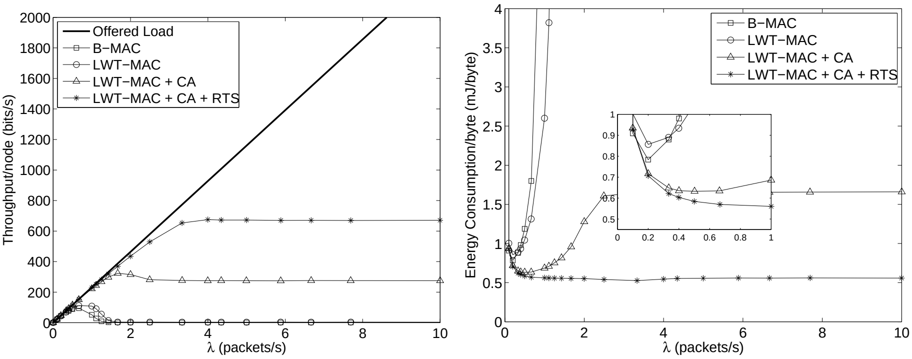
The results of the grid scenario are depicted in Fig. 11. Results are similar to the ones obtained in the 3-hidden scenario but the saturation throughput of LWT-MAC with CA Timer feature considered (both with and without the *RTS if DATA fails* option) is reduced as, in this case, transmitting nodes should share the channel with three other nodes. For B-MAC and LWT-MAC without CA Timer we obtain an increase in the saturation throughput. This is because in this scenario the effect of the hidden terminals is reduced as their channel access probabilities are decreased due to the transmissions of the neighbouring nodes. It is important to note that for small values of throughput the value of the energy consumption per number of bytes received fluctuates which is caused by the small number of packets received.



(a) 1-Hidden - Throughput (b) 1-Hidden - Consumption

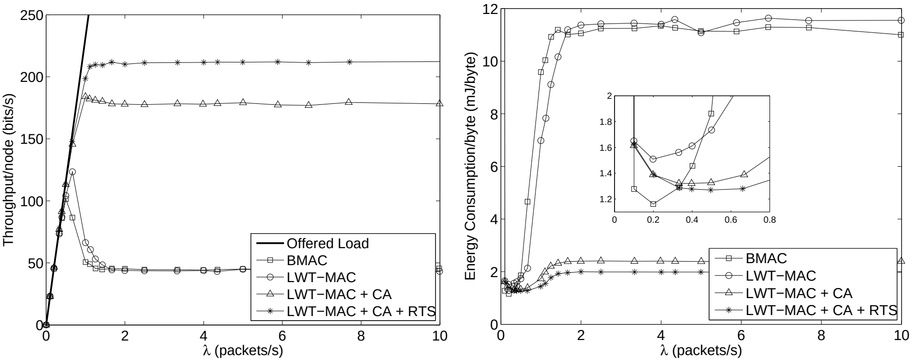


(c) 2-Hidden - Throughput (d) 2-Hidden - Consumption



(e) 3-Hidden - Throughput (f) 3-Hidden - Consumption

Fig. 10. Performance results in the n_{h}-hidden scenario. The insets show a zoom of the lower loaded region ($0 \leq \lambda \leq 1$ packets/s)



(a) Throughput

(b) Consumption

Fig. 11. Performance results in the grid scenario. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 0.8$ packets/s)

5.4 Random Scenario

In this scenario two different traffic profiles are simulated: periodic and event-based. Sensor nodes send messages periodically to the sink trying to model a typical monitoring application. At each sensor node the time between periodic message generation is constant and it has been set equal to 1 hour[1]. Moreover, if sensor nodes detect an event they notify the sink immediately by sending an event-based message of length equal to 29 bytes. The event radius is considered equal to 30 m, therefore nodes with a distance to the event center smaller than this value will detect the event and will generate a message. Although the average coverage range (that is equal to $r' = 98$ m given the transmission power, path loss propagation model and sensitivity) is higher than the event radius, hidden terminal problems can still happen in the path to the sink.

A network formed by 300 nodes randomly located in a 175×175 m^2 area in which sensor nodes continuously send periodic messages to the sink is simulated. After 90 minutes from the simulation start, when all nodes have at least sent a periodic packet, an event is generated in a randomly selected position. In that moment, sensor nodes inside the radius of the event (27.7 sensor nodes on average) try to send a message to the sink (placed in the middle of the topology). The percentage of periodic and event-based packet

---

[1] The first generation instant is randomly selected to avoid synchronization effects.

losses (mean and standard deviation) of 500 independent simulations, with duration equal to 120 minutes, are shown in Table 3 and 4 respectively.

Results show that the basic LWT-MAC protocol is not able to significantly improve the performance of the network compared to B-MAC, the percentages of both periodic and event-based lost packets are approximately the same. On the contrary, the LWT-MAC with the CA Timer feature enabled allows to considerably reduce the periodic loss percentage and notably improve (a 9.34% gain) the percentage of event-based packets correctly received at sink. Finally, as found in the previously analyzed scenarios, the *RTS if DATA fails* mechanism is able to slightly improve the results of the LWT-MAC with the CA Timer feature active.

Table 3. Mean and Standard Deviation of the Periodic Packet Loss Percentage in the Random Scenario

Protocol	Mean	Standard Deviation
B-MAC	0.34%	0.30%
LWT-MAC	0.37%	0.31%
LWT-MAC+CA	0.05%	0.09%
LWT-MAC+CA+RTS	0.02%	0.06%

Table 4. Mean and Standard Deviation of the Event-based Packet Loss Percentage in the Random Scenario

Protocol	Mean	Standard Deviation
B-MAC	16.94%	12.66%
LWT-MAC	14.67%	11.78%
LWT-MAC+CA	7.60%	8.90%
LWT-MAC+CA+RTS	7.04%	8.67%

5.5 Summary

From the performance results obtained in this section the following general conclusions can be drawn:

- The effects of hidden terminals are noticeable using the basic preamble sampling technique when the network load is high or due to instan-

taneous increases in the traffic load caused by the event detection at nearby sensor nodes.
- The use of the LWT-MAC without the CA Timer is able to improve the performance until a certain extent. As soon as the number of hidden terminals increases results are close to the basic preamble sampling protocol (B-MAC).
- The combination of the LWT-MAC and the CA Timer allows to considerably improve the performance of the network in all the scenarios evaluated.
- The combination of the LWT-MAC with the CA Timer and the *RTS if DATA fails* mechanism slightly improves the performance of the network if compared to the LWT-MAC using the CA Timer alone.

6 Conclusions

In this work the ability of the LWT-MAC protocol to alleviate hidden terminal problems has been evaluated in a set of key scenarios and in a more realistic case with periodic and event-based traffic profiles. Results have shown that the use of the LWT-MAC protocol combined with the CA Timer provides a higher network performance in presence of hidden terminals if compared to B-MAC. Moreover, the combination of the RTS/CTS after a data transmission failure in the unscheduled phase provides even better results.

The LWT-MAC protocol shows a considerable performance gain if compared to B-MAC in a simple and low consuming manner, however some considerations have to be taken into account. The first one is that retransmissions must be enabled, otherwise some packets would not be correctly delivered because of *desynchronization* among transmitter and receiver, i.e., the transmitter is trying to send a message in the scheduled phase but the receiver is sleeping because it has not overheard the last transmission. Apart from that, the value of the CW shall be set with the goal to increase the probability of correctly receiving the RTS messages in the scheduled phase in presence of hidden terminals. Finally, it has to be considered that the proposed approach consumes hardly more energy at low loads because nodes wake up at the end of every transmission but the probability that any other node wants to transmit is small. However, as soon as the load increases the energy consumption is considerably improved.

References

1. IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11*, 1999 Edition (Revised 2007).
2. A. Bachir, D. Barthel, M. Heusse, and A. Duda. Hidden Nodes Avoidance in Wireless Sensor Networks. In *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*, 2005.
3. A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications (ICC 06)*, pages 3365–3370, 2006.
4. M. Buettner, G.V. Yee, E. Anderson, and 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 (Sensys '06)*, pages 307–320, 2006.
5. C. Cano, B. Bellalta, J. Barceló, and A. Sfairopoulou. A Novel MAC Protocol for Event-based Wireless Sensor Networks: Improving the Collective QoS. In *Proceedings of the 7th International Conference on Wired / Wireless Internet Communications (WWIC)*, 2009.
6. C. Cano, B. Bellalta, A. Sfairopoulou, and J. Barceló. A Low Power Listening MAC with Scheduled Wake Up after Transmissions for WSNs. *IEEE Communication Letters*, 13(4):221–223, 2009.
7. G. Chen, J. Branch, M.J. Pflug, L. Zhu, and B. Szymanski. SENSE: A Sensor Network Simulator. *Advances in Pervasive Computing and Networking*, pages 249–69, 2004.
8. Chipcon. CC100. Single Chip Very Low Power RF Transceiver. http://focus.ti.com/lit/ds/symlink/cc1000.pdf.
9. A. El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad hoc Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications. (ICC 02)*, volume 5, pages 3418–3423, 2002.
10. A. El-Hoiydi and J.D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. *Algorithmic Aspects of Wireless Sensor Networks*, pages 18–31, 2004.
11. J. Hill and D. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, pages 12–24, 2002.
12. B. Jang, J.B. Lim, and M.L. Sichitiu. AS-MAC: An Asynchronous Scheduled MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. (MASS 2008)*, pages 434–441, 2008.
13. R. Jurdak, P. Baldi, and C. Videira Lopes. Adaptive Low Power Listening for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 6(8):988–1004, 2007.
14. P. Karn. MACA-A New Channel Access Method for Packet Radio. In *Proceedings of the ARRL/CRRL Amateur Radio Computer Networking Conference*, 1990.
15. K. Kredo II and P. Mohapatra. Medium Access Control in Wireless Sensor Networks. *Computer Networks*, 51(4):961–994, 2007.
16. S. Mahlknecht and M. Bock. CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks. In *Proceedings of the IEEE International Workshop on Factory Communication Systems*, pages 73–80, 2004.
17. J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.

18. I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, 2008.
19. Y. Sun, O. Gurewitz, and 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 (Sensys '08)*, pages 1–14, 2008.
20. F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-access and the Busy-tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.
21. A. Woo and D.E. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proceedings of the International Conference on Mobile Computing and Networking*, page 235, 2001.
22. W. Ye, F. Silva, and J. Heidemann. Ultra-low Duty Cycle MAC with Scheduled Channel Polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, page 334, 2006.