Ph.D. Thesis

Daniele Messina

Design and Implementation of a WSN Based Reliable Data Gathering System for Large Scale Applications

The beginning of know ledge is the discovery of something we do not understand.

Frank Herbert (1920 - 1986)

Table of Contents

Chapter 1

Introduction

1.1 Motivations and goals

In large-scale data-gathering applications a fundamental importance is assigned to those queries with which the user requests aggregated data, that refer to large geographical regions and long time intervals, rather than queries that request single measures from specific locations and times. Because of this typical employment, wireless sensor networks-based systems used for these tasks are open to various implementation approaches and optimizations including spanning trees, which often employ aggregation techniques. The primary limit of the spanning tree approach is its vulnerability against communication errors and malfunctioning or energy depletion of nodes. In fact, in case of errors, depending on the point of the tree where they occur, it is possible to lose large amounts of data, typically all the measurements coming from the nodes of a whole sub-tree. A possible strategy consists of the use of robust routing techniques, which rely on multiple paths, however, an important issue arises, as multiple paths generate duplicates of data, which, for some types of queries, such as counts, sums or averages, lead to altered results. Researches over the recent years have formalized a data aggregation framework, known as *synopses diffusion*, which can help to solve the duplicates problem. This technique allows to build digests of measurements, namely synopses, obtaining the exact same result independently from the order in which data are included in the digest and from the number of times each single datum is considered. The independence of the result from all the possible sequences that may occur in a multi-path scenario enables the separation of the two problems, data aggregation and robust routing, thus making the use of multi-path techniques compatible with data aggregation. Moreover, synopses provide implicit acknowledgment of receipt, which may be used instead of traditional acknowledgment packets.

The objective of this work has been to obtain robust and seamless operation in

a large scale data-gathering system through the use of multi-path routing, relying on implicit acknowledgments for detecting communication errors, and on caching with controlled retransmissions for actuating the necessary recovery procedures. Ready recovery mechanisms allow the applications to continue working, collecting and routing data, when node or communication failures occur, during the time new routing paths are determined. At the same time, routing protocols reactivity to errors can be reduced, in order to not react to repeated communication errors, likely to occur because of critical channel conditions, which may be mistaken for node failures and trigger unnecessary path recalculations.

The implementation of the designed communication scheme has been designed to work on top of a standard IEEE 802.15.4 MAC Layer, which has been chosen to meet the requirement of having a standard interface capable of using commercial off-the-shelf technologies. IEEE 802.15.4 has been selected for its energy-efficient design (the protocol can exploit the periods of inactivity which are present in the communication scheme and put nodes in a low energy consumption state), as well as for its configurability, which the communication agent exploits in order to obtain the necessary behavior of the nodes during the several phases of its operation.

The considered scenario, where hundreds of sensor nodes may be deployed, makes the collision avoidance problem a primary concern, especially because of the well known hidden node problems, which have catastrophic effects on the through- put of very large networks. At the origin of the problem is the fact that the local perception of the wireless channel as idle, available at the transmitting node, does not guarantee that a destination node can correctly receive a communication and it is not interfered by other sources unknown to the sender. As detailed in the body of this document, the use of acknowledgments and retransmissions must be limited, as they consume additional energy and may worsen channel congestion. Moreover, the considered data gathering applications exhibit a periodic communication, hence the requirements for the collision avoidance mechanism were to be able to exploit the feedback provided by previous transmissions and limit the overhead and delay typically introduced by MAC Layer's backoff mechanisms, aiming to obtain an hybrid scheduled/non-deterministic access scheme.

Lastly, while simulations offer a valuable support during the design process and validation of protocols, several common assumptions made in most simulation environments limit the reliability of simulation results in assessing the actual performance of the tested protocols. Such assumptions include:

- A radio's transmission area is circular.
- All radios have equal range.
- The communication channel is symmetric (i.e. If I can hear you, you can hear me).

- The communication channel has a cut-off connectivity characteristic (i.e. If I can hear you at all, I can hear you perfectly).

The connectivity model resulting from these assumptions does not reflect the real- world conditions, where factors like terrain, tree density, 3-D antenna location, foliage types, wavelength, external source of radio-frequency interference, etc. have a consistent impact on radio communications. Accordingly, in order to carry out a reliable evaluation of protocols' performance, and to develop solutions which work in real life applications, the design process should be complemented by the implementation and testing with real hardware devices.

1.2 Contributions

The main contributions of this work are:

- A communication model resulting from the integration of data aggregation techniques and multi-hop routing strategies.
- An application-controlled collision avoidance scheme which is adaptive, distributed and does not require control communication overhead.
- A standard interface based on the use of the primitives of the IEEE 802.15.4 specifications, allowing for a straightforward implementation of the data- gathering system over widely available off-the-shelf hardware devices, such as Crossbow's MicaZ and TelosB.
- A versatile simulation tool, based on Network Simulator 2.
- An implementation of the data-gathering system for Crossbow's MicaZ and TelosB motes, using TinyOS and NesC.

1.3 Dissertation outline

The remainder of the dissertation is organized as follows:

- Chapter 2 introduces the characteristics and issues typical of communication in the wireless networks, focusing on aspects related to Medium Access Control (MAC). Important concepts and protocols adopted for traditional wire- less networks, as opposed to wireless sensor networks are presented. Lastly, the characteristics of wireless sensor networks are discussed, highlighting the similarities and differences with traditional wireless network, and presenting the new challenges of this type of networks.

- Chapter 3 presents a classification of MAC protocols for wireless sensor net- works, identifying two main classes, scheduled and unscheduled protocols. Characteristics, advantages and issues of the two approaches are discussed, as well as for several representative protocols for each class.

- Chapter 4 presents the characteristics and performance of IEEE 802.15.4, which has been adopted as the MAC-PHY protocol stack for the work presented in this thesis.

- Chapter 5 introduces the cooperative reliable communication system, pro- viding a description of the network protocol, which addresses power management, synchronization and link reliability, and provides enhanced robustness in IEEE 802.15.4-based sensor networks. The chapter also contains the results of a performance study, carried out by means of simulations.

- Chapter 6 presents a collision avoidance technique, which can be implemented on standard IEEE 802.15.4 networks, and extends the functionalities provided by the IEEE 802.15.4 MAC protocol. The description of the collision avoidance techniques complete the presentation of the communication system.

- Chapter 7 presents the software implementation of the system, developed with the use of Network Simulator 2, and employed to carry out performance evaluation.

- Finally, Chapter 8 describes the implementation of the communication sys- tem and the collision avoidance techniques with actual devices, which has been developed with the use of TinyOS.

Chapter 2

Communication in Wireless Sensor Networks

2.1 Medium Access Control Issues in Wireless Sensor Networks

The Medium Access Control (MAC) layer sits directly on top of the Physical layer. It manages different mechanisms involved in the communication process, including framing, error control, reliable data exchange, medium access with collision avoidance, and flow control.

- Framing defines the frame format and performs data encapsulation and de-capsulation for communication between devices.
- Error control uses error detection or error correction codes to control the amount of errors present in frames delivered to upper layers.
- Reliability ensures successful transmission between devices. Most commonly accomplished through the use of Automatic Retransmission Request algorithms algorithm (ARQ) which employ acknowledgment (ACK) messages and retransmissions when necessary.
- Medium access controls which devices participate in communication at any time. Medium access is a main function of wireless MAC protocols because of the broadcast nature of the communication channel which easily causes data corruption through collisions.
- Flow control prevents frame loss due to overloaded recipient buffers.

In WSN the radio utilization typically draws more energy than other activities, such as computing, hence MAC protocols, which have the most direct control over the usage of the radio, most often focus on energy efficiency instead of meeting traditional goals for wireless MAC design such as fairness, delay, and bandwidth utilization.

Main sources of energy waste at the MAC Layer are collisions, idle listening, overhearing, and control packet overhead.

- Collisions waste energy in that if ARQ techniques are being used, they trigger retransmissions.
- Idle-listening refers to listening to an idle channel, waiting for a potential packet to arrive.
- Overhearing refers to the reception of packets destined to other nodes.
- Finally packet overhead refers to headers, control messages and every other protocol data exchanged by the protocols, besides protocol payloads, in order to achieve their correct operation.

A common strategy used to save energy is to turn-off the radio when it is not needed for communication. This approach opens a series of problems to be addressed, primarily because the MAC Layer has to ensure that the devices willing to communicate are simultaneously active when the communication occurs. For this purpose, several coordination and scheduling techniques exist.

The MAC Layer protocol is often also required to present an abstract view of the connectivity and topology of the network to the upper layers.

Protocols that function based on some form of time synchronization must take into consideration that clock drifts become significant over a sensor network's lifetime. Synchronization is in fact a problem within sensor networks since the requirement for low cost devices often necessitates the use of lower precision hardware.

Scalability is also a problem for protocol designers. Sensor networks may operate with many hundreds to thousands of devices, hence centralized protocols have a major disadvantage due to the overhead associated with information distribution. Distributed algorithms, even sub-optimal ones, are typically the best choice for the characteristics, in terms of functionality and platform, of sensor networks.

Mobility and fluctuating quality of the wireless links determine a requirement for protocols to be adaptive, and to react to changes in topology, connectivity and availability of nodes without disruption of network operation.

Finally sensor network application requirements and characteristics exhibit large variability, however any single application has often very specific features,

for instance in terms of traffic patterns, nodes deployment as well as reliability requirements. This circumstance may be exploited by the MAC protocol designer in order to simplify and optimize the protocol operation with respect to some performance metric defined for the particular application, although this may limit the protocol's generality.

As a consequence there is no clear trend indicating that medium access for sensor-nets is converging towards a unique best solution.

2.2 MAC Protocols for Wireless Networks

Wireless networks have been thoroughly investigated in the past, their typical issues are well known and many MAC protocols have been proposed over the years. Unfortunately, the amount of work that has been done on this subject can only partially be exploited and exported to the specific research area of wireless sensor networks. The main reason behind this limit is the different set of constraints that affect WSN, for which the primarily concern is typically the lifetime of the network and therefore the energy consumption.

2.2.1 CSMA

Carrier Sense Multiple Access (CSMA) is a very popular form of medium access control. CSMA has several variants and elements of CSMA techniques are present in many MAC protocols. In non-persistent CSMA, a devices senses the channel before a transmission to determine if another device has already started transmit- ting. When the channel is sensed busy, the device initiates a backoff procedure, waiting for a certain amount of time and then attempting to transmit again. If no activity is detected, then the transmission starts immediately. A device that only needs to transmit packets may keep the radio not active during the backoff time, while devices which wait for incoming packets have to maintain the radio in receive mode. With p -persistent CSMA, a device that senses activity on the channel starts its backoff by continuing to sense the channel, until the activity terminates, when it decides for an immediate transmission with probability p or delays the transmission with probability $1\text{-}p$. This variant of CSMA requires the device to keep the radio in receive mode during the backoff time. The timers involved in the operation of the protocol may use continuous values for unslotted CSMA or discrete time values for slotted CSMA.

Forms of p -persistent CSMA with $p < 1$ implement what is called CSMA with Collision Avoidance (CSMA/CA), by reducing the probability that two or more devices start a transmission simultaneously, as soon as the activity on the channel ends. Collision Detection, which is used in wired networks and allows the sender

of a message to detect collisions, is not used in wireless networks for two reasons: first, detecting collisions would require the use of two half-duplex transceivers, or an equivalent full-duplex transceiver, in order to both transmit and receive at the same time, which poses cost and implementation problems; second, packets are corrupted when a collision occurs at the receiver, while a sender may not receive any colliding packet and vice versa.

Some versions of CSMA use an exchange of control packets in order to ensure that there is not any activity on the channel at both the ends of the communication. Namely, when the CSMA algorithm has determined a transmission time, the sending device transmits a small Request To Send (RTS) packet instead of the data packet. If the receiving node is able to receive the RTS and senses the channel idle, it responds with a Clear To Send (CTS) packet. Upon a successful reception of the CTS, the data packet is finally transmitted, followed by an acknowledgment packet if requested.

The RTS and CTS packets may also carry information about the duration of the whole packet exchange, thus informing all the neighboring devices reached by at least one of the two packets that they should not start any transmission during that time. This technique is known as Virtual Carrier Sense.

The use of RTS and CTS reduces the impact of collisions when a dominant part of the network traffic consists of large packets, whereas its benefits in the area of WSN, where the overhead introduced by the additional control packets may become not negligible, have to be carefully evaluated, taking into account traffic conditions, wireless channel characteristics, and network topology.

2.2.2 MACA

MACA [18] is a protocol that aims to solve some inefficiencies of CSMA. It uses RTS and CTS, and Virtual Carrier Sense. With MACA, devices that receive an RTS message destined for another device, but do not receive the expected CTS message, are allowed to begin a data exchange. This rule intends to prevent the so called *exposed terminal* problem, which occurs when a device that is physically near to the transmitting device would be able to initiate a transmission of its own without causing interference at the receiver, but it is blocked by the rules of CSMA.

When the receiving device is intended to send an acknowledgment back to the transmitting device, as in the case of MACAW [6], a device that has received an RTS but not the corresponding CTS cannot be allowed to initiate a transmission, as it could interfere with the reception of the acknowledgement by the neighbor device. MACAW uses a third control message, namely a Data Sending message (DS), transmitted by the sending device after the reception of a CTS, to inform the neighbor nodes of an incoming transmission.

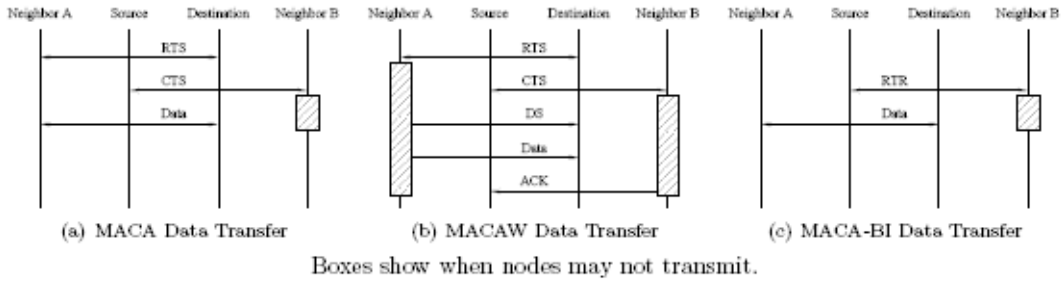

Figure 2.1: Data-transfer in MACA, MACAW and MACABI.

Neighbors may still start a transmission if they have received an RTS but not the corresponding DS. However, the improvements of throughput and communication reliability introduced by MACAW may not be as valuable as the extended network lifetime achievable by not transmitting or receiving the DS message, or even any of the control messages.

MACA-BI uses a different approach to medium access control by having the destination devices initiate the data transfer process. A device that expect an incoming transmission sends a Ready To Receive (RTR) message to inform the source of information that it may begin the data packet transmission. The performance of the protocol depends on the destination's ability to predict the data it will receive. The destination exploits an optional field within the data message that indicates the number of messages queued at the source. This version of MACA may find application in networks where devices continually generate data. The reduced overhead makes MACA-BI more applicable to sensor networks, however it still requires that devices constantly sense the channel and it cannot be adopted in its original form. MACA-BI is interesting because it shows an example of protocol optimization based on the particular characteristics of a set of application scenarios.

Figure 2.1 shows data transfers for the MACA, MACAW, and MACA-BI protocols. For each protocol, boxes indicate when neighboring devices may not transmit because they defer access in the presence of a previous communication.

CSMA, MACA, and their variants meet the requirement of simplicity for sensor networks, but require the transceiver to operate continuously, which would make sensor nodes consume their energy far too quickly to make the deployment useful.

2.2.3 IEEE 802.11

The most used access protocol in wireless network is part of the IEEE 802.11 standard, which contains specifications for Wireless LAN Medium Access Control

(MAC) and Physical Layer (PHY) [2]. IEEE 802.11 provides two modes of operation for wireless devices: an infrastructure mode and an ad-hoc mode. In infrastructure mode devices communicate through a central entity called an access point (AP) using the point coordination function (PCF); in ad-hoc mode devices communicate with each other directly using the distributed coordination function (DCF). Both the PCF and DCF use channel access mechanisms which are based on slotted CSMA/CA, acknowledgments are used for reliability, and channel utilization is tracked through physical carrier sensing as well as virtual carrier sensing, using the information included in the protocol header of packets. When devices read the communication length information they update a counter, named Network Allocation Vector (NAV) which decrements periodically. For the purpose of determining channel activity, IEEE 802.11 devices consider the channel busy as long as the NAV contain a non-zero value or they physically detect some activity. RTS and CTS may be also used.

DCF is a form of p-persistent CSMA with non-stationary probability p. For a device to consider the channel idle, it must not detect activity for a time period called DCF interframe space (DIFS). When first trying to transmit a message, a device senses the channel and, if the channel is free for a DIFS, it transmits the message. If the channel is determined busy, a device defers the access and perform the backoff algorithm by randomly selecting a number of time slots to wait and storing this value in a backoff counter. The backoff counter is decremented of one unit for each time slot where the device senses no activity on the channel. When the backoff timer reaches zero the device starts its transmission. If a device detects activity on the channel during its backoff, it halts the countdown, and waits until the channel is clear for a DIFS; afterwards, it resumes the countdown. When acknowledgments are used, a receiving device transmits the acknowledgment after short period of time following the end of the reception of the incoming packet. The adopted period of time is called Short Interframe Space (SIFS) and is shorter than the DIFS, in order to prevent any device from sensing an idle channel and potentially transmit and collide with the acknowledgment.

DFC is represented in Figure 2.2.

The PCF operation is based on the same key logic. The AP coordinates collision-free time periods by broadcasting beacon messages, including the duration of the next collision-free period and a list of devices to receive data. During the contention-free period the AP transmits messages to the devices listed in the beacon or it transmits polling messages to devices, which allows the devices to initiate data transfer with the AP. The AP uses an interframe space, namely PCF interframe space (PIFS), which is shorter than a DIFS and longer than a SIFS. A PIFS is used as the timeout after a polling message receiving no response, and between consecutive messages by the AP. In this way, the AP maintains the control of the channel, by preventing that a device operating according to the DFC rules

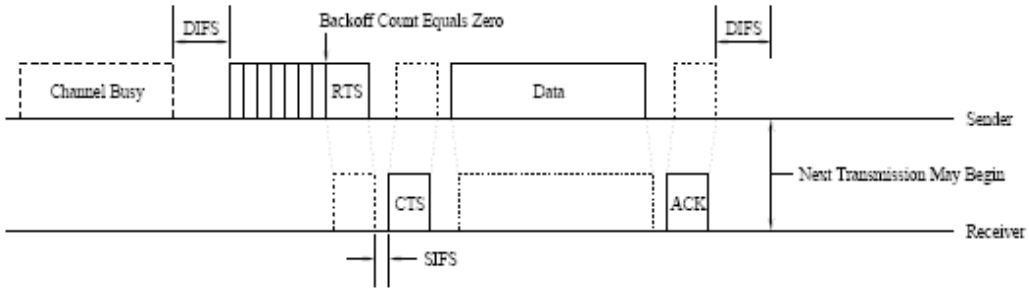

Figure 2.2: IEEE 802.11 DCF Backoff Algorithm and Message Transfer showing the RTS-CTS packet exchange and the use of acknowledgments.

gains access to the channel, but still allowing devices to send replies, such as CTSs and ACKs, for which a SIFS is used.

The main reason why IEEE 802.11 does not fit the requirements of WSN and their applications is that IEEE 802.11 devices consume large amounts of energy due to the high percentage of time spent listening without receiving messages. IEEE 802.11 provide an energy management capability, called a power save mode, to devices operating according to the PCF. Devices may enter sleep mode when they do not have messages to receive or transmit, and wake up to receive beacon messages from the AP to determine if they must receive messages during the following contention-free period. However, power save mode only operates in infrastructure mode, which creates a scalability problem, as all the devices must be in the radio range of the AP, while most deployments of WSNs have large network diameters and use multi-hop communication. Additionally, the protocol overhead in IEEE 802.11, which local networks can tolerate, becomes very large when used in sensor networks where applications may only generate a few bytes of data per message.

2.3 The Challenges of Wireless Sensor Networks

The protocols discussed in the previous section, designed to provide high throughput, low latency, fairness, and mobility management, are not satisfactory for the lack of strong energy management. Messages within sensor network applications often have a much smaller size when compared to traditional wireless networks, hence the protocol overhead may become a concern. Multi-hop operation is often not fully supported and infrastructure modes of operation consume too much energy for sensor networks deployed over large geographical areas, because the transmission power required to correctly receive a message increases with distance

d, typically with a power between d^2 and d^4, making a sequence of short-range transmissions more efficient. Another mean to achieve energy conservation consists of cycling the sensor node hardware between high power, or active, states and low power, or sleep, states. The fraction of time the sensor node spends awake is referred as duty cycle and it may be as low as one percent in many sensor network applications, dramatically increasing a sensor node's lifetime. Much of the research done for ad-hoc networks may also apply to sensor networks since both operate as multi-hop wireless networks with power constraints. Protocols for ad-hoc networks, however, focus on device mobility, while sensor networks normally have limited or no mobility. Several problems in wireless networks, such as the hidden terminal problem, exist in sensor networks as well, which add to the characteristics unique to sensor networks.

2.3.1 Collisions, Overhearing, Idle Listening

Collisions within sensor networks cause performance limitation and energy waste. Sensor network applications with low data rate requirements and high delay tolerances can accept a slight performance decrease, however the energy waste due to frequent collisions can significantly decrease a sensor node's lifetime. When a collision occurs and a device retransmits a message it uses its transceiver consuming multiple times the minimum energy required for that message. Sensor networks that do not require a reliable link layer may opt for not retransmitting messages. Since the wireless channel is inherently a broadcast medium and sensor nodes typically use omnidirectional antennas, several sensor nodes may receive the same transmission, possibly multiple times with retransmissions, even there is only one intended recipient. Both reception and processing consume energy and unintended receivers waste this energy.

It is also possible to end a reception early and enter the sleep state to limit the energy losses associated with overhearing messages. In order to implement this technique a device should be able to determine that the message belongs to another node, by reading and processing the destination address as the message is being received. Of course, the message format must include the destination address early in the packet header and the receiving device may require slightly complex hardware.

Some protocols turn what is commonly considered an issue into a useful resource, by exploiting overhearing to infer information such as sensor node availability, link status, and acknowledgment of receipt.

In the absence of synchronization techniques, nodes may spend the most time listening to the channel, waiting for incoming transmissions. This activity, referred to as idle listening, waste the energy consumed by the transceiver without any benefit for the node, often accounting for a significant portion of the energy a

sensor node consumes. Carrier sensing instead, should not be assimilated to idle listening, because it performs a necessary work for the MAC protocol.

2.3.2 Hardware Characteristics and Energy Consumption

Most sensor network transceivers consume the same energy in receive mode whether they are actually receiving a message or they are only receiving noise, i.e. they are on idle listening. Some transceivers possess an intermediate state between the full active state and the sleep state, which allows them to listen to the channel with very low power and save a great deal of energy normally expended on idle listening. Protocols that exploit this feature can have a large impact on power savings over the lifetime of the sensor network. Another benefit of intermediate states is the reduced delay for switching the transceiver to an active state, achieved by keeping critical circuits operational. Smaller delays allow more flexibility for the protocol designer and reduce the risk of violating protocol timing.

Synchronization, and timing in general, is also problematic because of the use of low accuracy oscillators to reduce sensor node cost.

Modulation schemes in a transceiver affect the bit error rate (BER) for a given transmission power, with more complex modulation schemes generally achieving better performance. However, complex transceiver can cost more and consume more energy. For these reasons, preferred modulation schemes for sensor networks are simple, such as On-Off Keying (OOK) and Binary Phase Shift Keying (BPSK), more often than complex ones, such as Direct Sequence Spread Spectrum (DSSS) and Ultra-Wide Band (UWB).

2.3.3 Computation and Storage Resources

Wireless Sensor Networks are characterized by limited computation and storage resources. Complex protocols may provide good energy savings and useful functionalities, such as clustering and topology estimation, which are likely to be implemented more efficiently than in the upper layers. MAC protocol proposals do not generally discuss the processing requirements for protocol operation, but a complex MAC protocol might consume a large fraction of the available processor time, both decreasing the time a sensor node can spend in the sleep state, and limiting the availability of the processor for the application and other protocols. Sensor nodes may store information that allows the protocol to conserve energy by adjusting the transmission power or decreasing collisions, but also leaving fewer memory resources available for data collected by the application, and program space. MAC protocol designers should then ensure that the functionality they provide fits the application requirements while avoiding unnecessary use of processor and memory.

2.3.4 Distributed and Centralized Algorithms

Distributed algorithms are typically preferred over centralized network organizations for several reasons. Centralized algorithms will likely need to collect information from the whole network, process this information, and communicate results and/or commands back to the network. The low data rate and multiple hops necessary to share information among the nodes limit the feasibility of the centralized approach: for instance, due to the high response time, the network conditions or events that triggered a network response may have changed by the time the central entity can determine a proper action. Clearly, sharing information also consumes large amounts of energy as the sensor nodes transmit and forward the control messages. MAC protocols for WSN are required to provide scalability, primarily in terms of network size, but also in sensor node density, and to support even hundreds or thousands of sensor nodes. This sets a strong orientation towards distributed algorithms. Centralized approaches may also be not practicable due to the limited processing and memory resources available at the nodes, and in some cases they may require to employ special purpose, more powerful devices, in coexistence with ordinary ones.

Chapter 3

MAC Protocols For Wireless Sensor Networks

This chapter discusses some representative MAC protocols for WSN, following a general classification that identifies two main classes: scheduled protocols and unscheduled, or random, protocols.

Scheduled MAC protocols organize sensor nodes so that communications occur according to a defined timing or schedule. Most scheduled protocols use time division multiple access (TDMA), defining a frame structure where every sensor node utilizes a time slot for transmitting and/or receiving messages. Scheduled transmissions may reduce collisions and message retransmissions at the cost of synchronization and state distribution.

Unscheduled protocols allow sensor nodes to operate independently, with a minimum of complexity, but do not eliminate collisions and idle listening.

Schedules may also control when nodes enter sleep states to conserve energy. Scheduled solutions generally allow nodes to maintain lower duty cycles, without affecting traffic capacity and latency.

3.1 Unscheduled MAC Protocols

Unscheduled MAC protocols meet the demand for simplicity, which is common to many WSN applications. Devices do not maintain and share state information, thus consuming fewer processing resources, requiring less memory space and saving the amount of energy they would spend transmitting control messages. The absence of an organized schedule of transmissions simplify and speed up the operation of joining the network by newly added nodes. For the same reason, events such as redeployments or movements do not require burdening and delaying procedures like obtaining the current schedule or running new resource allocations.

Unscheduled protocols also allow sensor nodes to adapt more easily to changing traffic conditions.

Since transmissions are not coordinated, unscheduled MAC protocols experience, in general, a higher rate of collision, idle listening, and overhearing because. These issues may be mitigated by introducing additional features including forms of synchronization and channel reservation, however the more rigid is the organization the more the protocols lose their characteristic benefits.

Fairness may be an issue in unscheduled MAC protocols as they generally lack specific mechanisms that equalizes the channel usage, which can be easily introduced in a scheduled MAC protocol.

3.1.1 Multiple Transceiver MAC Protocols

Using multiple transceivers generally increases the hardware complexity of the sensor nodes, which makes this approach seem counterproductive. A system with multiple transceivers (two in most proposals) must have the computational capacity to operate them simultaneously, and to process and communicate data on separate channels. Possible benefits of such solutions include increased bandwidth and shorter response times.

PAMAS

The Power Aware Multi-Access with Signaling (PAMAS) [39] protocol uses two transceivers: one for data messages and the other for control messages. The protocol aims to save energy by avoiding collisions of large data messages through the use of control messages on the signaling channel.

Figure 3.1 shows an exchange of packets involving two nodes. The sender nodes starts by transmitting an RTS message to the destination on the control channel. The destination responds with a CTS when it does not detect any activity on the data channel and it has not received control messages that inform it of ongoing communication which it might interfere with. If the sender does not receive a CTS in time, it starts a binary exponential backoff procedure, by choosing a number of timeslot to wait before a new attempt in a backoff window, whose size is doubled after every unsuccessful attempt. When the sender receives the CTS, it transmits the data message over the data channel. The destination starts transmitting a busy tone over the control channel once it starts receiving the data message. In this way it informs nearby nodes that they may not use the data channel. The busy tone is a message with twice the length of an RTS or CTS message. If the destination receives an RTS message or detects activity on the control channel during the data reception, it transmits the busy tone, in order to corrupt possible CTS message replies and prevent transmissions on the data channel.

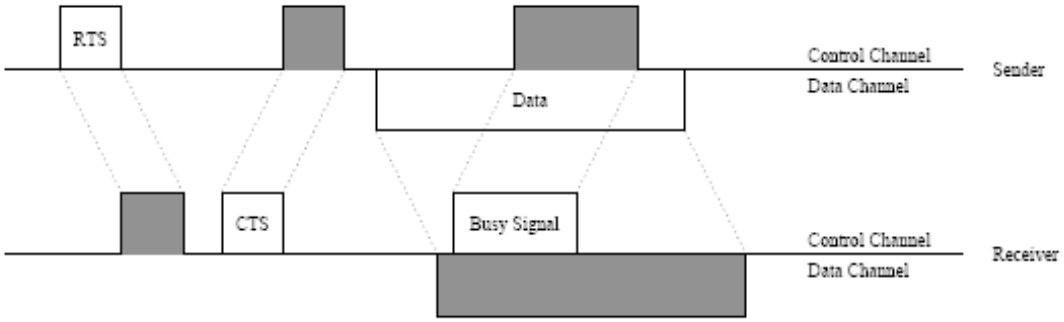

Figure 3.1: Data Transfer on PAMAS.

When a node does not have messages to transmit and it receives a transmission on the data channel, destined to another node, it can power down the transceiver, as it cannot receive other messages without corruption. In order to determine the length of time to sleep, data messages include the transmission duration in their header.

Conversely, when a node A has messages to transmit and a neighbor C is transmitting data, A may still be able to communicate with a destination B without collisions, therefore it is allowed to transmit an RTS on the control channel. If the corresponding CTS is received, that means the receiver D of the ongoing communication is not reached by node A's transmissions, and at the same time the receiver B of the new communication does not receive node C's transmission on the data channel. If the RTS is received by node D, it will respond with a busy tone informing node A that it cannot transmit without corrupting the ongoing communication, and then, without any possibility to either receive or transmit, it may power down the transceiver.

A node may awake during an ongoing message transmission, whose length it does not know. If the node does not have messages to transmit it sends a probe message on the control channel, to which the transmitting node will respond indicating the remaining transmission duration. When multiple nodes are transmitting, their response may collide and a collision resolution algorithm is applied. Once the transmission duration is known, the node may return to the sleeping state. If, on the contrary, the node wants to transmit a message, it may transmit an RTS as in previous scenarios. Should it receive a busy tone, this would indicate the remaining transmission time, allowing the node to sleep for the correct amount of time.

Improvements of PAMAS have been proposed, that introduce ACK messages, or eliminate the probe mechanism and keep the transceiver always active on control channel.

3.1.2 Multi-path MAC Protocols

Control messages and carrier sensing are part of the protocol overhead that may be the subject of simplifications. Some medium access technique exist which do not use them and only rely on backoff mechanisms in order to avoid collisions. Messages are transmitted after generally random delays, and to increase the probability of message delivery, many copies of each message may propagate through the network. Network characteristics and application requirements must be considered in order to determine a proper balance between protocol simplification and the cost of the extra traffic.

PFP

Probabilistic Forwarding Protocol [11] (PFP) is a simple routing protocol that may be used in conjunction with some multi-path MAC protocols. PFR is based on the assumption that nodes possess directional antennas, which make them able to detect the angle of arrival (AoA) of incoming messages. Each node is also supposed to generate traffic destined to the base station and to know the base station's direction. When a sensor node receives a message, it decides to broadcast the message with a certain probability, based on the angle formed between the message source, itself (the forwarding sensor node), and the base station. The closer the angle to 180 degrees, the higher the probability that the message is forwarded.

SRBP, ARBP, and RARBP

Simple Random Backoff Protocol [12] (SRBP) is the simplest of the protocols briefly presented in this section. SRBP does not use channel sensing nor control messages. Nodes simply transmit their messages after an initial random backoff . The backoff delay, t_b, is randomly selected from a backoff window, $[T_b\,min,\ T_b\,max]$, which remains the same during the network's lifetime. There is not any feedback mechanism in this protocol.

Adaptive Random Backoff Protocol [12] (ARBP) improves SRBP by adding a form of feedback that takes nodes density and traffic volume into consideration in order to adjust the backoffwindow. ARBP runs two sub-protocols that estimate the sensor node density, d_l, and the traffic density, T_l.
The density estimator sub-protocol determines the density of nodes based on the number of different node IDs read within received messages in the recent past. A node ID is removed from the list when a message with that ID is not received over
a defined time period. The traffic estimator sub-protocol updates a simple counter

for each received message. ARBP updates the maximum backoff value, $T_b max$, according to the function $T_b max = T_b^- + \alpha C_d + \beta C_t$.

$\alpha, \beta \in \{0,1\}$ are system parameters which may be set by the end user, whereas $C_d = max^- \frac{d_l - d_l^-}{d_l + d_l^-}$ and $C_t = max^- \frac{T_l - T_l^-}{T_l + T_l^-}$ depend on the relative variation from the previous estimation d_l^- and T_l^-.

Range Adaptive Random Backoff Protocol [12] (RARBP) uses an estimation of the distance from the source of a received message, to condition the backoff delay to be used in the forwarding process. Namely, if $D_e st$ is the estimated distance and $R_M AX$ is the maximum range, the random backoff value is selected from a normal distribution with mean $T_b min + (T_b max - T_b min)\frac{D_e st}{R_M AX}$ and standard deviation $\frac{1}{d}$, where d_l is a node density estimation. The key idea behind RARBP is that allowing farther sensor nodes to transmit earlier, the message latency can be shortened and each message may traverse fewer hops. A drawback of this protocol resides in its hardware requirements, as a node must be able to infer distances from the received signal or must be externally provided with location information.

The protocols considered in this section do not communicate information about transmission successes, thus spending energy transmitting the same message along multiple paths, and of course they do not provide guaranteed delivery. In conditions of high traffic, the collision rate may increase to the point of disrupting the protocol performance. In these cases increased backoff windows may be adopted, however it negatively affect message latency. x -RBP protocols appear then suit- able for some applications that generate light traffic, do not require reliable service, and employ nodes with limited computing resources.

3.1.3 Event-Centered Protocols and CC-MAC

In the wide scenario of sensor network applications, target detection has some peculiar characteristics. Such networks have very little traffic most of the time, but may produce relatively large volumes of data when an event of interest occurs. MAC protocols whose design is based on the assumption of constant traffic generation could waste energy when the sensor network, in the absence of events, produced little or no traffic. A MAC protocol employed for such applications could also save energy by taking into account some application requirements in order to limit the amount of traffic produced by a node. For example, a maximum number of reports to forward or an accepted latency, beyond which reports become useless, could stop the forwarding process for a node.

This approach is the starting point of Correlation-based Collaborative MAC [44] (CC-MAC). CC-MAC exploits the knowledge that that sensor nodes located near each other generate correlated measurements. The protocol reduces the number of messages transmitted in the sensor network by filtering messages originated by highly correlated sensor nodes. The reduced traffic volume results in lower wireless medium contention, and so fewer collisions. Nodes can also operate with lower duty cycles. The actual filtering algorithm is based on statistical information about the deployment of the nodes which can be deter- mined during the network initialization. The base station, which is supposed to have su cient computational resources, calculates the filtering parameter, called correlation radius. Sensor nodes closer than the correlation radius are assumed to produce correlated, and therefore redundant, readings while sensor nodes located farther than the correlation radius are assumed to generate independent results. The need to run a centralized algorithm and then to distribute its results throughout the network can be considered the main cost of the protocol.

CC-MAC consists of two components: Event MAC (E-MAC) and Network MAC (N-MAC).

E-MAC has the responsibility of filtering sensor node measurements, by making only nodes separated by at least the correlation distance generate data. In order to balance energy consumption within an area, nodes rotate the role of generating measurements. E-MAC traffic is identified by a First Hop (FH) bit set in the control packet headers. In general, the radio range does not match the correlation radius, hence the two cases where the transmission radius of the sensor nodes extends further than the correlation radius and where the correlation radius extends beyond the transmission range are discussed by the authors. The packet exchange resembles the RTS/CTS/DATA/ACK scheme in IEEE 802.11.

Nodes that do not generate measurements only participate in the forwarding process, which is governed by the N-MAC, and operate with lower duty cycles. N-MAC packets have the FH bit cleared and get priority access to the channel by means of reduced collision windows and interframe spaces, in the same way that the PCF in IEEE 802.11 receives preferential access to the wireless channel over the DCF.

The main limit of CC-MAC is represented by its complexity. Moreover when the sensing conditions change with time, the protocol must compute the new cor- relation radius and distribute the result throughout the network, this introducing an overhead which may become significant for large networks.

3.1.4 Encounter-Based MAC Protocols

When an unscheduled MAC protocol is used, and nodes adopt duty cycle operation, it is necessary to coordinate the nodes that must communicate. One

possibility is for a node to send probe messages until the neighbor awakes, and then, once the nodes that want to communicate encounter each other, the message transfer can begin. Several techniques exist, which differ from the developement of a network-wide synchronized schedule, as they basically only synchronize nearby sensor nodes when needed, and only for the duration of the transmission. Traffic characteristics such as patterns and predictability should be evaluated when deciding for the use of unscheduled encounter-based protocols over scheduled ones. The following sections briefly discuss some encounter-based solutions.

STEM, TICER and RICER

The two variants of the Sparse Topology and Energy Management Schurgers et al. (January 2002) (STEM) protocol show two different approaches to the problem of local coordination of nodes. With STEM, nodes are assumed to work with alternating sleep and wake states. When a node wants to transfer a message it uses signaling messages in order to wake up the destinations.

In STEM-B the message source will alternate between transmitting beacon packets and listening for a reply from the intended receiver. Nodes periodically sense the channel, therefore the destination node should be able to catch one of the beacon packets and reply to the source with a small acknowledgment packet.

In STEM-T the source sensor node transmits a long tone message instead of beacon messages. The length of the tone must be such that the destination has a high probability of sensing at least part of it during its awake period.

In both cases, once the nodes are synchronized, a full-functioned MAC protocol transfers the message.

Similar to STEM-B is the Transmitter Initiated Cycled Receiver (TICER) [22] protocol, in which sensor nodes with data to send periodically transmit RTS control messages and wait for a reply. Candidates for the reception of data periodically listen to the wireless channel and reply with a CTS when they detect an RTS message, thus enabling the data exchange to start.

The Receiver Initiated Cycled Receiver (RICER) [22] protocol operates in a dual fashion, by having receivers periodically transmit beacons at the beginning of their awake period. In RICER sensor nodes with data to transmit must listen to the channel until they hear a beacon from the intended receiver, and then start the data transmission.

Clearly, the performance of this family of approaches is heavily dependent on protocol parameters, such as the time between control messages and the duty cycles.

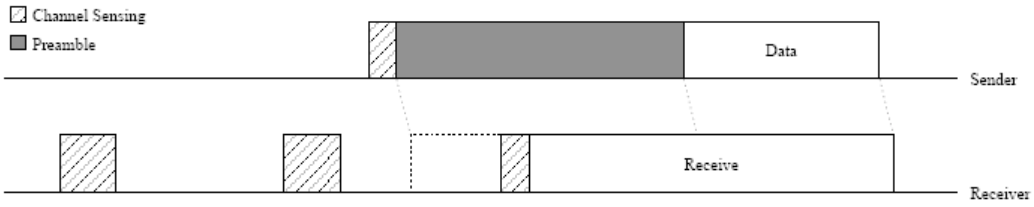

Figure 3.2: Message Transfer in B-MAC.

B-MAC

The Berkeley MAC (B-MAC) [34] protocol works in a similar fashion to STEM-T, with the tone message replaced by a long-preamble. In B-MAC sensor nodes follow independent sleeping schedules, aiming to a target duty cycle, and periodically wake up and sense the channel, in order to detect activity. When a sensor node sense a preamble on the channel, it remains awake to receive the upcoming message, otherwise it returns to sleep. Transmitters check the state of the channel before any access as in traditional CSMA schemes, and may incur performance degradation due to hidden terminal issues.

Figure 3.2 illustrates the message transfer in B-MAC, with a receiver period- ically sensing the channel and returning in the sleep state, till it awakes in the middle of the reception of a preamble and then stays active to receive the message.

Extensive flexibility, allowing to adjust protocol parameters, makes B-MAC suitable for a wide variety of scenarios.

WiseMAC

WiseMAC [14] is a protocol similar to B-MAC, which introduces a certain level of synchronization among neighbor nodes, by having sensor nodes remember the channel sampling schedule of each other. This is obtained through the use of an extra field in ACK packets, which indicates the time until a node's next channel sampling. With this information available, a node can properly delay a transmission so that it starts just as the receiver wakes up to sense the channel. WiseMAC can use shorter preambles, and it has less overhearing, at the cost of some extra control information exchange and the corresponding memory space. Figure 3.3 shows a message transfer using WiseMAC. The sample rate of the receiver is the same as in the case of B-MAC (see Figure 3.2), but since the sender knows when the receiver will wake up, it can minimize the duration of the preamble and save energy. The ACK packet is also shown, with the double purpose of confirming the correct message reception and announcing the next

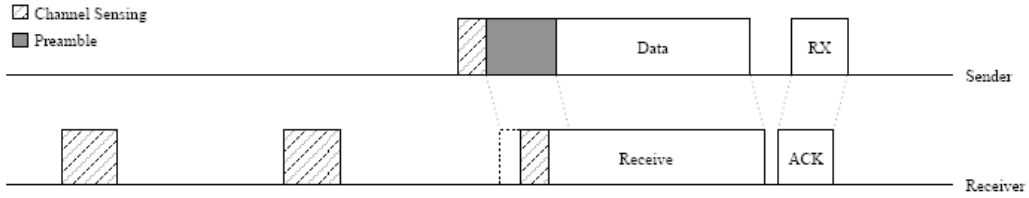

Figure 3.3: Message Transfer in WiseMAC.

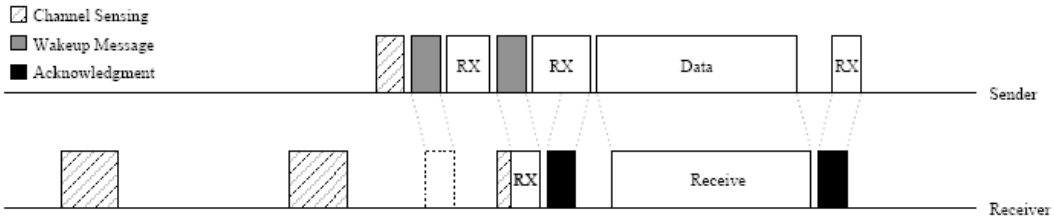

Figure 3.4: Message Transfer in CSMA-MPS.

channel sample time.

CSMA-MPS

The CSMA with minimal preamble sampling (CSMA-MPS) [26] protocol, similarly to STEM-B and TICER, uses small control messages instead of a long preamble. The use of small messages allows the nodes to determine the sampling schedule of the destination nodes without the need to exchange synchronization information. When a receiver node wakes up in time to receive a wakeup message and to reply with an ACK message to the sender node, the reception of the ACK performs two functions: it informs the source node that the data exchange can begin (wakeup and ack messages have equivalent roles as RTS and CTS messages), and it implicitly communicates the relative time off-set of the sampling schedule of the receiver. Figure 3.4 shows a message transfer where the receiver node responds to the second wakeup message transmitted by the sender node.

With CSMA-MPS, the first two control messages can also carry small amount of data. Implementing CSMA-MPS requires responsive control of the transceiver, which is used at high switching rates.

3.2 Scheduled MAC Protocols

Scheduled MAC protocols coordinate sensor nodes by means of a common schedule. TDMA is generally the form of multiple access with the most contained cost, as compared with frequency or code division which have greater power and hardware requirements, and therefore it is also the most common choice. The schedule indicates which sensor nodes should utilize the channel at any time, thus minimizing collisions, idle listening, and overhearing.

When nodes are not involved in message communication, which includes for- warding, they may enter the sleep state, and wake up when the schedule assigns them a 'slot' for a message transmission or reception.

Coordination can also be exploited to optimize energy consumption for the entire network or groups of nodes instead of having single nodes defining their activity based only on their personal goal of lifetime maximization. Criteria can be adopted in order to grant priority access to nodes with important traffic or with a larger backlog of messages, which considering the scarce memory resources may lead to queue overflow and loss of traffic. In general, many forms of fairness can be considered.

The main costs for this class of protocols derive from the need to create and maintain the schedule. Synchronization is typically critical, and it may require the transmission of periodic beacons, which increases the transceiver utilization, or the use of precise oscillators, which increases the sensor node cost.

From a functional point of view, some disadvantages exist, including the complexity of adjusting the schedule in the presence of node mobility, node redeployment or node death. Typically, some delay between the time a sensor node dies and the time the protocol reassigns its resources, is inevitable, so some resources may be wasted. Delay can also be an issue when new sensor nodes join the network.

How the protocol operates under situations where sensor nodes posses incorrect state is also a fundamental problem, as inconsistency may lead to errors and instability, so cancelling the benefits of the protocol.

Depending on the characteristics of the traffic generation, and how it changes throughout the network and with time, nodes may not receive enough resources compared to their needs, or, conversely, they may not fully use the assigned re- sources. Hence, resource management algorithms must possess enough flexibility and speed in order to deal with such situations, possibly comprising the use of extra resources. In TDMA-based MAC protocols the choice of the time slot length becomes crucial: reducing the time slot length may decrease the waste associated with short messages, but also decrease the maximum message length without fragmentation; longer time slot length may introduce unwanted latency.

3.2.1 Priority-Based MAC Protocols

In the following protocols, the access to the channel is deterministically decided based on priorities assigned to nodes or links. Priorities are derived by nodes' IDs through the use of pseudo-random functions. Nodes maintain information about their two-hop neighborhood by including control information in data messages.

NAMA

Node Activation Multiple Access (NAMA) [5] uses a TDMA-based access scheme, where time is divided in sections, each section is formed by parts, and finally each part is divided into slots. Nodes select a part and contend with the two-hop neighbor nodes which have selected the same part. Each node computes a priority value for every slot, based on its ID, as well as the priorities of the neighbor nodes for the same slot, and decides to use the slot when it has the highest priority among the neighbors. The last section of each block is reserved for signaling messages that allow sensor nodes to join the network.

LAMA

Link Activation Multiple Access (LAMA) [5] is a protocol that uses both topology information and code division to coordinate the access to the channel. The protocol assigns a Direct Sequence Spread Spectrum (DSSS) code to every receiver. A node which wants to communicate with a destination, will use a TDMA slot where it has the highest ID-based priority in its two hop neighborhood, and then will transmit using the DSSS code associated with the receiver.

PAMA

In Pairwise-link Activation Multiple Access (PAMA) [5] the protocol activates a link (u,v), allowing the source u to transmit towards the destination v, when the link has the highest priority among all links of nodes u and v, and node u has the highest priority of its two hop neighbors. Similarly to LAMA, the use of DSSS allows parallel non-interfering communications. The protocol makes so that the same code is not used in communications within a two-hop distance.

NAMA, LAMA and PAMA all require a sensor node to compute the priorities of each neighbor and for each time slot, which may be a resource consuming activity and constitutes the main drawback of the protocols. Moreover LAMA and PAMA

also require the sensor nodes to have complex radios, as they both use direct spread spectrum techniques.

3.2.2 Traffic-Based MAC Protocols

In those applications where the network produces low traffic most of the time, with peaks of large volumes of traffic, MAC protocols that adapt to changing traffic conditions may consume less energy, while still providing good throughput when needed. Traffic estimation and control information must be shared in order to promptly react to changes.

TRAMA

The Traffic-Adaptive Medium Access (TRAMA) Rajendran [36] protocol uses a TDMA frame with scheduled slots with no contention, used for longer data messages, and random access slots, for small control messages. The structure of the TDMA frame is shown in Figure 3.5. TRAMA uses three sub-protocols in order to adapt to traffic conditions, to learn the two-hop topology of their neighbors, and to assign the use of the slots.

The Neighbor Protocol (NP) is used to share the topology information. Sensor nodes use a random control slot and transmit a list of their one hop neighbors. All sensor nodes collect information from neighbors' control messages and determine the sensor network topology within a two-hop neighborhood. Control messages may experience collisions and require retransmissions; moreover, the number of control slots should be planned based on the expected number of two-hop neighbors.

The Schedule Exchange Protocol (SEP) allows nodes to share traffic information. It operates similarly to the NP, by transmitting schedule packets and summaries which inform the neighbors about the transmitting node's queued traffic. A schedule packet is transmitted during the last owned slot in each frame, containing the number and positions of the slots the protocol has assigned to the node in the next frame, and a bitmap of the intended receivers. Schedule sum- maries are appended to data packets and provide a backup mechanism against loss of schedule packets. They include a bitmap that indicates only the slots the sensor node plans to transmit in, during the current frame, but not the destinations' identities.

The Adaptive Election Algorithm (AEA) is the sub-protocol which selects the slots to use for data transfer based on the collected topology and traffic information. Slots where a sensor node has no planned transmission or reception may be used to enter a sleep state. Similarly to NAMA, the AEA protocol assign data slots by defining a node priority based on the sensor node's ID and the slot number, with the node with the highest priority within a two-hop neighborhood winning

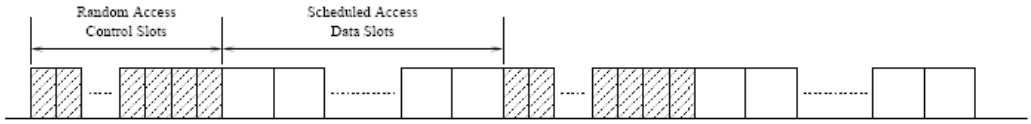

Figure 3.5: Medium access in TRAMA.

the slot. The owner of a slot has the right to transmit while a sensor node indicated as destination will attempt to receive the message.
Inconsistency in the determination of slots' ownership, due to different view of the two-hop neighborhood topology, may cause data loss or idle listening, and the authors discuss techniques to obviate this problem.
TRAMA efficiently reduces collisions and energy consumption through its adopted scheduled access, and quickly adapt to changes in network and traffic conditions
by providing the random access slots once per frame. Additionally, the use of information appended to data messages and bitmaps reduces the protocol overhead.
The disadvantages of TRAMA are the ones typical of a scheduled protocol, including a high level of complexity, memory requirements and overhead. TRAMA may experience decreased performance in case of inconsistent state among the nodes, which it attempts to minimize by using schedule summaries and having sensor nodes listen during a transmitter's final data slot. Requiring that the nodes stay awake during the control slot portion of each frame, TRAMA also has some limitations on the possible duty cycles a sensor node may adopt.

PMAC

The Pattern MAC (PMAC) [47] protocol uses an approach similar to TRAMA. As shown in Figure 3.6, PMAC defines a frame which consists of Data, Broadcast and Pattern Exchange slots.
Each node decides a pattern of sleep and awake slots that it will use in the upcoming frame, and transmits a bitmap ("1" indicating awake slots, and "0" sleep slots) which represents this pattern within the Pattern Exchange slots, using CSMA. A pattern has the format of zero or more sleep slots followed by an active slot. It contains the minimum information necessary to determine the activity on the entire frame, hence a sensor node with a 25% duty cycle would transmit a pattern 0001, which in a 10 slot frame would be expanded by the neighbors to
0001000100. Nodes compare their pattern with neighbors' patterns to determine the schedule of transmissions they will use, in a distributed manner.
A node updates its pattern based on the traffic it has to handle, increasing and

Figure 3.6: Medium access in PMAC.

decreasing its activity in a way that is similar to the scheme of growth of the TCP congestion window. Sensor nodes start with a pattern of 1, which corresponds to an entirely active frame, and decrease multiplicatively their duty cycle by doubling the number of sleep periods per active period. Therefore a possible sequence could be 1, 01, 001, 00001. When a node reaches a threshold duty cycle, it continues to decrease its activity linearly, by adding one single sleep slot at a time. All the times a sensor node has a message to send, it immediately increases its duty cycle to 100%.

The schedule a sensor node uses depends on the patterns of its neighbors along with the pattern it generates, and it consists of a sequence of three possible operations for each slot: transmit, listen, or sleep. A node with a message for a neighbor wakes up and transmits within a slot where that neighbor advertises a 1. A node listens during slots where it advertises 1. In case of absence of activity a node may also return to sleep after a short time, in order to conserve energy. In all the other cases a node will sleep for the entire slot.

PMAC is an hybrid scheduled/non-scheduled protocol, as data transmission within slots occurs using CSMA/CA with ACKs providing reliability.

During the Broadcast Slot, all the nodes remain awake, so that broadcast messages may be effectively transmitted. The Broadcast Slot may also be used to deliver messages to sensor nodes with very low activity schedules.

PMAC is a simple and quickly adaptive solution, however some disadvantages can be spotted. First, since sensor nodes exchange their patterns by using CSMA, collisions and errors may cause inconsistent information, leading to collisions, idle listening, and message loss. Moreover, since the protocol updates the pattern each time the sensor node operates in an active time slot, its processing requirements may become an issue during times of high traffic intensity.

3.2.3 Clustering-Based MAC Protocols

Grouping sensor nodes into clusters allows to efficiently perform some tasks, such as synchronization, in a local manner, without involving the whole network. State distribution can be limited to sensor nodes belonging to the same cluster, thus reducing the energy consumed for sharing this information, while still keeping

an advantage with respect to protocols where sensor nodes decide their behavior independent of other sensor nodes. Clustering-based MAC protocols are generally more scalable, as clusters may be viewed as single entities at a higher level of abstraction. Moreover, different types of traffic, i.e. local and global traffic, can be managed with different mechanisms, involving only intra-cluster communication or otherwise inter-cluster communication.

However, clustering typically require more message overhead. Sensor nodes acting as cluster heads, have managing duties and generally consume more energy than ordinary sensor nodes, which in turn are coordinated in such a way to reduce their average energy consumption. For this reason, protocols often rotate the cluster head functionality among the sensor nodes of a cluster, in order to evenly distribute the additional energy consumption.

Redeployments, mobility, and sensor node death complicate clustering proto- cols as frequent cluster formation and head assignment may be required. Since the execution of these tasks consumes energy and computational resources, a good trade-off must be found in terms of frequency of cluster reformation, considering the energy savings possible from cluster reformation and the application requirements.

LEACH

The Low-Energy Adaptive Clustering Hierarchy (LEACH) [17] protocol organizes the network according to a 2-levels hierarchy. As shown in Figure 3.7 the Base Station has direct communication links with cluster heads, which in turn exchange data with sensor nodes belonging to their cluster. The cluster head role rotates among the sensor nodes in order to equalize the energy consumption. Within each cluster the sensor nodes communicate using an orthogonal direct sequence spread spectrum (DSSS) code, hence interference with other clusters is limited. One code is reserved for the communication between the cluster heads and the base station.

The cluster formation is initiated by a node which transmits a cluster head announcement message accepting the cluster head role. Sensor nodes wait for a random delay before transmitting an announcement and becoming cluster heads. Sensor nodes which receive a cluster head announcement send a cluster join message to inform the new cluster head of their membership. When a sensor node receives cluster head announcements from multiple neighbors, it can select the cluster head that requires the lowest energy for communication.

Cluster heads compute and distribute schedule to the sensor nodes it controls, assigning time slots within which sensor nodes transmit their messages to the cluster head. Cluster heads perform message aggregation and forward the gathered data to the base station, using a single message. Data aggregation prevents over loading the communication links to the base station, making each

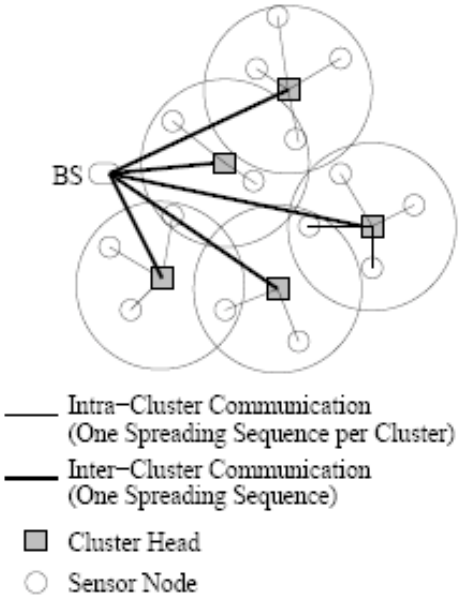

Figure 3.7: Network communication hierarchy in LEACH.

cluster produce a traffic equivalent to a single sensor node. The communication between the cluster heads and the base station is governed by CSMA.

The most important disadvantage of LEACH is that it requires a complex radio, which increases the sensor node cost. Another limitation, is the assumption that each sensor node can communicate directly with the base station, which drastically reduces the possibilities of application of the protocol to small geographical areas, or alternatively requires nodes to consume large amount of energy for long-range transmissions. As pointed out by the authors, evolutions of the protocols could address this drawback by adopting a multi-hop routing structure out of the cluster heads or using a multiple-level hierarchical structure of clusters.

Finally, cluster formation can take a long time during which the sensor nodes cannot perform any useful work.

LEACH-C

LEACH-C is a variant of LEACH, which uses the base station to select the cluster heads. During a network setup phase, each sensor node transmits its location and energy levels to the base station, which computes the optimal selection of clusters and transmits a list of sensor nodes that will act as cluster heads. The cluster formation process is then similar to LEACH with sensor nodes transmitting join

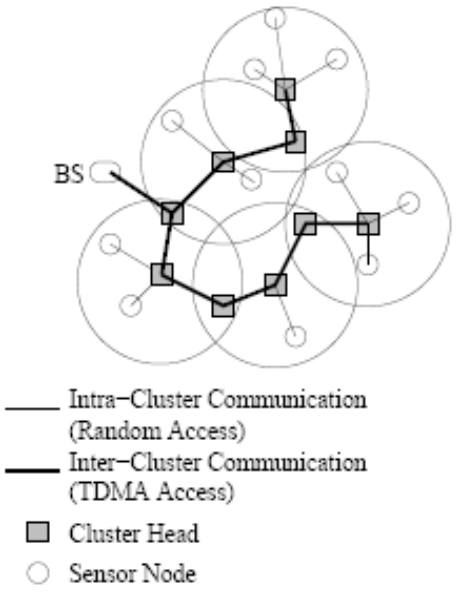

Figure 3.8: Network communication hierarchy in GANGS.

messages and cluster heads distributing schedules.

LEACH-C can conserve more energy than LEACH, however it requires nodes to determine their location by using hardware support (e.g. GPS) or range estimation algorithms, increasing the node cost and power consumption.

GANGS

GANGS [7] is a cluster-based protocol which defines a TDMA communication scheme for transmissions between cluster heads, and demands the organization of intra-cluster transmissions to external unspecified contention protocols.

Differently than in LEACH, sensor nodes are not assumed to be able to communicate directly with the base station, instead the cluster heads form a routing backbone in the sensor network, which is shown in Figure 3.8.

GANGS follows a two-steps process for the formation of its clusters: cluster head election and cluster connection.

During the first phase, sensor nodes share their energy resource level with neighbors. Nodes that have more energy left among their neighbors declare themselves cluster heads and transmits an announcement message.

During the second phase non-cluster head sensor nodes must join a cluster. If a sensor node has received a single cluster head announcement, it joins that cluster.

If it has received multiple cluster head announcements, it select the cluster head with the highest energy level. Astley, when a sensor node has not received any announcement, it sends a message to the neighbor with the highest energy level, which automatically is promoted to cluster head.

This procedure builds a clustered sensor network with connected cluster heads. Since cluster heads will eventually have lower energy resources than their neighbors, the cluster formation procedure is repeated.

TDMA slots used by cluster heads are assigned with the execution of a distributed algorithm. Each cluster head picks a random number between one and the number of neighbors it has plus one and transmits this number to its neighbors. When two neighboring cluster heads pick the same number they repeat the procedure by picking an unused number. At the end of this process each cluster head decides to use the time slot which corresponds to its number, thus defining the TDMA schedule. This mechanisms requires a frame length larger than the maximum expected cluster head connectivity, which in GANGS is fixed and equal in the whole network.

GANGS has about the same disadvantages as LEACH, including overhead, energy consumption and unavailability periods due to cluster formation and restructuring. As in LEACH, the extent and frequency of cluster reformation is a primary concern. Moreover, in GANGS, cluster reformation also affects routing and may lead to instability.

The way GANGS assigns slots to cluster heads is rather not efficient, as not all slots may get used. For example, sensor nodes within a cluster are supposed to use the frame's slots following the one used by the coordinator, however there will likely exist multiple unused slots between the slots assigned to cluster heads.

Compared with LEACH, GANGS is as flexible and less complex, also requiring less expensive sensor nodes. Compared with TRAMA, it requires much fewer computational resources. In sum, its characteristics make GANGS a suitable choice for small networks of low cost nodes.

Group TDMA

Group TDMA [38] attempts to optimize channel utilization by dividing sensor nodes into groups that can communicate simultaneously.

Each different group has a set of assigned TDMA slots, so that collisions between nodes of different groups do not occur. Clusters are formed based on topology information.

Similarly to GANGS, Group TDMA does not specify detailed message exchange rules, it provides functionalities for clustering and collision avoidance between clusters and it needs to be used coupled with a MAC protocol that arbitrates

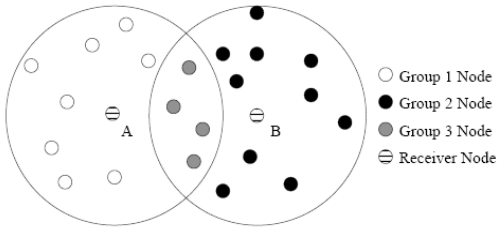

Figure 3.9: Receiver-Based Grouping in Group TDMA.

intra-cluster transmissions.

The formation of Group TDMA clusters occurs in a distributed manner. Each sensor node waits for a random amount of time, and then it transmits a message announcing it will act as a receiver. Neighbors which receive the message enter the cluster associated with that receiver node and become transmitters. When all sensor nodes have become transmitters or receivers, the protocol assigns time slots to each group.

Figure 3.9 shows an example topology. Node A and B transmit their announcement message and become receivers. All the sensor node receive at least one message and then the group formation ends, with three groups formed. Group
1 nodes can transmit only to node A, Group 2 nodes can transmit only to node B, and finally Group 3 nodes can transmit to either node A or node B. Group 3 is fur- ther divided into sub-groups. Group 3_1 will contain all Group 3 nodes with traffic for node A, while Group 3_2 will contain all Group 3 nodes with traffic for node B. Group TDMA can now assign three slots: Group 1 and 2 will transmit during the first slot, Group 3_1 will transmit during the second slot, and the third slot will be assigned to Group 3_2. Group TDMA allows sensor nodes to sleep during the slots of other groups if they do not have messages to transmit. In general, Group TDMA uses a distributed algorithm that approximates the link coloring problem, defining rules for groups formation and assignment of slots based on connectivity and traffic, with reuse of slots after proper spatial separation.

The set of sensor nodes that act as receivers is selected again after a number of frames, in such a way that all nodes can communicate.

The authors present methods to determine the slot length which maximizes the network throughput given the group organization and traffic distribution, as well as the number of frames between two group formations, which minimize energy consumption given the energy resources left in each group and their energy consumption rates.

Similarly to other scheduled protocols, Group TDMA can consume a large amount of energy and take a significant amount of time for the setup phase, therefore it may not work well for highly dynamic sensor networks. The state of receiver group membership and transmission schedules can consume large

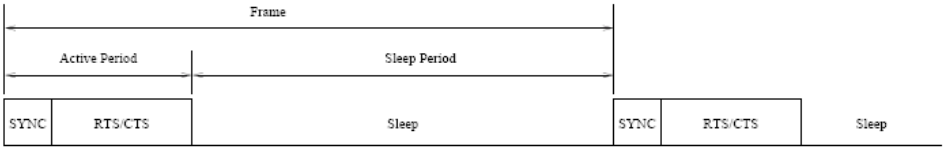

Figure 3.10: Frame Format in S-MAC.

amount of memory resources, while the involved processing operation are not resource demanding. Excessive latency may become an issue, as a sensor node must queue messages until the destination becomes an active receiver. Channel reuse makes the protocol reach higher levels of efficiency than other protocols, giving Group TDMA an advantage in large multi-hop networks.

S-MAC

Sensor MAC (S-MAC) [45],[46] is a very popular and perhaps the most studied scheduled MAC protocol for sensor networks.

S-MAC synchronizes sensor nodes, forming virtual clusters consisting of all the neighboring sensor nodes simultaneously awake. S-MAC defines a frame format, which is shown in Figure 3.10.

Each node with an already determined schedule transmits a SYNC message at the beginning of the active frame time, allowing the neighbors to synchronize by learning its schedule so they can wake up at the proper time to transmit a message. Each sensor node transmits SYNC messages performing a simple collision avoidance algorithm, based on a random backoff .A node willing to transmit a message will initiate a message exchange using RTS and CTS, during the portion of the frame following the interval used for SYNC messages.

A sensor node adopts the schedule of a neighbor and so joins its virtual cluster when it currently does not have a schedule and hears a SYNC message. If the neighbors of a given sensor node adopt multiple, sufficiently different schedules, that node adopts a schedule which is a merge of the neighbors' schedules and allows it to communicate with the different virtual clusters. Finally, a sensor node that does not hear any SYNC messages for a defined amount of time, chooses its own schedule.

When sensor nodes strictly follow their schedules, the probability of hearing new SYNC messages from neighboring nodes is as low as their duty cycle, hence in order to detect new schedules, they periodically listen for a longer time period.

Message transfer occurs using a particular RTS/CTS/DATA/ACK procedure, namely sensor nodes transmit the RTS and CTS messages during the Active

Period, while the data message is transferred during the Inactive Period. This allow the sensor nodes which read either the RTS or the CTS and are not involved in the communication to sleep, avoiding overhearing. This scheme requires all the sensor nodes to perform both physical and virtual carrier sensing. The RTS and CTS messages contain the message transmission time, including time for the ACK message, informing all the other sensor nodes about the time interval during which they may sleep, and cannot transmit without causing interference.

The original scheme of S-MAC clearly allows only to forward a message over one hop per frame time. The authors, in order to overcome this limitations, introduce the adaptive listening technique, where a next-hop node is announced within the CTS. If the next-hop node hears the CTS it can wake up at the end of the data transmission, while the sensor node that receives the message will attempt to forward it by starting a message transmission sequence after it sends an ACK to the original sender, overriding its schedule. As a result, message latency can be decreased. However, the adaptive listening technique only works within a virtual cluster, since sensor nodes outside the cluster are likely to not receive the CTS message.

The protocol also allows message fragmentation, which reduces the impact of collisions on the network throughput, as only fragments of large messages need to be retransmitted in case of collision or channel error. A single RTS/CTS exchange can used for the transmission of all the fragments of the message.

S-MAC provides a clustering functionality while only loosely and locally synchronizing sensor nodes, so reducing the problems which derive from approaches that aim to strict and network-wide synchronization. Local synchronization also allows the protocol to scale easily. S-MAC requires few processing and memory resources, relying only on o sets and timers, and adapts quickly to new conditions, as schedule and synchronization maintenance occur at each frame interval. Coordination is obtained through the use of SYNC messages, acting as beacons, and therefore sensor nodes do not have to forward or share large amounts of state information.

S-MAC, has been at the center of many studies, which have pointed out some disadvantages.

First, sensor nodes positioned along the borders of several virtual clusters adopt several schedules, multiplying their duty cycle, and reducing their lifetime. Premature nodes deaths lead to performance degradation and even network segmentation. A second disadvantage is that S-MAC uses a static duty cycle, which can be set by the send users based on expected application requirements, and may not change following traffic or density conditions, thus possibly consuming more energy than required or limiting the protocol's performance. Finally, S-MAC does not attempt to control virtual cluster size, which may have an important impact on the protocol's performance, as large clusters reduce the number of sensor nodes that need

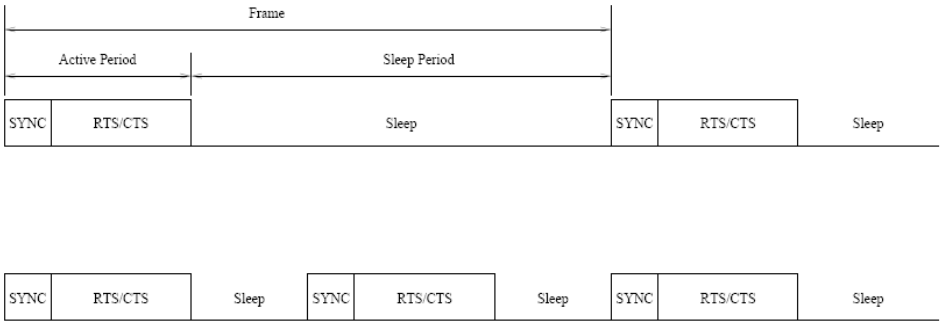

Figure 3.11: Frame Format in DSMAC.

to use multiple schedules, but increases the message latency by increasing the level of contention of the same period of activity.

The following section illustrates some attempts to improve S-MAC.

S-MAC Variants

The S-MAC protocol's limitation of static duty-cycle is dealt with in DSMAC [23], which adopts dynamic duty cycles based on traffic and energy considerations. DSMAC uses extra fields in SYNC and data messages to allow sensor nodes to increase their duty cycle, considering large per-hop data delays as indicator for a duty cycle that needs to be increased. Delays are measured from message reception to transmission completion, and added to future data messages in order to inform neighboring nodes. DSMAC also includes a limit on the maximum duty cycle a sensor node may reach, reducing the maximum energy consumption rate. As shown in Figure 3.11, sensor nodes within the same virtual cluster which decide to increase the duty cycle, do so by multiplying it by powers of 2. In this way these sensor nodes remain synchronized, as they can still receive SYNC messages sent by sensor nodes operating at lower duty cycles.

The T-MAC [41] protocol is an enhancement of S-MAC which uses a timer to indicate the end of the active period. As a result, the active portion of each frame may have a duration which depends on traffic conditions and save energy when needed. In Figure 3.12 two consecutive frames are shown, the first containing both a SYNC message and data transmission, followed by a second one, where only the SYNC message is sent, with a resulting shorter duration of the active period. Similarly to the adaptive listening technique of S-MAC, T-MAC introduces a future request to send message (FRTS), used to inform the next hop of a message that it has a future message transfer and reduce message latency. T-MAC also uses a flow control mechanism which

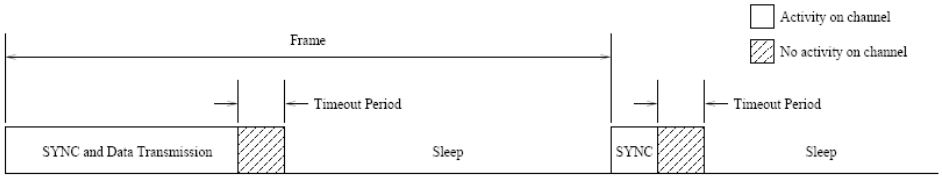

Figure 3.12: Frame Format in T-MAC.

gives priority to sensor nodes with buffers close to overflow, which receive a higher chance at transmitting their queued messages.

Finally, MS-MAC Pham and Jha [32] introduce a mechanism to decrease the time a sensor node needs to join a virtual cluster. Nodes record received signal strength values for each neighbor and use any changes as indications of sensor node movement. SYNC messages include the maximum speed a sensor nodes estimates among its neighbors, and nodes with a high mobility, as well as sensor nodes around a highly mobile sensor node, increase the rate at which they checks for new schedules.

3.2.4 TDMA MAC Protocols

Protocols based on time division multiple access (TDMA) provide mechanisms to reduce collisions and idle listening, as well as the possibility to introduce fairness among the sensor nodes. Issues derive the overhead information needed to coordinate the sensor nodes, which may affect the performance of large networks.

Overhead traffic is partially related to the synchronization functionality, which must exist to compensate the clock drift of the typically cheap oscillators.

Flexibility is necessary to avoid utilization problems and waste of energy during periods of low traffic.

EMACS, LMAC, and AI-LMAC

EMACS [42], LMAC [43], and AI-LMAC [9] are TDMA MAC protocols which share several similarities. All the protocols define a frame with time slots and slot assignment occurs by sensor nodes picking a random slot not controlled by a neighboring sensor node. Each sensor node transmits a control message during any time slot it owns, maintaining synchronization with neighboring nodes and notifying forthcoming data transmissions.

As shown in Figure 3.13, the EMACS's time slot has three sections: communication request, traffic control, and data. The communication request section is used by sensor nodes to request access to the data section of a time slot they do

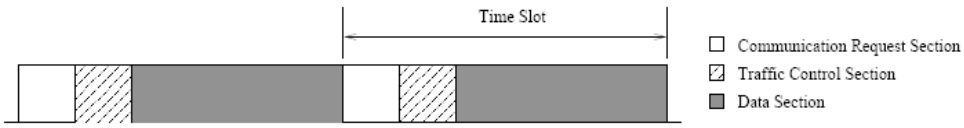

Figure 3.13: Frame Format in EMACS.

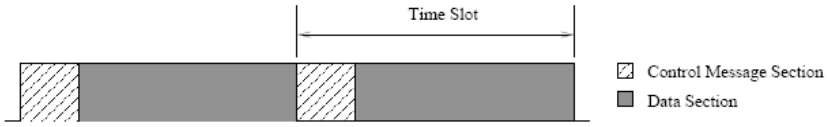

Figure 3.14: Frame Format in LMAC.

not own, with the time slot owner replying and possibly passing the ownership to the requesting sensor node within a control message. Control messages are transmitted during the traffic control section, when every node must listen for the control packet of the time slot owner neighbors. Lastly, during the data section, data transmissions occur.

The slot assignment process is initiated by the base station, which at the be- ginning owns all the time slots. Neighboring sensor nodes randomly pick a slot and request its ownership. Nodes which experience collisions, indicate the event within the control message they transmit during their time slot, notifying sensor nodes who have the incompatible ownership. This process propagates slot owner- ship through the sensor network with sensor nodes reusing slots at non-interfering distances.

EMACS provides varying levels of functionality, which allow the sensor nodes to conserve energy when the application does not require the entire population of nodes to be active in order to perform its tasks. Namely, sensor nodes may operate in one of three possible modes. Active nodes own one or more slots and transmit a control message within each slot they own. Passive sensor nodes do not own a slot and only transmit messages after requesting a slot from an active sensor node. Lastly, dormant sensor nodes do not participate in the communication and sleep.

In LMAC all sensor nodes own a slot, in the terms of EMACS, operate as active nodes. Figure 3.14 shows the frame structure of LMAC; compared to EMACS, since all sensor nodes own a slot, the communication request section is not present. LMAC adopts a simple hop count-based routing protocol used to forward messages towards the base station.

The method used to assign time slots by EMACS and LMAC is good for its simplicity, affordable by very limited devices. In large networks, however, network

setup may take considerable time, since the process starts at the base station and is possibly delayed by slot collisions, which may take several frames to resolve. The protocols also have large overhead and listening period used for slot maintenance, because of control information transmitted during the control portion of each slot.

Overcoming a limit of EMACS, the AI-LMAC protocol [9] introduces a varying number of slots a sensor node may own, which may be adapted based on traffic conditions. Each sensor node maintains a Data Distribution Table (DDT) which records statistics on the data which traverse a sensor node, such as values, originating node, and previous hop.

Sensor nodes are organized in a parent-child hierarchy, where parents, based on information within the DDT, may suggest that a child take control of more or less time slots. Since parent nodes formulate suggestions, fairness of slot assignment can be achieved, and it can also be ensured that aggregate child bandwidth does not exceed the parent sensor node's upward bandwidth.

A sensor node only transmits a control message in the first time slot it owns within a frame, including the time slots it owns and indicating any data messages it will transmit during the current frame.

Compared to LMAC, AI-LMAC introduces data message acknowledgments, transmitted inside control messages. A primary disadvantage is represented by the overhead required for the Data Distribution Tables, both in terms of extra information exchange and memory space. Keeping the DDTs updated may also consume computational and energy resources.

ZMAC

The Zebra-MAC (Z-MAC) protocol [37] follows a flexible approach which puts together CSMA and TDMA. Similar to other pure TDMA- based protocols, Z-MAC assigns sensor nodes a time slot, however it also allows sensor nodes to utilize slots they do not own through CSMA. In this way Z-MAC can perform similar to CSMA in low-traffic conditions, switching to a TDMA behavior when traffic requirements increase.

A distributed protocol assigns sensor nodes the time slots they may utilize for transmission, ensuring that two-hop neighbors do not get assigned the same slot number. The slot assignment procedure is repeated when the network topology changes. Assigning slots during network setup, introduces a large initial overhead, however it reduces the energy expended for communicating control information during the sensor network's lifetime. Node mobility or node re-deployment generate additional overhead as slot assignment is repeated.

All sensor nodes, including the slot owner, use CSMA to determine who may transmit during each time slot. However, the slot owner is given preference in channel access by means of a reduced initial backoff window, with respect to nodes

that do not own the slot. While the owner of the current slot selects a random backoff time of up to T_o, the other sensor nodes select a backoff time between T_o and T_{no}, where $T_{no} > T_o$, and all perform CSMA. The reason behind making the slot owner use a random backoff is the need to limit the e ects of incorrect synchronization among neighboring sensor nodes.

Sensor nodes receive messages according to the B-MAC protocol (Section 3.1.4) and maintain a receive schedule independent of the time slots.

Z-MAC provides a mechanism of congestion avoidance which basically moves the protocol behavior in the direction of TDMA. Namely, sensor nodes track the amount of time they spend in backoff because of failed carrier sensing, and send an explicit congestion notification (ECN) message to the neighbors they have messages for, which in turn broadcast the ECN message to their neighbors. The sensor nodes which receive the forwarded ECN message enter a high contention level (HCL) state. A node in the HCL state only attempts to transmit in its slot and those of its immediate neighbors, thus reducing contention between neighbors two hops apart and preventing hidden terminals from disrupting the communication from the original sender of the ECN message to the node which has forwarded it. After a time period without receiving any ECN messages, sensor nodes return to a low contention level (LCL) state. ECN messages can reduce contention within a local area, however they also introduce further overhead traffic on an already busy network.

Rapid adaptability to traffic conditions is the main advantage of Z-MAC, which can save large amount of energy by switching between its two operation mode. Z-MAC is also robust against synchronization errors.

Compared to other protocols, Z-MAC requires few processing and memory resources, but has a high protocol overhead, caused by the TDMA structure and slot assignment during network setup. Similar to any TDMA protocol, sensor nodes must also consume resources to maintain synchronization. Lastly, since Z- MAC uses B-MAC as its underlying communication mechanisms, it inherits its main disadvantages.

Chapter 4

IEEE 802.15.4

4.1 Characteristics of IEEE 802.15.4

4.1.1 IEEE 802.15.4

The IEEE 802.15.4 standard was created for low-rate, wireless personal area net- works (WPANs) [1]. IEEE 802.15.4 is targeted for low-cost, resource-constrained devices that are deployed for lengthy periods of time without maintenance or battery replacement. The application domain for the standard includes wireless sensor networks, industrial and commercial control and monitoring, and home automation. The standard is divided into two layers: the Physical (PHY) Layer and the Media Access Control (MAC) Layer.

Different network topologies are allowed, including star, mesh, and cluster tree networks. Figure 4.1 shows examples of topologies and indicates the communication flow.

IEEE 802.15.4 is designed to operate on two classes of devices: reduced function devices (RFDs) and fully functional devices (FFDs). FFDs have the capability to communicate with any device in a network within their communication range, while RFDs are only able to communicate with FFDs. A network consists of multiple FFDs and RFDs, with one of the FFDs designated as the personal area network (PAN) coordinator.

IEEE 802.15.4 PHY

The PHY layer specification dictates how IEEE 802.15.4 devices may communicate with each other over the wireless channel. It allows for the use of three frequency bands with varying data rates. The bit rates are 20 kb/s in the European 868 MHz band (868-868.6 MHz), 40 kb/s in the North American 915 MHz band (902-928
MHz), and 250 kb/s in the worldwide 2.45 GHz band (2.4-2.4835 GHz).

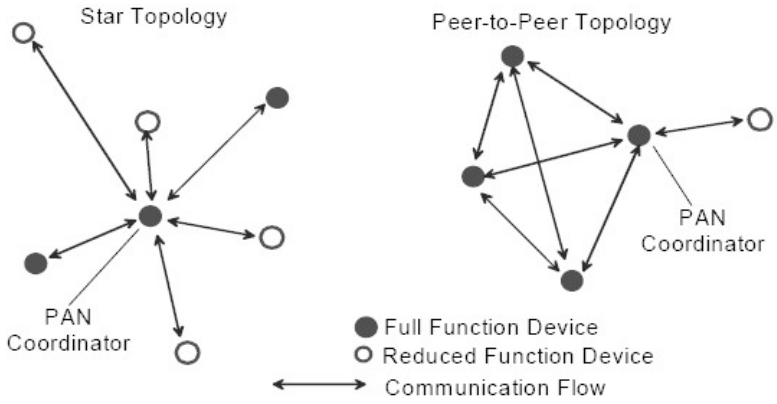

Figure 4.1: Star and peer-to-peer topology examples.

The IEEE 802.15.4 Physical Layer offers a total of 27 channels, one in the 868MHz band, ten in the 915MHz band, and, finally, 16 in the 2.4GHz band.

This layer is responsible for activation and deactivation of the transceiver, channel frequency selection, and data transmission/reception. In addition, it performs channel energy detection (ED), link quality indication (LQI) for received packets, and clear channel assessment (CCA) for the MAC carrier sense multiple access with collision avoidance (CSMA-CA) protocol.

IEEE 802.15.4 MAC

The MAC protocol specifies when devices may access the channel for communication. The basic services provided by the MAC are beacon generation and synchronization, supporting PAN association and disassociation, supporting optional device security, managing channel access via CSMA-CA, maintaining guaranteed time slot (GTS) communication, providing message validation, and providing message acknowledgments.

The standard defines four frame types:

- beacon frames;
- data frames;
- acknowledgment frames;
- MAC control frames.

Beacon frames are used by the coordinator to describe the channel access mechanism to other nodes. Data frames are used to send varying amount of payload

(2-127 bytes), while acknowledgment frames are used to increase reliability for data frame and control frame transmissions. Finally, the control frames are used to carry out network management functions, such as association to and disassociation from the network.

A PAN may be set up in one of two basic configurations: beacon-enabled and non beacon-enabled. In a non beacon-enabled network, devices may communicate with each other at any time after an initial association phase. Channel access and contention are managed using an unslotted CSMA-CA mechanism and any node-level synchronization must be performed at some higher layer.

In a beacon-enabled network, the PAN coordinator periodically transmits a beacon which other devices use both for synchronization and for determining when to enable transmission and reception of messages. This beacon message is used to define a superframe structure that all nodes in the PAN synchronize to. [1]

The beacon order (BO) subfield in beacon frames specifies the transmission interval of the beacon, called the beacon interval (BI), with $BI = B \cdot 2^{BO}$, where B is a base superframe duration, and $0 \leq BO \leq 14$. If $BO = 15$ the coordinator transmits beacon frames only when requested to do so, such as on receipt of a beacon request command.

The superframe is divided into several sections, the lengths of which are configurable. There is an active period, during which communication takes place, and an inactive period, during which devices may turn o their transceivers in order to conserve energy. The superframe order (SO) subfield specifies the length of time during which the superframe is active, called superframe duration (SD), according to the identity $SD = B \cdot 2^{SO}$ symbols. If $SO = 0$, the superframe following the transmission of the beacon frame is not active. The superframe structure is shown in Figure 4.2.

Figure 4.3 shows the durations of the beacon intervals and superframe durations corresponding to different values of BO and SO.

The active period is divided into 16 equally-spaced slots and may be further divided into a contention access period (CAP), and an optional contention free period (CFP). During the CAP, devices may communicate using a slotted CSMA- CA mechanism, similar to unslotted CSMA-CA, except that the back-off periods are aligned with slot boundaries. The CAP can contain from 9 up to all 16 slots.

[1] This definition thoroughly applies to star networks, where a single coordinator broadcasts beacons to all neighbor nodes. However, there may be scenarios with multiple coordinators that send beacons with different superframe structure and timing, in which case the receiving nodes only consider beacons transmitted by the coordinator they are associated with.

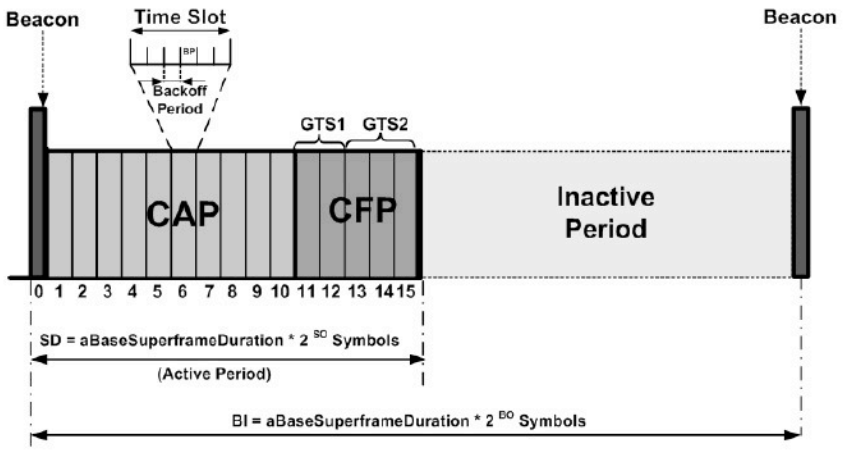

Figure 4.2: Structure of the superframe.

SO/BO	868 MHz	915 MHz	2450 MHz
0	0.048	0.024	0.01536
1	0.096	0.048	0.03072
2	0.192	0.096	0.06144
3	0.384	0.192	0.12288
4	0.768	0.384	0.24576
5	1.536	0.768	0.49152
6	3.072	1.536	0.98304
7	6.144	3.072	1.96608
8	12.288	6.144	3.93216
9	24.576	12.288	7.86432
10	49.152	24.576	15.72864
11	98.304	49.152	31.45728
12	196.608	98.304	62.91456
13	393.216	196.608	125.82912
14	786.432	393.216	251.65824

Figure 4.3: Beacon Interval Times/Super Frame Durations (in seconds) for available BeaconOrder and SuperFrameOrder settings.

A node computes its backoff delay based on a random number of backoff pe- riods, and performs two Clear Channel Assessments (CCAs) before accessing the medium. The operation of the slotted CSMA/CA backoff algorithm depends on the values of three variables:

- The Backoff Exponent (BE) is used in the computation of the backoff delay. Namely, the backoff delay is a random variable between 0 and $(2^{BE}$ 1).
- The Contention Window (CW) represents the number of backoff periods during which the channel must be sensed idle before accessing to the channel. The standard set the default initialization value to CW = 2 (corresponding to two CCAs). The CCA is performed during the first 8 symbols of a backoff period.
- The Number of Backoff s (NB) represents the number of times the CSMA/CA algorithm has attempted to access the channel for the current transmission. This value is initialized to zero (NB = 0) before each new transmission attempt and compared with a threshold to determine whether the algorithm can start a new backoff or it has to abort the transmission.

Figure 4.4 presents the flowchart of the slotted CSMA-CA algorithm.

1. First, the number of backoff s and the contention window are initialized ($NB = 0$ and $CW = 2$). The backoff exponent is also initialized to $BE = 2$ or $BE = min(2, macMinBE)$ depending on the value of the MAC attribute $BatteryLifeExtension$. $macMinBE$ is a constant defined in the standard, which is by default equal to 3.

2. Then, the algorithm starts counting down a random number of backoff periods (BPs) uniformly generated within $[0, 2^{BE} - 1]$. The countdown must start at the boundary of a BP.

3. When the timer expires, the algorithm performs one CCA operation at the BP boundary to assess channel activity.

4. If the channel is busy, CW is re-initialized to 2, while NB and BE are incremented. BE must not exceed $aMaxBE$ (which is 5 by default). If the maximum number of backoff s ($NB = macMaxCSMABackoffs = 5$) is reached, the algorithm reports a failure to the higher layer, otherwise, it goes back to step 2 and the backoff operation is restarted.

5. If the channel is sensed as idle, CW is decremented. The CCA is repeated if $CW = 0$. Performing two CCA operations aims to prevent potential collisions of acknowledgment frames. If the channel is again sensed as idle, the

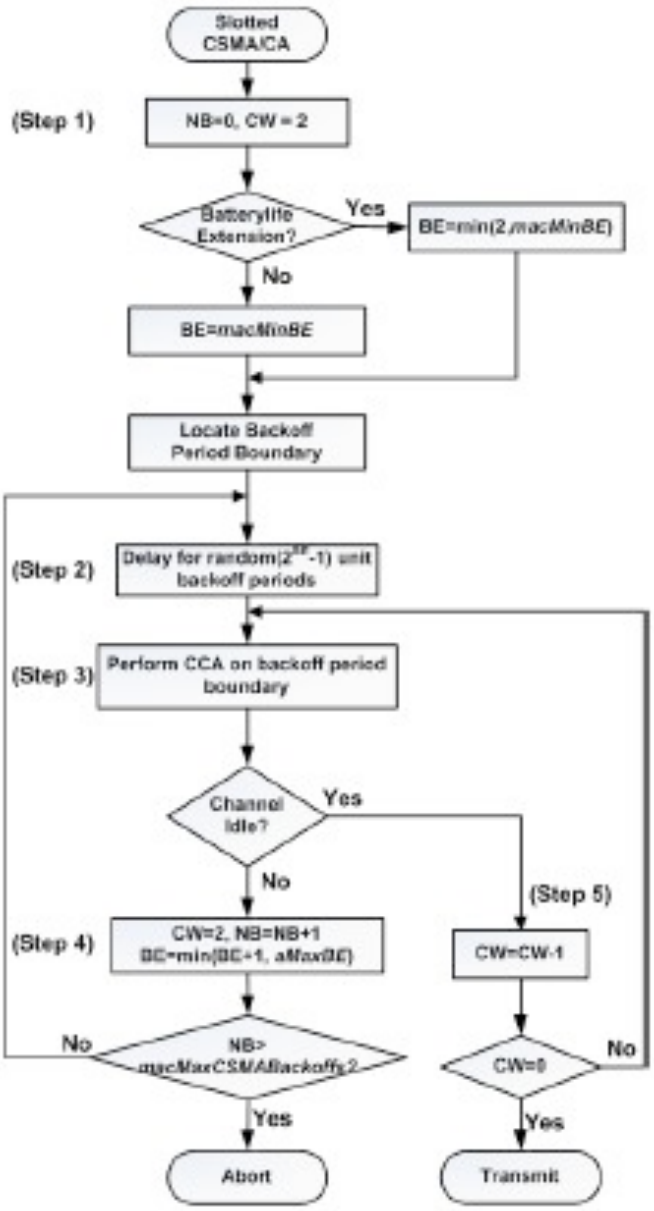

Figure 4.4: The slotted CSMA-CA algorithm.

node attempts to transmit, provided that the remaining BPs in the current CAP are sufficient to transmit the frame and the subsequent acknowledgment. If not, the CCAs and the frame transmission are both deferred to the next superframe. This is referred to as CCA deference.

During the contention free period, which, may last up to seven active period slots, devices are allocated Guaranteed Time Slots (GTSs) by the PAN coordinator. During a GTS a device has exclusive access to the channel and does not perform CSMA-CA. During one of these GTSs, a device may either transmit data to or receive data from its PAN coordinator. All GTSs must be contiguous in the CFP and are located at the end of the superframe active period. A device may disable its transceiver during a GTS designated for another device in order to conserve energy.

4.2 Performance of IEEE 802.15.4

4.2.1 Performance of the IEEE 802.15.4 Physical Layer

The IEEE 802.15.4 supports two PHY options. The 868/915MHz PHY, known as low-band, uses binary phase shift keying (BPSK) modulation, whereas the 2.4GHz PHY (high-band) uses o set quadrature phase shift keying (OQPSK) modulation. Both modulation modes offer extremely good bit error rate (BER) performance at low Signal-to-Noise Ratios (SNR). Figure 4.5 compares the performance of the IEEE 802.15.4 modulation technique to Wi-Fi and Bluetooth. The graph clearly illustrates that IEEE 802.15.4 modulation is anywhere from 7 to 18 dB better than the IEEE 802.11 and IEEE 802.15.1 modulations , which directly translates to a range increase from 2 to 8 times the distance for the same energy per bit, or an exponential increase in reliability at any given range.

When referring to communication reliability in terms of PER (Packet Error Rate), it is common to identify three different reception regions in a wireless link: connected, transitional and disconnected. While the connected and disconnected regions are characterized by, respectively, near to error-free reception and total absence of connectivity, the transitional region is characterized by high variances in the reception rates and by asymmetric connectivity. Being able to estimate the boundaries of the transitional region is important when planning the deployment of a sensor network, in order to make sure that most communication between nodes will occur within the connected region. Conversely, when the applications do not allow to plan the deployment of the nodes, protocols will have to cope with unreliable communications and adopt all the necessary measures. Figure 4.6 from [31] shows the packet reception rate (PRR=1-PER) vs. distance for an off-the-shelf receiver in a real indoor and outdoor environment.

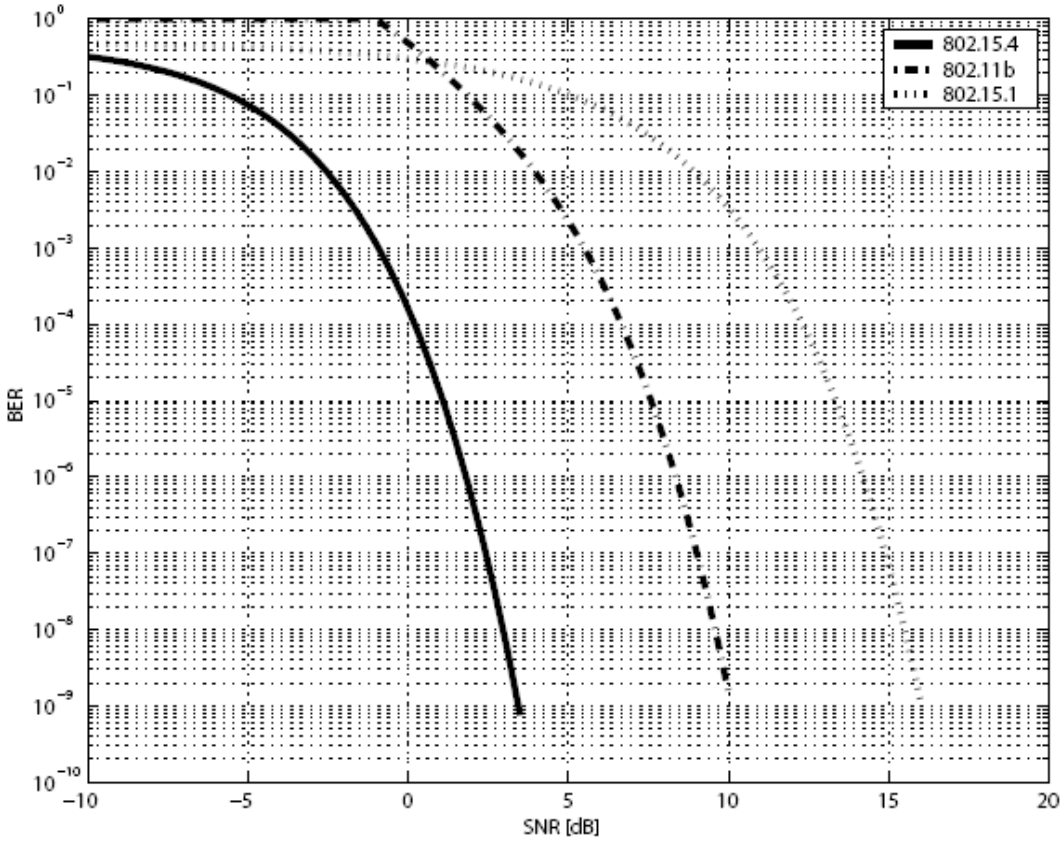

Figure 4.5: Theoretical bit error rate in an AWGN channel for IEEE 802.15.4, IEEE 802.11b and IEEE 802.15.1

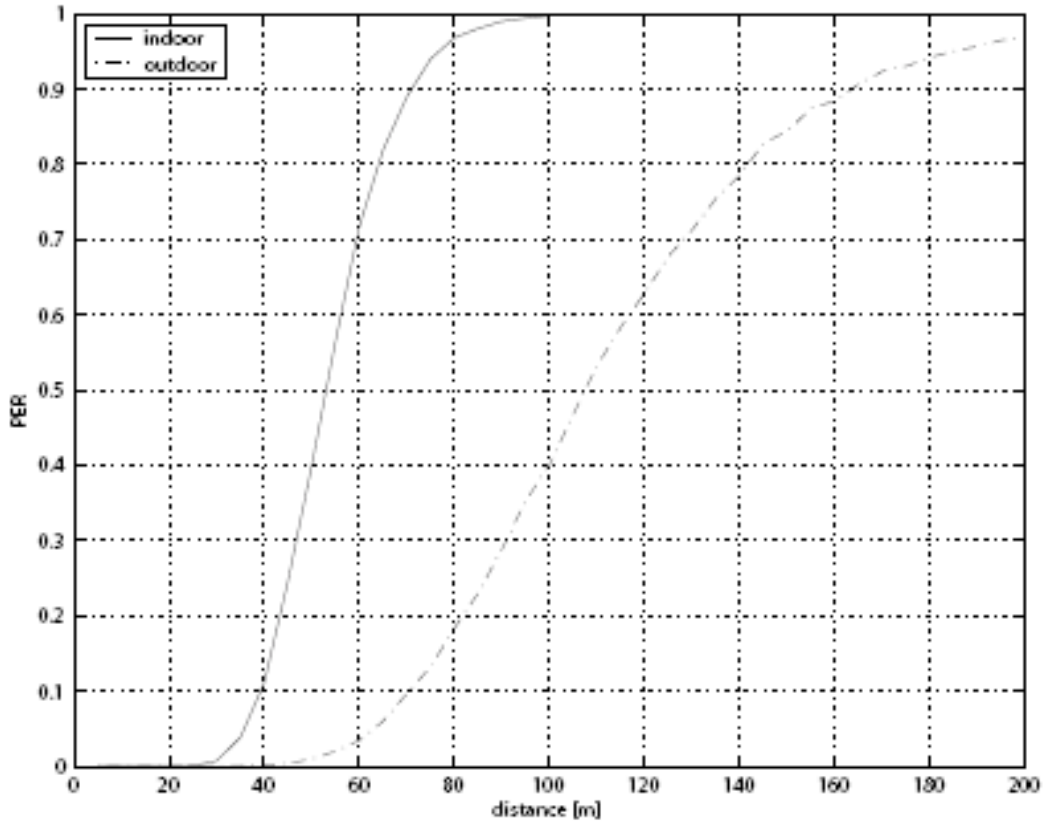

Figure 4.6: Transitional region for in a real indoor and outdoor environment.

The outdoor environment exhibits a rather large transitional region, resulting in highly unpredictable channel behavior.

A way to investigate how reception errors are temporally correlated is measuring the run lengths distribution, where a run is defined as a sequence of error-free receptions. The results discussed in [31] and reported in Figure
4.7, show how the Complementary Cumulative Distribution Function (CCDF) of the run lengths for an indoor environment, within the connected region (i.e. where the channel exhibits good reliability), can be very well reproduced with an independent (Bernoulli) and two-state Markov model. Similarly the two models can be used to describe the behavior in the outdoor environment for distances up to 20m, where the PER is less than 1%.

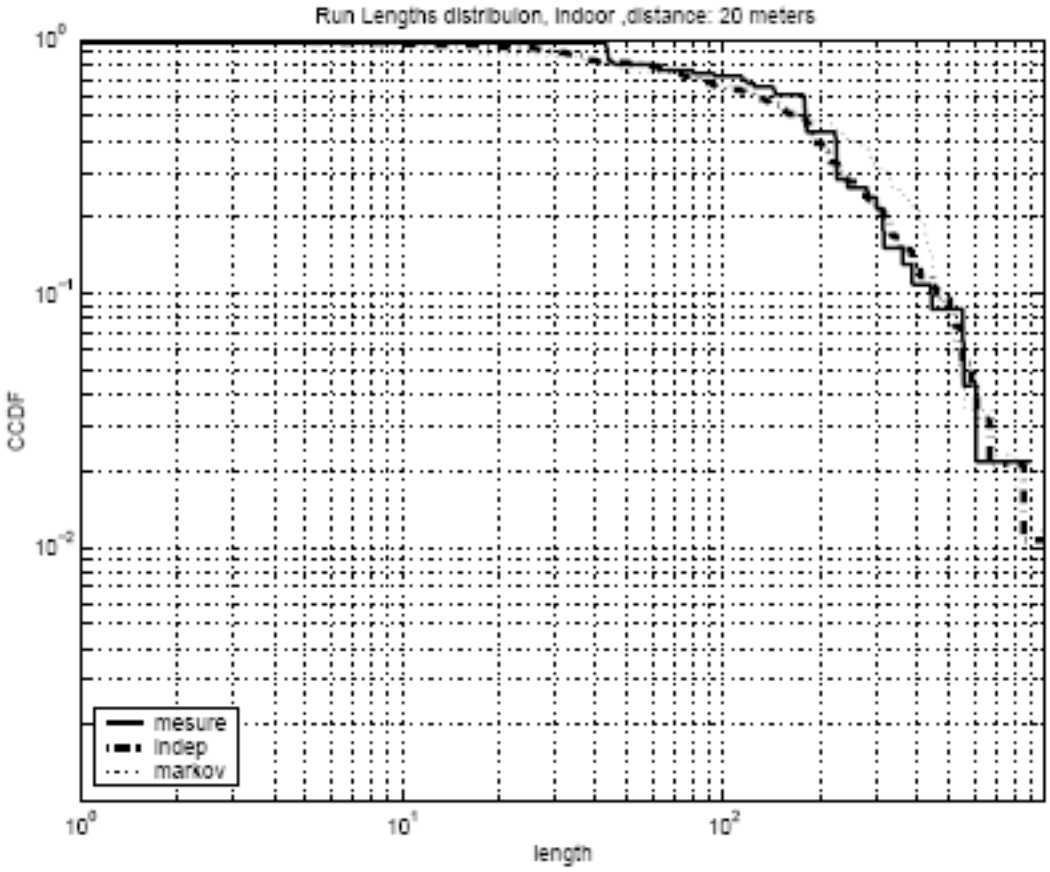

Figure 4.7: Run lengths distribution, indoor, 20 m.

4.2.2 Performance of the IEEE 802.15.4 MAC Layer

A performance study using simulation is presented in [19], where the impact of the IEEE 802.15.4 MAC attributes (BO, SO, and BE) on the performance of slotted CSMA/CA is addressed. Performance is studied in terms of throughput, average delay and success probability.

The study is carried out using the OPNET simulator, adopting Free Space as the path loss model and no hidden-node problems. A surface of (100 m x 100 m) with one PAN coordinator and 100 identical nodes is considered. Nodes are randomly spread and generate Poisson distributed arrivals, with the same mean arrival rate and constant size. The PAN coordinator periodically generates beacon frames according to the BO and SO parameters, with $SO = BO$ unless otherwise specified. Unacknowledged transmissions are considered, and the slotted CSMA/CA attributes are set to their default values (i.e. CW = 2, macMaxCSMABackoffs =5
and macMinBE = 2).

The performance of the slotted CSMA/CA mechanism is evaluated as a function of the offered load G in the network. Among the performance metrics analyzed in this study, the following will be considered here:

- Network Throughput (S), defined as the fraction of traffic correctly received by a network analyzer operating in promiscuous mode, normalized to the overall capacity of the network (250 kbps).
- Average delay (D), that is the average delay experienced by a data frame from the start of its generation by the application layer to the end of its reception by the analyzer.

Impact of SO and BO

Figure 4.8 shows the network throughput for different values of SO (and BO = SO) as a function of the offered load G. Low SO values produce lower network throughput, mainly due to two factors. First, the overhead of the beacon frame is more significant for lower SO values, since beacons are more frequent. Second, shorter superframes make CCA deference more frequent, which leads to more collisions at the start of each superframe. An increase in the superframe order from SO equal to 0 up to 3 has a considerable impact on the network throughput, as the probability of simultaneous CCA deference in multiple nodes decreases. Further increments of SO have little to no impact on the network throughput, as the probability of deference is already quite low. It can be noticed that, for high offered loads, the network throughput reaches a stable saturation throughput (around 62%).

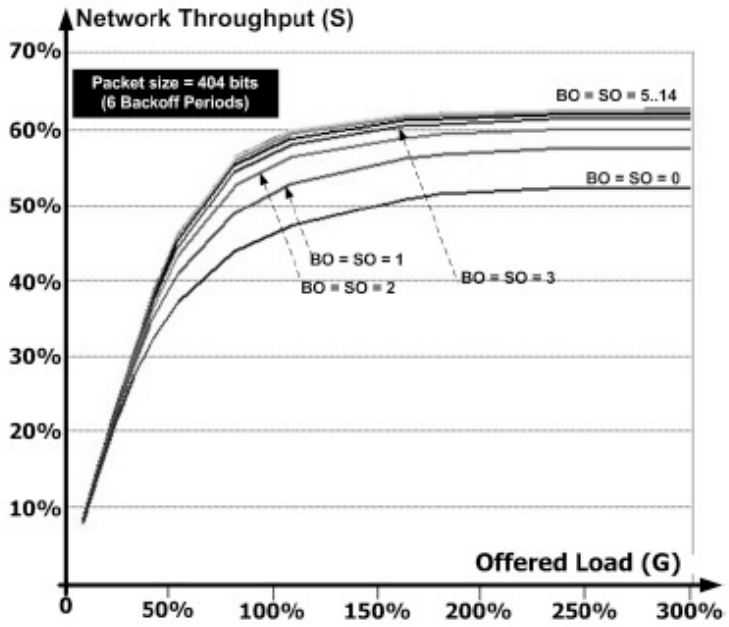

Figure 4.8: The network throughput as a function of the offered load for different BO and SO values.

In Figure 4.9 the effect of different BO and SO values on the average delay is shown. For offered loads larger than 50%, the results show an increasing trend of the average delay as BO and SO increase. This trend can be explained considering the impact of the CCA deference. CCA deference causes the last portion of the superframe to be unused with high probability, hence nodes performing the CCA find an idle channel, even when there are other nodes willing to transmit. Ac- cording to the MAC algorithm, these nodes reset the backoff exponent and try to transmit at the beginning of the next superframe. As mentioned above this may cause collisions more frequently, but the access delay is kept shorter. Conversely, when the superframe size is larger, the effect of CCA deference is less significant, and nodes performing the CCA find the channel busy with a probability that is more closely related to the offered load, and less influenced by the occurrence of the CCA deference. Accordingly, for high traffic loads, nodes will increment the backoff exponent BE and will wait for longer periods of time. In some sense, the backoff algorithm operates more properly, reducing the collision rate, but, on the other hand, increasing the access delay.

Figure 4.10 shows the impact of SO and BO for values of the offered load less than 50%. It can be seen that larger values of SO and BO result in larger delays. The explanation of this behavior is that with low offered loads, backoff s occur more rarely, and the additional delay in which nodes that perform CCA deference incur

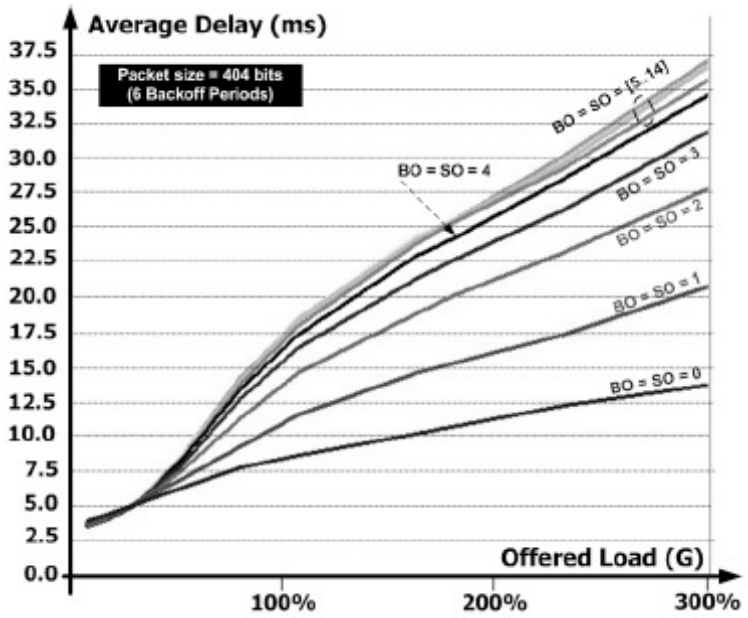

Figure 4.9: The average delay as a function of the o ered load for di erent BO and SO values.

becomes more significant.

Impact of *macMinBE*

The Backoff Exponent (BE) used in the computation of the random backoff delay before trying to access the channel, is set to an initial value, denoted as $macMinBE$, at the beginning of the backoff algorithm. $macMinBE$ is 3 by default, but can be set differently by the MAC sublayer in the range $[0, 5]$. When $macMinBE = 0$ the collision avoidance is disabled during the first iteration of the algorithm.

In Figure 4.11 it is observed that the network throughput is completely independent from the initial value of the backoff exponent macMinBE. This result is due to the fact that, in the considered 100-nodes scenario, the probability that a medium is busy is high, which leads to nodes increasing BE following negative CCAs. The backoff interval will tend to [0,31] in all the nodes waiting to access the medium and, as a result, the backoff delay distribution will not depend too much on the initialization value of mac MinBE.

The results in Figure 4.12 show that the average delay increases with macMinBE for a given offered load. For low offered loads (G 50%), the variance of the average delays for different macMinBE is not significant, whereas, for high offered loads (G 50%), the impact of macMinBE is significantly more visible.

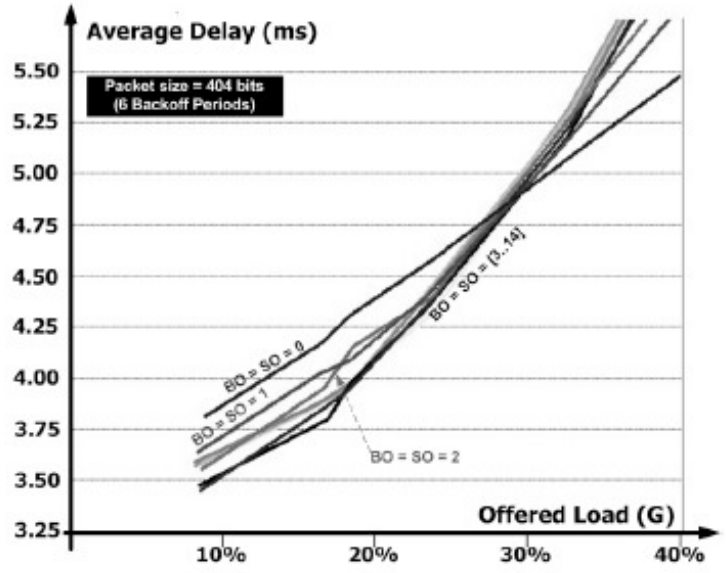

Figure 4.10: The average delay as a function of the offered load for different BO and SO values, and $G < 40\%$

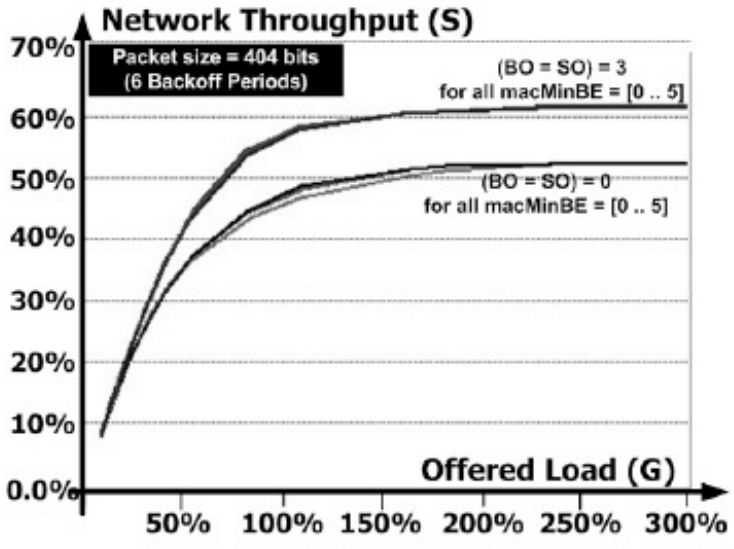

Figure 4.11: The network throughput as a function of the offered load for different values of $macMinBE$.

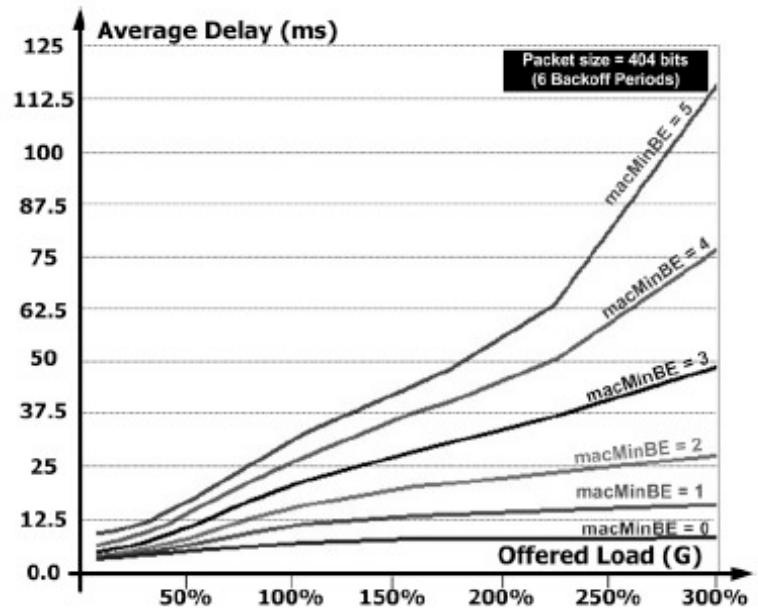

Figure 4.12: The average delay as a function of the o ered load for di erent values of $macMinBE$.

Chapter 5

Cooperative Reliable Communication in Cluster Tree IEEE 802.15.4 Wireless Sensor Networks

5.1 Introduction

This chapter describes a network protocol for enhanced robustness in IEEE 802.15.4- based sensor networks, which also addresses typical MAC Layer issues, including power management, synchronization and link reliability.

The characteristics of the wireless channel, the non-negligible probability of node malfunctioning, as well as battery power depletion are likely to introduce highly dynamic topologies and demand both adaptiveness and self-configurability from the network. In such specific scenario the traditional "pure" layered approach is not fully suitable; recent research has instead focused on the design of algorithms that exploit a higher degree of integration between layers. On one hand, Medium Access Control (MAC) protocols, which traditionally manage power saving, are designed to be application-aware to some degree; for example, they may provide service differentiation for data, query and management packets; on the other side, network protocols and applications may in turn be aware of power-management, for instance by taking sleep/listen schedules into account.

The Network Layer protocol discussed here uses a single-path strategy in error- free scenarios and resorts to using alternative paths when communication errors are detected. It exploits implicit acknowledgment of reception, a feature which may be provided by data aggregation when a broadcast medium such as the wireless channel is used. Therefore, MAC Layer acknowledgments are not used and errors

recovery relies on a caching and retransmission strategy. The protocol requires synchronization among the nodes, which also allows the implementation of power saving techniques such as sleep/listen schedules.

The performance of the proposed approach is evaluated through simulations, in which the overall network reliability is studied and the energy requirements are quantified, with different network sizes and protocol parameters.

In the remainder of this chapter, Section 5.2 describes the proposed reliable data gathering protocol whose implementation over IEEE 802.15.4 is detailed in 5.3. Simulation results are finally discussed in Section 5.4, while Section 5.5 presents conclusions.

5.2 PERLA

Based upon the ideas presented in [16] a network protocol has been developed called PERLA (Power Efficient Routing with Limited Latency). PERLA relies on IEEE 802.15.4, it takes power management into account and addresses some specific issues related to the adoption of this standard such as synchronization among nodes. The protocol relies on a spanning tree for ordinary routing operations, and resorts to exploit alternative paths only when a malfunctioning is detected.

5.2.1 Scenario and Motivations

Typical causes of errors in a WSN are failures in links or nodes. The former occur when a transmitted packet is not correctly received by the recipient; they are mainly caused by channel errors and collisions, or secondarily by wrong synchronization between the sleep/listen schedules of the nodes. Generally they are characterized by a temporary nature and they are not explicitly handled by the Network Layer. Node failures, on the other hand, have a permanent nature and may be caused by malfunctioning, battery depletion or other environmental factors; they introduce dead routes that need to be detected by the routing layer in order to provide the necessary changes in the topology.

During the time elapsed for the process of node failures detection, routing tables are not consistent with the real topology, and data will likely incur in partial or total loss. Although latency may not be a primary concern for all sensing applications, it is desirable that the network timely reacts to permanent failures that generate topology changes. Increased responsiveness of the routing layer protocol would address this issue, but might cause excessive fluctuations when repeated link failures, which are very common in highly populated WSNs and interfered environments, are mistaken for a node failure.

PERLA specifically targets link failures, while relying on a traditional counter- based approach for handling node failures.

If the Network Layer is capable of recognizing link failures, it may adopt and tune a specific procedure, thus avoiding the risk of overreacting with permanent route changes. Moreover the same measures may also be invoked when node failures occur, during the time that the new routes are established.

Instead of relying on link-layer acknowledgments and retransmissions, the protocol discussed here makes use of the implicit acknowledgment technique for link failure detection and implements a caching and retransmission strategy in neighbor nodes in order to provide an immediate recovery procedure. Alternative approaches for solving the problem of the wireless link unreliability could consist in delegating the matter to the MAC Layer, using acknowledgments and retransmissions, or in building on an unreliable link layer service and adopting a multi-path routing. With respect to a pure multi-path approach, the proposed retransmission strategy, while adding robustness to the protocol, involves fewer nodes, so that the overall traffic in the network is reduced. This leads to a better utilization of resources, especially when only a fraction of the nodes actually produces data and a considerable amount of nodes are involved only in the relaying process. PERLA exhibits some advantages also with respect to the adoption of link-layer acknowledgments, as will be discussed in Section 5.4.

5.2.2 Network Operation

During the network initialization phase, while growing the tree, nodes are assigned a level representing the hop-distance between themselves and the sink node, with higher levels typically corresponding to larger geometric distances.

Although data may be asynchronously generated by sensor nodes at different levels, they traverse the network with a defined timing, similarly to what described in Madden et al. (2002), and in [24]. As shown in Figure 5.1, it is assumed that the collection of data from all source nodes to the sink must be completed within a specific time, called *epoch* and indicated by e in the figure. Assuming that the tree depth is n, each level will be assigned a time e/n (also referred to as *sub-epoch*) to complete its transfer.

Figure 5.2 shows the detailed timing of the data forwarding process between adjacent levels. The sub-epoch is divided into eight *phases* during which different actions are performed. The sub-epochs corresponding to adjacent levels are shifted so that the correct phase coupling is achieved. The main transmission and reception phases, which are the only ones used in an error-free situation, are highlighted in dark grey and labeled as `TX` and `RX`. During the `RX` phase, nodes receive packets from the lower level. Data from own child nodes and data from others' child nodes are separately aggregated and cached. In an error-free scenario data coming from

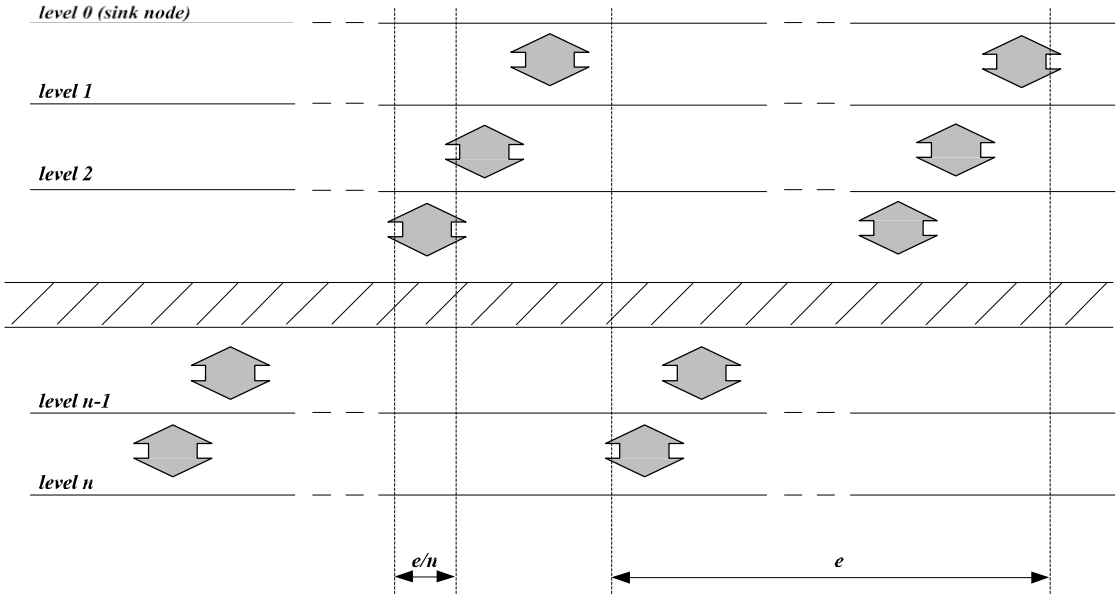

Figure 5.1: Data-transfer timing showing the epoch-based scheme.

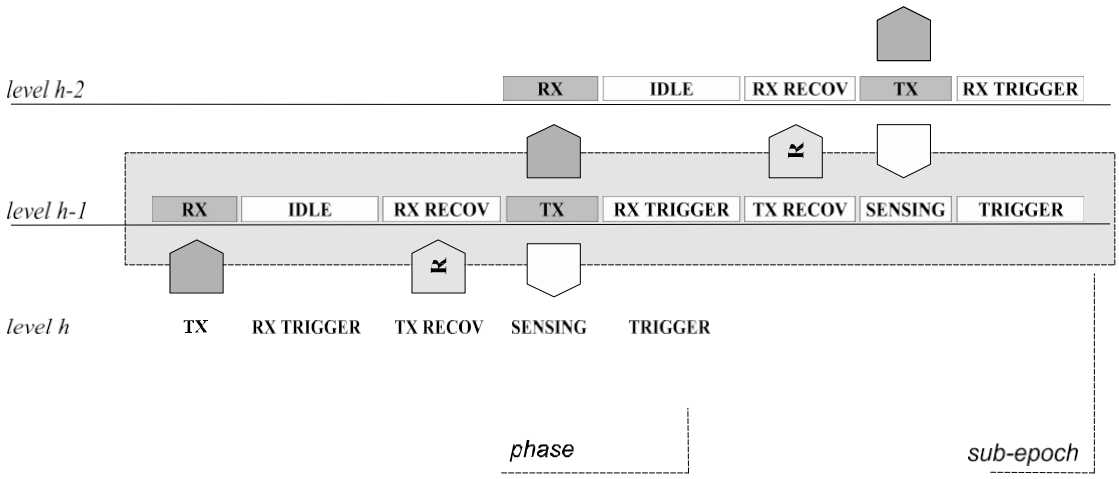

Figure 5.2: The organization of the phases of PERLA.

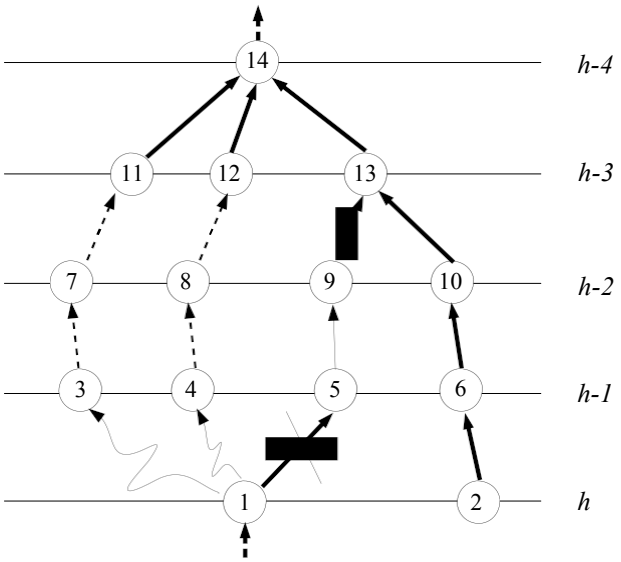

Figure 5.3: A retransmission scenario exploiting multiple alternative paths.

the child nodes are the ones which are forwarded during the following TX phase.

Error Recovery Procedure

Figure 5.3 shows a branch of the routing tree containing node 14 and its child nodes; nodes in the same row belong to the same level, indicated by the label on the right. Let us assume that node 1 collected data from its children and node
2 is the source of new data; both nodes are at level h and their data need to be routed toward the base station through node 14. Solid lines represent the path that packets would follow in an error-free situation.

Assume that, during the RX phase assigned to level $h-1$, node 6 correctly receives a packet from node 2, while node 5 experiences an error while receiving the packet from node 1; nodes 3 and 4 are able to overhear the packet sent by node 1 and to cache it, as indicated by the wavy lines. During the TX phase, nodes at level h 1 forward data based on packets received by their own child nodes. Since implicit acknowledgments are used, the sending nodes do not expect any acknowledgment for the sent packets. Note that when a node of level $h-1$ is in the TX phase, all of its child nodes at level h are performing a SENSING phase. Node
5 transmits its data packet, which does not contain any data originating from node
1. Node 1, upon sensing the channel, becomes aware that some fault happened and that a retransmission is needed; however it does not try to retransmit the packet but rather waits until the TRIGGER phase begins.

Afterwards, nodes 1 and 2 enter their TRIGGER phase, which corresponds to

Table 5.1: Superframe parameters

	RX	IDLE	RX-REC	TX		TX-REC	SENSING	TRIGGER
No of superframes	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2
macBeaconOrder	b_0	b_0	b_0	b_0	b_0	b_0	b_0	b_0
macSuperframeOrder	s_0	15	s_0	15	s_0	15	15	15
macPromiscuousMode	T	F	F	F	F	F	T	F

RX-TRIGGER for nodes 3-6. During this phase, node 1 broadcasts a trigger packet that enables nodes at level h 1 to act "on behalf of" node 5 and to forward the data they have previously overheard. Transmissions from nodes 3 and 4 take place during the upcoming TX-RECOVERY phase. As shown in Figure 5.2, when nodes in level h 1 are in TX-RECOVERY phase, nodes in level h 2 already entered their RX-RECOVERY phase and may thus receive the recovered packets. As already mentioned, recovered data may follow multiple paths, as indicated by dashed lines in Figure 5.3; data originating from node 1 and node 2 will reach node 14 during the sub-epoch dedicated to level h 4, thus terminating the recovery procedure.

5.3 An IEEE 802.15.4-based implementation

This Section describes how PERLA can be implemented over the IEEE 802.15.4 MAC Layer, without any modification to the standard.

IEEE 802.15.4 allows to control the duty cycle of the nodes as well as to limit link-layer retransmissions and to disable acknowledgments; in our implementation both retransmissions and acknowledgments are turned off.

PERLA is implemented on the peer-to-peer topology of IEEE 802.15.4 and works in beacon-enabled mode with the use of superframes in order to achieve synchronization between levels; the Contention Free Period of the IEEE 802.15.4 superframe is not used. PERLA also exploits the possibility of defining the number of children per node and of switching the devices to promiscuous mode as necessary.

5.3.1 Network setup

As already mentioned, upon initialization IEEE 802.15.4 nodes automatically organize themselves into coordinator-child relationships, forming a tree of MAC-Layer associations. At the Network Layer, nodes also need to select a parent and to establish the level to which they belong within the routing tree. In PERLA, a node chooses its MAC Layer coordinator as parent at the Network Layer. The base station and the nodes which have already selected a parent periodically broadcast

PERLA management packets, containing their level in the tree and their MAC address. A non-configured node collects these packets and determines its own level and its parent's Network Layer address by selecting the packet sent by its MAC Layer coordinator. Hence, once the setup is completed, the routing tree matches the tree of MAC Layer associations. The topology of the latter may be controlled by the upper layer according to predefined criteria, so the described approach does not represent a limitation; moreover it allows the nodes of a certain level to be driven by parent nodes through the beaconing process.

It is worth noting that since the association procedure requires a bidirectional packet exchange, only symmetric links are selected as is also required for the implicit acknowledgment mechanism. Finally, in order to control the tree branching factor, fine-tuning of the *macAssociationPermit* IEEE 802.15.4 parameter is performed.

5.3.2 Synchronization and phases

Each of the phases depicted in Figure 5.2 has been implemented using multiple superframes, gathered in groups sharing the same settings. Phase synchronization is achieved through the beaconing process and the transmission of Beacon Sequence Numbers (BSNs) within the beacons: each phase is assigned a range of BSNs and the nodes determine the current phase for their level by reading the beacons they periodically receive from coordinators. When a node receives a beacon, it waits for a small random amount of time and transmits a beacon in turn, whose BSN value is obtained by increasing the received BSN by an opportune o set. The o set accounts for the shift of the phases between adjacent levels, while the random delay has been introduced in order to reduce the probability of collisions among beacons. However, since collisions may still occur, a support timer is adopted in order to compensate for missing beacons. Synchronization has finally been reinforced by disabling the transmission of packets in the first and the last superframes of each phase.

Table 5.1 summarizes the IEEE 802.15.4 parameters that are a ected by PERLA settings. *macBeaconOrder* (BO) and *macSuperframeOrder* (SO) values, which are transmitted in each beacon, respectively determine the time period between two successive beacons and the duration of the active portion of the superframe. A reserved value for SO (15) indicates an *inactive* superframe and is used to pre- vent child nodes from transmitting, as for instance during the `IDLE` and `TX` phases. During all other phases it is possible to control the duty-cycle of the nodes by statically or adaptively setting SO to an actual value indicated by s_0; varying s_0 is functionally transparent with respect to PERLA.

The *macPromiscuousMode* parameter must be set to *true* whenever the MAC Layer is required to forward packets to the PERLA agent regardless of the intended

recipient node. Finally, not all phases can have their durations set independently; namely, phases which are constrained to occur simultaneously must share the same duration, thus allowing at most three independent durations n_1, n_2 and n_3.

5.3.3 Power management

Nodes may save energy by regulating the duration of the Inactive period of the superframe through SO, and by controlling the *macRxOnWhenIdle* IEEE 802.15.4 parameter, which determines whether the radio is switched o when not transmitting or receiving packets. PERLA uses both mechanisms to put the nodes in a sleep state when out of the sub-epoch and in specific intervals during some phases.

5.4 Simulation Results

PERLA has been implemented as a module for the *ns-2* simulator [4] commonly adopted by the networking scientific community, using the IEEE 802.15.4 implementation provided with the standard distribution for version 2.29. Chapter 7 provides more details about the implementation.

In order to model the energy requirements, the power consumption levels re- ported in [8] have been adopted. The energy model of *ns-2* does not completely support *macRxOnWhenIdle*, and it considers the radio switched o only during the `IDLE` phase and out of each sub-epoch; therefore *ns-2* reports a higher power consumption than PERLA actually needs.

The energy required by PERLA may be controlled acting on the durations of the `TX-RECOVERY` and `TRIGGER` phases; however, the shorter the duration of a phase, the lower the probability that all nodes will complete the transmissions in time. In the considered simulations, phase durations were always set to be longer than the time needed for all nodes to complete their transmissions, so that an upper bound to the performance offered by PERLA could be evaluated regardless of energy constraints. Parallel simulations were performed using the computing facilities available on the grid system of the Department of Computer Engineering at the University of Palermo.

Performance of a sensor network strongly depends on the chosen MAC/PHY layers; moreover, in the case of IEEE 802.15.4, it also depends on the specific settings for all customizable parameters. As, to my best knowledge, literature does not present any comprehensive description for a widely accepted routing layer which works with the peer-to-peer topology of IEEE 802.15.4 networks, no direct comparisons can be made to PERLA; in this analysis the performance of the proposed approach is compared to two alternatives, thus resulting in the following algorithms:

Table 5.2: Settings for the simulations.

Scenario	max children	b_0	s_0	n_1	n_2	n_3
1	5	2	2	12	4	4
2	5	2	2	12	4	4
3	2	2	2	12	4	4

a) cache-and-retransmit strategy (PERLA);

b) unacknowledged transmissions ("base");

c) link-layer acknowledgments and retransmissions ("ACK").

In all cases the data gathering model described in Section 5.2 has been adopted. For the last two algorithms the structure shown in Figure 5.2 has been simplified by considering only the `TX` and `RX` phases, while nodes are kept in sleep state during the other phases.

All simulations assume that nodes have a transmission range of 10 meters, which corresponds to an area $A_f \cong 314.16m^2$. Nodes are randomly placed according to a uniform distribution and are all assumed to be generating data. Node failures were not considered, and an ideal channel is assumed, so that channel errors do not occur; however link errors may still be present as there is the possibility of collisions.

Performance is measured in terms of reliability and connectivity. Reliability is defined as the ratio between the number of originators of the aggregate received by the base station and the total number of nodes. Connectivity is defined as the number of nodes which were able to join the network and synchronize. Moreover, the energy spent per epoch is measured. All the reported values are averaged on the overall number of epochs of the simulation.

The algorithms have been tested on three different scenarios; in the first one, the network contains 40 nodes and, in order to assess the behavior of the algorithms with respect to scalability, the results are compared with a second scenario where the number of nodes is increased to 100. In a third scenario, again with a 40 nodes network, the maximum number of children was limited to 2 per node, in order to study the influence of the branching factor.

Furthermore, a deeper insight into the internal operations of PERLA is pro- vided by analyzing the amount of trigger packets sent by the nodes at each level, in order to point out any potential weakness of the proposed strategy, and to suggest possible improvements.

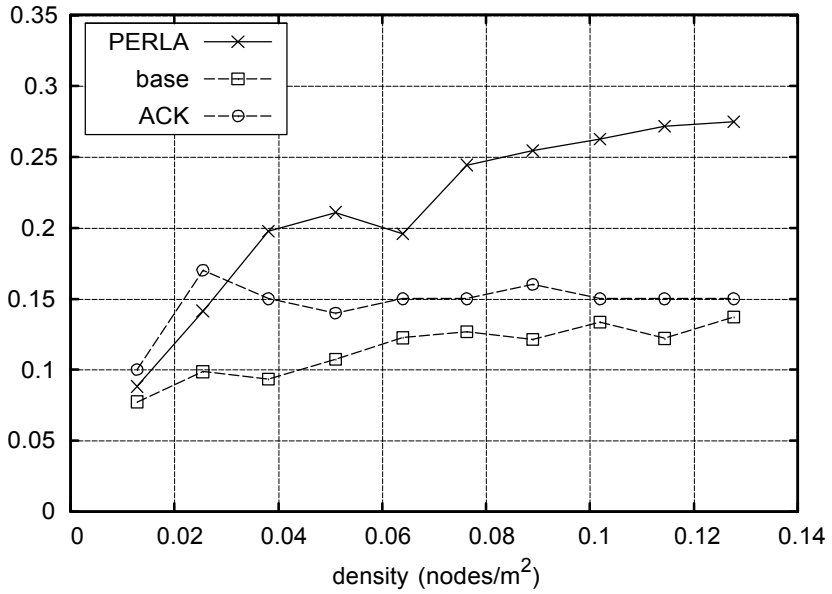

Figure 5.4: Reliability as a function of the nodes density, in the 40-node scenario.

5.4.1 Performance evaluation

Figure 5.4 shows the results in the 40 nodes scenario. The reliability is plotted against the density of nodes, in a range from 4 to 40 nodes per area A_f.

The performance reached by algorithm "base" is very poor; it just slightly increases with the density of nodes, but never reaches 15%. This unsatisfactory behavior is due to the high amount of collisions, while the trend may be explained with a lower number of levels in the routing tree as the density increases. When acknowledgments and retransmissions are used (algorithm "ACK") the reliability improves and is on average higher than 15%, not showing significant variations with the density. The performance of PERLA is remarkably better than with the other two approaches with a reliability in a few cases beyond 25%, that is twice over "base". The highest values are achieved for high density, since there is a higher degree of cooperation among the nodes. The advantage of PERLA over "ACK" is justified by two main factors: while MAC-Layer acknowledgments and retransmissions contribute to raise the number of collisions, PERLA retransmissions are scheduled in a separate, dedicated phase and do not negatively affect the first transmissions; in addition, PERLA retransmissions do not take place at the tree depth in which the link failure occurred, instead they exploit the fact that the packet may have been cached by nodes closer to the base station and start from there. Figure 5.5 shows that the connectivity is always over 80%, with PERLA performing slightly better. In Figure 5.6, the energy spent per epoch is reported;

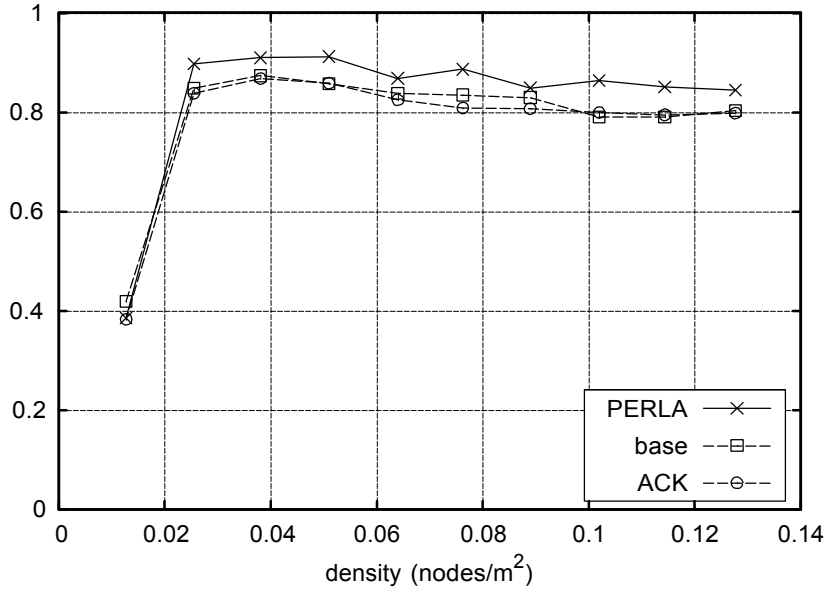

Figure 5.5: Connectivity as a function of the nodes density, in the 40-node scenario.

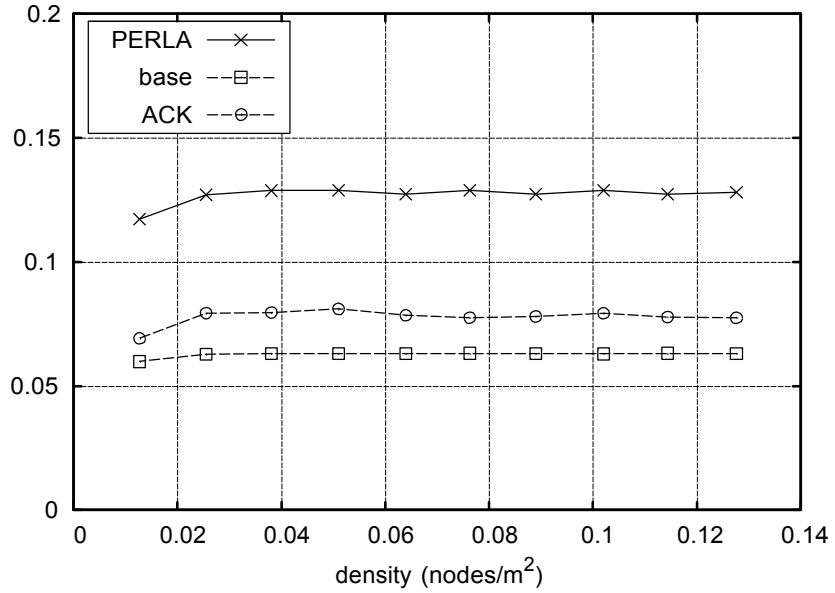

Figure 5.6: Energy consumption as a function of the nodes density, in the 40-node scenario.

Reliab

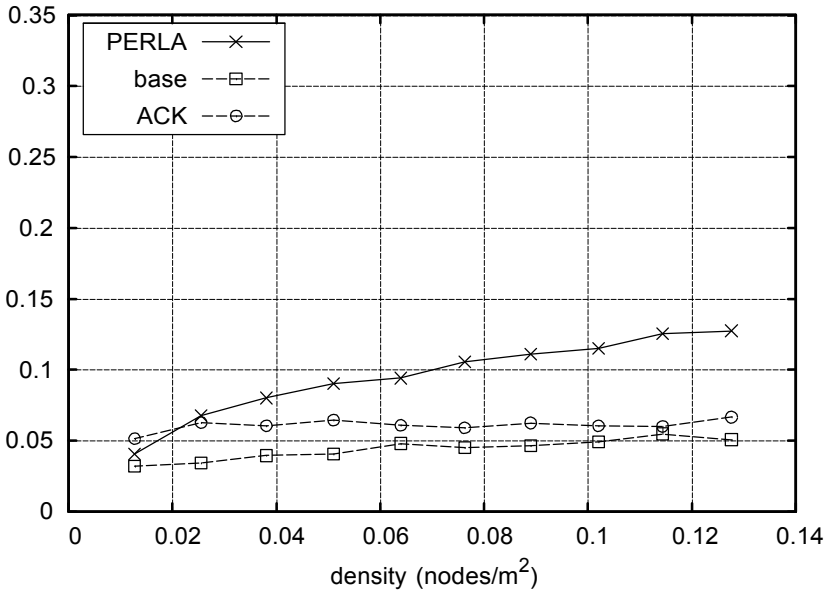

Figure 5.7: Reliability as a function of the nodes density, in the 100-node scenario.

the doubling of the reliability shown in Figure 5.4 may be obtained by PERLA with less than twice the energy of "base".

In Figures 5.7, 5.8 and 5.9 the results from the second scenario are displayed. The number of nodes has been increased up to 100, and the network field resized in order to leave the density unchanged. The reliabilities, shown in Figure 5.7, are reasonably lower than the previous scenario, while the same qualitative behavior may be observed. A reduction of about 10% on the values of connectivity is visible in Figure 5.8. The levels of energy, which are reported in Figure 5.6, do not show variations since all the three approaches are essentially independent of the network size as far as energy is concerned.

In the last scenario, it is shown how the branching factor of the tree affects the performance of the network. A network of 40 nodes is considered and the limit to the maximum number of children a node can accept is changed from the default value of 5 to 2. This case shows the highest possible number of levels, and this negatively affects the reliability, since more hops are required. However the number of nodes that simultaneously access the channel is the lowest, and the number of collisions drops. In Figure 5.10 it can be seen that the two factors result in overall reliability enhancement with respect to scenario 1. Performance of "ACK" is comparable to that of PERLA, with the latter performing slightly better at higher densities. For all algorithms, connectivity and energy consumption do not appear to be significantly affected by the branching factor.

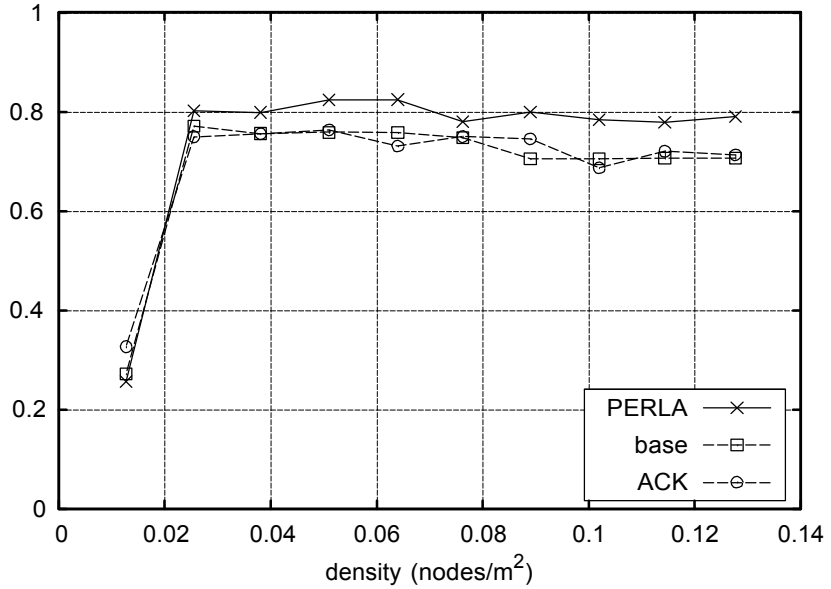

Figure 5.8: Connectivity as a function of the nodes density, in the 100-node scenario.

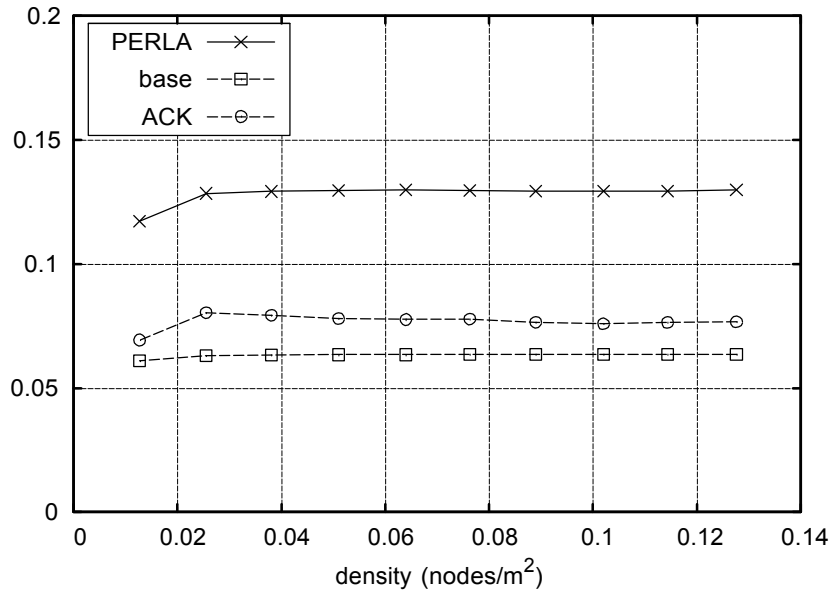

Figure 5.9: Energy consumption as a function of the nodes density, in the 100-node scenario.

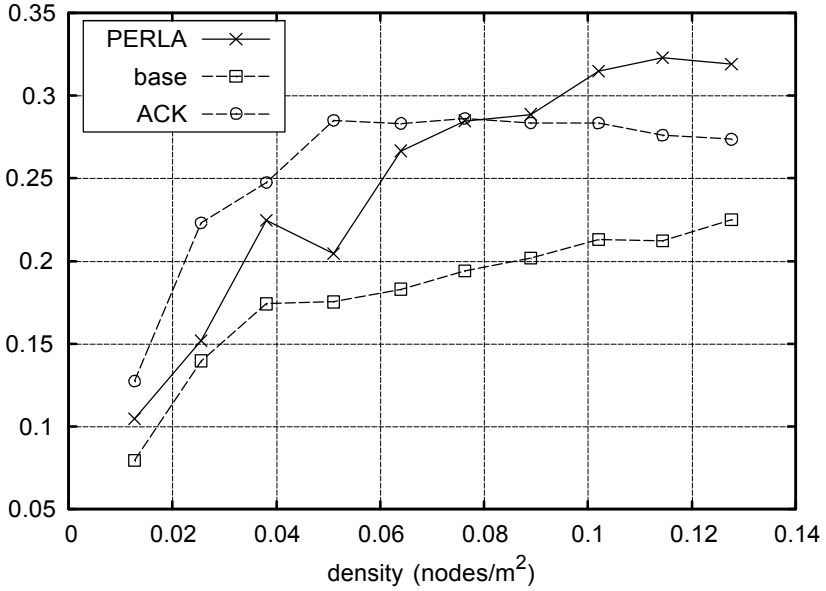

Figure 5.10: Reliability as a function of the nodes density, in the 40-node scenario, with a maximum number of children set to 2.

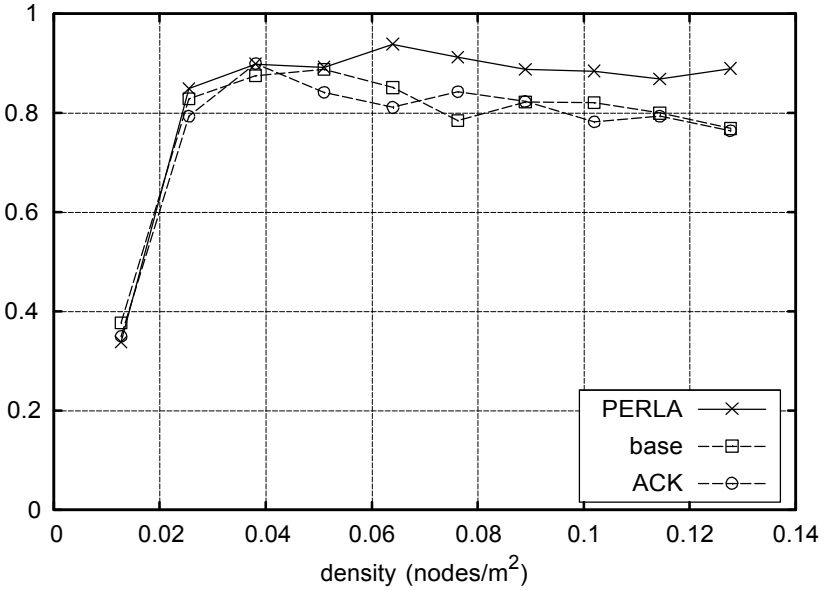

Figure 5.11: Connectivity as a function of the nodes density, in the 40-node scenario, with a maximum number of children set to 2.

Energ

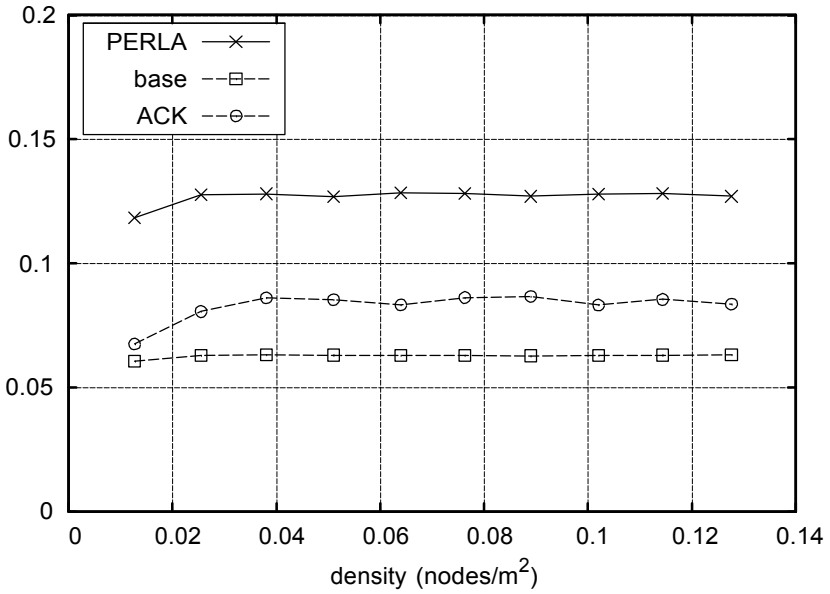

Figure 5.12: Energy consumption as a function of the nodes density, in the 40-node scenario, with a maximum number of children set to 2.

5.4.2 Analysis of per-level trigger probabilities

The *h*-level trigger probability is defined as the ratio of the amount of trigger packets sent by all nodes belonging to level *h* during one epoch, and the total number of those nodes, averaged on the overall number of epochs. This quantity gives an indication of the probability of successful transmission during the `TX` phase. The lower the trigger probability, the higher the probability that a packet from level *h* is correctly received by the intended recipient at level *h-1*, without the need for retransmissions.

Figures 5.13, 5.14, 5.15 show the trigger probabilities for the three considered scenarios, for the first 20 levels, when using PERLA. Ten curves are plotted in each figure, representing the trigger probabilities for increasing node densities. Densities are expressed as multiples of *d*, which is the lowest value considered in the simulations.

Figure 5.13 shows that the trigger probability is largest at the lowest levels of the network, whereas it decreases as the hop distance from the base station increases, becoming almost null after level 14. Clearly, the trigger probability is zero for the first two levels, which represent the base station and its child nodes, that do not send trigger packets. The observed behavior is in accordance with the intuition that nodes tend to fill the levels closest to the base station with higher probability, hence a larger number of nodes try to access the channel simultaneously.

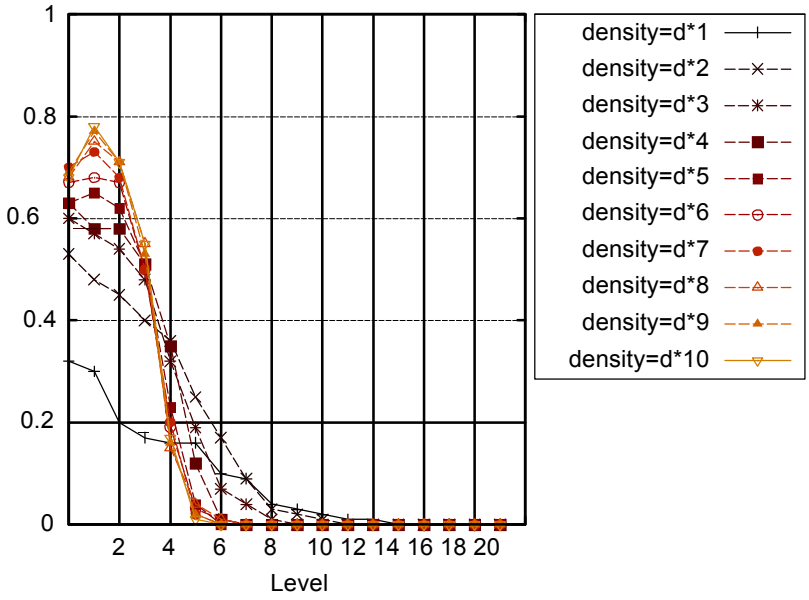

Figure 5.13: Per-level trigger probability for Scenario 1.

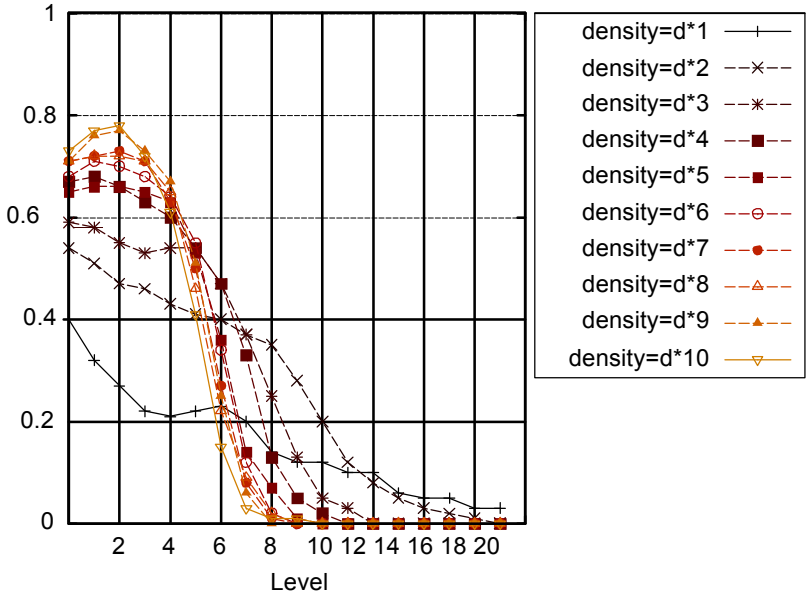

Figure 5.14: Per-level trigger probability for Scenario 2.

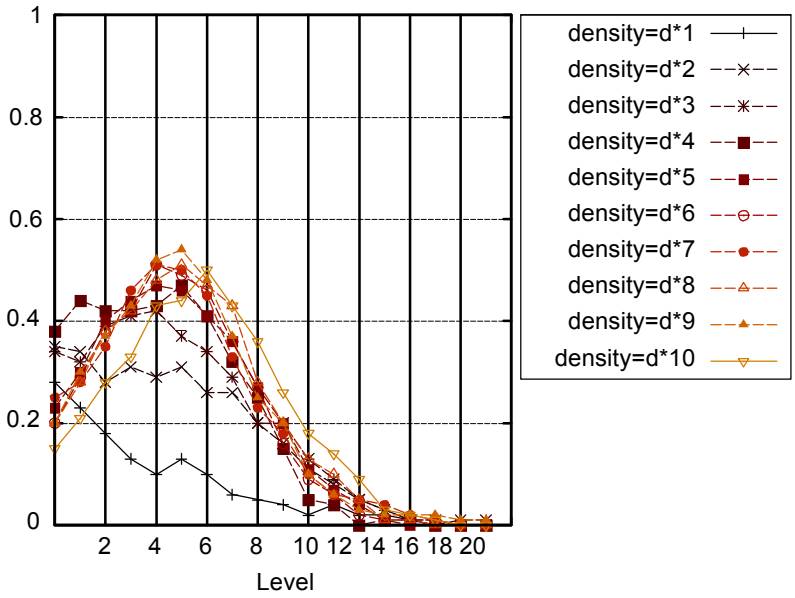

Figure 5.15: Per-level trigger probability for Scenario 3.

Note that, since the branching factor is set to 5 in Scenario 1, up to 25 nodes may belong to the second level. Moreover, as density increases, the trigger probability decreases at the lowest levels, whereas it increases at the highest ones; this trend can be interpreted as an effect of the distribution of nodes among the levels, as well.

Figure 5.14 refers to Scenario 2, which considers a network with 100 nodes. It shows a similar trend; nodes send trigger packets with high probability up to levels 6-8, which contain more nodes than in Scenario 1.

In Figure 5.15 the effect of reducing the branching factor to 2 can be analyzed. In most cases the peaks of the curves are located between levels 6 and 8, and the values of trigger probability are lower than in Scenario 1, in accordance with the increased reliabilities observed in Figure 5.10.

Lowering the branching factor has the side-effect that nodes spread through a larger number of levels; since the epoch duration needs to be dimensioned accordingly, this affects the minimum allowed latency. Finally, from these results it can be inferred that an improvement of the performance of PERLA can be achieved by providing a lower amount of contention, especially at the lowest levels, where a collided packet causes the loss of a larger amount of information. Currently, a technique that adaptively limits the branching factor is under study.

5.5 Conclusions

Additional tests have shown that a considerable amount of collisions is caused by the beacons, which are transmitted without CSMA. It would be desirable to reduce the number of beacons, and to adopt longer superframes; however, beacons are necessary for the process of synchronization, and this might weaken its robustness, therefore on-going research is being conducted to address this issue. Moreover, phase durations dimensioning for optimized power consumption is under study, as well as suitable settings of IEEE 802.15.4 parameters for better performance, including the ones influencing the branching factor.

Finally, different portions of this work have appeared in publications [27],[28],[29].

Chapter 6

Application Controlled Collision Avoidance over IEEE 802.15.4

6.1 Introduction

Wireless Sensor Networks differ from wireless ad-hoc networks because of intrinsic factors including hardware limitations and energy constraints. Typical network sizes pose scalability and manageability challenges, which are further problematic because of hidden-terminal issues, deriving from the limited radio ranges and the extension of the deployment fields. Moreover typical applications introduce specific characteristics in terms of traffic models and requirements [15].

In many applications, the utilization of the transceiver is recognized as the most energy-consuming activity. As a result, energy-efficient communication has been the subject of extensive studies. The MAC-layer protocol, which governs how multiple nodes access the wireless channel and provides functionalities such as collision avoidance, has a fundamental role in controlling the activity of the transceiver, hence many authors have researched solutions that involve the use of specifically designed, non-standard MACs.

As discussed in greater detail in Chapter 3, a general classification of sensor net- works MAC protocols can be done by distinguishing between random (or non deter- ministic) protocols, and scheduled protocols [21]. The former are less complex and can be fully distributed, hence they are generally more scalable. In addition, low complexity and the absence of shared information, or 'state', reduce memory and processing requirements as well as control overhead. Most non deterministic protocols are modeled after CSMA/CA, and exploit the in- formation that is directly available through the node radio, therefore being able to avoid collisions only at the sender's side [33]. The introduction of RTS/CTS control packets and virtual carrier sensing has been proposed to

specifically address the hidden terminal problem, however a common criticism is that such approach is not general, as it is based on the assumption of symmetric links, and cannot be applied to the case of broadcast transmissions. Access control can be integrated by a dedicated control channel [39], even though the cost and the power consumption of multiple transceivers often exceed the sensor networks constraints; solutions which exploit multiple paths can be very simple but do not generally perform well under high traffic loads [10]. Scheduled MAC protocols organize nodes for transmitting according to a common schedule and provide the capability of reducing energy waste due to collisions, *over-hearing* and *idle-listening*. Higher complexity, state information distribution, and synchronization overhead can contribute to raise power consumption, because of the resulting computational load and the exchange of control messages; additionally, depending on the complexity of the state information, some protocols also require non-neglectable use of memory space. Schedule maintenance is complicated by node mobility and failures, network segmentation, and incomplete information available at the nodes. Potential inconsistency of the MAC state among the nodes has to be taken into account during the design process in order to limit its effects on the network performance. Most scheduled proto- cols are based on TDMA, [35], [5], which is simpler than FDMA or CDMA and generally requires less expensive hardware. Protocols based on clustering, such as LEACH [17], can provide more scalability and reduced power consumption, as the MAC state has to be shared only locally. LEACH, however, has some relevant limitations, as it requires a complex radio and assumes each sensor to be within the radio range of the base station. A drawback that is common to all MAC-based approaches, which prevents their widespread adoption, is the incompatibility with existing devices. In the recent past the IEEE 802.15.4 standard for WSNs, which includes MAC and PHY-layer specifications, has attracted increasing attention. The diffusion of IEEE 802.15.4-based devices has motivated our research towards a different approach to the problems of multiple access and collision avoidance.

In this chapter a collision avoidance technique is presented which can be implemented on standard IEEE 802.15.4 networks, and extends the functionalities provided by the IEEE 802.15.4 MAC protocol. The proposed collision avoidance technique is distributed and characterized by low complexity, hence it is suitable to the limited computation resources of wireless sensor nodes, moreover it exhibits good scalability.

The reference scenario is a multi-hop WSN which runs a monitoring application, where nodes periodically generate packets directed to a data collecting center. This includes the network described in Chapter 5. The collision avoidance technique exploits the periodic nature of the monitoring traffic, and sets up a global periodic schedule of packet transmissions, which aims to avoid collisions. The computation

of a network-wide transmissions schedule can be a rather complex and resource demanding problem if performed in a centralized fashion (e.g. requiring knowledge of nodes position, radio-ranges, and so on). However an analogous, sub-optimal task can be performed in a distributed fashion with much less e ort. Accordingly, with the proposed collision avoidance technique, each node controls the instants of its own transmissions from the Application Layer, through the introduction of proper delays when passing packets from the Application to the MAC Layer. The Application Layer processes transmission successes and failures, and applies a decision algorithm in order to determine whether to change or keep the last used delay for the next transmission. In this way the global schedule is cooperatively updated and converges to an arrangement of packet transmissions that consistently reduces the amount of collisions. Two decision algorithms have been devised: the first one changes the adopted delay when the amount of consecutive transmission failures exceeds a configurable threshold; while the second one relies on a slightly more sophisticated and flexible filtering that operates on the past few transmission results.

In the following, a network model derived from the one presented in Chapter 5 is assumed as reference application, in order to provide an example of concrete implementation of the technique. As mentioned above, the collision avoidance technique processes past transmission events, hence it needs a mechanism to detect successes and failures. In the considered application, which does not make use of MAC-Layer acknowledgments and employs synopses diffusion for data aggregation, implicit acknowledgments can be used in order to detect successful transmissions or failures.

A similar approach has been adopted in [40], where the authors propose two techniques that exploit detected collisions in order to adapt the transmission rate and to avoid collision. The collision avoidance technique is based on the detection of large periods without packet receptions and on sending this information, within NACK packets, back to the sources, which use proper o sets to transmit during the suggested time. As opposed to the solutions proposed in [40], which apply to a scenario with single receiver node and have been evaluated in small-size single-hop networks, the ideas presented here apply to a multi-hop network with multiple sources and receivers and an extensive performance evaluation study is presented, in which the proposed algorithms have been tested in large networks, including 40 up to 200 nodes. As detailed in the following of this chapter, restricting this technique to single-hop networks is trivial, and a quantitative comparison with the techniques in [40] is not meaningful.

In the rest of this chapter, the considered communication model is reviewed in Section 6.2; Section 6.3 describes the proposed collision avoidance mechanisms, while Section 6.4 provides some details about the implementation of the system.

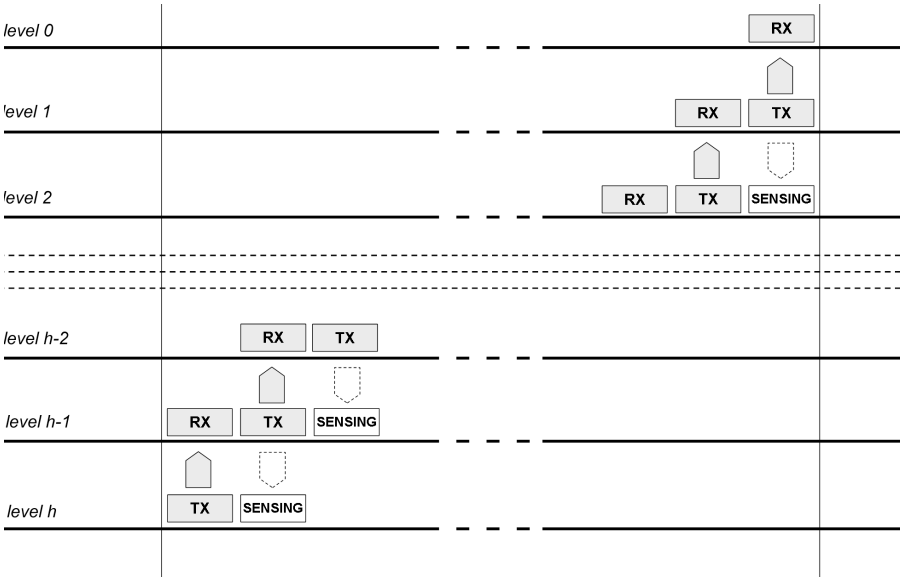

Figure 6.1: Communication model.

In Section 6.5 the performance evaluation is presented based on simulation results. Finally, Section 6.6 concludes the chapter.

6.2 Communication Model

For the evaluation of the collision avoidance algorithms proposed here, a network with a tree-based topology is considered. The tree is rooted at the data-collecting center, or base station (BS), and nodes communicate through multi-hop routing. The communication process is governed by a global periodic schedule, whose duration is termed *epoch* ; it defines periods of activity (*phases*), during which nodes at different levels in the tree are allowed to receive and transmit data packets. This process is represented in Figure 6.1.

Each intermediate node groups its child nodes into a cluster, and acts as a coordinator and data aggregator for them. At the same time it is also part of its coordinator's cluster. The coordinator's level determines the timing for the communications of its cluster, i.e. it establishes the arrangement of the phases with respect to the epoch's boundaries. Every node performs a sensor reading at the beginning of the epoch; it collects data from child nodes, computes an aggregate, and transmits a single packet per epoch toward its coordinator. Packets flow from levels farther from the BS towards the root of the network and communication within clusters at the same level is simultaneous. Figure 6.1 also shows the three phases used to perform data transfer.

Nodes acting as coordinators receive packets during the RX phase. During the TX phase, as child nodes, they transmit the aggregate to the cluster coordinator, and finally listen for the coordinator transmission during a third phase called SENSING. The alignment between phases of adjacent levels is such that the TX and SENSING phases overlap respectively with the RX and TX phases of the lower-level cluster. The goal of the SENSING is to provide error control. In this system, error control does not rely on MAC-Layer acknowledgments and retransmissions, so that the level of contention on the channel can be reduced and the collisions minimized. On the other hand, feedback on the outcome of a transmission is provided to a node by the packet captured during the SENSING phase, through the adoption of the implicit acknowledgment technique provided by synopses diffusion.

Implicit acknowledgment can also be implemented with very low complexity by introducing an header field where the sender specifies the nodes from which the forwarded data have been received. Note that the use of local identifiers can contribute to keep the extra overhead limited.

This approach based scheduled periods of activity allows to reduce the energy consumption of the nodes, which may enter a sleeping state when they do not need to actively partecipate to the communication process. In this particular case a rationalization of the accesses to the channel is achieved, for only a limited subset of nodes contend during each TX phase. On one side, during the TX phases of deeper levels, nodes belonging to different clusters are likely to be too far apart to interfere with each other; on the other side, at the higher levels the expected population of nodes is smaller; hence this network organization helps reducing contention on the channel. The next section provides the details of the proposed MAC algorithms used within the TX phases, and explains how the peculiarities of the standard IEEE 802.15.4 MAC protocol are exploited.

6.3 Collision Avoidance

The collision avoidance scheme presented here extends and improves the operation of the IEEE 802.15.4 MAC protocol. The proposed approach is based on application-initiated actions and maintains the underlying layer unmodified, thus making this solution viable for implementation on devices compliant with the standard IEEE 802.15.4 protocols. The underlying idea is based on two following main assumptions. First, every node transmits a maximum of one packet during each TX phase, and the packet is ready at the beginning of the TX phase. Second, every node maintains a synchronization so that the Application Layer can accurately determine the beginning edge of the TX phase. The latter assumption is discussed in greater detail in Section 6.4.

As a prerequisite for the application of this collision avoidance scheme, the

MAC protocol parameters are configured in such a way that more accurate control over packet transmission times is given to the Application Layer. The IEEE 802.15.4 MAC uses CSMA/CA with a binary exponential backoff, whose operation introduces a non-deterministic delay between the reception of a packet from the Application Layer and the actual transmission over the channel. In order to limit the effect of the backoff algorithm it is necessary to reduce the dynamic range of the MAC delay, so that the final transmission time can be predicted with reasonable accuracy. IEEE 802.15.4 uses a collision window equal to $[0, 2^{BE} - 1]$, where the minimum value of the backoff exponent (BE) is controlled by the configurable parameter $macMinBE$. By setting $macMinBE$ to 0, collision avoidance is disabled during the first iteration of the backoff algorithm, i.e. the MAC protocol performs channel assessment and attempts to transmit outgoing packets immediately after the reception from the upper layer. The transmission is deferred only if the channel is sensed busy, according to the rules defined in the specifications. Although the possibility of many deferrals still exists, and the MAC delay mantains a certain variance, these settings give the Application Layer the highest control over the timing of the transmissions, and constitute the basis for the effective application of the collision avoidance algorithm.

More specifically timing of the transmissions is based on the introduction of an application delay, applied between the beginning edge of the `TX` phase and the time when the Application Layer passes the packet to the MAC Layer for transmission. The application delay can consist of up to $MaxDelaySlots - 1$ MAC-layer backoff slots. A node will initially adopt a random application delay D_A, and will decide whether to keep it or change it for the next epoch according to a set of rules which depend on the outcome of the transmissions. The basic idea is that a node will tend to keep a delay which results in transmission successes, and will change it in case of failures. The introduction of this memory mechanism, together with the reduced dynamic range of the MAC delay D_{MAC}, trigger an adaptive process that converges to the selection of application delays minimizing the probability of collisions.

The specific rules adopted in the decision process have to be defined considering the following factors: the control over the transmission times is not complete, as D_{MAC} cannot be determined in advance; moreover collisions are not the only cause of transmission failures; finally, a minimum level of persistence is needed to ensure that collected results are meaningful for future transmissions, so as to avoid instability. In the light of these considerations, the decision of changing the application delay should not be based on single failure events, but instead on the detection of a trend of negative results.

In the following two decision algorithms are considered, *FailuresCount* and *WeightedAverage*, which use memory of past transmission results, and a third algorithm, *RandomDelay*, used for comparison purposes.

6.3.1 Algorithm 1: RandomDelay

This algorithm is a basic solution which can take advantage of the extended backoff window resulting from the introduction of application delays. According to this algorithm, a different random application delay is used for each transmission, and the outcome of the transmission is disregarded for the purposes of the decision. The effects of this algorithm are similar to what could be obtained by increasing the MAC collision window; however, by operating at the Application Layer, it is possible to test values of the window falling outside of the legal range specified in the standard. This algorithm will be used as a term of comparison in Section 5.4. Note that the effect of setting a zero range for the application delay is that the algorithm exactly reproduces the behavior of the standard MAC.

6.3.2 Algorithm 2: FailuresCount

The pseudocode for algorithm *FailuresCount* is reported in Figure 6.2. The algorithm uses a counter for keeping track of consecutive transmission failures (*TxFailCount*), and compares it with a configurable threshold (`MAX_TX_FAIL`) in order to decide on the application delay reuse.

During initialization the counter is set to a null value, and an initial random value for the application delay is chosen ($D_{A,old}$).

Since the coordinator of the first-level nodes is the Base Station, they do not overhear any forwarded packet, so they adopt a new random application delay at every epoch; in other words algorithm *RandomDelay* is used for them. For all other nodes, when the `TX` phase begins, *TxFailCount* is compared with `MAX_TX_FAIL`. If *TxFailCount* < `MAX_TX_FAIL`, the previous delay ($D_{A,old}$) is mantained; otherwise, a new random D_A is selected and the counter is also reset, so that the results achieved with the new D_A can be tested. The adopted delay is stored and *TxFailCount* is incremented, i.e. a failure is assumed by default.

If a positive implicit acknowledgment is obtained during the `SENSING` phase, *TxFailCount* is reset. On the other hand, if the implicit acknowledgment indicates a failure or if no packet is overheard at all, *TxFailCount* is not changed.

6.3.3 Algorithm 3: WeightedAverage

As already mentioned, the backoff algorithm performed by the MAC Layer introduces a certain variance in the overall delay ($D_A + D_{MAC}$), so it can be expected that the same settings, in terms of application delays, may lead to different results in terms of collisions from one epoch to another. The approach of algorithm *FailuresCount* has the advantage of simplicity, but it would allow situations in which collisions occur at an intolerable rate, while never exceeding the configured

Algorithm 2 Failures Count.

```
procedure Initialize
    TxFailCount ← 0;
    D_{A,old} ← random(0, MaxDelaySlots − 1);
end procedure

procedure sendData                                          - Phase TX
    if level = 1 then
        D_A ← random(0, MaxDelaySlots − 1);
    else
        if TxFailCount < MAX_TX_FAIL then
            D_A ← D_{A,old};
        else
            D_A ← random(0, MaxDelaySlots − 1);
            TxFailCount ← 0;
        end if
        D_{A,old} ← D_A;
        TxFailCount ← TxFailCount + 1;
    end if
    wait for D_A MAC backoff slots;
    pass the packet to the MAC Layer;
end procedure

procedure recvDataFromParent                                - Phase SENSING
    if packet includes data transmitted during phase TX then
        TxFailCount ← 0;
    end if
end procedure
```

Figure 6.2: Pseudocode describing delay assignment through Algorithm 2.

thresholds. Under these circumstances the nodes do not change the application delays, and the network keeps running with poor performance. Obviously lower thresholds could be configured to prevent such situations, but this might introduce instability, with nodes unable to find a steady schedule. To overcome these short- comings, a third approach has been devised which is based on a weighted average over the last transmission results.
The algorithm *WeightedAverage* uses a binary vector, $\overrightarrow{tx\_res}$, which stores the last n transmission results (1 indicating a failure and 0 for a success), a set of configurable real weights, $\overrightarrow{tx\_w}$, and a threshold, `TX_FAIL_THR`. The algorithm is described by the pseudocode in Figure 6.3.

An initial backoff value ($D_{A,old}$) is selected at initialization, and the vector of

Algorithm 3 Weighted Average.

```
procedure Initialize
    tx_res ← 0⃗;
    D_{A,old} ← random(0, MaxDelaySlots − 1);
end procedure

procedure sendData                                          - Phase TX
    if level = 1 then
        D_A ← random(0, MaxDelaySlots − 1);
    else
        if Σ_{i=1,...,n} tx_res ⇥ tx_w < TX_FAIL_THR then
            D_A ← D_{T,old};
        else
            D_A ← random(0, MaxDelaySlots − 1);
            reset tx_res to all zeros;
        end if
        D_{A,old} D_T;
        rightShift(tx_res)
        tx_res[0] ← 1;
    end if
    wait for D_A MAC backoff slots;
    pass the packet to the MAC Layer;
end procedure

procedure recvDataFromParent                                - Phase SENSING
    if packet includes data transmitted during phase TX then
        tx_res[0] ← 0;
    end if
end procedure
```

Figure 6.3: Pseudocode describing delay assignment through Algorithm 3.

results ($\overrightarrow{tx\_res}$) is filled with zeros.

During the TX phase, the average of the values in $\overrightarrow{tx\_res}$, weighted by $\overrightarrow{tx\_w}$, is compared with the threshold TX_FAIL_THR. In analogy with Algorithm 2, when the computed quantity is smaller than the threshold, $D_{A,old}$ is reused, otherwise, a new D_A is selected and the vector of results is reset. Afterwards, the adopted delay is stored in $D_{A,old}$ and $\overrightarrow{tx\_res}$ is right-shifted, in order to make room for the new transmission result. Since a transmission failure is assumed by default, $tx\_res[0]$ is set to 1.

Note that the exception about level 1 nodes also applies to algorithm *WeightedAverage*.

A node receiving a positive implicit acknowledgment during the SENSING phase,

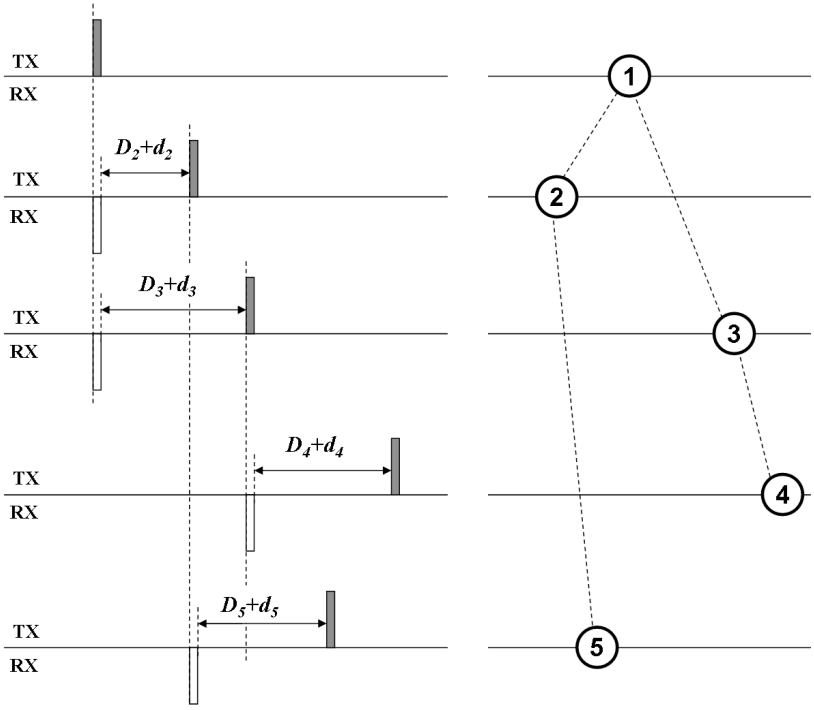

Figure 6.4: Collision avoidance for the beaconing process.

resets $tx\_res[0]$, while $tx\_res[0]$ remains 1 in case of negative acknowledgment or in the absence of a captured packet.

6.4 Implementation Details

As stated in Section 6.3, the Application Layer of all the nodes needs to be able to locate the beginning boundary of the `TX` phase with reasonable accuracy, hence a synchronization mechanism is needed.

According to the standard, in a single-cluster network, a coordinator which operates in *beacon-enabled* mode transmits beacons with a period called *Beacon- Interval*, and the time bounded by two consecutive beacons is referred as *super- fame*. The capability to synchronize to the beaconing process is embedded in IEEE 802.15.4 nodes, and it can be exploited by the network to limit the portion of the superframe to be used for packet exchange in order to achieve a low duty cycle and save energy. When the network spans over a large area, as in most actual WSN applications, it is unlikely that all the nodes will be able to receive beacons from a single coordinator, thus making the single-cluster option unfeasible.

For these cases the IEEE 802.15.4 specifications define a beacon-enabled *cluster-tree* topology, which refers to a tree-based network where any intermediate node can form a cluster and become its coordinator, broadcasting beacon frames to neighboring devices. The standard, however, does not provide guidelines for an actual implementation, and does not address the several issues which arise in a cluster-tree network, including clustering rules and network-wide collision-free beaconing. The beacon frame collision problem has been addressed by the Task Group 15.4b 15., but none of the proposed approaches has been included in the most recent standard; more recently the problem has been discussed in [20], where the authors proposed a beacon scheduling based on Time Division. However, this approach is centralized and requires knowledge of node locations; on the contrary, a low-complexity and decentralized solution has been adopted, as described in the following.

6.4.1 Beacon Collision Avoidance

In the considered communication model every intermediate node, which is both a child node and a coordinator, defers the transmission of its beacons by adopting a small delay $D_{BEACON} = D + d$ relative to the reception of the beacons from its own coordinator. The following ranges have been adopted:

$\{ D \in [2, 15];$
$d \in [\ 2, +2]$

where D and d are expressed in terms of MAC backoff intervals. D is determined upon the first beacon reception and is not modified during the lifetime of the network, while d is randomly selected at the beginning of each new epoch. The random term d has been introduced to prevent a statically selected D_{BEACON} from causing systematic collisions.

The whole process is represented in Figure 6.4, which shows a branch of the tree-topology spanning through three levels and the beaconing process of the considered nodes. The schedule introduced in Section 6.2 has been implemented as an overlay structure based on this process, namely by grouping adjacent superframes to form the activity periods, or phases, as illustrated by Figure6.5. The mapping is built by the Application Layer which, according to the standard, receives an indication from the MAC Layer upon the reception of every beacon, and uses it together with the sequence number of the beacon in order to locate the beginning of the `TX` phase.

It should be noted that the use of the variable term d introduces relative variations in the time reference of the clusters belonging to the same level, and partially reduces the effectiveness of the collision avoidance scheme among their nodes. For this reason other deterministic or adaptive approaches are being considered to eliminate this inconvenience.

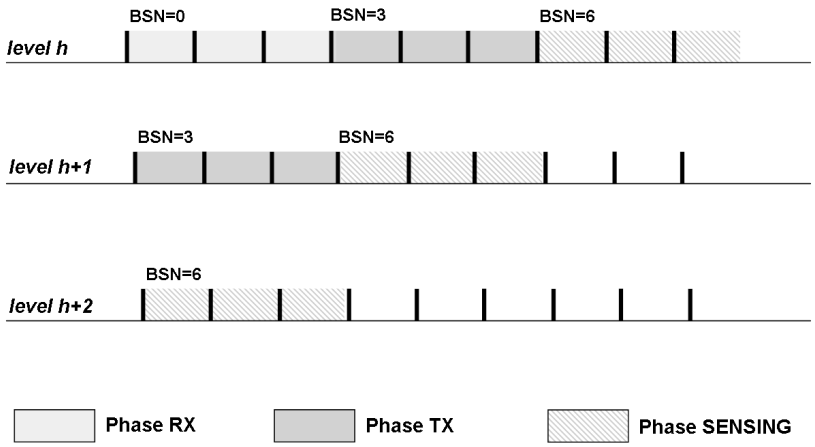

Figure 6.5: Correspondence between phases and IEEE 802.15.4 superframes.

6.5 Performance Evaluation

The collision avoidance schemes discussed in Section 6.3 have been evaluated through simulations using the developed simulation tool. It is based on the widely adopted *ns-2* simulator Authors (2000) and uses an IEEE 802.15.4-compliant MAC- and PHY-Layer protocols implementation provided by third parties.

6.5.1 Simulation Setting and Performance Metrics

Nodes are assumed to have a 10-meters transmission range, which corresponds to an error-free reception area $A_f \cong 314.16m^2$. They are randomly deployed according to a uniform spatial distribution, with network density ranging from 4 to 40 nodes/A_f. An ideal channel is assumed, i.e. collisions are the only source of packet corruption. The clustering process is automatically managed by the *ns-2* implementation of 802.15.4. After the network formation, all the nodes begin to generate and forward data. The duration of all simulations is set to 3000 seconds, and measurements are collected after a 500-seconds transient time, which accounts for network formation and allows the collision avoidance algorithms to reach a steady state. All the results have been averaged over 100 simulations, and the error bars on the charts represent 95% confidence intervals.

The performance is evaluated in terms of average delivery ratio $D_{r,avg}$. The delivery ratio D_r is defined as the ratio between the number of different sensor readings in the final digest computed by the base station, and the total number of nodes. D_r is calculated at the end of each epoch, and $D_{r,avg}$ is the average across all the epochs.

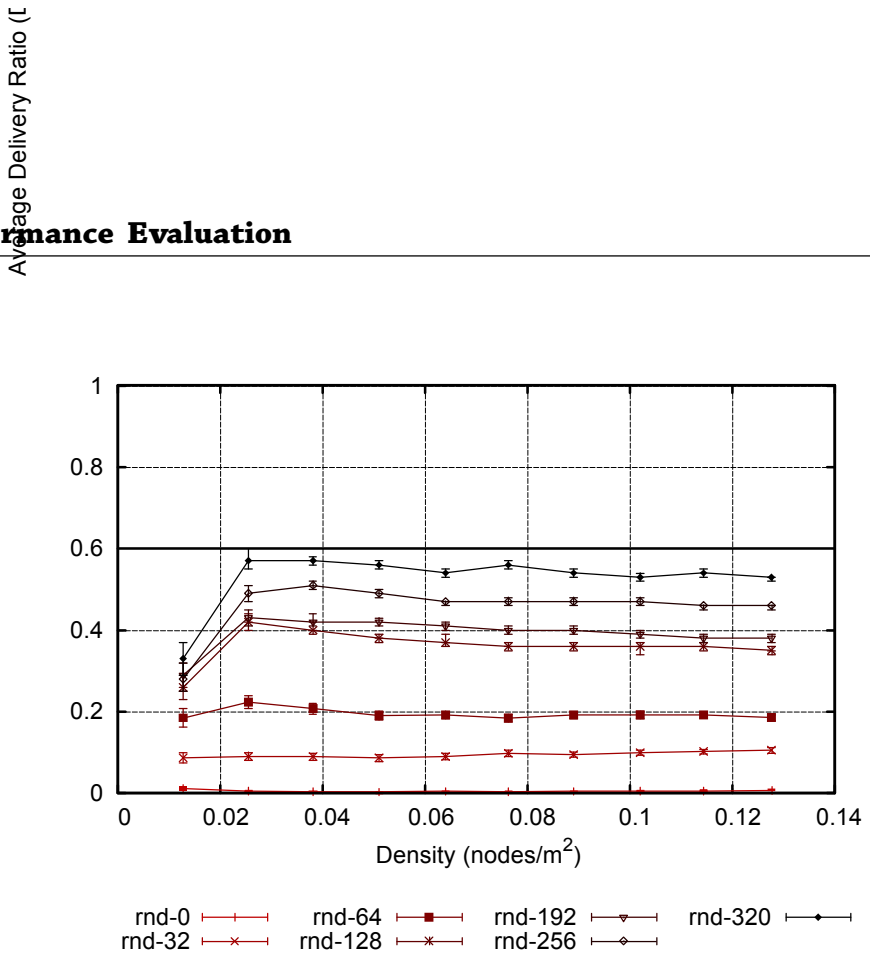

Figure 6.6: Performance of the *RandomDelay* algorithm in a 40-nodes network.

As far as energy consumption is concerned, it can observed that when the nodes employ either the *RandomDelay* algorithm or the standard MAC, which do not make use of implicit acknowledgments, they only need to keep their radio on during the two phases `RX` and `TX`; on the other side, when one of the closed-loop algorithms is used, the radio is on during the `SENSING` phase as well, in order to capture forwarded packets. It can be assumed that during the `TX` phase, nodes need to turn on the radio only for the small amount of time needed to complete one transmission; during the `RX` phase, nodes listen to the channel for incoming packets for the whole duration of the phase; finally, during the `SENSING` phase nodes can turn o the radio after the capture of their parent's packet, which results into using the radio for half the duration of the phase on average. Based on this analysis, the energy consumed during the `TX` phase can be neglected and it can be assumed

Table 6.1: Settings for the simulations presented in Figure 6.6.

	Nodes	*PD*	*Algorithm*	*MDS*	*Thr*
rnd-0	40	3	Random	0	-
rnd-32	40	3	Random	32	-
rnd-64	40	3	Random	64	-
rnd-128	40	3	Random	128	-
rnd-192	40	4	Random	192	-
rnd-256	40	4	Random	256	-
rnd-320	40	4	Random	320	-

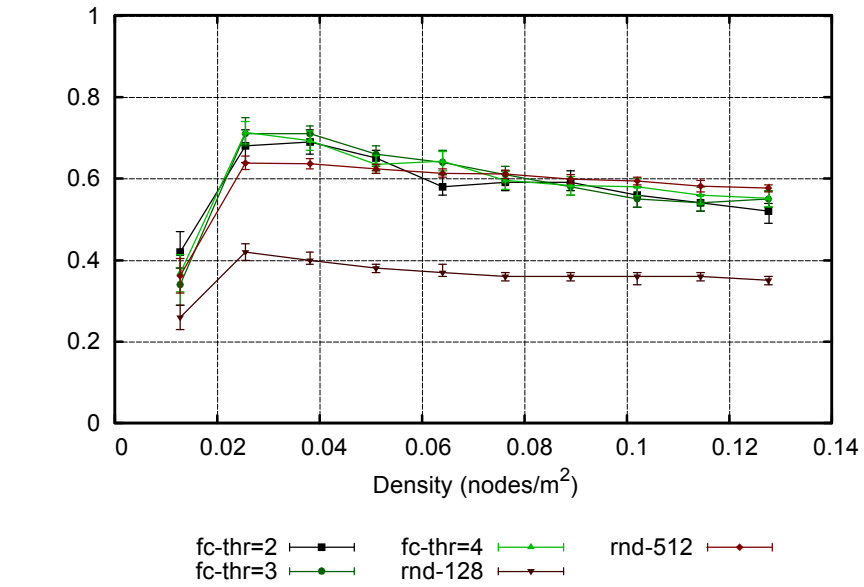

Figure 6.7: Performance of the *FailuresCount* algorithm in a 40-nodes network.

that, for a given phase duration (PD), the closed-loop algorithms consume 50% more energy than the *RandomDelay* algorithm or the standard MAC.

In general, energy consumption also depends on the values of *MaxDelaySlots* (MDS). Larger MDS will require longer phase durations, in order to avoid the occurrence of late packets, i.e. those packets deferred over the `TX` phase boundary which would be discarded by the application. Of course longer phases imply proportionally higher energy consumption. In order to make a fair comparison among the considered alternatives, in the following simulations the duration of the phases is the minimum required by the adopted value of *MaxDelaySlots*. Namely, for each value of MDS, increasing phase durations have been tested and the minimum duration beyond which any further increment has negligible effects on performance has been selected.

Concluding, the analysis of energy consumption will be based on the comparison of the values of PD, taking into account a 1.5 factor for the *FailuresCount* and *WeightedAverage* algorithms.

6.5.2 Medium-size Networks

In order to evaluate the benefits of adopting application-controlled backoff delays, a first set of simulations is considered where the *RandomDelay* algorithm, described in Section 6.3.1, has been applied. The network size is 40 nodes, and several

values of the maximum additional backoff delay, *MaxDelaySlots*, are tested. The complete settings are reported in Table 6.1, where the column *Thr* indicates the threshold used by the considered algorithm, i.e. `MAX_TX_FAIL` for the algorithm *FailuresCount* and `TX_FAIL_THR` for the algorithm *WeightedAverage*; the results are shown in Figure 6.6.

For *MaxDelaySlots* = 0 (see the result set *rnd*-0), the observed delivery ratio represents the one achieved by the standard MAC Layer without any additional collision avoidance mechanism. As shown in the chart, the performance is heavily impaired by the large amount of collisions among both data packets and beacons. A consistent improvement is already obtained with *MaxDelaySlots* = 32 (*rnd*-32), which increases the average delivery ratio up to about 0.1, maintained through the whole tested range of network densities. Performance further improves when increasing the value of *MaxDelaySlots* up to 320. As reported in Table 6.1, in the *PD* column, the use of up to 128 slots consumes as much energy as the standard MAC ($rnd - 0$), while using 192 or more slots requires longer, more energy consuming phases (4 versus 3 superframes). The proposed closed-loop algorithms address this problem by optimizing channel utilization and keeping *MaxDelaySlots* limited for a target delivery ratio.

Figure 6.7 shows the results obtained with the *FailuresCount* algorithm, presented in Section 6.3.2, in the same 40-nodes scenario. The value of *MaxDelaySlots* has been fixed to 128 slots, and the algorithm is compared with the *RandomDelay* algorithm. For the *FailuresCount* algorithm, di erent values of the threshold `MAX_TX_FAIL` are considered. Detailed settings for this simulations are reported in Table 6.2.

The proximity of the three result sets *fc-thr*=2, *fc-thr*=3 and *fc-thr*=4 shows that the algorithm performance is only marginally affected by the value of the threshold. Comparing these results with the result set *rnd*-128, it can be observed that the delivery ratio achievable with the *FailuresCount* algorithm is definitely higher with respect to the *RandomDelay* algorithm with the same *MaxDelaySlots*. The extent of the improvement ranges from about 80% to 40% as the node density increases, showing that the algorithm is mainly affected by the number of nodes contending for channel access and less by hidden node issues, which typically arise with lower density. This result was expected because the algorithm

Table 6.2: Settings for the simulations presented in Figure 6.7.

	Nodes	*PD*	*Algorithm*	*MDS*	*Thr*
fc-thr=2	40	3	Fail. Count	128	2
fc-thr=3	40	3	Fail. Count	128	3
fc-thr=4	40	3	Fail. Count	128	4
rnd-128	40	3	Random	128	-
rnd-512	40	5	Random	512	-

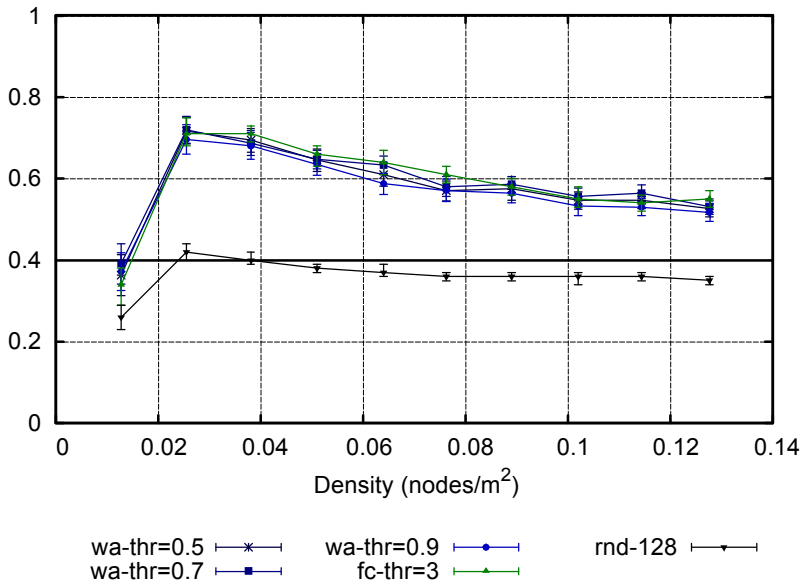

Figure 6.8: Performance of the *WeightedAverage* algorithm in a 40-nodes network.

makes use of a feedback that is indirectly provided by the receivers, and thus obviates to the incapacity of the senders to determine collisions at the receiver side, which is at the origin of the hidden node problem. The *FailuresCount* algorithm shows comparable performance to the *RandomDelay* algorithm using 512 slots. As shown in Table 6.2, 512 slots require almost twice the phase duration (5 versus
3), however at the same time the *RandomDelay* algorithm uses only two of the three phases. Based on the evaluation of the energy consumption during the three phases, discussed earlier, the *RandomDelay* algorithm with 512 slots will consume more energy than the *FailuresCount* algorithm with 128 slots. This means that the extra time and energy used by the nodes to capture forwarded packets and get the implicit acknowledgment is better rewarded, in terms of performance, than the time used by the *RandomDelay* algorithm with extended collision window. When compared with the standard MAC (result set *rnd* 0 in Figure 6.6), the *FailuresCount* algorithm shows a performance improvement well worth the 50%

Table 6.3: Settings for the simulations presented in Figure 6.8.

	Nodes	*PD*	*Algorithm*	*MDS*	*Thr*
wa-thr=0.5	40	3	Weighted Avg.	128	0.5
wa-thr=0.7	40	3	Weighted Avg.	128	0.7
wa-thr=0.9	40	3	Weighted Avg.	128	0.9
fc-thr=4	40	3	Fail. Count	128	4
rnd-128	40	3	Random	128	-

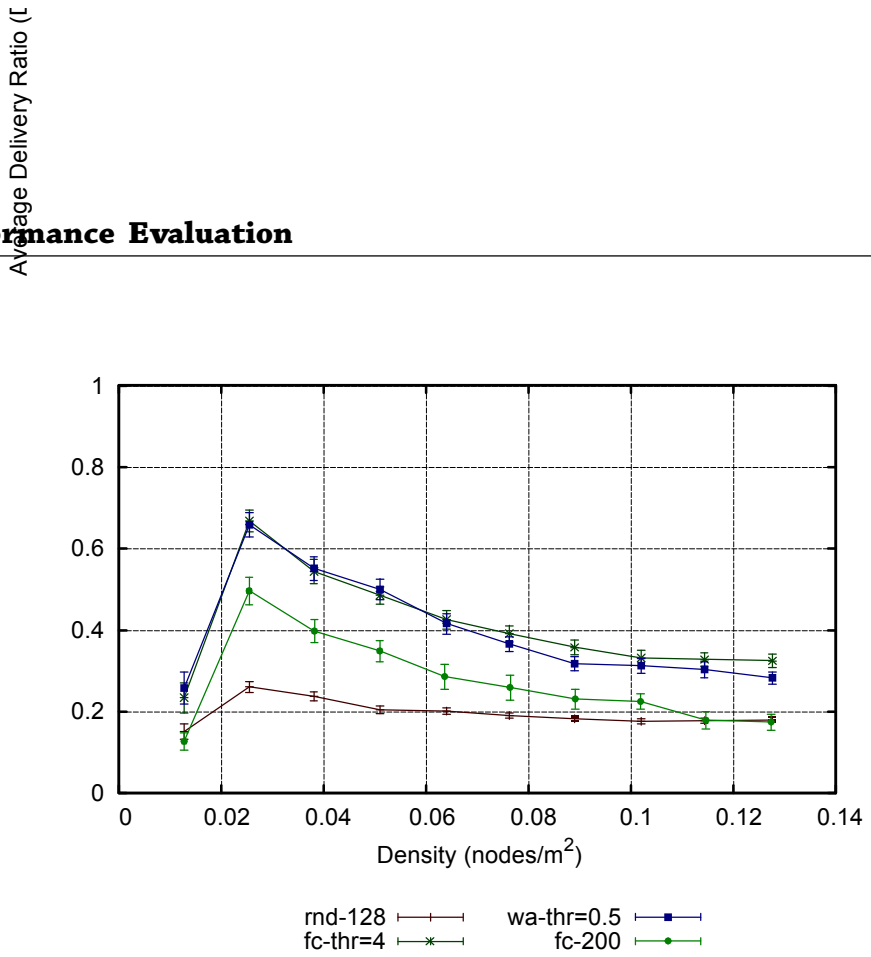

Figure 6.9: Performance of the collision avoidance algorithms in networks of 100 nodes.

extra energy spent.

The perfomance of the *WeightedAverage* algorithm has been evaluated with a third set of simulations, whose results are reported in Figure 6.8, along with the result sets *fc-thr*=2 and *rnd*-128 from Figure 6.7. The algorithm operates on the last 6 transmission results, with a constant vector of weights. Different values of the threshold `TX_FAIL_THR` are considered. $MaxDelaySlots$ has been set to 128, while the other parameters are shown in Table 6.3.

The curves *wa-thr*=0.5, *wa-thr*=0.7, and *wa-thr*=0.9 in Figure 6.8 indicate delivery ratios that are nearly equivalent to those obtained with the *FailuresCount* algorithm, with the same dependence on node density and also little impact of the threshold. All of the above considerations about the energy consumption of the *FailuresCount* algorithm also apply to the *WeightedAverage* algorithm, as their use of the radio is identical.

Table 6.4: Settings for the simulations presented in Figure 6.9.

	Nodes	*PD*	*Algorithm*	*MDS*	*Thr*
rnd-128	100	3	Random	128	-
fc-thr=4	100	3	Fail. Count	128	4
wa-thr=0.5	100	3	Weighted Avg.	128	0.5
fc-thr=6	200	3	Fail. Count	128	6

6.5.3 Large-size Networks

Further simulations have been carried out in order to evaluate the scalability of the proposed algorithms. The results presented in Figure 6.9 refer to a network size of 100 nodes, and the size of the collision window has been fixed to 128 slots. The detailed settings for the considered scenarios are summarized in Table 6.4. The *RandomDelay* algorithm (result set *rnd*-128) proves to be able to deliver an average of about 20% of the generated sensor readings in the 100-nodes scenario, with slightly worsening performance as the node density increases. In the same scenario algorithms *FailuresCount* (*fc-thr*=4) and *WeightedAverage* (*wa-thr*=0.5) achieve consistently better results. Their performance are similar, with a peak of the delivery ratio corresponding to the lower density value that guarantees full network connectivity, and, in both cases, the delivery ratio decreases with increasing density and seems to level o the 35%. The fourth considered result set (*fc*-200) has been reported to show preliminary results which have been obtained with the *FailuresCount* algorithm in a network of 200 nodes, that suggest that the algorithm is also showing good scalability property.

6.6 Conclusions

This chapter described an approach to collision avoidance for IEEE 802.15.4 wire- less sensor networks employed for periodic monitoring tasks. The basic idea of this approach is to build an adaptive schedule of transmissions, controlled by the application through the introduction of additional backoff delays, without changes in the standard IEEE 802.15.4 protocols. The system exploits the periodic nature of the traffic, and by collecting information about past transmission failures and successes, adapts the adopted delays in order to reach a schedule which minimizes the collision rate. An open-loop solution has been discussed which only applies random delays and can be set up to reproduce the standard 802.15.4 behavior. This has been compared with two adaptive protocols which operate in response to transmission results. Simulation have shown that the proposed adaptive techniques converge to a steady schedule of transmissions that reduces collisions, at the expense of some additional energy consumption. Simple and distributed, the proposed approach performs well in large scale networks and is a good candidate for dynamic environments. Current research e orts are focused on the design of self-configuring algorithms which automatically tune their parameters based on network conditions. The possibility to apply the adaptive closed-loop paradigm to the beacon frames collision problem is also under study.

Parts of this work have appeared in publications [30].

Chapter 7

Implementation with Network Simulator 2

7.1 Introduction

The performance of the techniques presented in the previous chapters have been evaluated with the help of simulations, performed with a software implementation based on the widely adopted framework Network Simulator 2 (ns-2)Authors (2000). ns-2 is a general purpose network simulator with the capabilities of simulating a very wide range of protocols and networks, as well as many functionalities, including the ones reported in the following list:

- Applications: Telnet, FTP, Ping;
- Traffic Source Behavior: www, CBR, VBR;
- Multicasting;
- Transport Agents: UDP/TCP;
- Network Topology;
- Router queue Management Techniques DropTail, RED, CBQ;
- Simulation of wireless networks: Terrestrial (cellular, adhoc, GPRS, WLAN/802.11, 802.15.4, BLUETOOTH), Satellite;
- Mobile-IP, and ad-hoc routing protocols such as DSR, TORA, DSDV and AODV;

Modularity, extensibility and easy network topology setup are among its main characteristics.

From a technical point of view, ns-2 is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the OTcl interpreter (also called the interpreted hierarchy). The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. Users create new simulator objects through the interpreter, and these objects are closely mirrored by a corresponding object in the compiled hierarchy.

The implementation of the Cooperative Reliable Communication system, discussed in Chapter 5, and the Application Controlled Collision Avoidance algorithms, discussed in Chapter 6, has required to extend the default class hierarchies already present in ns-2 with the introduction of new classes and changes in some of the existing ones.

The following list shows the main new classes which have been added to the compiled hierarchy:

- *DGRrtAgent* : this class represents the Network-Layer protocol, which controls several aspects of the communication including routing, synchronization, and error recovery. The DGRrtAgent is responsible for requesting data from the simulated sensors with the proper timing, and processing data by means of the functionalities offered by the synopses library, inferring errors in message reception and requesting retransmissions when needed. The DGRrtAgent controls the IEEE 802.15.4 MAC Layer through the interface offered by the implementation embedded in ns-2, which includes a subset of the primitives of the Service Specific Convergence Sub-Layer (SSCS) defined in the IEEE 802.15.4 standard Authors (2000).

- *BitVector* is the class that implements the synopses library. It offers the typical functionalities of synopses including the generation of new digests, the order- and duplicate-insensitive merge operation, and the extraction of the set of the sources of the data in a digest.

The logical architecture of a sensor node used in the simulations is represented in Figure 7.1.

The next sections describe the process of configuring and launching simulations and the contents of the trace files.

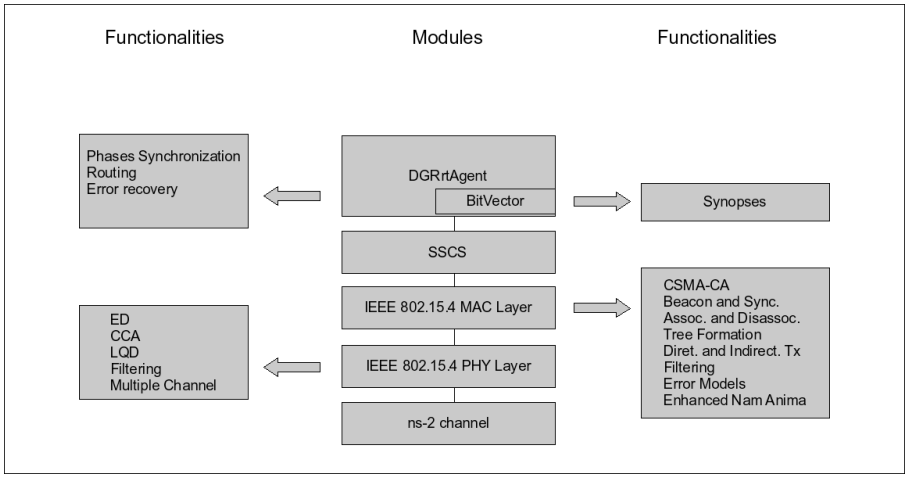

Figure 7.1: Basic stack of modules of a sensor node and respective functionalities.

7.2 Simulation Setup

The script *perl_go* is used to launch a set of simulations in multiple scenarios. *perl_go* coordinates the whole process by calling other scripts and passing them the proper parameters as inputs.

1. For each scenario it launches NUM_SIM iterations by calling the script *sim.sh*.

2. The script *sim.sh* launches a single simulation and receives options according to the *getopts* format. In order to generate the random topology for the simulation, *sim.sh* runs the command *setdest*, thus producing a file of sensor nodes coordinates, *nodes.tcl*, and then it generates a tcl script, *test_d9.tcl*, by editing a template script, *test_d9.tcl.in*.

3. As the single simulation ends, *perl_go* regains control by calling a Perl script, *compute_reliability.pl*, which processes raw simulations results, which are stored in *traceMotes.out*. The script *compute_reliability.pl* requires the number of active sources and the transient time duration as inputs.

4. The standard NS-2 trace file, *NS_DGR.tr*, is processed to extract the number of dropped packets: a temporary file, *NS_DGR.dropped.tr*, is generated and processed by the Perl script *extract_collision.pl*, which writes its output on *collisions.tr*. Afterwards *perl_go* calls the script *stats.sh*, which extracts further statistics from the trace files *NS_DGR.tr*, *NS_DGR.dropped.tr* and *collisions.tr*, and puts its output on *sims_report.tr*. This concludes the operations

needed for a single simulations. If less than NUM_SIM simulations have been completed, the process repeats from step 2.

5. When a block of NUM_SIM simulations for a given scenario is completed, the *update_trace.sh* script analyzes the results in *sims_report.tr* and writes a row in *trace.tr*. A copy of *sims_report.tr* is saved and the following block of simulations, if any, is launched.

7.3 Files

7.3.1 *trace.tr*

trace.tr is a file containing the results from all the simulations launched by *perl_go.sh*. Each row lists the results averaged over NUM_SIM iterations with the same set- tings. Rows' format includes tags indicating some of the simulation parameter for the simulations each row refers to, including

- the number of nodes (n_sens);
- the limit over the maximum number of children a node may have (max_ch);
- the size of the simulation area (L);
- the percentage of nodes that generate readings ($conn\_p$);
- the probability of starting a retransmission after detecting an error ($retx\_p$);
- a boolean flag indicating the use of power management optimizations (opt_en);
- ($bcn_d\, p$);
- ($bcn_d\, s$);
- the adopted *beacon order* (bcn_o);
- the three independent phase durations ($rx1\_sn$,$rx2\_sn$,$rx3\_sn$) expressed in number of superframes;
- a flag indicating the use of IEEE 802.15.4 MAC acknowledgments (ack);
- a flag indicating the use of IEEE 802.15.4 MAC acknowledgments only in the communication between first level nodes and the base station (ack_lv1);
- the values of the thresholds used by the collision avoidance algorithm ($fail\_thr$, $fail\_thr2$);

- the value of *macMinBE* used by the MAC-Layer backoff algorithm (*min_be*).

All results are averages over the NUM_SIM iterations, hence, in the following, the explicit use of the term "average" refers to the arithmetic mean over the epochs of a simulation, while the term global refers to the arithmetic mean over all the nodes. Confidence values refer to the average over the iterations. Results include:

- average reliability, with its maximum and minimum confidence values, and the adopted confidence level;
- minimum and maximum reliability;
- connectivity, with its maximum and minimum confidence values, and the adopted confidence level;
- average reliability normalized to connectivity;
- average amount of energy spent during one epoch, with its maximum and minimum confidence values, and the adopted confidence level;
- global traffic statistics as detailed in the following section;
- average amount of energy spent during one epoch and traffic statistics for each level of the tree;
- amount of collisions.

The original data from which the average values are calculated are stored in *sims_report.tr*.

7.3.2 *sims_report.tr*

sims_report.tr is a file that stores the results relative single simulations. Each row refers to a different simulation and its contents are:

- average, minimum and maximum reliability registered during the epochs of a simulation;
- number of reports collected from the base stations, which produces one report per epoch;
- average amount of energy consumed by nodes during one epoch, individual for each level of the tree and global;

- statistics on the traffic exchanged during the different phases of the communication, individual for each level of the tree and global; reported statistics are:
  - average number of data packets received in the main reception phase during one epoch;
  - average number of data packets received in the retransmission phase during one epoch;
  - average number of retransmission requests received during one epoch;
  - average number of distinct sources of the data collected during one epoch;
  - average number of data packets transmitted in the main transmission phase during one epoch;
  - average number of data packets retransmitted during one epoch;
  - average number of retransmission requests sent during one epoch;
- number of sensor nodes which joined the network (connectivity);
- average reliability normalized to the value of connectivity;
- statistics on dropped packets and collisions.

collisions.tr contains data for all the collisions registered after the end of the transient time.

Chapter 8

TinyOS implementation

8.1 Introduction

TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library in-cludes network protocols, distributed services, sensor drivers, and data acquisition tools, all of which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model enables fine-grained power management yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces. TinyOS has been ported to over a dozen platforms and numerous sensor boards. A wide community uses it in simulation to develop and test various algorithms and protocols.

For this thesis, TinyOS (version 2.0.2) has been used on the Berkeley/Crossbow motes, namely MicaZ and TelosB motes, in order to build an implementation of the collision avoidance techniques presented in Chapter 6.
The TinyOS's library does not include an implementation of the IEEE 802.15.4 standard, hence the open-source open-ZB implementation of IEEE 802.15.4 for TinyOS v2.0, developed within the context of the ART-WiSE [3] (Architecture for Real-Time communications in Wireless Sensor networks) art research framework, has been adopted. This implementation supports:

- CSMA/CA algorithm - slotted version;
- GTS Mechanism;
- Indirect transmission mechanism;
- Direct / Indirect / GTS Data Transmission;

Figure 8.1: Crossbow MICAz mote.

- Beacon Management;
- Frame construction - Short Addressing Fields only and extended addressing fields in the association request;
- Association/Disassociation Mechanism;
- MAC PIB Management;
- Frame Reception Conditions;
- ED and PASSIVE channel scan.

8.1.1 MICAz Motes

A MICAz mote (Figure 8.1) has the following features:

- ATMEL ATmega128L 8-bit microcontroller
- CC2420 RF transceiver
- 128 KB of Program memory (in-system reprogrammable flash);
- 4 KB of EEPROM;

Figure 8.2: Crossbow TelosB mote.

it can be connected with several sensor boards and interacts with a programming board by means of a UART communication port. The programming board is also the means by which the mote can send information to a PC.

8.1.2 TelosB Motes

A TELOSB mote (Figure 8.2) has the following features:

- TI MSP430 16-bit microcontroller
- CC2420 RF transceiver
- 48 KB of Program memory (in-system reprogrammable flash);
- 10 KB of EEPROM;

Differently than MICAz, TelosB motes include a temperature and a light sensor, and can communicate with and be programmed directly by a PC through an UART communication port (USB).

8.1.3 Interface Boards

As already mentioned, MICAz motes need an interface board in order to be programmed and to communicate with a PC. There are three available interface boards (Figure 8.3), equipped with different interfaces on the PC side, namely:

Figure 8.3: MIB510, MIB520 and MIB600 boards.

- the MIB510 board has a a serial RS-232 interface;
- the MIB520 board has an USB interface;
- finally, the MIB600 board has an RJ-45 E thernet interface with an implementation of the full TCP/IP protocol.

For this work, the MIB600 board has been used.

8.2 Description of the code

The software architecture of the open-ZB implementation of IEEE 802.15.4 is visible in Figure 8.4, which shows the implemented modules and interfaces as well as the components which were already present in TinyOS. In this organization, the module that implements the collision avoidance techniques is represented by the element called NWL APP. A detailed description of the open-ZB implementation is outside of the scope of this document, the interested reader is referred to the reference guide [13] available at the IPP HURRAY project's site. Since the open-ZB implementation lacks of some IEEE 802.15.4 functionalities which are necessary for this work, some changes had to be introduced, which are described in the next section. In the rest of the chapter the root TinyOS directory will be indicated with $TOSROOT, while $TOSDIR stands for $TOSROOT/tos.

8.2.1 Changes to the open-ZB code

MacM

path:$TOSDIR/ieee802154/mac/MacM.nc

The MacM module has been modified in order to introduce support for the promiscuous mode of operation, controlled by the MAC PIB parameter *macPromiscuousMode*. Additionally a new provided interface, namely an ASYNC_TIC interface has been introduced, with the purpose of providing a clock signal with the

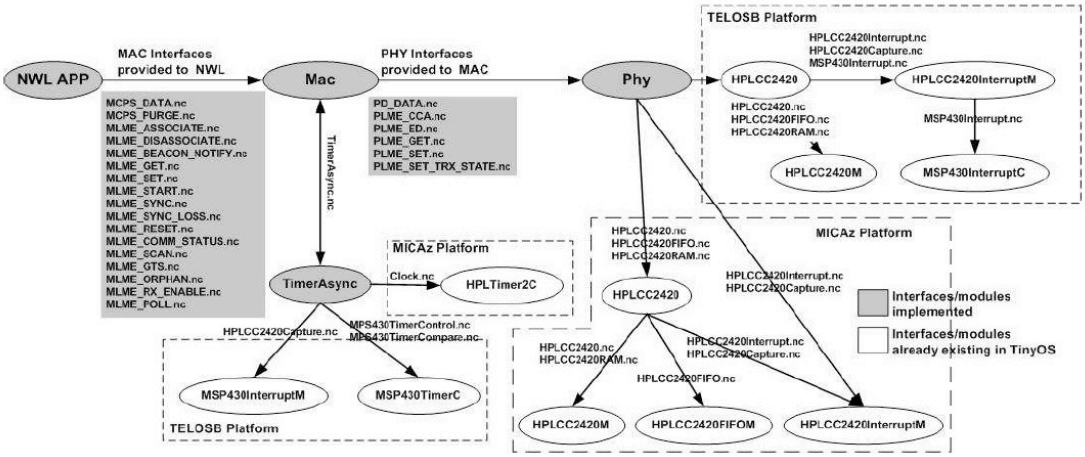

Figure 8.4: IEEE 802.15.4 open-ZB implementation diagram.

granularity of a backoff slot duration (it fires an event call at the beginning of every backoff slot).

Mac

path:$TOSDIR/ieee802154/mac/Mac.nc

The Mac configuration has been modified with the introduction of a new pro- vided ASYNC_TIC interface, which is linked to the interface provided by the included MacM component and used by the PerlaApp configuration.

CC2420ReceiveP

path:$TOSDIR/chips/ieee802154/CC2420ReceiveP.nc

The CC2420ReceiveP module has been modified in order to allow sensor nodes to accept beacons from any beaconing neighbor, by removing the filter which dis- cards a beacon whose source address does not match with the address set in the *ver_macCoordShortAddress* variable. Additionally, in order to enable the promiscuous mode of operation, the filter which discards data and command packets whose destination address differs from the sensor node's address stored in the *ver_macShortAddress* variable, has been disabled as well.

8.2.2 Description of the main events, and functions of PerlaApp

path:$TOSROOT/apps/PerlaApp.nc

In the nesC language, the application that runs on motes, is a *configuration* which contains modules or other configurations, together referred as components, and defines their *wiring*, i.e. the connections between users and providers of inter- faces. For this implementation, this configuration is called PerlaApp and contains the following components:

- the timers Timer_Setup, Timer_Epoch, Timer_BeaconingStop, Timer_Start, Timer_Scan, Timer_Associate; as well as the two following timers included for debugging and logging purposes: Timer_Serial and Timer_Print;
- MainC, which provides the events necessary to start the application;
- LedsC, which provides functions to control the mote's on-board leds;
- Mac, i.e. the open-ZB implementation of IEEE 802.15.4;
- PerlaM, which implements the Application Layer and controls data generation, data aggregation, synchronization as well as the collision avoidance mechanisms.

This section provides a description of the main events, and functions of the PerlaM component.

Boot.booted

This event

- initializes the node ID (*myAddr_*);
- computes the number of backoff slots in a superframe and the equivalent time in milliseconds;
- initializes the application controlled delay, setting a duration larger than a superframe;
  - the base station sets its level to zero and launches the Start timer;
  - ordinary nodes either
    - if scanning is disabled, invoke the preset function which substitutes scanning and association,
    - or launch the Start timer otherwise.

Timer_Start.fired

This timer

- sets the ID as the node's short address (*my_short_address*)
  - the base station
    * sets its short address and the PAN ID at the MAC Layer, and reset the beacon sequence number;
    * starts the beaconing;
    * starts the timer that indicates the beginning of each epoch (Timer_Epoch), and the timer that periodically stops the beaconing (Timer_BeaconingStop);
    * the timer that governs the sending of setup packets is started as well.
  - ordinary nodes, if scanning is enabled, begin the scanning by starting the proper timer (Timer_Scan).

preset

This function is used only when scanning is not enabled. It sets the ID as the node's short address (*my_short_address*) and sets it as well as the PAN ID at the MAC Layer. It sets the ID of the sensor node that will be selected as coordi- nator (*macCoordAddr*), and starts the association with it by calling the primitive MLME_ASSOCIATE.request.

Timer_Epoch.fired

This timer starts the Timer_BeaconingStop timer and reprograms itself. It puts the node in the RX phase.

- the base station resets the BSN and, it restarts the beaconing, by calling the MLME_START.request primitive and setting the *pendingBeaconingStatus* variable accordingly.
- ordinary nodes reset the BSN and, if beaconing suspension is enabled, re- trieve PAN information such as ID and logical channel from the current element in the list of PAN descriptors, and restart the beaconing in the same way as the base station.

Timer_BeaconingStop.fired

This timer performs operations needed at the end of each epoch, including incrementing the epoch counter.

- The base station
  - after the transient epochs have passed, updates the statistics about the average number of different generators of the received data;
  - if beaconing suspension is enabled, it interrupts the beaconing, by calling the MLME_START.request primitive and setting the *pendingBeaconingStatus* variable accordingly.
- Ordinary nodes, if beaconing suspension is enabled, interrupt the beaconing, by calling the MLME_START.request primitive and setting the *pendingBeaconingStatus* variable accordingly.

Timer_Scan.fired

This timer is present only if scanning is enabled. It starts the scanning (passive scanning) and sets the *scanningStatus* accordingly.

MLME_START.confirm

This is a primitive called by the MAC Layer. It signals the outcome of a previous request of start or interruption of beacons transmission and sets the *beaconingStatus*.

MLME_SCAN.confirm

This is a primitive called by the MAC Layer. The MAC Layer provides a list of PAN descriptors. Each descriptor includes information regarding a detected PAN, such as *CoordPANId*, *CoordAddress*, *LogicalChannel*, superframe characteristics and link quality indicator (*lqi*), obtained from the collected beacons. This primitive starts the association process, by calling the MLME_ASSOCIATE.request primitive and starting a timer (Timer_Associate) for more attempts in case of unsuccessful association.

MLME_BEACON_NOTIFY.indication

This primitive is called upon the reception of every beacon from the coordinator the sensor node is associated to.

- When a sensor node is associated but not yet synchronized to the schedule of phases, the time o set with which the node must start to transmit its beacons is determined and the Timer_Epoch timer is started.

- When a sensor node is both associated and synchronized a new time o set is determined only upon the reception of the beacon with $BSN = 0$, and the Timer_Epoch timer is updated accordingly. This prevents sensor nodes from keeping transmitting colliding beacons. This primitive also starts the transmission of setup packets, by controlling the Timer_Setup timer. Finally, it calls managePhases, which is described next, and passes it the received BSN.

managePhases

This is one of the most important function of the application, as it receives a beacon sequence number as input and, based on it, it determines the current phase and starts all the necessary operations.

- When the sensor node is in the TX phase, before sending a data packet, it applies the selected collision avoidance technique, by determining a new delay to be applied or keeping the old one according to the rules discussed in Chapter 6. A counter (*backoff sToDataPkt_*) is used to store the number of backoff slots to go before the transmission.

- All sensor nodes but the base station and level-1 nodes, during the second phase, listen to incoming transmissions in order to get the implicit ack. First, since the data packet is supposed to have been already sent during the TX phase, the current list of generators is reset, then *managePhases* put these nodes in promiscuous mode, while the base station and any level-1 nodes are put in a sleeping phase.

- When the received BSN indicates that the second phase has ended, the promiscuous mode of operation is suspended and sensor nodes enter a sleeping phase. For every BSN received during the TX phase, a node adjusts its own BSN by applying an o set that equals the duration of a phase.

MCPS_DATA.indication

This primitive is called by the MAC Layer protocol upon the reception of data packets. There are two types of payloads: DATA and SETUP.

- SETUP packets are processed only by associated sensor nodes, which com- pare the packets' source address with their coordinator's address. Only

SETUP packets from the coordinator are further processed. In this case, if the sensor node still does not have its level configured, it sets its level as that read in the packet incremented by one.

- DATA packets are processed only by sensor nodes with their level already configured.
  - During the RX phase, a sensor node updates the list of current generators by adding the ones read in the received packet.
  - During the SENSING phase, a sensor node only processes packets coming from its coordinator, in order to get the implicit acknowledgment and update the algorithms used for collision avoidance.

8.3 Results

Experiments have been conducted with simple tree networks. The considered scenario is represented in Figure 8.5, which shows a tree network consisting of a base station, one first-level node and six second-level nodes. Constant settings are reported in Table 8.1. Different values for the maximum number of backoff slots (MDS), used for the application controlled delay, have been tested.

Algorithms RandomDelay and FailuresCount have been evaluated, with the results summarized in Table 8.2.

macBeaconOrder	5
macSuperframeOrder	5
RX phase duration	8 sf
TX phase duration	8 sf
SENSING phase duration	8 sf

Table 8.1: Experiments setup.

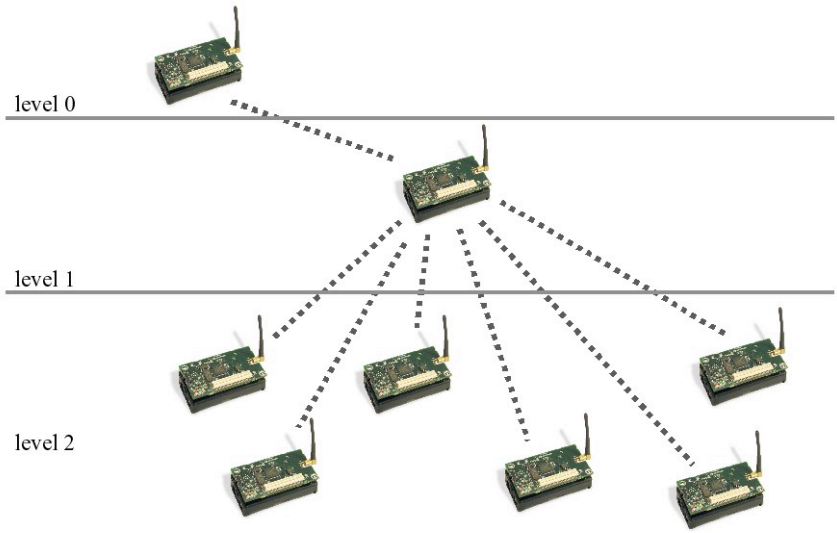

Figure 8.5: Experiments topology.

C.A. Algorithm	MDS	Avg Delivery Ratio (%)	Epochs
FailuresCount	9216	86.83	612
RandomDelay	9216	85.63	500
FailuresCount	4608	94.77	93
FailuresCount	4608	98.90	104
RandomDelay	4608	92.04	79
RandomDelay	4608	93.30	111

Table 8.2: Average Delivery Ratio for the RandomDelay and FailuresCount algorithms, with varying MDS.

Bibliography

[1] *IEEE 802.15 WPAN Task Group 4b*.
http://grouper.ieee.org/groups/802/15/pub/TG4b.html.

[2] *IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 1999.

[3] *ART-WiSe: Architecture for Real-Time communication in Wireless Sensor networks*. http://artwise.cister-isep.info/.

[4] Various Authors. *ns-2, network simulator (ver. 2)*. http://www.isi.edu/nsnam/ns/, 2000.

[5] Lichun [5] and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 210–221, New York, NY, USA, 2001. ACM. ISBN 1-58113-422-3. doi: http://doi.acm. org/10.1145/381677.381698.

[6] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw: A medium access protocol for wireless lans. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, pages 212–225, 1994.

[7] Saad Biaz and Yawen Dai Barowski. Gangs: An energy e cient mac protocol for sensor networks. In *Proceedings of the Annual Southeast Regional Conference*, pages 82–87, April 2004.

[8] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene. Energy e ciency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. In *Proc of Design, Automation, and Test in Europe (DATE)*, 2005.

[9] S. Chatterjea, L.F.W. van Hoesel, and P.J.M. Havinga. Ai-lmac: An adaptive, information-centric and lightweight mac protocol for wireless sensor networks.

In *Proceedings of the Intelligent Sensors, Sensor Networks, and Information Processing Conference*, pages 381–388, December 2004.

[10] Ioannis Chatzigiannakis, Athanassios Kinalis, and Sotiris Nikoletseas. Wireless sensor networks protocols for e cient collision avoidance in multi-path data propagation. In *PE-WASUN '04: Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 8–16, New York, NY, USA, 2004. ACM. ISBN 1-58113-959-4. doi: http://doi.acm.org/10.1145/1023756.1023759.

[11] Ioannis Chatzigiannakis, Tassos Dimitriou, Marios Mavronicolas, Sotiris Nikolet- seas, and Paul Spirakis. A comparative study of protocols for e cient data propagation in smart dust networks. In *Parallel Processing Letters*, pages 615–627, December 2003.

[12] Ioannis Chatzigiannakis, Athanassios Kinalis, and Sotiris Nikoletseas. Wireless sensor networks protocols for e cient collision avoidance in multi-path data propagation. In *Proceedings of the ACM International Workshop on Perfor- mance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 8–16, October 2004.

[13] A. Cunha, M. Alves, and A. Koubaa. An ieee 802.15.4 protocol implementation (in nesc/tinyos): Reference guide v1.2. http://www.hurray.isep.ipp.pt/.

[14] Amre El-Hoiydi and Jean-Dominique Decotignie. Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks. In *Proceedings of the In- ternational Workshop on Algorithmic Aspects of Wireless Sensor Networks (Al- gosensors)*, pages 18–31, July 2004.

[15] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *Proc. of Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.

[16] L. Gatani, G. Lo Re, and M. Ortolani. Robust and e cient data gathering for wireless sensor networks. In *Proc. of the 39th Annual Hawaii International Conference on System Sciences*, pages 235–243, Hawaii, Jan. 2006.

[17] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-e cient communication protocol for wireless microsensor networks. In *Proc. of the Hawaii Int. Conference on System Sciences*, Jan. 2000.

[18] Phil Karn. Maca - a new channel access method for packet radio. In *Proceedings of the ARRL Computer Networking Conference*, 1990.

[19] A. Koubaa, M. Alves, and E. Tovar. A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks. *Factory Communication Systems, 2006 IEEE International Workshop on*, pages 183–192, June 27, 2006.

[20] Anis Koubaa, Andre Cunha, and Mario Alves. A time division beacon scheduling mechanism for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks. *Real- Time Systems, 2007. ECRTS '07. 19th Euromicro Conference on*, pages 125–135, 4-6 July 2007. ISSN 1068-3070. doi: 10.1109/ECRTS.2007.82.

[21] II Kurtis Kredo and Prasant Mohapatra. Medium access control in wireless sensor networks. *Comput. Netw.*, 51(4):961–994, 2007. ISSN 1389-1286. doi: http://dx.doi.org/10.1016/j.comnet.2006.06.012.

[22] En-Yi A. Lin, Jan M. Rabaey, and Adam Wolisz. Power-e cient rendez-vous schemes for dense wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 7, pages 3769–3776, June 2004.

[23] Peng Lin, Chunming Qiao, and Xin Wang. Medium access control with a dynamic duty cycle for sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1534–1539, March 2004.

[24] G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-e cient and low-latency mac for data gathering in sensor networks, 2004. URL `citeseer.ist.psu.edu/lu04adaptive.html`.

[25] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. In *OSDI*, 2002.

[26] Stefan Mahlknecht and Michael B¨ock. Cmsa-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, September 2004.

[27] D. Messina, M. Ortolani, and G. Lo Re. Achieving robustness through caching and retransmissions in IEEE 802.15.4-based WSNs. In *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN 2007)*, pages 1117–1122, 2007a.

[28] Daniele Messina, Marco Ortolani, and Giuseppe Lo Re. A network protocol to enhance robustness in tree-based wsns using data aggregation. In *MASS 2007*, 2007b. ISBN 1-4244-1455-5.

[29] Daniele Messina, Marco Ortolani, and Giuseppe Lo Re. Reliable data gathering in tree-based ieee 802.15.4 wireless sensor networks. In *IWASN 2007*, 2007c. ISBN1-4244-1025-8.

[30] Daniele Messina, Marco Ortolani, and Giuseppe Lo Re. Adaptive collision avoidance through implicit acknowledgments in wsns. In *Wina 2008*, 2007d.

[31] M. Petrova, J. Riihºarvi, P. Mahonen, and S. Labella. Performance study of ieee 802.15.4 using measurements and simulations. *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, 1:487–492, 0-0 2006. ISSN1525-3511. doi: 10.1109/WCNC.2006.1683512.

[32] Huan Pham and Sanjay Jha. An adaptive mobility-aware mac protocol for sensor networks (ms-mac). In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 214–226, October 2004.

[33] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM. ISBN 1-58113-879-2. doi: http://doi.acm.org/10.1145/1031495.1031508.

[34] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 95–107, November 2004.

[35] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy- efficient, collision-free medium access control for wireless sensor networks. *Wirel. Netw.*, 12(1):63–78, 2006. ISSN 1022-0038. doi: http://dx.doi.org/10.1007/ s11276-006-6151-z.

[36] Venkatesh Rajendran, Katia Obraczka, and J.J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 181–192, November 2003.

[37] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. Z-mac: A hybrid mac for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 90–101, November 2005.

[38] Yalin Evren Sagduyu and Anthony Ephremides. The problem of medium access control in wireless sensor networks. In *IEEE Wireless Communications*, pages 44–53, December 2004.

[39] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Optimizing sensor networks in the energy-latency-density design space. In *IEEE Transactions on Mobile Computing*, pages 70–80, January 2002.

[39] Suresh Singh and C. S. Raghavendra. Pamas—power aware multi-access protocol with signaling for ad hoc networks. *SIGCOMM Comput. Commun. Rev.*, 28(3):5–26, 1998. ISSN 0146-4833. doi: http://doi.acm.org/10.1145/293927.293928.

[40] T. Stathopoulos, R. Kapur, D. Estrin, J. Heidemann, and Lixia Zhang. Application-based collision avoidance in wireless sensor networks. *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 506–514, 16-18 Nov. 2004. ISSN 0742-1303. Doi:10.1109/LCN.2004.24.

[41] T°s van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 171–180, November 2003.

[42] L.F.W. van Hoesel and P.J.M. Havinga. Poster abstract: A tdma-based mac protocol for wsns. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 303–304, November 2004.

[43] L.F.W. van Hoesel and P.J.M. Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *Proceedings of the International Conference on Networked Sensing Systems (INSS)*, June 2004.

[44] Mehmet C. Vuran and Ian F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. In *IEEE/ACM Transactions on Networking*, pages 316–329, April 2006.

[45] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, pages 214–226, June 2002.

[46] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. In *IEEE/ACM Transactions on Networking*, pages 493–506, June 2004.

[47] Tao Zheng, Sridhar Radhakrishnan, and Venkatesh Sarangan. Pmac: An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium*, pages 65–72, April 2005.