Università degli Studi di Catania
Dipartimento di Ingegneria
Informatica e delle Telecomunicazioni

DOTTORATO DI RICERCA IN
Ingegneria Informatica e delle Telecomunicazioni
XXIII Ciclo

Industrial Wireless Sensor Networks: Research Challenges and Novel Solutions

Ing. Emanuele Antonino Toscano

Coordinatore
Prof. O. Mirabella

Tutor
Prof. L. Lo Bello

Contents

1 Introduction 1

1.1 Overview of the communication protocols for classical WSNs 3

1.1.1 Optimization-based protocols 3

1.1.2 Data-centric protocols 5

1.1.3 Cluster-based protocols 5

1.1.4 Location-based protocols 6

1.1.5 Topology management protocols 7

1.2 Differences between classical WSNs and industrial WSNs . . 8

1.2.1 Architecture 8

1.2.2 Requirements 10

1.3 Research challenges and possible solutions 11

2 Assessment of cross-channel interference in IEEE 802.15.4 networks 13

2.1 Coexistence of wireless networks 14

2.2 On cross-channel interference in IEEE 802.15.4 16

2.3 Testbed and Methodology 18

2.3.1 Methodology 19

2.4 Case study and experimental results 23

2.4.1 The IEEE 802.15.4 platform 23

2.4.2 Preliminary Assessments 25

2.4.3 Worst Case PER validation 27

2.4.4 Interference by a single node 28

2.4.5 Interference from multiple nodes 31

2.4.6 Influence of MAC parameters 34

2.5 Concluding remarks 36

3 Multichannel Superframe Scheduling for IEEE 802.15.4 39
3.1 The IEEE 802.15.4 protocol 40
3.2 Cluster-tree topologies and beacon frame collisions 42
3.3 Approaches for beacon frame collision avoidance 44
3.4 The SDS Algorithm 46
3.5 The multichannel approach 48
3.6 Multichannel Superframe Scheduling 49
3.7 Analysis of the MSS algorithm 53
3.7.1 MSS Schedulability 54
3.7.2 Frequency constraints 57
3.8 Implementation issues 58
3.8.1 Changes to the IEEE 802.15.4 MAC 58
3.8.2 Adding MSS Support to the upper layers 60
3.9 Experimental Testbed 62
3.10 Concluding remarks 67

4 A Topology Management protocol for RT-WSNs 69
4.1 Approaches to improve WSN performance 70
4.2 Network model 72
4.3 Design Principles 73
4.4 The proposed topology management protocol 75
4.4.1 InitState 76
4.4.2 Cluster Head. 78
4.4.3 Relay Node. 79
4.4.4 Common Node. 80
4.5 Discussion and Protocol Analysis 80
4.6 Performance Evaluation 87
4.6.1 Energy Efficiency of the proposed solution 87
4.6.2 Effect of the delay bound on CH to RN transmissions 89
4.6.3 Effect of the proposed topology management mechanism on the SPEED routing protocol 91
4.7 Concluding remarks 94

5 An improved dynamic topology management protocol for RT-WSNs 95
5.1 Benefits and limitations of the static approach 96
5.2 The dynamic approach 98
5.2.1 Super-frame structure 98

5.2.2 Node Initalization 99
5.2.3 AU organization 102
5.2.4 Lifetime Estimation 103
5.2.5 Dynamic energy balancing 104
5.3 Performance evaluation 106
5.3.1 Effect of the node exchange policy on network lifetime 106
5.3.2 Effect of the topology management protocol on SPEED real-time performance 109
5.4 Concluding remarks 111

6 A chain-based routing protocol for industrial WSNs 113
6.1 Chain-based routing in WSNs 114
6.1.1 Linear scheme 114
6.1.2 Binary-combining scheme 115
6.1.3 Multiple-chain scheme 115
6.2 The Circular Chain Data Forwarding (CCDF) protocol . . . 116
6.3 Fault Tolerance Mechanisms 119
6.4 Distributed Chain Creation 120
6.4.1 Sub-chain creation algorithm 123
6.5 Protocol Analysis 124
6.5.1 Node Traversal Time (NTT) 124
6.5.2 Chain Traversal Time 126
6.5.3 Average Chain Trip Time 128
6.6 Performance evaluation 131
6.6.1 Validation of theoretical results 132
6.6.2 Comparative assessments 135
6.7 Concluding remarks 137

7 Conclusions 139

List of Figures

1.1 Architecture of a typical industrial network. 9

2.1 Structure of the testbed. 19
2.2 Model for overlapping transmission probability. 22
2.3 Preliminary Assessments 27
2.4 Packet Loss and expected PER versus the varying transmission period of the interferer node. 28
2.5 Packet Loss with equidistant nodes. 29
2.6 Expected worst case PER with equidistant nodes. 30
2.7 Estimated PER as a function of the difference between interferer and source received power level. 31
2.8 Spectra of the three interferer nodes seen by the receiver antenna. 32
2.9 Packet Loss using multiple interferer nodes. 33
2.10 The effect of the CCA Threshold. 35
2.11 The effect of the Minimum Backoff Exponent on delay under cross-channel interference. 36

3.1 Network topology. 43
3.2 Direct and indirect beacon collisions. 44
3.3 Example scenario where two grouped coordinators may have interfering nodes. 48
3.4 Scheduling the clusters in alternate timeslices (TS1 and TS2). 49
3.5 MSS superframe scheduling. 51
3.6 Scheduling of a superframe set that is unfeasible using SDS. 54
3.7 Superframe structure. 60
3.8 Association of a (non PAN-) coordinator. 61
3.9 Obtaining radio channel and offset from the PAN Coordinator 62

3.10 Software architecture of TKN15.4 63
3.11 Testbed Scenario. 65
3.12 Temporal trace of the experiment: association of C5. 66
3.13 Temporal trace of the experiment: association of C4. 66
3.14 Temporal trace of the experiment: steady state network operations 67

4.1 State diagram of the proposed topology management protocol. 76
4.2 The proposed topology management protocol 81
4.3 Design space of the proposed topology management protocol 86
4.4 Mean AU power consumption vs. varying super-frame length. 89
4.5 Success Ratio for CH-to-RN synchronous communications. . 91
4.6 End-to-end delay. 92
4.7 Packet loss ratio. 93
4.8 SPEED Hit ratio. 93

5.1 Super-frame structure of the static topology management protocol. 96
5.2 Super-frame structure of the dynamic topology management protocol. 99
5.3 Graphical Representation of AU lifetime. 104
5.4 Example of node exchange between two AUs. 105
5.5 AU configuration after the initialization phase (a) and AU lifetime distribution after one minute (b). 107
5.6 Dynamic evolution of the WSN in terms of AU lifetime. . . 108
5.7 Network lifetime with dynamic AU adaptation. 109
5.8 Performance comparison. 110

6.1 Network Architecture. 117
6.2 Chain-based schemes. 118
6.3 Network Setup. 121
6.4 Node Traversal Time. 125
6.5 Chain Traversal Time. 128
6.6 Time to recover lost frame. 130
6.7 Chain Trip Times. 134
6.8 Comparative performance assessment. 136

List of Tables

2.1 In-air jamming resistance obtained with 2 m distance from transmitter to the receiver. 18

2.2 Basic Testbed configuration 25

3.1 Set of SI and BI values. 52

Chapter 1

Introduction

A Wireless Sensor Network (WSN) is a collection of nodes organized into a cooperative network, typically operating in an unattended environment. Each node is equipped with a processing element, a radiofrequency transceiver (usually with a single omnidirectional antenna), a number of sensors and actuators, memories (data, program, flash) and a power source. From the functional point of view, nodes can be classified as sources, sinks and routers. Sources are sensor nodes that monitor a defined phenomenon (e.g. temperature) and transmit data, whereas sink nodes are those which collect and process data. Routers are nodes that are in charge of forwarding data from sources toward the sink(s). Nodes can play multiple roles at the same time, e.g., sources may also act as routers. Moreover, multiple nodes can take part in data processing before delivering data to the final destination.

When a WSN is designed, multiple conflicting requirements should be met simultaneously. On the one hand, nodes should have sufficient computing and storage capabilities and enough bandwidth for transmission, they should be able to work autonomously and may have different QoS requirements (e.g. limited end-to-end delay). On the other hand, devices should have low costs and limited energy consumption, so that long lifetime can be achieved. Although the correct trade-off between these conflicting requirements is dependent on the specific WSN application, most of the research on WSNs focuses on how to increase the network lifetime.

In order to meet the long-lasting requirement, WSN nodes typically feature low-power processors and very small memories. However, this is not sufficient, as the energy consumption in WSNs is typically dominated by the node communication subsystem costs rather than by processing costs.

In order to prolong the nodes' lifetime, and thus the lifespan of the network as a whole, as much as possible, strategies aiming at reducing energy consumption have to be implemented at all the different levels of the network protocol stack. This is why literature offers many communication protocols aiming at reducing energy consumption, implemented at the various levels of the protocol stack, from the physical up to the application layer, and even cross-layer approaches to save energy are found in the literature. An overview of the existing literature on WSN communication is given in Section 1.1.

Industrial applications can take advantage of the lower cost and easier deployment of WSNs as compared to traditional industrial networks [1]. While the deployment of a traditional industrial network infrastructure is costly and time-consuming, WSNs only need a minimal infrastructure, if any. In addition, WSNs allow greater flexibility and scalability than traditional industrial networks. In industrial scenarios a WSN may be used to reduce the networking cost of less critical controls and/or to connect different network cells (i.e., dedicated fieldbuses) for monitoring purposes. However, industrial WSNs have both different requirements and different architectures than traditional WSNs [2]. In industrial WSNs the most important requirement is to achieve a predictable behaviour and bounded latency. Energy still plays a role, but is less crucial than in traditional WSN, as industrial WSNs are not supposed to be unattended for long periods. Concerning network architecture, unlike traditional WSNs which typically have to work without any infrastructure, an industrial WSN is usually connected to a real-time wired backbone (e.g., Industrial Ethernet or a real-time fieldbus), because data flows required by critical control loops cannot be transmitted over the wireless medium. A more detailed explanation of the differences between classical and industrial WSNs is given in Section 1.2. Such differences make the existing protocols for classical WSNs unsuitable, or just inconvenient, for the implementation of industrial WSNs.

This thesis investigates novel approaches at different levels of the protocol stack, which are explicitly developed for industrial WSNs. As it will be explained in Section 1.3, the proposed mechanisms and protocols address different challenges (e.g., robustness to the interferences, better bandwidth exploitation, energy efficiency, bounded end-to-end transmission delay), but all of them pursue the common objective of making WSN technology ready for the demands of modern flexible industries.

1.1 Overview of the communication protocols for classical WSNs

As sensor nodes are typically battery-operated, energy saving is a major design issue in classical WSNs. It has been proven that the communication cost for sensor nodes is much higher than the computational cost. For this reason, when deploying a WSN, the network topology, and thus the distance between communicating nodes, is a crucial aspect. In some cases sensors can be put in place in a controlled way, so the WSN can be built in an energy-efficient way if a suitable node placement strategy is followed. However, in most practical cases sensor nodes are randomly scattered over the field, so WSNs are self-organizing and deployed in an ad hoc fashion, and the network topology cannot be set according to any strategy targeting energy consumption. As a result, in order to prolong the network's lifetime as much as possible, approaches aiming at reducing energy consumption have to be implemented at all the different levels of the network protocol stack, from the physical up to the application layer, and even cross-layer approaches to save energy are found in the literature.

The strategies working at the physical layer try to reduce system-level power consumption through hardware design or by means of suitable techniques, such as Dynamic Voltage Scaling or duty cycle reduction. The approaches operating at the data link layer typically exploit low-power MAC protocols aimed at reducing the main causes of energy wastage, i.e., collisions, overhearing, idle listening and the protocol overhead due to the exchange of a high number of control packets. At the network layer energy consumption is mainly dealt with in data routing.

Energy-saving routing protocols for WSNs can be classified into four main categories, i.e., optimization-based, data-centric, cluster-based, and location-based. Such categories are not necessarily disjoint, and some examples of routing algorithms matching multiple categories can be found. Examples given here are the TEEN [3] and the APTEEN [4] protocols, which are both data-centric and cluster-based.

1.1.1 Optimization-based protocols

A broad spectrum of routing algorithms for WSNs aiming at reducing the energy consumption of sensor nodes are present in the literature. Some of them take energy into account explicitly when routing sensor data, and for most of them the main goal is the optimization of some metric. For this reason, we will henceforward refer to them as *optimization-based* energy-aware routing protocols. Example of metrics to be minimized are the energy consumed per message, the variance in the power level of each node, the cost/packet ratio, or the maximum energy drain of any node.

Trying to minimize the energy consumed per message may lead to poor routing choices, as some nodes could be unnecessarily overloaded and thus could quickly extinguish their batteries. A more effective option, if all nodes are equally important for the WSN to operate correctly, is to try to balance the battery power remaining in the nodes, as there is no point in having battery power remaining in some nodes while the others have already run out of power. Minimization of the cost/packet ratio involves labeling different links with different costs and then choosing the best option so as to delay the occurrence of network partitioning as long as possible. On the other hand, the idea of minimizing the maximum energy drain of any node derives from the consideration that network operations start to be compromised when the first node exhausts its battery, so it is advisable to minimize battery consumption in this node.

A number of optimization-based power-aware routing approaches try to maximize network lifetime. They target *network survivability*, meaning that their goal is to maintain network connectivity as long as possible. To achieve this goal, “optimal” routes that avoid nodes with low batteries and try to balance the traffic load are chosen [5]. The use of optimization techniques to find the *minimum cost path*, where the cost parameter takes energy (alone or combined with other metrics) into account, is proposed. However, the minimum cost path approach has a drawback in terms of network lifetime in the long term. In fact, a protocol which, once it has found an optimal path, uses only that path for routing will eventually deplete the energy of the nodes along the path. As large differences in the energy levels of the WSN nodes could lead to undesired effects such as network partitioning, suitable solutions have been developed. A notable example is the Energy-Aware Routing protocol [6], where network survivability is pursued by choosing not a single optimal route, but a set of good routes, i.e., sub-optimal paths which are selected in a probabilistic way.

1.1.2 Data-centric protocols

Unlike the optimization-based routing algorithms described above, other routing protocols for WSNs obtain low power consumption for sensor nodes without explicitly dealing with energy considerations when performing route selection, but implementing mechanisms which reduce energy wastage. One of the main causes of energy wastage in WSNs is *data redundancy*, which derives from a combination of a lack of global identifiers (as no IP-like addressing is possible in WSNs) and the random deployment of sensors, which in many cases makes it difficult, if not unfeasible, to select a specified set of sensors within a given area. To solve this problem, *data-centric* routing approaches were introduced. In these approaches, data is named using high-level descriptors, called *meta-data*, and data negotiation between nodes is used to reduce redundancy. Another approach to reduce data redundancy (and the consequent energy wastage) is by performing *data aggregation* at the relaying nodes, which consists of combining data from different sources and eliminating duplicates, or applying functions such as average, minimum and maximum. Data aggregation also overcomes the *overlap* problem, which arises when multiple sensors located in the same region send the same data to the same neighbour node. Thanks to data aggregation significant energy savings can be achieved, as computation at sensor nodes is less energy-consuming than communication. When performed through signal processing techniques, data aggregation is referred to as *data fusion*. According to the kind of routing protocol, data aggregation may be a task performed by special nodes or any node in the network. Notable examples of data-centric routing protocols which perform data aggregation for energy-saving purposes are SPIN [7] and Directed Diffusion [8], which in turn inspired several other protocols.

1.1.3 Cluster-based protocols

Another critical aspect for energy consumption is the presence of nodes which, being either closer to the sink or on the optimal (e.g. minimum-cost) path to the sink, perform more relaying than the other nodes, thus depleting their energy reserve faster than the others. When such nodes run out of energy, network survivability is compromised, and when all the nodes closest to the sink die, the sink itself becomes unreachable. To avoid this problem, *hierarchical* or *cluster-based* routing was introduced. In cluster-based routing, special nodes called *cluster heads* form a wireless backbone to the sink. Each of them collects data from the sensors belonging to its cluster and forwards it to the sink. In heterogeneous networks, cluster heads may be different from simple sensor nodes, being equipped with more powerful energy reserves. In homogeneous networks, on the other hand, in order to avoid a quick depletion of cluster heads, the cluster head role rotates, i.e., each node works as a cluster head for a limited period of time. Energy saving in these approaches can be obtained in many ways, including cluster formation, cluster-head election, etc. Some of these approaches also perform data aggregation at the cluster-head nodes to reduce data redundancy and thus save energy. Notable examples of cluster-based routing protocols are LEACH [9] and its extensions such as TEEN [3] and APTEEN [4].

Derived from the cluster-based protocol is a communication model where nodes are not explicitly grouped into clusters, but each node only communicates with a close neighbour and takes turns to transmit to the base station, thus reducing the amount of energy spent per round. This is called *chain-based* approach, as data goes across a chain of nodes, from the sources to the final destination. This class of protocols will be discussed in a more detailed way in Chapter 6, Section 6.1.

1.1.4 Location-based protocols

Location-based routing protocols use position information for data relaying. Location information can be exploited for energy-efficient data routing in WSNs as, based on both the location of sensors and on knowledge of the sensed area, a data query can be sent only to a particular region of the WSN rather than the whole network. This feature of location-based routing protocols may allow for a significant reduction in the number of transmissions and thus in the power consumption of sensor nodes.

The Geographic and Energy-Aware Routing (GEAR) protocol, described in [10], which uses an energy-aware metric along with geographical information to efficiently disseminate data and queries across a WSN. Unlike other geographical protocols not specifically devised for sensor networks, such as the well-known Greedy Perimeter Stateless Routing (GPSR) protocol [11], this protocol addresses the problem of forwarding data to each node inside a target region. This feature enables GEAR to support data-centric applications.

SPEED [12, 13] ia another well-known location-based protocol which combines feedback control and non-deterministic geographic forwarding to achieve to manage the QoS. The basic idea is to maintain a desired delivery speed across the sensor network. A similar approach is used in RPAR [14], where transmission power adaptation is used to find a trade-off between delivery speed and energy efficiency.

1.1.5 Topology management protocols

Topology management protocols are a slightly different approach to saving energy than standard routing protocols, as they do not directly operate data forwarding. These protocols run at a lower level of the network stack, i.e. just under the network layer. Their objective is to improve the energy efficiency of routing protocols for wireless networks by coordinating the sleep transitions of nodes. Several routing protocols in fact try to enhance network lifetime by reducing the number of data transmissions or balancing the transmission power, but neglect idle power consumption. However, several measurements, e.g. in [15, 16], show that idle power dissipation should not be ignored, as it could be comparable to the transmitting or receiving power. Therefore, in order to optimize energy consumption, nodes should turn off their radios. Topology control protocols exploit redundancy in dense networks in order to put nodes to sleep while maintaining network connectivity. They can be applied to standard routing protocols for ad-hoc networks or for WSNs that do not directly handle sleep schedules. Although some of them are designed for wireless ad-hoc networks rather than WSNs, the typically high redundancy of sensor nodes and the need for maximum energy saving make WSNs perhaps the most suitable type of networks for taking advantage of these protocols.

The Geographic Adaptive Fidelity (GAF) [17] protocol, in order to put nodes into low-power sleep states without excessively increasing the packet loss rate, identifies groups of nodes that are “equivalent” in terms of routing cost and turn off unnecessary nodes. This is achieved by dividing the whole area into *virtual grids*, small enough that each node in a cell can hear each node from an adjacent cell. Nodes that belong to the same cell coordinate active and sleep periods, so that at least one node per cell is active and routing fidelity (which requires that in any cell at any one time there is at least one node able to perform routing [18]) is maintained.

In [19], another distributed coordination protocol for wireless ad-hoc networks, called Span, is presented. The objective of the Span protocol is to reduce energy consumption without significantly reducing network capacity or the connectivity of a multi-hop network. To achieve this, Span elects in rotation some *coordinators* that stay awake and actively perform multi-hop data forwarding, while the other nodes remain in power-saving mode and check whether they should become coordinators at regular intervals. Coordinators form a forwarding backbone that should provide as much capacity as the original network.

The Sparse Topology and Energy Management (STEM) protocol presented in [20] is a topology control protocol specifically designed for WSNs. The assumption of STEM is that nodes in a WSN may spend most of the time only sensing the surrounding environment waiting for a target event to happen. Thus, unlike other topology management schemes that coordinate the activation of nodes during the transmission phase, STEM optimizes the energy efficiency of nodes during the monitoring state, i.e. when no one is sending data. STEM exploits the fact that, while waiting for events, the network capacity can be heavily reduced, thus resulting in energy savings.

1.2 Differences between classical WSNs and industrial WSNs

There are important differences between classical WSNs, which are addressed by the protocols discussed in Section 1.1, and the industrial WSNs which are addressed in this work. As previously mentioned, such differences involve both the requirements and the architecture of the networks. The most relevant aspects concerning the different architecture and requirements are discussed in Sections 1.2.1 and 1.2.2, respectively.

1.2.1 Architecture

Classical WSNs are independent deployments of ad-hoc networks, which typically run just one collaborative monitoring application. They typically comprise a large number of nodes capable of monitoring a certain phenomenon (e.g. temperature, luminosity, etc.), processing the relative data and exchanging it amongst themselves as well as with a base station via a Sink node. The nodes in a WSN are generally located in the proximity of or inside the phenomenon they are monitoring. The environments involved are often remote or hostile to humans and in some cases the nodes are placed in their environment in ways that are far from being ordered and

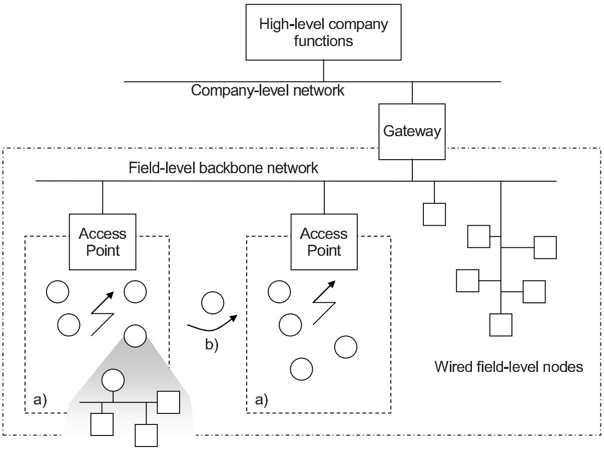


Figure 1.1: Architecture of a typical industrial network.

predictable. A WSN therefore has to be autonomous, and able to configure itself automatically and to function without human intervention for as long as possible. Moreover, typical WSNs cannot rely on any other infrastructure.

Industrial WSNs, on the contrary, are always coupled with wired industrial networks, such as fieldbuses or industrial Ethernet. The reason is that wireless networks differ substantially from wired fieldbuses in two respects. Firstly, a wireless channel experiences much higher bit error rate than a wired one. Secondly, the wireless medium is shared with other networks, thus it is subject to external interferences. As a result, it is not always feasible to replace wired networks with current wireless technologies. Rather, industrial WSNs integrate with wired networks, as they can greatly improve flexibility and open new possibilities for industrial applications. These include deployment of sensor nodes in settings where realizing a wired network is not feasible or it would need prohibitively expensive safety certifications. As shown in Figure 1.1, typical industrial networks are hybrid and exhibit a hierarchical architecture, with one or multiple wired segments and one wireless segment which is used for the less critical monitoring and control tasks and/or to interconnect multiple wired segments. The main consequence is that industrial WSNs do not need to be independent and autonomous like classical WSNs. Rather, industrial WSNs can exploit the presence of a wired infrastructure in order to provide better performance in terms of both latency and predictability.

1.2.2 Requirements

As discussed in Section 1.1, the most important requirement in typical WSNs is *energy efficiency*, followed by the *self-configuration* and *self-adaptation capabilities* which are required in unattended deployments. Other common requirements are *high scalability* and *low cost* of the nodes. All these characteristics are appreciated also in industrial WSNs, especially scalability. In fact, large factories may include a very large number of nodes and high node density. Moreover, while such networks should cover a large area the radio coverage of sensor nodes is typically small. As a result, sensor nodes must be able to perform routing in order to interconnect multiple wireless cells. However, in order to make WSNs suitable for factory communication, there are other requirements that have to be met.

Predictability is probably the most important requirement for industrial communications. An industrial network shall provide tools allowing the end user to simulate his network environment and determine in advance end-to-end performances of the system such as end-to-end latency, the relevant absolute jitter and network throughput. For this reason, an industrial WSN has to make it possible to obtain (at least statistical) upper bounds on the delivery time for application data over the network.

Resistance to the interferences is also a major concern. In fact, industrial WSNs operate in harsh environments with large metallic parts (machines) and should consider factors like high temperature, dust, vibrations, humidity, metallic surroundings, etc. The network should tolerate potential interferences and high variation of the radio signal strength.

Finally, it is worth recalling that industrial WSNs cannot completely supersede wired factory communication systems, because they cannot compete with wired networks in terms of performance and predictability. Rather, the aim of industrial WSNs is to complement them and to allow a flexible wireless extension of preexisting wired networks. As a consequence, another important requirement of industrial WSNs is the *ability to integrate with wired industrial networks.*

1.3 Research challenges and possible solutions

All the above mentioned requirements represent research challenges, to which current literature has provided only partial solutions, if any. Because of the variety and the complexity of such requirements, it is not possible to address all of them within one single communication protocol. On the contrary, a suite of protocols working at different layers is needed which collaborate to achieve common goals. A possible solution is the application of the *Divide and Conquer* paradigm, where each layer of the protocol stack addresses just one requirement, or a few of them, while the careful combination of multiple techniques working at different levels leads to the desired results. This work goes in that direction, providing different techniques and protocols working at different layers of the protocol stack and addressing once at a time the requirements discussed in Section 1.2.2.

Chapter 2 addresses the physical layer, in particular the robustness of IEEE 802.15.4 networks to cross-channel interference. The chapter provides a better understanding of cross-channel interference in co-located IEEE 802.15.4 industrial networks and proposes a general methodology for the assessment of IEEE 802.15.4 performance under different cross-channel interference conditions. This methodology allows a network designer to perform on-site but accurate assessments and can be easily deployed in real industrial environments to perform measurements directly in the environment-under-test. Finally, a case study based on COTS IEEE 802.15.4 devices is presented to show how to apply our methodology to a real scenario and to discuss the results obtained with one or multiple interferers and varying some MAC level parameters.

Chapter 3 addresses the scalability problem at the MAC layer. The chapter proposes a novel multi-channel approach to the beacon collision avoidance problem. The novel approach enhances scalability of cluster-tree IEEE 802.15.4 networks while allowing contention-free scheduling, thanks to the use of multiple radio channels in the same network. Moreover, a Multichannel Superframe Scheduling (MSS) algorithm is presented that, following the multichannel approach, can outperform the algorithms offered by current literature, which use just one channel.

Chapters 4 and 5 address the problem of reducing energy consumption while introducing a predictable delay and follow an innovative approach that is based on a topology management protocol which resides between the MAC and the routing layer of sensor nodes. The topology management protocol presented in Chapter 4 rules both the active/sleep cycle of sensor node, taking care of the energy efficiency, and data transmission schedule, avoiding collisions and ensuring that the delay introduced by the sleep cycles is predictable. It also provides routing fidelity, but it follows a static approach. Chapter 5 extends such work, presenting a dynamic topology management protocol that overcomes the limitations of the static approach introducing support for event-driven data transmissions and node joining at run-time and providing a novel adaptive technique for energy balancing among nodes to further increase network lifetime. The chapter provides a detailed description of the dynamic protocol and simulation results on network lifetime and routing performance with comparative assessments.

Finally, Chapter 6 addresses predictable data delivery at the Routing layer and integration between the industrial WSN and the wired industrial infrastructure. In particular, this chapter proposes a network architecture and a communication protocol, called Circular Chain Data Forwarding (CCDF), that not only supports integration with a wired industrial infrastructure, but also takes advantage of such integration to deliver real-time performance, even to nodes that could not be directly covered by a sink. To achieve this goal, a chain-based mechanism is used, which integrates data forwarding with the channel access strategy. Theoretical results, confirmed by in-depth simulations, are provided to analyze the performance of the protocol in the case of both error-free and error-prone channels.

Chapter 2

Assessment of cross-channel interference in IEEE 802.15.4 networks

The IEEE 802.15.4 protocol [21,22] is generally considered as one of the most promising options for low-cost low-power communications in industrial environments [23]. As industrial WSNs usually comprise a large number of sensors and actuators and typical applications require small delays, scalability is a key issue [24]. A viable solution is splitting a large network into several smaller networks, interconnected through a wired or a wireless backbone. In order to support the requirements of industrial applications and obtain reliable communications, the interference between the different networks has to be taken into account. A possible option is the use of different radio channels for the different networks, thus implementing a cellular architecture. A similar approach has been presented in [25]. The IEEE 802.15.4 standard is suitable for this solution, as the physical layer can use up to 26 different radio channels on three different bands (although the majority of Commercial Off-The-Shelf (COTS) IEEE 802.15.4 radios only support the 16 channels defined on the 2.4 GHz band). However, when a similar solution is implemented, it is important to estimate the effect of cross-channel interference. Although in IEEE 802.15.4 there is no overlapping between adjacent radio channels, the work [26] shows that actually some interference is present, due to spurious emissions caused by the O-QPSK coding. In that work, cross-channel interference is evaluated through both experimental results and theoretical considerations on the coding of the IEEE 802.15.4 physical layer. The technique described in this chapter is based on the work in [26], but extends it in several respects. While [26] mainly discusses the results of measurements performed in a specific IEEE 802.15.4 deployment, here the following contributions are provided:

- A discussion on the current "best practices" to cope with cross-channel interference in IEEE 802.15.4 networks, that pinpoints the main limitations of such approaches.

- A generic methodology for the evaluation of cross-channel interference between IEEE 802.15.4 networks in industrial environments, which allows for on-the-fly but accurate on-site assessments. As this methodology relies only on standard IEEE 802.15.4 primitives and components, it is generic and easy to adopt in real deployments.

- A case study, which shows how to apply the proposed methodology to a real scenario. The case study platform, which is based on COTS IEEE 802.15.4 devices, is described and the results obtained are discussed.

This chapter is organized as follows. Section 2.1 gives an overview of relevant literature. Section 2.2 introduces the problem of cross-channel interference in 802.15.4 networks and the current best practices suggested by IEEE 802.15.4 hardware manufacturers. Section 2.3 describes the methodology proposed in this chapter and the associated testbed. Section 2.4 presents and discusses the results of measurements performed on a case study platform based on COTS IEEE 802.15.4 devices. Finally, Section 2.5 gives some concluding remarks.

2.1 Coexistence of wireless networks

Interference between wireless networks has been extensively addressed in recent literature. In 2003, the IEEE published a document of recommended practices [27] in which the problem of co-existing 802.15.1 and 802.11b networks is analyzed through both simulations and analytical models. The problem of wireless link assessment in industrial environments is addressed in [28] for IEEE 802.11 communications. Theoretical and experimental works exist which address interference in Bluetooth networks used in industrial environments [29, 30]. Delay performance and the packet loss probability caused by a number of co-located interfering piconets are analyzed in [31] and an upper bound on the packet error rate is analytically derived. In [32] the effect of transient interference under TDMA protocols is evaluated for dependability purposes. In [33] the impact of an IEEE 802.15.4 network on an IEEE 802.11b one is studied. In [34] the influence of IEEE 802.11 on IEEE 802.15.4 is analyzed and a model to estimate the packet error rate obtainable in interference conditions is given. In [35] the model is extended, deriving the packet error rate of IEEE 802.15.4 networks under combined interference from WLANs and Bluetooth networks. Empirical evaluations of the co-existence of IEEE 802.15.4 with IEEE 802.11, Bluetooth and microwave ovens are presented in [36]. The work [37] assesses the impact of CSMA/CA parameters on the IEEE 802.15.4 performance in the presence of interference coming from IEEE 802.11, Bluetooth or the same IEEE 802.15.4, but it emulates a simple industrial control task to evaluate application-specific performance and does not aim at providing a general method to obtain accurate on-site performance assessments. In addition, it does not deal with cross-channel interference, as the interfering IEEE 802.15.4 networks are deployed in the same channel. In [38], a simulator that takes into account coexistence issues between IEEE 802.11 and IEEE 802.15.4 is used to calculate the packet error rate of both networks. Concerning cross-channel interference, various experimental studies exist, which mainly focus on the IEEE 802.11 protocol family [39, 40]. In [41] the impact of cross-channel interference and other factors (such as beacon frames and overhead caused by both access points and WLAN adapters) on the performance of IEEE 802.11g networks is experimentally analyzed. In [42] the authors investigate the correlation between spatial distance and channel spacing to deal with interference between concurrent transmissions in a multichannel WSN. Their results, although interesting, are hardware-specific, as they refer to a proprietary platform. Moreover, the authors do not target a real industrial scenario, so their results are not directly applicable to IEEE 802.15.4 industrial networks. No methodologies are given to obtain application-related figures, such as packet error rate or latency values, through on-site assessments.

Cross-channel interference in IEEE 802.15.4 networks is also addressed by some application notes [43, 44] relevant to specific devices (Texas Instruments CC2420 and Freescale MC1319x, respectively). Both technical notes address the receiver jamming resistance (i.e., the degree to which interferers will impact the receiver) and quantify the receiver performance in the presence of interferers through interference rejection measurements, which show the compliance of the addressed radios with the IEEE 802.15.4 specifications. However, all the measurements are performed in lab, connecting the transmitter and the receiver through cables and attenuators to eliminate all the other sources of interference. Furthermore, no in-air testing is performed in [43], while some in-air assessment is outlined in [44], but it is only a rough estimation of the interference rejection obtained with varying frequency offsets (< 25 Mhz or > 25 MHz, respectively) between the desired carrier and the interferer. On the contrary, the work [26] gives an insight on the effects of cross-channel interference in a specific IEEE 802.15.4 deployment, providing both analytical results and experimental measurements. Differently from [26], in this chapter we provide a generic methodology to accurately assess the effect of cross-channel interference in industrial IEEE 802.15.4 networks. Thanks to the combination of descriptive statistics and error propagation theory, our methodology allows to obtain not only a realistic performance assessment of real industrial networks through on-site measurements, but also the accuracy of packet loss and worst-case PER measurements in terms of confidence intervals. The proposed methodology is truly generic, as it only relies on a simple testbed that uses only standard IEEE 802.15.4 features and that can be easily deployed "on-site" in industrial environments.

2.2 On cross-channel interference in IEEE 802.15.4

The IEEE 802.15.4 physical layer defines three different radio bands, each with a different data rate and a different coding technique. Today, the most widely used is the 2.4 GHz band, which belongs to the ISM band. Sixteen different data channels are defined around the 2450 MHz frequency, each of them having a 2 MHz bandwidth. The distance between two adjacent channels is 5 MHz. Nevertheless, because of the Offset Quadrature Phase Shift Keying (O-QPSK) modulation used at the physical layer, a small fraction of the signal is spread as spurious emission outside the 5 MHz bandwidth, as shown in [26]. In order to limit cross-channel interference, the IEEE 802.15.4 specifications [21] impose a transmit power spectral density (PSD) mask, which defines the upper bounds on the average spectral power of a device measured with a 100 kHz resolution bandwidth in frequencies distant more than 3.5 MHz from the center frequency as 20 dB (relative to the peek) and -30 dBm (absolute limit), respectively. The IEEE 802.15.4 standard also defines the minimum jamming resistance for the receiver so that the Packet Error Rate (PER) is less than 1% as 0 dB for an interferer in the adjacent channel and 30 dB for an interferer in the alternate channel[1], respectively. According to the IEEE 802.15.4 standard, such a jamming resistance should be calculated using 20 byte packets with a desired signal power of -82 dBm and only one interferer. The procedure to compute the jamming resistance for an IEEE 802.15.4 transceiver according to the standard is described in some application notes, such as [43] and [44], which refer to specific devices. In [44] the jamming resistance obtained from in-lab measurements is used to calculate the minimum distance of the interferer so that the PER keeps under 1%. This relation is obtained using the path loss equation to calculate the power of the desired signal given the distance between the transmitter and the receiver. Then, using the inverse formula, the distance of the interferer that results in the desired jamming resistance value is obtained for the given transmitter/receiver distance. We computed the jamming resistance for the adjacent channel as described in [44], using three Maxstream XBee modules, equipped with the same transceiver as in [44]. The interferer transmitted a continuous[2] modulated pattern of pseudo-random data. Differently from [44], we performed in-air measurements in a real scenario reproducing the working conditions typically found in industrial contexts and used the path loss equation in [21] to compute the actual attenuation of the signals, i.e.,

$$L_p\left(d\right) = \begin{cases} 40.2 + 20 \log d, & d < 8m \\ 58.5 + 33 \log \frac{d}{8}, & d > 8m. \end{cases} \tag{2.1}$$

The distance between transmitter and receiver was fixed to 2 m. The results of our measurements, given in Table 2.1, show that the jamming resistance increases with the distance between the interferer and the receiver. In all our measurements the obtained jamming resistance is far better than the minimum value of 0 dB imposed by the standard. In the case of 1.5 m distance, we were not able to calculate the exact value, as the obtained packet

[1] The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel [21, 22].

[2] Using the spectrum analyzer in air, a 98.8% duty cycle was assessed.

Interferer Distance (m)	1.50	1.25	1.00	0.63	0.50
Jamming Rejection (dB)	>23	23	19	15	8

Table 2.1: In-air jamming resistance obtained with 2 m distance from transmitter to the receiver.

error rate (PER) was less than 1% even with the maximum interferer power. This means that the jamming rejection was certainly higher than the 23 dB value obtained with a 1.25 m distance from the interferer. These results also show that there is a significant difference between the jamming resistance values obtained through in-lab measurements, shown in [44], and the ones measured on site. We conclude that current best practices that use in-lab jamming resistance and the path loss formula to obtain the minimum distance between the PER and the interferer give only a rough information to the network designer. For this reason, it is advisable to perform testing in the real working scenario under realistic conditions. However, to perform on-site accurate assessments on cross-channel interference, a suitable methodology has to be carefully devised and the corresponding experimental testbed has to be deployed. This is exactly the main contribution provided by this chapter.

2.3 Testbed and Methodology

The approach proposed in this chapter requires a simple testbed made up of portable and affordable components. The testbed consists of a personal computer (PC), in charge of controlling the transmitter (T) and receiver (R) nodes through a serial connection, and one or more interferer nodes (N_i) configured in such a way to autonomously send frames on different channels at the same time. An auxiliary receiving antenna connected to a portable spectrum analyzer (S), if available, may be useful to detect external sources of interference. Such a testbed is generic, as it does not require either a particular kind of radio modules or a specific environment, as no assumptions on the environment are made (e.g., on the presence/absence of obstacles, on their shape, material, etc.). It is possible to deploy such a testbed using any IEEE 802.15.4 COTS modules, as long as they support the standard IEEE 802.15.4 primitives.

An ordinary PC is connected to the board on which the wireless nodes

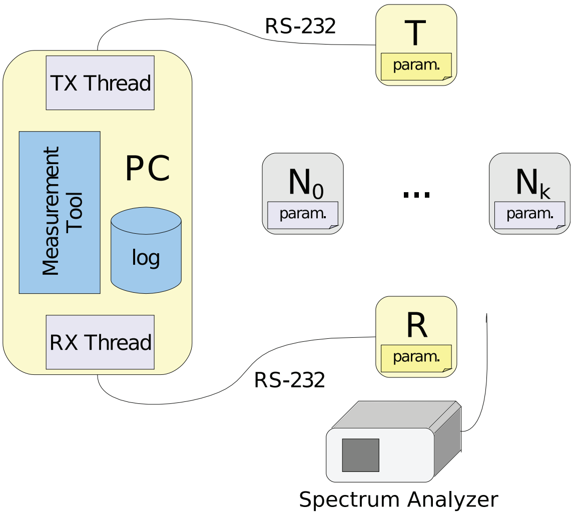


Figure 2.1: Structure of the testbed.

reside through a USB or RS232 port and can send commands to either modify the network parameters or send data frames or read received frames. As in typical industrial scenarios the presence of periodic interfering packets is a realistic assumption [45], in our testbed interferer nodes periodically transmit the same packet for the duration of the measurement campaign, without the need to attach a PC to the interferer nodes.

2.3.1 Methodology

The choice of the parameters to be taken into account in the measurements is based on the sensitivity assessments made in [26], where the sensitivity of the testbed to the RSSI value returned by the IEEE 802.15.4 module versus distance and the packet loss ratio versus interference power level were analyzed. The results obtained showed that the experienced RSSI values are directly related to the distance and are also quite stable, as the coefficient of variation was below 2% in almost all the performed measurements. This agrees with other studies on the characterization of IEEE 802.15.4 link quality and signal strength, such as [46]. However, in [26] it was also shown that RSSI is not a good indicator of the link quality in noisy environments, as it does not distinguish between the signal and interference power. Moreover, on the factory floor meeting the application-related constraints is mandatory and thus drives the WSN design choices. As a result, reliability and timeliness are the crucial requirements to be taken into account. For this reason, the performance indicators adopted here are latency, packet loss and worst case packet error rate (PER). They can be obtained as follows:

Latency estimation

When dealing with wireless industrial communications, given the typical time-critical requirements of the exchanged traffic, latency is an important parameter to be assessed. An estimate of the one-way latencies of data frame transmissions can be obtained by comparing the logs of sent and received frames. To guarantee the temporal coherence of timestamps, the measurements have to be performed on the same PC, therefore with a common clock reference. Another important detail to be considered when evaluating latencies is that, as the transmitter and receiver modules are connected to the PC through a serial connection, an additional latency is introduced in both the transmission and the reception of a frame. As the amount of data to be transmitted is known and there is no contention for the medium access, this delay can be estimated and subtracted from the one-way delay. In particular, if a L_{data} octet data frame has to be transmitted through the wireless connection, and a L_{ov} octet overhead is needed to send the transmission (or reception) command, the time spent for the transmission (or the reception) of a frame over the serial link is

$$T_{RS232} = \left\lceil \frac{8\,(L_{ov} + L_{data})}{L_{byte}} \right\rceil \frac{(L_{start} + L_{byte} + L_{parity} + L_{stop})}{D_{RS232}} \tag{2.2}$$

where L_{byte} is the number of bits in every frame of the RS232 protocol, L_{start}, L_{parity} and L_{stop} are the number of start, parity and stop bits respectively, and D_{RS232} is the baud rate of the serial connection. Considering that the propagation time can be neglected, the latency can be calculated as

$$T_{frame} = t_{rx} - t_{tx} - T_{RS232_{rx}} - T_{RS232_{tx}} \tag{2.3}$$

where t_{rx} and t_{tx} are the time instants of the frame reception and transmission, respectively, while $T_{RS232_{rx}}$ and $T_{RS232_{tx}}$ are the overheads for transmitting and receiving a frame, respectively. However, the delay calculated with (2.3) includes some overheads introduced by the operating system and communication controllers. To limit such a jitter, it is advisable to reduce the computational load on the PC as much as possible and to keep in RAM the proper data structures to track the sending and receiving of data frames, so that the jitter caused by blocking I/O functions is avoided. Moreover, when a very high degree of accuracy in delay measurements is required, it is advisable to run the software under a real-time kernel.

Packet Loss estimation

In our testbed, experiments are run by repeatedly sending packets from the transmitter T to the receiver R and counting the times a packet sent by T is not received by the receiver R. Suppose that, given a defined transmitting power and a defined kind of interference, each packet has a fixed probability $(1 - PL)$ to be successfully received by the destination, and a probability PL to be lost. This assumption can be considered realistic in a well air-conditioned environment with no moving obstacles [46]. Under this assumption the packet loss event will happen according to a Bernoulli distribution, where the PL parameter represents the probability to have a packet loss.

The best approximation of the PL probability is given by the sample mean $\widehat{PL} = \frac{1}{n}\sum_{i=1}^{n} X_i$, where n is the number of packets transmitted in the whole experiment and X_i are the results of a single packet transmission (1 means that the packet has been lost, 0 means that the packet has been successfully received). Moreover, if the number of packets that are sent in each experiment is large, the confidence bounds for PL can be obtained through the formula

$$PL = \widehat{PL} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\widehat{PL}(1 - \widehat{PL})}{n}} \tag{2.4}$$

where $z_{1-\frac{\alpha}{2}}$ is the z-score of the standard normal distribution that determines the desired interval of confidence [47], e.g., 1.96 for 95% confidence.

Worst Case PER estimation

To obtain the worst-case packet error rate, a constant cross-channel interference should be considered. As it is fully described in [26], even with an interferer node that transmits data packets periodically, our testbed makes it possible, under proper assumptions, to approximately assess the PER under constant interference conditions. Considering an IEEE 802.15.4 network working in non-beacon enabled mode, let T_i be the period of the interferer

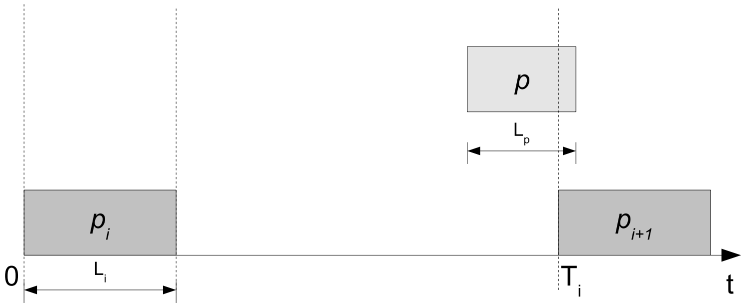


Figure 2.2: Model for overlapping transmission probability.

node, L_i the interferer frame length and L_p the length of the packet we are interested in, such that $L_p \leq L_i$, and $L_i \ll T_i$. Referring to Figure 2.2, a packet p does not overlap with a packet of the interferer if $L_i < t < T_i - L_p$, where t is the arrival time of p. Therefore, the probability that no overlap will occur between these packet is $\left(T_i - L_p - L_i\right)/T_i$. So, the probability that a packet will overlap with an interferer data frame is

$$P\left(C\right) = \frac{L_p + L_i}{T_i}. \tag{2.5}$$

Let L be the lost packet event and C the collision event. Assuming L as our event, and C together with “any other cause than a collision” as our set of mutually-exclusive and all-inclusive causes of the event, we can calculate the PER using the Bayes theorem. Under the assumption that every transmission overlap causes a collision event, irrespective of the fraction of packet overlapping, we have

$$\text{PER} = P\left(L|C\right) = \frac{P\left(C|L\right) \cdot P\left(L\right)}{P\left(C\right)}. \tag{2.6}$$

In Formula (2.6), $P\left(L\right)$ is exactly the packet loss obtained through our measurements, $P\left(C\right)$ is the probability obtained in (2.5) and $P\left(C|L\right)$ represents the probability of a packet being lost because of a collision given that the packet is lost. A packet loss may be due to either a collision with the interferer node or a different cause (anything other than a collision). We can assess the packet loss ratio obtained in the same conditions but without any interferer node, namely PL_0, and calculate $P\left(C|L\right)$ as $1 - \text{PL}_0$. If PL is the packet loss ratio obtained with those parameters and PL_0 the packet error rate obtained without any interferer, the worst case PER, i.e., the PER in the case a packet collides with an interferer packet, can be approximated as

$$\mathrm{PER} = \frac{T_i(1 - \mathrm{PL}_0)\,\mathrm{PL}}{L_p + L_i}. \tag{2.7}$$

The estimation of the worst case PER for a given scenario can be useful in contexts where a defined reliability has to be maintained, such as industrial automation. However, in order to be useful, even these results should include the confidence intervals. As there are two different parameters in (2.7) that are derived from measurements, the error propagation has to be calculated using the error propagation theory. As an imprecision in PL_0 may also affect the measurements of PL, it is safe to use the conservative estimation of the confidence interval for a product, given by the sum of the relative confidence intervals of the two factors [47]. As a result, if $u_c(PL)$ and $u_c(PL_0)$ are the confidence intervals for PL and PL_0 respectively, a conservative estimation of the confidence interval is

$$u_c(PER) = \frac{T_i}{L_p + L_i}\left[PL \cdot u_c(PL_0) + u_c(PL) \cdot (1 - PL_0)\right]. \tag{2.8}$$

In order to assess the effectiveness of our methodology, we ran some experiments using our testbed. The experimental results obtained, as it will be shown in the case study addressed in Section 2.4, are compliant with our estimations according to (2.7) and (2.8).

2.4 Case study and experimental results

Using our testbed, a broad series of in-air measurements to experimentally assess the impact of cross-channel interference under different operating conditions can be run. In the following, the methodology proposed in the previous section is explained through a case study. Several test scenarios were built in order to reproduce the typical working conditions of industrial environments. Results obtained in these scenarios with one or multiple interferers will be presented.

2.4.1 The IEEE 802.15.4 platform

In our case study, measurements were performed using the MaxStream XBee / XBee Pro [48] modules. These nodes follow the IEEE 802.15.4

standard specifications and work exclusively within the 2.4 GHz ISM band. Both these two types of modules are equipped with a MC9S08GT60 microcontroller and an MC13193 802.15.4 RF transceiver. They are pin-compatible, so for the connection with the PC the same development boards, i.e., MaxStream XBIB-U-DEVs and MaxStream XBIB-R-DEVs, have been used. The only difference between these modules is the transmitting power, which is up to 0 dBm for the XBee modules, while it is up to 18 dBm for the XBee Pro ones. The original XBee firmware (ver. 10A5) in API mode [48] was used in the transmitter and the receiver node, while for the interferer we developed a customized firmware using the Freescale Codewarrior for HC(S)08, the implementation of IEEE 802.15.4 provided by the Freescale Beekit and the XBee Development Toolkit publicly available in [48]. However, when the Freescale IEEE 802.15.4 implementation is used on the XBee Pro modules, the maximum transmitting power does not coincide with the one of 18 dBm obtainable using the original firmware. For this reason our customized firmware was run only when the continuous transmit mode was needed, while in all the other cases the original XBee firmware in the Transparent Operation mode was used.

To coordinate the IEEE 802.15.4 wireless nodes and the PC, a specific software was developed. The software allows us to set various parameters of the nodes that make up the testbed (i.e., transmission period, data packet size, channel, presence of the interferer, etc.), as well as to drive the transmitter node and monitor the traffic of a generic receiver node. A spectrum analyzer is used for monitoring purposes, to ensure that no interference from uncontrolled wireless devices occur during our measurement campaigns.

In all the experiments the interferer nodes transmit periodic packets, while T transmits packets “almost” periodically, i.e. with an interarrival time of $100 \pm \delta$ ms where δ is a random value chosen in the interval $[-5, 5]$, introduced to avoid the occurrence of repetitive patterns of interference. On the other hand, no jitter was explicitly added to the interferer period, to keep a fixed collision probability. The default settings of all the nodes in our testbed, when only one interferer is present, are shown in Table 2.2. Both transmitter and interferer nodes always use the non beacon-enabled mode. The 16-bit addressing mode is used, so a 17 byte header has to be added to the payload shown in Table 2.2. If not stated otherwise, the T and R nodes are fixed 1 m apart from each other, while the interferer nodes are in the middle, at a distance of 0.5 m from R. No obstacles are present between nodes. All the experiments comprise a large number of samples

	Transmitter	Receiver	Interferer
TX power	0 dBm	0 dBm	0/18 dBm
CCA Threshold	−44 dBm	−44 dBm	−44 dBm
macMinBe	0	0	0
Channel	11	11	12
Tx. Period	100 ms	n.a.	Variable
Jitter	5 ms	n.a.	No
Payload	30 bytes	n.a.	100 bytes
ACKs	No	No	No

Table 2.2: Basic Testbed configuration

(3000 packets sent by T, if not specified differently) and were performed in a real-life indoor environment. We tried to minimize all the other sources of interference, e.g. from WLANs operating nearby, by shutting down any electronic equipment under our control capable of emitting radio waves in nearby areas. Moreover, we monitored the environment through a Wi-Spy 2.4x portable spectrum analyzer, in order to assure that no interference from uncontrolled wireless devices occur during our experiments.

2.4.2 Preliminary Assessments

In order to verify that the obtained results will not be affected by hardware failures or imperfections, it is important to perform preliminary testing of the testbed components. Several components may lead to biased results, e.g., packet loss in the serial line connecting the PC to either the transmitter or the receiver, imperfections on the transceivers (or non-compliance to the IEEE 802.15.4 standard) or even different orientations of non-omnidirectional antennas.

In our case study, we used XBIB-R-DEV boards connected to the PC through a USB-to-serial adaptor featuring a PL-2303HX chipset and XBIB-U-DEV boards directly connected to the PC through a USB port. In both cases, the serial connection was tested by transmitting 10000 packets in the best possible conditions for the wireless channel, i.e., T and R were placed at 1 m with no obstacles in between and without any interferer. They were set to use a 0 dBm transmitting power and acknowledged transmissions, and the spectrum analyzer was used to verify that no other interference occurred during the test. In such conditions, there was no packet loss.

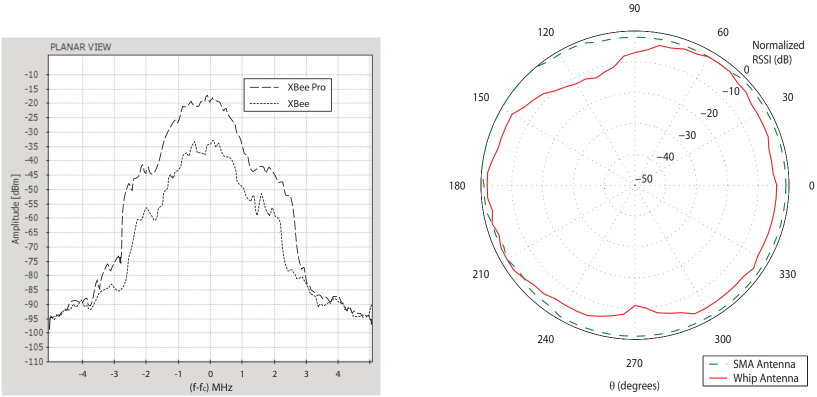
The testing of the MC13193 transceiver embedded in the XBee and XBee Pro modules is addressed in [44]. Nevertheless, we verified the compliance to the standard specifications of our devices, in terms of both the PSD mask and jamming resistance. The results in terms of jamming resistance were already discussed in Section 2.2. The PSD mask was measured in air setting the 100 kHz resolution bandwidth as indicated in [21], with both an Anritsu MS2668C and a Wi-Spy 2.4x portable spectrum analyzer. The output of latter is shown in Figure 2.3a. In that figure it is easy to notice that, for frequencies distant 3.5 MHz or more from the carrier, the measured signal never exceeds the -20 dBm relative threshold neither the -30 dBm absolute one. As a result, even the transceiver successfully passed the compliance test.

Two different types of antennas were used in our testbed, i.e. standard SMA-connectorized monopole antennas and integrated whip monopole antennas. In particular, a standard SMA-connectorized antenna was used for the receiver, while the transmitter and the interferer nodes were equipped with the integrated whip antennas. The measured radiation pattern of both types of monopole antennas used are publicly available on [49] and are closed to the ideal ones, i.e., they are almost omnidirectional (ripple of ±10 dB), in the monopole H-plane as expected. However since many external factors may influence the radiation pattern, we performed our pattern measurements in our testbed environment. We used XBee modules for both the transmitter and the receiver. The transmitter node was placed at the same height, but 2 meters away from the receiver. The transmitting power was set to -2 dBm. The receiver was kept fixed, while the transmitter angle was changed in steps of 5 degrees scanning the antenna H-plane. For each angle, 100 sample packets were sent, and the average value was taken. The received power is depicted in Figure 2.3b, which shows that:

1. with the same nominal transmitted power, the received power level (RSSI) was slightly higher when a standard SMA-connectorized antenna was used;

2. both types of antennas have radiation patterns that, with a fairly good approximation, can be considered omnidirectional (ripple of about ±6 dB).

The last result is quite relevant, as it indicates that small angle variations that might be introduced by rotating the interferer nodes do not have a remarkable effect on the power received by the receiver node.



(a) Power Spectral Density of XBee and XBee Pro modules

(b) Radiation diagrams (received power under different angles in the H-plane)

Figure 2.3: Preliminary Assessments

2.4.3 Worst Case PER validation

After the preliminary assessment, some experiments were run through our testbed to assess the effectiveness of the model for the estimation of the worst case PER. The default configuration of our testbed was used with an XBee Pro as the interferer node. The period of the interferer was changed, and for each value both the experienced PL and the expected $PER \pm u_c(PER)$ were obtained according to formulas (2.7) and (2.8). In addition, to experimentally assess the worst case PER, we used our modified firmware that sends continuously a data frame, so that the channel utilization is close to the worst case, i.e., 100% channel utilization. The results of this experiment are shown in Figure 2.4, where the first value (marked as “none” on the x-axis) is the one experienced (i.e., measured) without interference, while the last one (marked as “Continuous TX” on the x-axis) is the value experienced with continuous transmissions from the interferer (about 98.8% duty cycle). Notice that, in the latter case, no expected PER is given, as the worst case PER coincides with the PL experienced with continuous interfering transmission. Figure 2.4 shows that the number of lost packets increases with the decreasing period of the interferer node. This is because the probability that a packet is lost is higher when its channel occupancy

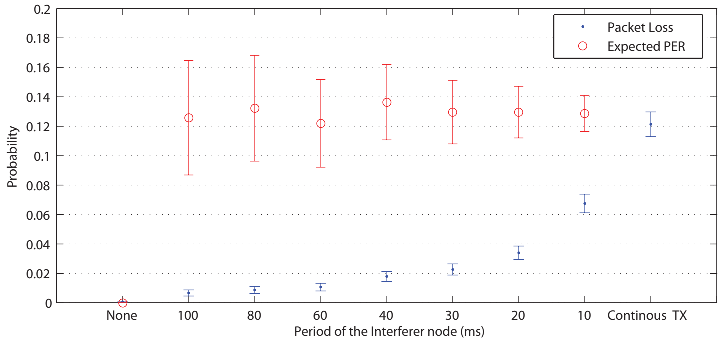


Figure 2.4: Packet Loss and expected PER versus the varying transmission period of the interferer node.

increases. However, the average PER values are very similar in all the trials. Moreover, the analytical PER matches the experimental one. This gives a significant evidence of the effectiveness of the model we used to calculate the worst case PER, although the size of the 95% confidence interval is larger when the period of the interferer is large. This is expected, as the confidence interval is proportional to the interferer period (T_i).

2.4.4 Interference by a single node

To assess the level of interference on a communication caused by an interferer working on an adjacent channel we used a simple scenario with a transmitter, a receiver and an interferer located 1 m apart from each other, each of them being the vertex of an equilateral triangle with a side of one meter. The configuration of the transmitter and receiver nodes is that in Tab. 2.2, where an interferer node transmits a payload of 100 bytes with a constant period of 100 ms. Both the transmitter and the interferer belong to the XBee family and their transmission power is 0 dBm. Six experiments were run, in which the transmitting channel of the disturbing node is varied. The results, shown in Figure 2.5 (with a 95% confidence interval), show that, although the power contribution on the adjacent and on the following channel is a very small fraction of that emitted by the interferer node, it is enough to determine a non-null packet loss, which means that cross-channel interference is non-negligible. We highlight that to calculate such confidence intervals, eq. (2.4) is not adequate, as it is not accurate for very large or

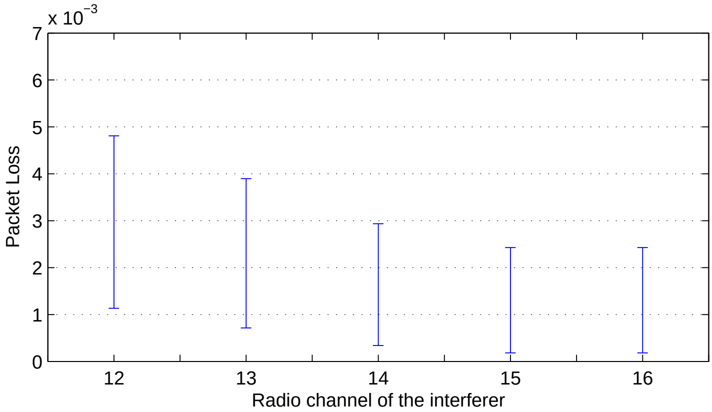


Figure 2.5: Packet Loss with equidistant nodes.

very low observed proportions [50], as in the case of the results obtained in this experiment. For this reason, the 95% confidence intervals in Figure 2.5 were obtained through a different method, given in [50], i.e., the lower and the upper bounds are calculated as $(A-B)/C$ and $(A+B)/C$, respectively, where

$$A = 2 \cdot n \cdot \widehat{PL} + z^2_{1-\frac{\alpha}{2}}, \tag{2.9}$$

$$B = z_{1-\frac{\alpha}{2}} \sqrt{z^2_{1-\frac{\alpha}{2}} + 4 \cdot n \cdot \widehat{PL}(1-\widehat{PL})}, \tag{2.10}$$

$$C = 2(n + z^2_{1-\frac{\alpha}{2}}). \tag{2.11}$$

The expected value of the worst case PER was calculated using equation (2.7), and the results are shown in Figure 2.6. Here we can notice that the effect of cross-channel interference clearly depends on the channel of the interfering node. This is an expected result, as spurious emissions of the interfering signal decrease with the channel offset. However, as long as the energy received by the receiver from the transmitter and interferer node is similar, only a limited packet loss occurs. In this case the worst case PER is always lower than 4.5%, that exceeds the 1% imposed by the standard for 0 dB jamming rejection. We underline that, as our purpose here was not to assess the jamming resistance, we did not use 20 byte packets as foreseen in the IEEE 802.15.4 standard, but 47 byte packets (payload=30 bytes, header=17 bytes). This explains why we obtained a PER>1%, although

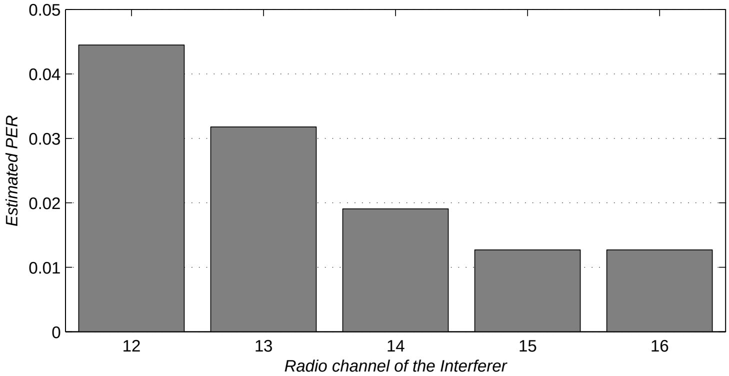


Figure 2.6: Expected worst case PER with equidistant nodes.

our devices are fully compliant with the standard specifications, as it was shown in Section 2.2.

The second question we tried to answer is what happens when the power of the interfering signal significantly exceeds that of the valid signal. This may occur either when there are nodes transmitting with a greater power than others, or when two distant nodes communicate in the presence of a close IEEE 802.15.4 network working on an adjacent channel. The two cases may also occur at the same time. In order to assess such a scenario we changed the interferer node to an XBee Pro module, that transmits with 18 dBm power, against the 0 dBm of the XBee. Considering the attenuation due to the path-loss, in this scenario the power of the signal received by the transmitter is about -40 dBm, while, as the shortest distance on which measurements were performed is 0.2 m, the power of the interfering signal on the adjacent channel is about -8 dBm. The difference between the power received from the transmitter and the interferer, henceforward referred as signal to interference ratio (SIR), causes a noticeable increase in the expected PER, as shown in Figure 2.7. When the distance between the interferer and the destination is about 1 m, corresponding to a SIR of about -18 dB, the PER is very low, but then it rapidly increases. With a 0.6 m distance (corresponding to a SIR of about -22 dB), the PER is over 20%, and when the distance decreases to 0.2 m (corresponding to a SIR of about -32 dB), the PER value is over 60%. Anyway, we can notice that maintaining the SIR above -20 dB, the worst case PER is lower than 10%,

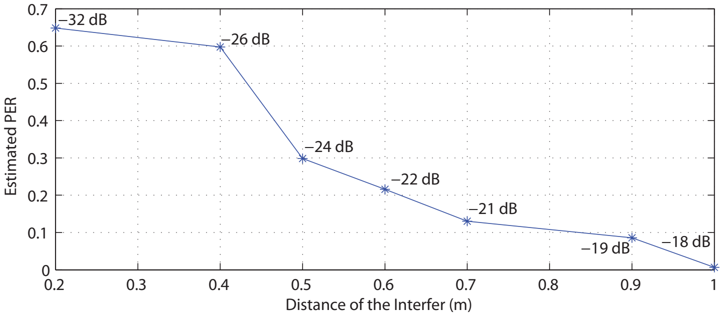


Figure 2.7: Estimated PER as a function of the difference between interferer and source received power level.

that is an acceptable value for most non-critical applications. However, we have to emphasize that this PER is pessimistic, because it assumes that each packet "collides" with a transmission in the adjacent channel. In such conditions, unacknowledged data transmission gives unfavorable results, but better results could obtained by enabling ACKs.

2.4.5 Interference from multiple nodes

The results of previous sections show how cross-channel interference can degrade network performance in terms of packet loss probability or worst case PER. Here we show the results obtained in our case study in the case of multiple interfering nodes. In order to understand the effect of multiple interferer nodes and multiple networks, here some scenarios featuring two or three interferer nodes have been set up. We configured these scenarios so that the receiver node receives exactly the same amount of energy from each interferer. To this aim, we connected the SMA-connectorized antenna of the receiver XBee module to a Wi-Spy 2.4x portable spectrum analyzer, while remaining in the exact location, and we performed small corrections on the location of the three different interferer nodes to obtain their spectrum masks alignment, as shown in Figure 2.8. We analyzed three different scenarios featuring multiple interferers, i.e.,

1. two interferers in the same adjacent channel;
2. two interferers in two different adjacent channels;

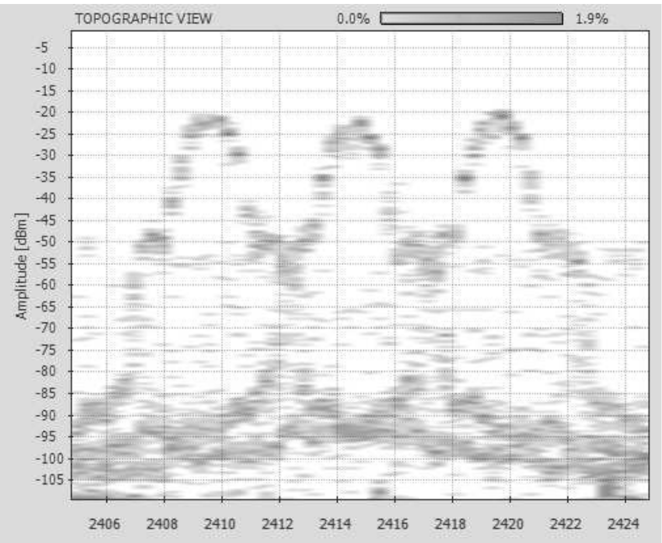


Figure 2.8: Spectra of the three interferer nodes seen by the receiver antenna.

3. three interferers in the same adjacent channel.

The total amount of data sent in all the multi-interferer scenarios is maintained the same. So, in the scenarios featuring two interferers each one sends a 100 byte payload with 40 ms period, while in the scenario featuring three interferers each one sends the same packet with 60 ms period. The results are compared with those obtained by using a single interferer on the adjacent channel, that sends the same packet with a period of 20 ms, 40 ms and 60 ms. In the first case the total amount of traffic is the same of the multi-interferer scenarios. The interferers transmitted independently, without any synchronization between them. In the scenarios where the interferers are in the same channel, the settings shown in Table 2.2 were used, i.e. channel 11 for the T and R, 12 for the interferer. In the scenario where both the adjacent channels are used, T and R transmit on channel 12, while the two interferers are on channels 11 and 13, respectively.

The results of these scenarios are shown in Figure 2.9, which shows that there is a clear correlation between the number of the interferers on the same channel and the packet loss probability. When the interferers are on the same channel , i.e. (1/20), (2/40) and (3/60), the effect of the

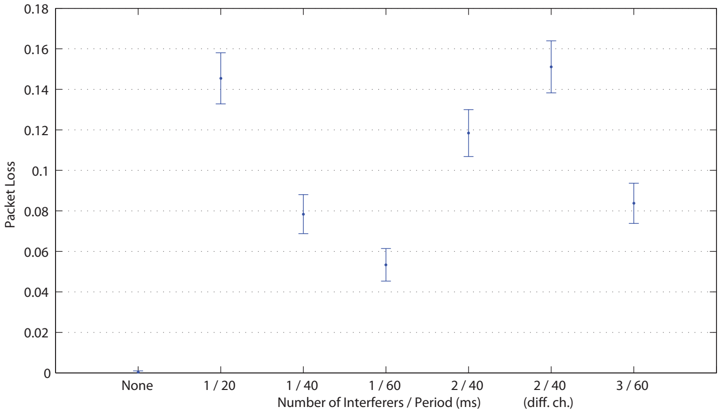


Figure 2.9: Packet Loss using multiple interferer nodes.

interference decreases as the number of nodes on the same channel increases. The reason for this result is found in the CSMA algorithm. As the number of nodes increases, the success probability of the CSMA algorithm decreases, due to the failed CCAs and the backoff delays. Every time the channel is found busy, the beacon exponent is increased, thus the average time between consecutive CCAs increases and so does the time between two packets sent on the medium. For this reason, being equal the total amount of traffic, the interference caused by the overall network decreases. This means that it is possible to have a pessimistic assessment on the performance degradation by transmitting the total amount of traffic from only one interferer, i.e., the one featuring the highest received power on the receiver. Such an assessment might be made when deploying an industrial network, in order to ensure that even in the worst conditions an acceptable network performance is still maintained. However, a different effect can be noticed in Figure 2.9 for the scenario with two interferers on both the adjacent channels, i.e., 2/40 (diff. ch.). Here the packet loss probability is very similar to the case of a single interferer with a 20 ms period (1/20), and it is about twice the one found with a single interferer with a 40 ms period (1/40). The reason for this is that, as the two interfering bands are 2-channels away from each other, they do not significantly affect each other. As a result, the transmissions of interfering networks are statistically independent, so the packet loss probability under their composite interference is the sum of the ones obtained with each single network, i.e., twice the packet loss probability of the scenario featuring a single transmitter with a 40 ms period.

2.4.6 Influence of MAC parameters

In this section we analyze the effect of the Clear Channel Assessment (CCA) threshold and the Minimum Backoff Exponent (macMinBe) in our case-study network.

CCA Threshold

In CSMA protocols, the CCA is performed before each transmission, in order to determine whether the channel is available for communication or not (it is busy). From the three CCA modes defined in [21], only the Mode 1 is supported by the XBee modules, i.e., the medium is reported busy if any energy above the CCA threshold is detected on the channel, thus we adopted this one. The CCA threshold of these modules ranges from −80 dBm to −36 dBm. We used the basic configuration of our testbed with a single XBee Pro interferer. The CCA threshold is changed from its minimum to its maximum value, in both the transmitter and the interferer nodes, in such a way that they always have the same threshold. In this way, none of them could take advantage of a higher threshold, otherwise, if the interferer node had a higher CCA threshold, it might send a packet while the transmitter in the same conditions would find the medium busy. The results, depicted in Figure 2.10, which gives the packet loss as a function of the CCA threshold, show that small changes of the CCA threshold do not have a significant impact on the packet loss. However, it is possible to notice that, with a 95% confidence level, the packet loss obtained using a −60-dBm threshold is lower than the one obtained with a −40-dBm threshold, and that the worst performance was obtained using the −80-dBm threshold. The reason is that under such conditions the (Mode 1) CCA is less reliable, because it is more likely that some noise in the channel causes the CCA to report a busy medium, and after a defined number of failed CCAs the packet is discarded. On the other hand, when the CCA threshold is set to high values, the medium may be erroneously reported as free, because the interfering power detected on the adjacent channel does not exceed the threshold.

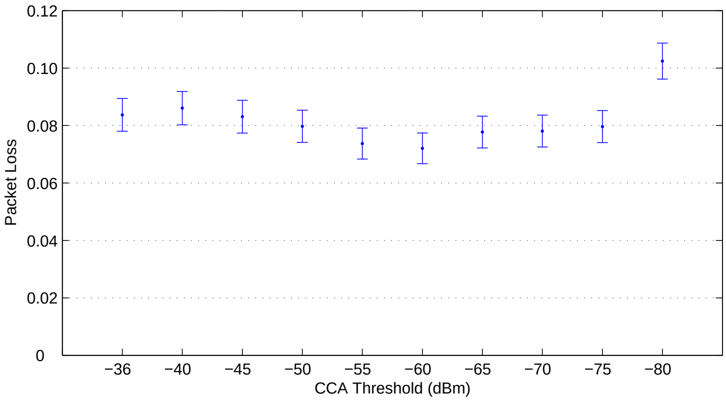


Figure 2.10: The effect of the CCA Threshold.

Minimum Backoff Exponent

The second MAC parameter we analyzed is the minimum backoff exponent, called *macMinBe* in the IEEE standard [21]. According to the IEEE standard, this value ranges from 0 to 3. When the macMinBe parameter is set to 0, the collision avoidance is disabled during the first iteration of the CSMA algorithm. As the collision avoidance and the macMinBe parameters may have a different impact on the network performance depending on the workload, three different traffic configurations for the interferer network, that use the same packet size but different periods, i.e., 20 ms, 40 ms and 80 ms, were assessed. The macMinBe parameter is varied from 0 to 3 on both the transmitter and the interferer nodes. In Figure 2.11 it is easy to notice that delay, calculated with (2.3), is strongly related to the macMinBe value. Here, are shown only four lines for the different macMinBe values, as the delays obtained using the same macMinBe but different interferer periods follow the same distribution. Obviously when the macMinBe is larger, the delay value also increases. However, with the 0 and 1 values the delay distributions have the same shape. On the other hand, when macMinBe is set to 2, also the distribution becomes wider, i.e., the deviation from the average value is larger. These results are expected, as the 1 exponent only enables collision avoidance, while larger backoff exponents spread random delays. In terms of packet loss, no significant difference was measured when the macMinBe parameter was changed. Based on our results, when short

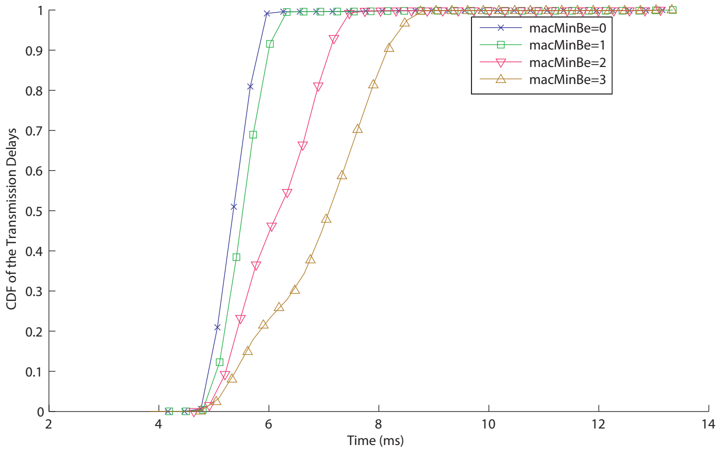


Figure 2.11: The effect of the Minimum Backoff Exponent on delay under cross-channel interference.

delays are sought, as in the case of industrial environments, it would be advisable to set the macMinBe for high priority real-time traffic to zero.

2.5 Concluding remarks

In industrial environments, the deployment of different co-located IEEE 802.15.4 networks on separate channels requires an effort on the designer side when sizing the whole system, in terms of carefully choosing the transmitting power and distances between nodes. Given the particular context dealt with, it is advisable to perform testing in the real working scenario, under realistic conditions, instead of relying only on the outcome of in-lab experiments. However, to perform on-site but accurate assessments on cross-channel interference, a suitable methodology has to be carefully devised and the corresponding experimental testbed has to be deployed. This chapter extensively addressed cross-channel interference with the objective of providing both a better understanding on this phenomenon and useful hints to plan the effect of cross-channel interference at design time. This chapter described a general methodology to evaluate cross-channel interference and a generic testbed devised for experimental on-site assessments in industrial networks. A case study is presented with the purpose of explaining how to set the testbed to assess the impact on cross-channel interference of one or multiple interferers and the effect of some MAC level parameters under cross-channel interference.

Chapter 3

Multichannel Superframe Scheduling for IEEE 802.15.4

The IEEE 802.15.4 MAC protocol [22] is designed for low-rate and low-power communications, it is particularly suitable for low-energy embedded devices. The protocol allows for varying nodes' duty cycles from 100% to a minimum of about 0.1%. Moreover the IEEE 802.15.4 features also collision-free time slots suitable for transmitting real-time traffic, called the Guaranteed Time Slots (GTS). The allocation of one or more GTSs allows to guarantee a defined bandwidth and a maximum access delay for a node. In [51] analytical relations that express the bandwidth and the delay guaranteed by n GTSs as a function of the superframe parameters are provided. Thanks to these relations, it is possible to obtain an upper bound on the delay of data transmission from a node to its coordinator. Such a delay, in the case of star topology, also coincides with the end-to-end delay. In [52] a methodology to extend such an analysis to a multi-hop cluster-tree network is presented. These analytical results show that an upper bound on the delay that a frame may experience from the source to the coordinator can be obtained from the network parameters. Such bounded delay capabilities enable the use of IEEE 802.15.4 cluster-tree networks to support time-constrained traffic, and make it attractive for industrial applications, such as remote sensor/actuator control in production automation and monitoring applications in factory automation. However, the IEEE standard does not solve the problem of beacon frame collisions in cluster-tree topologies, that may lead to loss of synchronization and disconnections, thus affecting communication reliability. Although in [53] it was shown that multi-hop beacon-enabled networks are feasible when the beacon order is larger than one, the distribution of coordinators is not very dense and the traffic is low, the non-negligible probability of losing the synchronization may not meet the stringent reliability requirements of typical wireless industrial networks [23, 45]. An algorithm to schedule the superframes of a cluster-tree network in a contention-free fashion, i.e., the Superframe Duration Scheduling (SDS) algorithm, was presented in [54]. While this algorithm solves the beacon frame collision problem, it limits the network scalability [55], as no parallel communication is allowed unless coordinators are distant enough not to collide.

This chapter describes a novel technique to schedule the superframes of cluster-tree IEEE 802.15.4 networks over multiple channels, so as to avoid beacon frame collisions as well as GTS collisions between multiple clusters. A novel algorithm is proposed, called a Multichannel Superframe Scheduling (MSS), that instead of operating only a time division between the different clusters, allows multiple clusters to schedule their superframes simultaneously on different radio channels. This way, it is possible to schedule sets of superframes which were non-schedulable using a single channel.

The chapter is organized as follows: Section 3.1 gives an overview of the standard IEEE 802.15.4 protocol, while Section 3.2 discusses the beacon frame collision problem in cluster-tree topologies. Section 3.3 gives a general overview of the current approaches to avoid beacon collisions, while Section 3.4 discusses the SDS algorithm. Section 3.5 gives the basic idea under the multichannel approach we proposed. Section 3.6 gives a detailed description of the MSS algorithm proposed in this chapter. Section 3.7 provides analytical considerations on the schedulability under MSS. Section 3.8 discusses the implementation issues of the proposed approach, while Section 3.9 describes our working implementation under TinyOS and describes some experimental results obtained through our testbed. Finally, Section 3.10 gives some concluding remarks.

3.1 The IEEE 802.15.4 protocol

An IEEE 802.15.4 network is composed by three different kinds of nodes: end devices, coordinators and Personal Area Network (PAN) Coordinator. End devices can produce data, but they have to interact necessarily with coordinators. On the contrary, coordinators may also perform network management and routing. Each network must have a PAN Coordinator, that is the main network controller. Nodes can be organized in three different topologies, i.e., star, peer-to-peer and cluster tree. In star topologies there is only a PAN coordinator and all the other nodes must communicate with it. In peer-to-peer (or mesh) topologies, each node can communicate with any other in its radio range. Finally, in cluster-tree topologies the network is organized in clusters, each one with a coordinator. Coordinators are hierarchically connected to form a tree, rooted at the PAN coordinator.

The IEEE 802.15.4 MAC protocol features two operating modes: a non-beacon-enabled mode, in which nodes access the channel using a classical (unslotted) CSMA/CA mechanism and a beacon-enabled mode in which time is subdivided in superframes, with a slotted CSMA/CA mechanism.

When nodes operate in beacon-enabled mode, they subdivide their time into Beacon Intervals, that are delimited by Beacon Frames periodically broadcast by each coordinator. Each beacon interval is divided into an active section, called superframe, and an inactive section, during which nodes do not transmit and may enter low-power states. The duration of these sections determines the nodes' duty cycle. The duration of the Beacon Interval (BI) and the Superframe Duration (SD) depends on two parameters, the Beacon Order (BO) and Superframe Order (SO), according to the relations

$$BI = aBaseSuperframeDutation \cdot 2^{BO} \tag{3.1}$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO}, \tag{3.2}$$

where *aBaseSuperframeDuration* is a constant defined in the standard [22] that denotes the number of symbols that form a superframe when SO is 0, and $0 \leq SO \leq BO \leq 14$.

The duty cycle (DC) of nodes is

$$DC = \frac{SD}{BI} = 2^{SO-BO} = 2^{IO}, \tag{3.3}$$

where IO is called an Inactivity Order.

Each superframe is divided into 16 equally-sized slots that form two different periods with different medium access mechanisms. They are the Contention Access Period (CAP), where the access mechanism is a slotted CSMA/CA, and the Contention-Free Period (CFP), where the access is regulated by the Guaranteed Time Slots (GTS) mechanism. The latter mechanism is the most suitable one for real-time traffic, as here frame transmission uses a time division access to the wireless channel which is more predictable than the CSMA protocol.

Each GTS may consist of one or more superframe slots and is assigned for transmission or reception to a single node. Each node can request to the coordinator the allocation or the de-allocation of a GTS of a defined length. The coordinator on each Beacon Interval decides which allocation requests are still valid and how to allocate them, then it informs the nodes of its cluster through the beacon. Each node receiving the beacon knows whether its allocation request has been accepted or not. In the first case, the node waits for its reserved slot to transmit/receive without collisions, whereas in the second case it may try to transmit/receive during the CAP. In each superframe a maximum of seven GTSs may be allocated. Moreover, the CFP duration cannot exceed a maximum value, i.e., the superframe duration minus the minimum CAP length defined in the standard. A node willing to transmit on its GTS checks whether it has enough time to complete the transmission within the GTS, considering also the waiting time for the ACK reception and an Inter Frame Spacing (IFS). In that case, it starts the transmission, otherwise it will schedule the transmission on the next CAP or GTS.

3.2 Cluster-tree topologies and beacon frame collisions

In the cluster-tree topology the network comprises multiple coordinators, also called ZigBee Routers and henceforth referred as “routers”. Routers periodically generate beacon frames to synchronize the nodes belonging to their cluster. In a cluster-tree network there can be several levels of parent-child relations between routers, up to the downmost level, that determines the tree height. For instance, Figure 3.1 represents a cluster-tree network where C5 is the parent of C6, while being child of the PAN coordinator (C1) that is also the root of the tree. It is easy to notice that, if the transmission of the beacon frames is not properly synchronized, i.e., if it is not properly scheduled, a beacon frame may collide either with other beacon frames from different coordinators or with data frames from different clusters. Nodes not receiving beacon frames may lose the synchronization with their coordinator and thus get disconnected from the network.

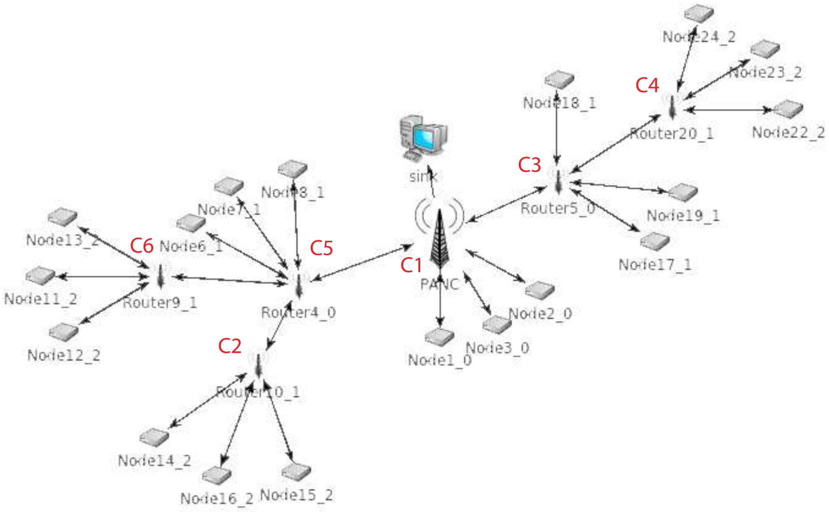


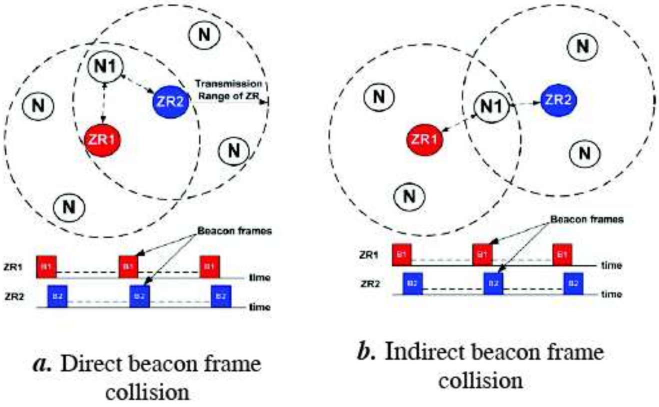
Figure 3.1: Network topology.

In particular, there are two different types of beacon collisions, i.e., direct and indirect ones.

A direct beacon frame collision happens when two ore more coordinators are within the respective transmitting ranges and transmit their beacon frame at the same time, as shown in Figure 3.8a, where N1 should receive the beacon frame from its parent ZR1, but also ZR2 sends its beacon frame approximately at the same time. This result in a beacon collision.

An indirect beacon collision is the situation depicted in Figure 3.8b, where ZR1 and ZR2 are not within their respective radio range so they cannot communicate to each other. However, their transmitting ranges intersect, so that nodes lying on the intersection, such as N1, may experience indirect beacon frame collision.

Collisions may also happen between beacon frames and data frame, when a router transmits its beacon frame during the active period of an adjacent cluster.



a. Direct beacon frame collision

b. Indirect beacon frame collision

Figure 3.2: Direct and indirect beacon collisions.

3.3 Approaches for beacon frame collision avoidance

Two generic methods have been proposed by the 15.4b Task Group [56] to avoid direct beacon frame collisions, i.e., the time division approach and the beacon-only period approach. In the time-division approach, each coordinator schedules its superframe during the inactive period of the other coordinators. This can be obtained by setting in each coordinator a proper offset for the beacon frame transmission, so this approach requires only a small modification to the current IEEE 802.15.4 standard. On the other hand, in the latter approach, the superframe structure is modified, as a period is introduced at the beginning of each superframe, during which the coordinators transmit their beacon frames. Such a period is called Beacon-Only Period, and it is the task of each coordinator to select a proper time slot so that its beacon frame does not collide with the ones from adjacent coordinators. This approach allows multiple clusters to share the active period, so it is more scalable than the time division approach. However this way it is not possible to allocate GTSs. This can be a serious limitation for time-sensitive networks such as typical industrial sensing/control WSNs.

To avoid also indirect beacon frame collisions, not only the overlapping of beacons with the adjacent coordinators is to be avoided, but also the overlapping with the ones that are two-hops away. To achieve this, two alternatives were proposed by the Task Group 15.4b, i.e., the reactive and the proactive approaches. When using a reactive approach, coordinators do not take into account indirect collisions during the association phase. Only when beacon collisions are detected they start a recovery procedure to solve the conflict. On the contrary, when using a proactive approach, coordinators should inform their parent of their offset, so that the information about potentially conflicting superframes can be collected by coordinators during the association phase. This way it is possible to completely avoid beacon frame collisions, but this method is quite complex to implement.

The 2006 IEEE 802.15.4 standard [22] introduced the support of the time-division approach by adding the *StartTime* parameter in the MLME-START primitive, which specifies the time offset between the parent and the child superframes. However, the actual mechanisms to schedule superframes in such a way that beacon collisions are avoided are not defined in the IEEE standard.

A distributed mechanism to avoid beacon frame collisions is given in [57], where a contention-based allocation of superframes is proposed, in which coordinators firstly wait for a backoff period before sending their beacon, then they send their beacon only if no other beacon were heard, otherwise they wait for three more beacon periods. Here, unlike in the IEEE 802.15.4 specifications, beacon frames are sent using Clear Channel Assessment (CCA) [22]. In [58], a distributed beacon synchronization mechanism is proposed, that builds a Beacon Schedule Table (BST) by listening to neighbours' beacons during the association phase, and then uses the CAP to request the neighbours' neighbours list. After all the data is collected, the node can determine its own schedule period. A similar mechanism is defined in the ZigBee specification [59], where a neighbours table is built in the process of joining the network, based on the information collected during the MAC scan [22]. Moreover, the value of the StartTime parameter is included in the beacon payload of every router. In this way, it is possible to select a time offset that does not overlap with either the superframes of the neighbours or the ones of neighbours' parents.

While the above mentioned distributed protocols are suitable for WSN applications in home and building automation, a centralized approach may be more suitable for industrial sensing/control WSNs, for two main reasons. The first is that their local knowledge may not be enough to avoid interferences between different clusters, as at some distance nodes may be too far to successfully communicate with each other but not enough to avoid interferences. The second reason is that using such distributed approaches the schedule of a given set of superframes is non-deterministic, as it will depend on the arrival order of the beacon requests.

A centralized algorithm to schedule IEEE 802.15.4 superframes using the time division approach is the Superframe Duration Scheduling (SDS) [54]. As the algorithm we propose in this chapter is inspired to SDS, a detailed discussion on such an algorithm is given in the following section.

3.4 The SDS Algorithm

In [54] it is theoretically proven that for a given set of superframe durations and beacon intervals, if a cyclic feasible schedule exists, than the minimum cycle length is the least common multiple of all the beacon intervals along the trees, called a *major cycle.* As can be noticed from rel. (3.1), each beacon interval is a multiple of the lower beacon intervals, thus the major cycle coincides with the maximum BI. As a result, the SDS algorithm analyses the schedulability and provides the scheduling of the superframe durations only within a major cycle.

The SDS algorithm can be described as follows:

1. The *minor cycle* is identified as the greatest common divisor of the beacon intervals, that, due to the rel. (3.1), coincides with the minimum beacon interval.

2. The set of all the clusters is ordered in increasing order of BI. The ties are broken in decreasing order of SD.

3. Time is divided into slots, the length of which is the minimum superframe duration.

4. The first beacon interval of the cluster set is considered. Its superframe duration is scheduled by searching the first amount of consecutive time slots able to contain the specific superframe duration. If such an available space is found, the superframe duration is allocated both there and periodically after each BI interval since the first activation.

5. Point 4 is repeated until either all the superframes have been scheduled (i.e., the superframe set is schedulable) or when there is no longer

enough available space within the major cycle (i.e., the set is not schedulable).

As in the pure time division approach each superframe is allocated slots in an exclusive way and there are no simultaneous communications, a necessary condition for a superframe set to be schedulable is that the sum of all the duty cycles is lower than one [54], i.e.,

$$\sum_{i=1}^{N} DC_i = \sum_{i=1}^{N} \frac{SD_i}{BI_i} \leq 1, \tag{3.4}$$

where N is the number of clusters in the cluster-tree topology. While scheduling each superframe at different times prevents collisions between different clusters of the cluster-tree topology, the network scalability is drastically limited. Such scalability issues may prevent the use of IEEE 802.15.4 cluster-tree topologies to realize large industrial WSNs.

To increase network scalability, in [54] the SDS algorithm is extended so as to exploit some spatial re-use of the wireless channel. Coordinators that are far enough so that their transmission ranges do not overlap may schedule their beacons at the same time. As a consequence, if r is the maximum transmitting range of coordinators, grouping of coordinators that may transmit simultaneously can be modeled and solved as a vertex colouring problem [60], where coordinators represent the vertexes and links between coordinators that are distant more than $2r$ represent the edges. Then, the SDS is run taking into account groups of superframes which can be scheduled simultaneously instead of the individual superframes. This way, it is possible to schedule even some sets of superframes for which the sum of duty cycles exceeds one.

However, it is worth noticing that, while beacon frame collisions are avoided, this solution does not prevent data frames sent during the GTSs of a cluster from interfering with other data frames from a parallel cluster. For instance, in the scenario depicted in Figure 3.3, the coordinators C3 and C4 do not overlap their radio ranges, so they can be grouped and share all or a part of their superframe durations. The end-devices D1 and D2 are associated with C3 and C4, respectively. However they are very close to each other. As a consequence, if either C3 or C4 allocates a GTS to its end-device, data transmission within the CFP actually will not be contention-free, as a transmission from the end-device of the other cluster may cause interference.

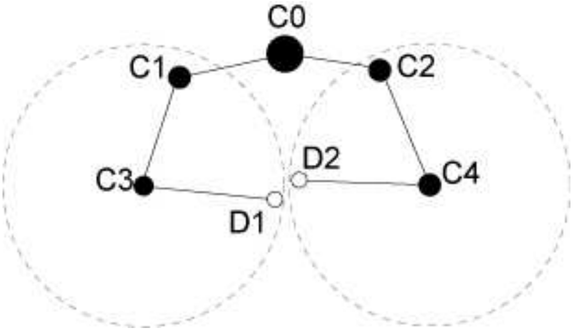


Figure 3.3: Example scenario where two grouped coordinators may have interfering nodes.

3.5 The multichannel approach

The multichannel time division approach we propose in this chapter aims to overcome the limitations of both the pure time-division approach (the simple SDS algorithm) and the time-division approach with spatial re-use (the SDS algorithm with cluster grouping). The capabilities of the IEEE 802.15.4 to support multiple radio channels are exploited by the proposed technique to provide higher scalability and to support contention-free transmission in the GTS with limited interference from other clusters.

The use of multiple channels within the same cluster-tree network is not trivial, as direct communication between two nodes can take place only if nodes are in the same radio channel. For instance, considering the topology in Figure 3.3, C3 is the coordinator of a cluster, while being also a member of the cluster coordinated by C1 (the PAN Coordinator). If C1 transmitted its beacon frame on a given radio channel while C3 were scheduling its superframe on a different channel, then C3 would lose the beacon frames from C1. As a result, C3 and C1 would not be able to communicate to each other.

The simplest solution to this problem would be to provide C3 (and all the other coordinators) with two different transceivers that can be individually set to two different radio channels, i.e., the channel of its cluster and that of the parent. Unfortunately, this solution would require custom hardware, as COTS IEEE 802.15.4 modules include a single transceiver.

However, as data transmission is performed hop-by-hop, a better solution to avoid the above mentioned problem is to give C1 and C3 a proper schedule so that, while C1 is transmitting, C3 avoids transmitting but it is still able to receive on the C1's channel. On the contrary, C4 may transmit simultaneously with C1 on a different channel, as C4 is not intended to communicate directly with C1. The multichannel approach to avoid beacon frames collisions is based on that consideration.

In general, the problem of enabling adjacent clusters to communicate although they use two different channels for their intra-cluster communications may be solved by scheduling adjacent clusters in two alternate timeslices, so that when a coordinator schedules its superframe, its adjacent coordinators are prevented from scheduling their ones. However, all the coordinators which are twohops-away may transmit in the same timeslice. For instance, the clusters of the topology in Figure 3.1 will be assigned the timeslices as shown in Figure 3.7. The coordinator C4 will schedule its superframe in the first time slice (TS1), simultaneously with C1, C2 and C6 but on different radio channels (unless a cluster is so far that no significant interference may be experienced by any of the cluster members). In the following time slice (TS2), C3 and C5 can schedule their superframes. However, the coordinators C4, C2 and C5 will remain active and switch to the radio channel used by their parents. This way, they can receive the beacon of their parent as well as communicate with nodes of the parent clusters.

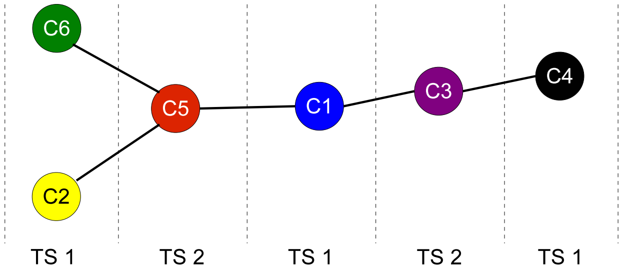


Figure 3.4: Scheduling the clusters in alternate timeslices (TS1 and TS2).

3.6 Multichannel Superframe Scheduling

After explaining the basic idea under the multichannel approach to the beacon (and GTS) frame collisions, we explain in detail the steps of the Multichannel Superframe Scheduling (MSS) algorithm.

1. Schedulability is analysed within the major cycle, after which all the scheduling is cyclically repeated. The major cycle is defined as the least common multiple of the beacon intervals of all the clusters, but it always coincides with the greatest BI due to relation (3.1).

2. The major cycle is divided into smaller time intervals called minor cycles. The minor cycle is the greatest common divisor of the beacon intervals of all the clusters, but it always coincides with the smallest BI due to relation (3.2).

3. The clusters are subdivided into two different groups. The first group contains the PAN Coordinator and all the clusters that can reach it in an even number of hops, i.e., all the clusters featuring an even tree depth. All the other clusters, i.e., those featuring an odd tree depth, are assigned to the second group.

4. The clusters of the second group are ordered in increasing order of BI. The ties are broken in decreasing order of SD.

5. All the clusters of the first group are scheduled at time zero, according to their superframe duration. Moreover, each superframe is allocated in the following minor cycles according to its beacon interval.

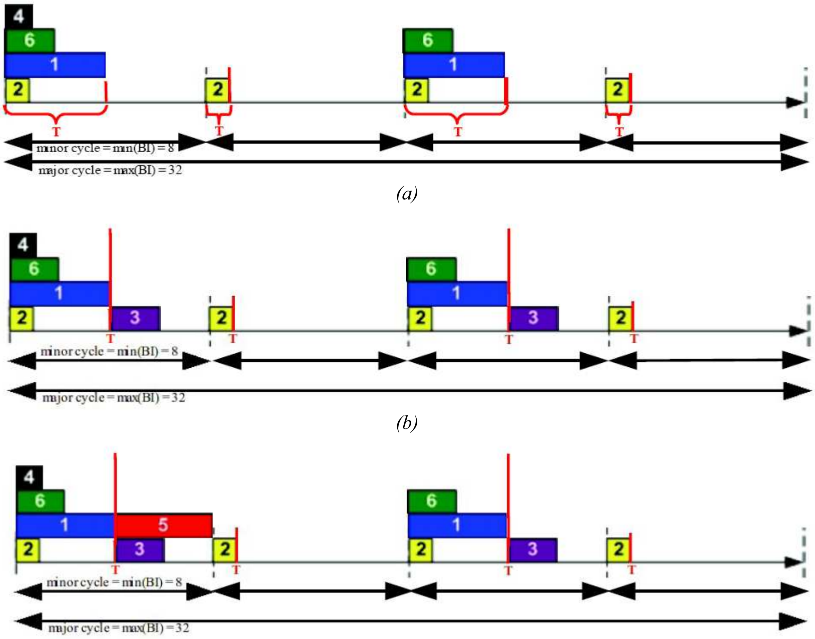
6. For each minor cycle i, the boundary between the first and the second timeslice, T_i, is defined as the time when the last superframe of each minor interval ends. The value of T_i corresponds to the greatest superframe duration scheduled in each minor cycle.

7. For each cluster in the second group, the algorithm tries to allocate the superframe duration starting from the first minor cycle. However, the exact starting time is determined by the largest timeslice boundary among the ones needed by each instance of that superframe within the major cycle. This means that, if the coordinator i has to schedule multiple instances of a given superframe within the major cycle, the starting offset of the second timeslice will be the same in all the minor cycles, and its value is

   $$t_i^{start} = \max_{j \in MC_i} (T_j), \tag{3.5}$$

   where MC$_i$ is the set of all the minor cycles where a superframe of this cluster should be allocated, according to its superframe interval. If there is enough space to allocate the whole superframe duration of the cluster in all the minor cycles of MC_j, then the superframe is scheduled there, otherwise, the algorithm goes to the next minor cycle and so on. If a time is reached when the number of the remaining minor cycles is lower than the superframe interval and no space is found that fits the superframe duration, than the algorithm concludes that scheduling is not feasible.

In Figure 3.5 an example is given, which shows how the MSS algorithm works in the scenario depicted in Figure 3.1, with the superframe set given in Table 3.1. The major cycle is 32, while the minor cycle is 8. Following



(a)

(b)

(c)

Figure 3.5: MSS superframe scheduling.

step 3 of the MSS algorithm, two groups of clusters are identified. The first group contains C2, C1, C6 and C4, while the second group only contains C3 and C5. Following step 5, all the clusters of the first group are allocated together, starting from the time t=0 (Figure 3.5a). The timeslice boundary is set according to step 6 and in that chart is referred as T. Now it is possible

Coordinator	SD	BI
C1	4	16
C2	1	8
C3	2	16
C4	1	32
C5	4	32
C6	2	16

Table 3.1: Set of SI and BI values.

to schedule the second group of clusters. The first node to be scheduled is C3. According to step 7, the algorithm tries to allocate C3 in the first minor cycle and in the third (according to its BI value), starting from the maximum of the timeslices between the first and the third minor cycle. As the greatest timeslice between the first and the third minor cycle starts at time is 4, i.e., the value of t_3^{start} is 4, four other time slots remain within the minor cycle, which are sufficient to allocate the whole superframe duration of C3 The resulting scheduling is shown in Figure 3.5b. The last cluster to allocate is C5. As the beacon interval of C5 is 32, it needs only one minor cycle to be allocated. Again, the allocation is performed according to step 7 of the algorithm. Even in this case, the algorithm tries to allocate the superframe in the first minor cycle, after the timeslice boundary of the first minor cycle (notice that in this case it is sufficient to check the timeslice of only one minor cycle, as the beacon interval is 32). As that timeslice is large enough to contain the superframe of C5, that superframe can be scheduled. As a result, the scheduling of the superframe set succeeds as shown in Figure 3.5c.

Suppose a node in C4 has to transmit some data to the PAN coordinator. Communication between multiple cluster occurs as follows.

1) As the superframe of C4 is scheduled in the first timeslice of the first minor cycle, the node belonging to C4 will transmit there its data to the coordinator of C4. As C3 and C5 do not schedule their superframes simultaneously, while the others do it on different channels, there is no risk of either beacon or GTS collisions. After the end of the superframe, the coordinator of C4 will switch to the channel used by C3.

2) In the second timeslice there is the schedule of C3. However, also the coordinator of C4 is active on the channel of C3, so it can forward the data coming from its node to the coordinator of C3. Even here, the only clusters which are scheduled simultaneously transmit on a different radio channel, thus beacon frames (and GTS) collisions are avoided. After the end of the superframe, the coordinator of C3 switches to the channel of C1.

3) After two *minor cycles*, C1 will be scheduled again. Now, as the coordinator of C4 is active on the radio channel of C1, it can forward the data coming from C4 directly to the PAN Coordinator.

3.7 Analysis of the MSS algorithm

The MSS algorithm schedules as many superframes as possible simultaneously on different channels, in order to exploit the multichannel capabilities of IEEE 802.15.4 radios as much as possible. This way, it is likely that the farthest minor cycles remain unused. While it may seem a waste of available space, decreasing energy consumption in a real-time WSN can be beneficial. In fact, if a coordinator is neither scheduling its superframe nor being active to communicate with the parent, then it may go to sleep. As a result, if a particular timeslice of a given minor cycle is unused, all the coordinators may go to sleep until the start of the following timeslice.

As the MSS algorithm schedules simultaneously superframes of different clusters, there is no need to satisfy formula (3.4). This can be shown through an example. Consider the network in Figure 3.1, using the parameters of Tabletab:mss1:1, for all the clusters but C6, which, instead of a SO=1 (SD=2) is given a SO=3 (SD=8). The sum of all duty cycles of the new superframe set is greater than 1 (precisely it is 1.15625). As a result, this set of superframes is not schedulable using the SDS algorithm, but it is schedulable using the MSS algorithm, as the schedule returned by the algorithm will be the one shown in Figure 3.9. Notice that in this case the second timeslice of the first minor cycle has zero length, as C6 occupies the whole minor cycle. Nevertheless, there is room to allocate both C3 and C5 in the second major cycle. Notice that, when using coordinator grouping, also the SDS may schedule sets of superframes where (3.4) does not hold. However, unlike in MSS, it is not possible to violate (3.4) in the same collision domain. The possibility of scheduling sets of superframes whose sum

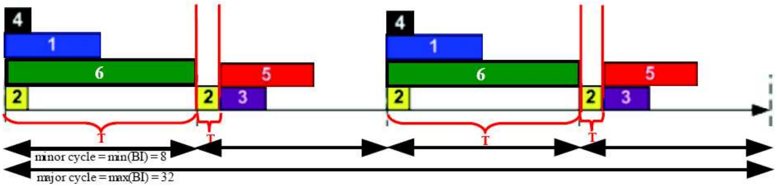


Figure 3.6: Scheduling of a superframe set that is unfeasible using SDS.

of duty cycles is greater than one in the same collision domain is a great benefit over classical single-channel superframe scheduling, as it widely extends the space of schedulability. By using simultaneous communications, it is possible to schedule a larger number of superframes than when using a single-channel, thus the scalability of cluster-tree topologies is highly enhanced. However, not any set of superframes is schedulable using the MSS, thus there are some restrictions on the superframe duration and superframe intervals that will be analysed in the following.

3.7.1 MSS Schedulability

The MSS algorithm can be run offline to both verify the feasibility of a given set of superframes and obtain the cyclic schedule within the major cycle. It is possible to identify some conditions that have to be satisfied to produce a feasible schedule. If these conditions are not met, it is needless to run the scheduling algorithm, as it will always fail. The first condition naturally derives from the need of each cluster to communicate with other clusters and states that there cannot be any coordinator which duty cycle is one.

Theorem 3.7.1 Let S be the set of superframes to be scheduled using MSS and let DC$_1$, DC$_2$, . . . , DC$_n$ be the duty cycles of the clusters 1, 2, . . . , *n*, respectively, being $n>1$. Necessary condition for a set of superframes in order to be schedulable using the MSS algorithm is that

$$DC_i < 1 \forall i \in S \tag{3.6}$$

Proof: The proof is made by contradiction. Suppose a set of n superframes with n>1 that is schedulable with the MSS algorithm and where at least a coordinator k has a DC$_k$=1, i.e., SD$_k$=BI$_k$,.Then it must be that one of the two alternate timeslices has zero duration. As the MSS uses the alternate timeslice to communicate with the adjacent cluster, there is only a set of superframes that is schedulable in MSS with a single timeslice, i.e., the set made up of only k. But this contradicts our hypothesis that n>1.□

A direct consequence is that necessary condition for a set of superframes in order to be schedulable using the MSS algorithm is that

$$DC_i \leq 0.5 \forall i \in S \tag{3.7}$$

Proof: The proof comes directly from the Theorem 3.7.1 and the definition of duty cycle in formula (3.3). As BO and SO are both integer so that SO < BO, the maximum duty cycle less than 1 is obtained for BO=SO+1, which leads to a duty cycle of 0.5. □

Another condition on the duty cycle of nodes comes from the multichannel technique we proposed, that needs two non-overlapping timeslices to enable adjacent clusters to communicate. This is explained in Theorem 3.7.2.

Theorem 3.7.2 Let S be the set of superframes to be scheduled using MSS and let DC$_1$, DC$_2$, ..., DC$_n$ the duty cycles of the clusters 1, 2, ..., *n,* respectively, being n>1. Let us group the clusters in two timeslices TS1 and TS2 as described by step 3 of the MSS algorithm. Necessary condition for a set of superframes in order to be schedulable using the MSS algorithm is that

$$\underset{i \in TS1}{max}(DC_i) + \underset{j \in TS2}{max}(DC_j) \leq 1 \tag{3.8}$$

Proof: Let H be the major cycle of the set S. Suppose that (3.8) does not hold. Then there must be at least a coordinator t in the first timeslice and a coordinator k in the second so that the sum of their duty cycles exceeds one. According to the definition in step 1 of the MSS algorithm, each major cycle will contain H/BI$_t$ superframes of t and H/BI$_k$ superframes of k. As SD$_t$ and SD$_k$ belong to different timeslices they cannot overlap, so they have to be scheduled sequentially. The minimum time needed to schedule those superframes will be

$$\frac{H}{BI_t}SD_t + \frac{H}{BI_k}SD_k = H(DC_t + DC_k). \tag{3.9}$$

If the hypothesis in (3.8) is false, $H(DC_t + DC_k)$ is greater than H, which means that a cyclic schedule would need more than a major cycle. But, as proven in [54], a feasible schedule cannot require a cycle greater than the major cycle. As a result, the set S cannot be schedulable. □

As superframes on different timeslices cannot be either overlapped or interrupted like tasks in preemptive operating systems, there is another necessary condition, stating that BI values must be large enough to fit all the superframes in the alternate timeslice. This is enunciated and proved in Theorem 3.7.3.

Theorem 3.7.3 Let S be the set of superframes to be scheduled using MSS, let SD_1, SD_2, ..., SD_n the superframe durations and let be BI_1, BI_2, ..., BI_n the beacon intervals of the clusters 1, 2, ..., *n*, respectively, being n>1. Let us group the clusters in two timeslices TS1 and TS2 as described by step 3 of the MSS algorithm. Necessary condition for a set of superframes in order to be schedulable using the MSS algorithm is that

$$\underset{i\in TS1}{max}(SD_i) < \underset{j\in TS2}{min}(BI_j) \tag{3.10}$$

Proof: The proof is made by contradiction. Suppose a set of n superframes with n>1 that is schedulable with the MSS algorithm and in which there is at least a cluster k in the second timeslice so that

$$BI_k \leq \underset{i\in TS1}{max}(SD_i) \tag{3.11}$$

Let j be the cluster which superframe duration is the maximum among all the clusters in the first timeslice. As BI_k is smaller than SD_j there must be at least one occurrence of SD_i within each superframe of j. This means that the clusters j and i partially overlap in time, i.e., they belong to the same timeslice. However, this contradicts our hypothesis that i and j belong to different timeslices. □

Finally, in Theorem 3.7.4 we provide a sufficient (but not necessary) condition for the schedulability of a set of superframes.

Theorem 3.7.4 Let S be the set of superframes to be scheduled using MSS, let SD_1, SD_2, ..., SD_n the superframe durations and let be BI_1, BI_2, ..., BI_n the beacon intervals of the clusters 1, 2, ..., *n,* respectively, being $n>1$. Let us group the clusters in two timeslices TS1 and TS2 as described by step 3 of the MSS algorithm. Sufficient condition for a set of superframes in order to be schedulable using the MSS algorithm is that

$$\underset{i\in TS1}{max}(SD_i) + \underset{j\in TS2}{max}(SD_j) \leq \underset{k\in S}{min} BI_k \tag{3.12}$$

According to step 2 of the MSS algorithm, the minimum BI is taken as minor cycle. As in the step 4 all the superframes in TS1 are scheduled since t=0, then the maximum timeslice boundary t_i^{start} will be equal to $\underset{i\in TS1}{max}(SD_i)$. As a result, the minimum available space in any minor cycle will be $\underset{k\in S}{min} BI_k - \underset{j\in TS1}{max}(SD_j)$, which is greater than the maximum superframe duration of any cluster in the second timeslice by hypothesis. This means that all the superframes of the second timeslice will be successfully scheduled by the MSS algorithm. □

The hypothesis of Theorem 3.7.4 is only sufficient and it is rather pessimistic, as it is satisfied only by the sets of superframes in which the MSS algorithm will schedule all the superframes of the second timeslice in the first minor cycle. For instance, this condition holds in the example of Figure 3.5, where $SD_1 + SD_5 = BI_2$, but it does not hold in the example of Figure 3.9, that despite this is schedulable.

3.7.2 Frequency constraints

The MSS algorithm uses a different channel for each superframe in the same timeslice. If the number of clusters to be scheduled in the same timeslice is smaller than the number of available channels, each cluster can be assigned a random channel from the set of the unused ones. Otherwise, some mechanism to achieve spatial re-use of the channels is needed to apply the given schedule. A possible approach is to group the clusters that are distant enough to use the same radio channel without interfering significantly and assign them to the same channels. For this purpose it is possible to use the same technique adopted in [54] for the SDS algorithm. However, in order to avoid interferences between clusters using the same radio channel, the condition discussed in Sect. 3.4 and depicted in Figure 3.3 is to be avoided. This means that the minimum distance between the clusters to be taken into account by the vertex colouring problem [60] should be set as two times the maximum distance at which it is possible to experience a non-negligible interference between any of the node of two clusters, and not only by the coordinator. Assuming all nodes having an interfering range of r_i, then a safe distance between two clusters sharing the same frequency is $4r_i$.

3.8 Implementation issues

This section discusses how it is possible to achieve a real implementation of the technique proposed in this chapter. Firstly, we discuss the modifications to the standard IEEE 802.15.4 MAC protocol that are required to implement the multichannel approach addressed in Section 3.6. Secondly, we discuss the functionalities that have to be provided by the upper layers to implement the MSS algorithm.

3.8.1 Changes to the IEEE 802.15.4 MAC

Implementing the multichannel approach to avoid beacon frame collisions needs only minor changes to the standard IEEE 802.15.4 MAC protocol. In fact, it is possible to use the same superframe structure as defined by the 2006 version of the standard [22], where for coordinators that are not the PAN coordinator two different active periods are defined, namely the Incoming and the Outgoing Superframes. In the former the coordinator receives beacons from its parent coordinator, while in the latter the coordinator transmits its own beacon frames. The only modification to be done in such a structure to avoid the beacon collision problem using the multichannel approach is storing two different radio channels, namely the Outgoing and the Incoming Radio Channels, and switching to the other channel before the start of the incoming and the outgoing superframe respectively, as shown in Figure 3.7. In the case the two radio channels coincide, nodes behave exactly as in the standard protocol. As the superframe structure is unchanged, the proposed approach is backward compatible with the standard IEEE 802.15.4 protocol. It has to be noted that it is possible to use standard modules for nodes that are not intended to work as coordinators. Concerning the primitives of the MAC layer, implementing the multichannel approach does not require either new primitives or changes in the names or in the parameters of the existing ones. However, slight modifications in their behavior are needed. To understand what such modifications are, let us consider the association phase of a coordinator (which is not the PAN coordinator), shown in Figure 3.8. When an *MLME-ASSOCIATE.request* is triggered at the MAC, the node associates with the parent coordinator. In that phase, the node has to store the parent's radio channel in a novel attribute of the PAN Information Base (PIB), that we call *macIncomingChannel*. After the coordinator has obtained the offset and the channel for its outgoing superframe (how such information is obtained will be discussed in Section 3.8.2), the MAC is triggered by the upper layer the *MLME-START.request* primitive,which contains, among its parameters the *StartTime* and the *LogicalChannel* parameters. The former is used according to the standard specifications, while the latter is to be stored in another novel MAC PIB attribute, that we call *macOutgoingChannel*. If this parameter coincides with *macIncomingChannel,* i.e., the parent channel, then the coordinator behaves in the standard way, otherwise it enables the multichannel beacon collision avoidance mechanism. At this point the coordinator has all the necessary information, and therefore it can start tracking the parent's beacon frame on the incoming radio channel. Upon the reception of the beacon frame, two timer events have to be set, one to switch to the outgoing radio channel after the end of the incoming superframe and one to switch to the incoming radio channel after the end of the outgoing superframe. However, as the radio channel switch may take a non-negligible amount of time depending on the adopted transceiver, two relations have to be satisfied to ensure that nodes will exhibit the desired behavior, i.e.,

$$SD_{incoming} + T_{switch} \leq StartTime \tag{3.13}$$

$$StartTime + SD_{outgoing} + T_{switch} \leq BI_{incoming} \tag{3.14}$$

where T_{switch} is the channel switching time, $BI_{incoming}$ is the beacon interval of the incoming superframe, $SD_{incoming}$ and $SD_{outgoing}$ are the superframe durations of the incoming and outgoing superframes, respectively.

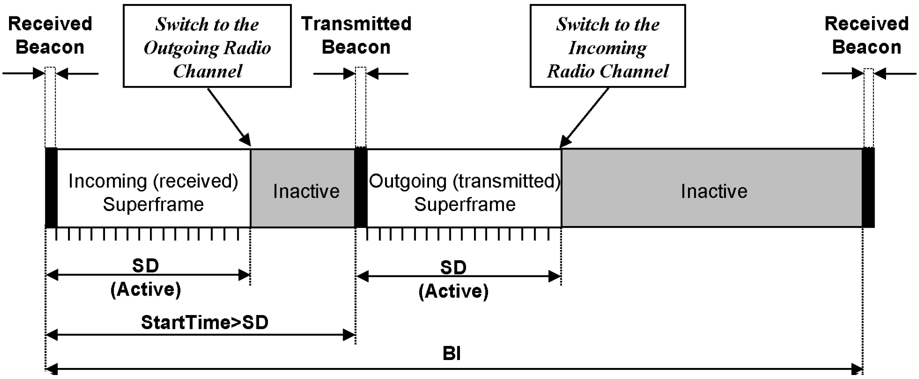

Received Beacon
Switch to the Outgoing Radio Channel
Transmitted Beacon
Switch to the Incoming Radio Channel
Received Beacon
Incoming (received) Superframe
Inactive
Outgoing (transmitted) Superframe
Inactive
SD (Active)
SD (Active)
StartTime>SD
BI

Figure 3.7: Superframe structure.

A common implementation issue for both the time division and the multichannel approach is that a packet left in the transmission buffer after the end of the outgoing superframe may block all the packets queuing to be sent in the incoming superframe, and vice versa. To avoid this problem, it is sufficient to use two different buffers to store the packets to be sent during the incoming and the outgoing superframes, respectively.

3.8.2 Adding MSS Support to the upper layers

As cluster-tree topologies involve multi-hop communications, it is necessary to provide nodes with a network layer in charge of data forwarding. In particular, the ZigBee protocol stack [59] supports a tree routing mechanism which is suitable for cluster-tree IEEE 802.15.4 networks. As a result, it is possible to implement the multichannel superframe scheduling algorithm at the application layer, on top of the ZigBee stack. In this way, the association phase remains compliant with the ZigBee and the IEEE 802.15.4 specifications, with the small add-ons described in Section 3.8.1. As shown in Figure 3.8, the application layer starts the association phase calling the *NLME-JOIN.request* primitive of the ZigBee Network Layer, which in turn triggers the *MLME-ASSOCIATE.request* primitive at the MAC. During the ZigBee association procedure, the node is given the short address by its coordinator and becomes a member of its cluster. While the node is not yet a ZigBee Router (ZR), it can still communicate with the other nodes as a ZigBee Device (ZD). As a result, it can use the standard *NLDE* and *MCPS* primitives to obtain the information about the radio channel and

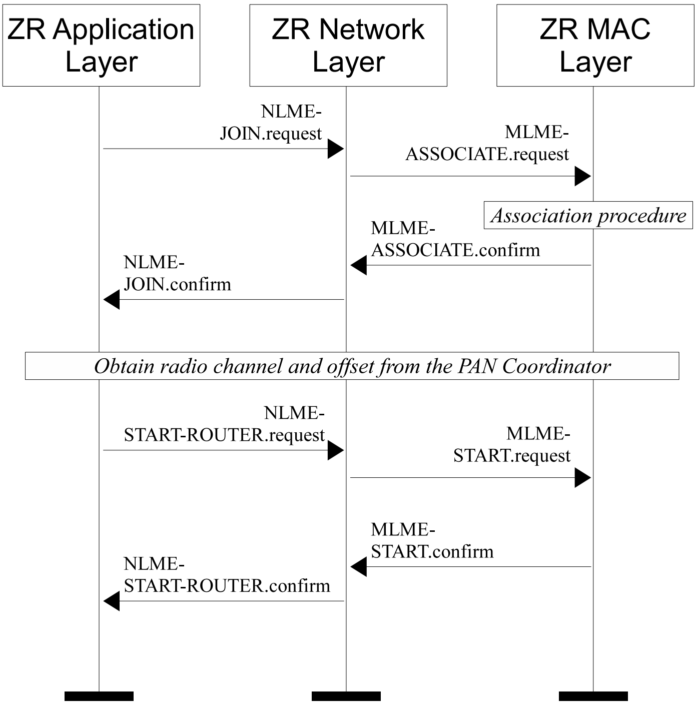


Figure 3.8: Association of a (non PAN-) coordinator.

offset from the PAN Coordinator. After that, the application layer can call the *NLME-START-ROUTER.request* primitive to make the node a ZR. However, according to the current ZigBee Specification [59], this primitive does not take as arguments the radio channel and the time offset, thus to add the support of the multichannel beacon collision mechanism the *NLME-START-ROUTER.request* primitive has to be modified by adding these two arguments. In this way, the network layer can call the *MLME-START.request* of the MAC with the right arguments provided by the PAN Coordinator running the MSS algorithm. The MSS scheduling algorithm can be run either offline, i.e., at network design time, or at run time. In the former case the information about the outgoing channel and the time offset can be hardcoded in each coordinator, thus there is no need to either exchange other data or call other primitives besides the ones described in Figure 3.8. This solution is feasible when the exact requirements and composition of clusters is known a priori, as usually happens in industrial

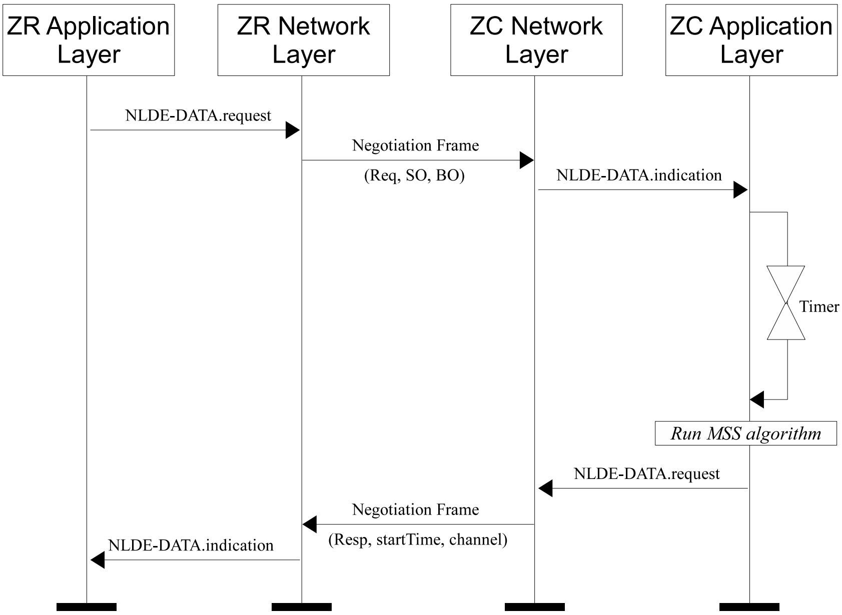


Figure 3.9: Obtaining radio channel and offset from the PAN Coordinator

applications. On the other hand, if it is not possible to plan the clusters at design time, it is possible for a node to use the standard *NLDE-DATA* primitives to obtain the information about superframe scheduling from the PAN coordinator. Coordinators can still be appointed and set up offline with the proper SO and BO values, otherwise they will perform the association in the standard way and then use the same superframe parameters as their parent. As shown in Figure 3.9, each coordinator sends a negotiation request frame including the requested BO and SO values. Upon the reception of such frames, the PAN coordinator (re-)starts a timer to avoid rerunning the scheduling algorithm many times in the initial network setup. As the timer expires, it runs the MSS algorithm and then sends back to the appointed coordinator a negotiation response frame containing the time offset and the outgoing radio channel for that coordinator.

3.9 Experimental Testbed

In order to show the feasibility of the proposed approach using COTS hard-

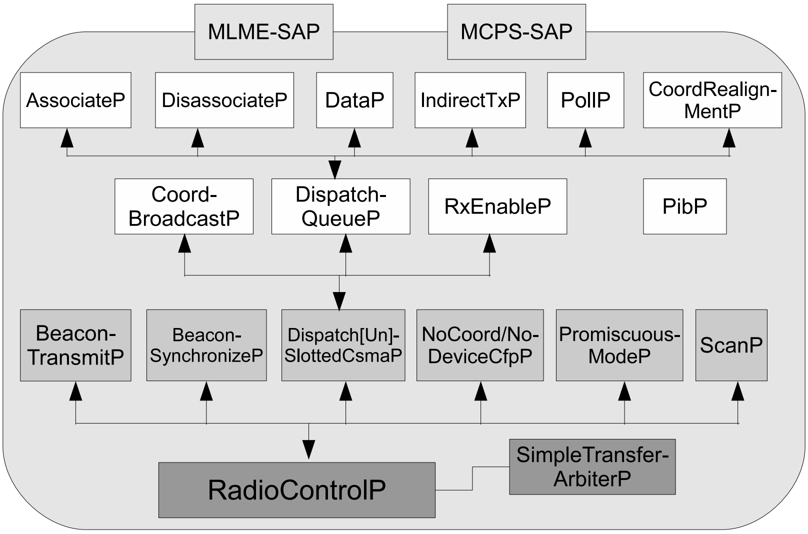


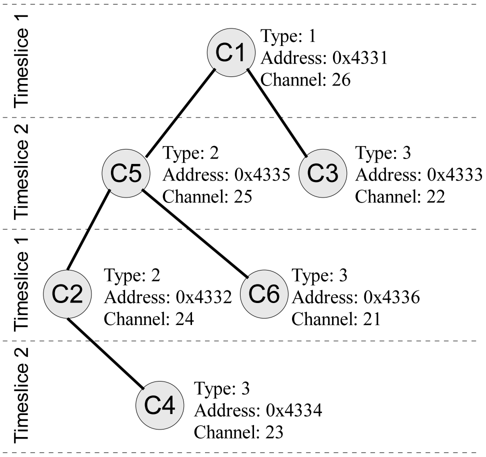
Figure 3.10: Software architecture of TKN15.4

ware, we implemented it on the well-known TinyOS operating system [61] and tested it using TelosB modules from Crossbow [62]. In particular our implementation is based on TKN15.4 [63], a platform independent IEEE 802.15.4-2006 MAC implementation for the 2.1 release of the TinyOS. Such an implementation of the IEEE standard is open source and follows a modular design, so it allows the easy modification of the protocol. In TKN15.4, the MAC functionalities are mapped to software components as shown in Figure 3.10. In particular, the TKN15.4 MAC can be divided into three layers. At the lowest layer there is the *RadioControlP* module, which acts as an arbiter to control which one of the upper components is allowed to access the radio and at what time. The components at the second level are the ones that implement the CSMA and the different parts of the superframe. For example, the *BeaconTransmitP/BeaconSynchronizeP* components handle the transmission/reception of the beacon frame, the *DispatchSlottedCsmaP/DispatchUnslottedCsmaP* components handle the transmission and reception of frames using the slotted/unslotted CSMA, while the *NoCoordCfpP/NoDeviceCfpP* components implement the CFP. These components implement the basic communication mechanisms that are used by the top level components to provide the MLME and MCPS services.

To support the multichannel beacon collisions avoidance mechanism, we modified the two modules at the second level that manage transmission and reception of beacon frames, i.e., *BeaconTransmitP* and *BeaconSynchronizeP*. The first module, which implements the *MLME-START.request* primitive and the transmission of beacons, is modified so as to call the *MLME-SET.phyCurrentChannel* primitive before sending beacon frames in order to set the radio channel to the incoming channel. The second module, which implements the *MLME-SYNC* primitives that are used to synchronize a node with a coordinator, is modified so as to call the *MLME-SET.phyCurrentChannel* primitive when the node is preparing to receive the beacon from the parent coordinator, in order to set the radio channel to the outgoing channel.

Besides modifying the TKN15.4 implementation of the IEEE 802.15.4 MAC, we have also written the modules that implement the upper layers. However, as our objective here was not implementing the whole ZigBee stack but testing the mechanisms proposed in this chapter, at the application layer only the functionalities needed for our purposes were implemented. In particular, three different applications were developed, which identify three types of nodes. Type 1 represents the PAN coordinator, which only features the outgoing superframe, during which it keeps listening for data from the other nodes. Type 2 nodes represent coordinators which only forward packet, but without producing any data. They send and receive beacons according to the multichannel scheduling provided by the MSS algorithm. Upon the reception of data from the associated nodes, a type 2 node stores the packets in a buffer and sends them later in the incoming superframe. Type 3 nodes are similar to type 2, but they also produce data during their incoming superframe. In particular, a type 3 node produces a data packet each time it receives a beacon from its coordinator.

Using those software modules, we deployed a cluster-tree network composed of six nodes, each hosting a different cluster. The network topology and the configuration of nodes are shown in Figure 3.11. In order to make our results easier to examine, we used a different radio channel for each outgoing superframe, and the same superframe durations and beacon intervals for all the clusters. In particular, we set for each coordinator BO=7 and SO=6, which result in a BI of 122880 symbols (corresponding to 1.966 s) and a SD of 61440 symbol (corresponding to 0.983 s). Notice that such a scenario is not feasible using the time division approach, as the sum of the duty cycles of all the coordinators is 3. However, using the multichannel approach and the MSS scheduling algorithm a feasible schedule for these



(a) Network Topology.

Parameter	Value
BO	7
BI	122880
SO	6
SD	61440

(b) Superframe parameters.

Figure 3.11: Testbed Scenario.

superframes is found. In our experiments we performed offline scheduling of the clusters and hardcoded the information about time offset and channel of the outgoing superframe in the coordinators. According to the MSS algorithm, both the major and the minor cycles are equal to the unique BI value. Moreover, the timeslice boundary is equal to the unique SD value. As a result, the *startTime* of all the coordinators of type 2 and 3 is set to 61440 symbols. In order to verify that nodes in the cluster-tree network behave correctly, we used an additional six TelosB modules working in promiscuous mode, each sniffing packets on a different channel of the deployed cluster-tree network. All these modules were connected to a single PC, so that timestamps of all the received packets are obtained from the same clock. We recorded both the received packets and their timestamps in a log file for each sniffer, then we put log files together to reconstruct the sequence of events.

In our experiment the first node to be switched on was C1, so it immediately started sending beacons in the first timeslice on channel 26. Then, at about time 25, we switched on C5, which started sending beacons on the second timeslice on channel 25 after the association phase. It should be noted that, as superframe scheduling information is set up offline, there is no need for any data exchange after the association. This phase of the network

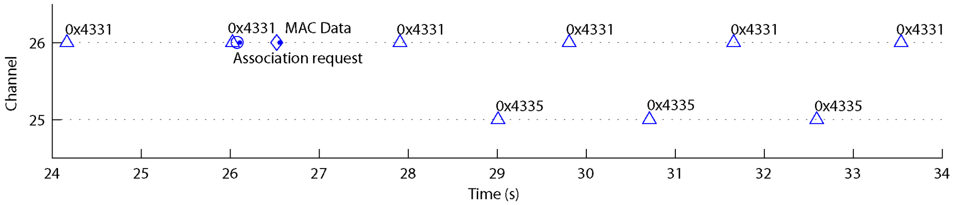


Figure 3.12: Temporal trace of the experiment: association of C5.

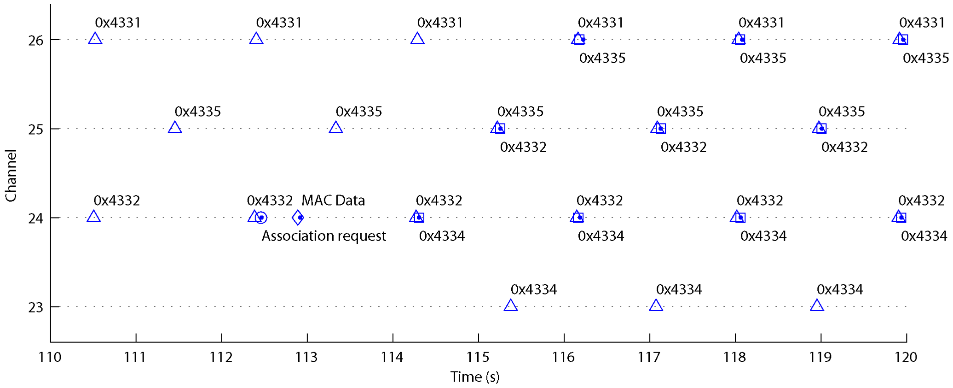


Figure 3.13: Temporal trace of the experiment: association of C4.

operation is shown Figure 3.12, where beacons are represented as triangles, association requests as circles, MAC data transmissions as diamonds and acknowledgements as points. Moreover, the hexadecimal number shown above beacons in Figure 3.12 is the short address of the source node. Exactly the same events occurred when we switched on C2 on channel 24, so we omit the plot for the sake of brevity. Among the type 3 nodes, the first to be switched on was C4 (about time 112), as shown in Figure 3.13, where data packets are represented as squares and the address of their source nodes is the hexadecimal number depicted below. As soon as C4 was associated with C2, it started transmitting data packets on channel 24, just after the incoming beacons (e.g., at time 114.3). The data packets were then forwarded by C2 and C5 in the respective incoming superframes, e.g., at times 115.3 and 113.2, respectively. Moreover, as C4 is also a coordinator, it starts transmitting its beacons in its outgoing superframe on channel 23. Notice that the beacons from C1 are aligned with the ones from C2, while beacons from C5 are aligned with the ones from C4. The reason is that C1 and C2

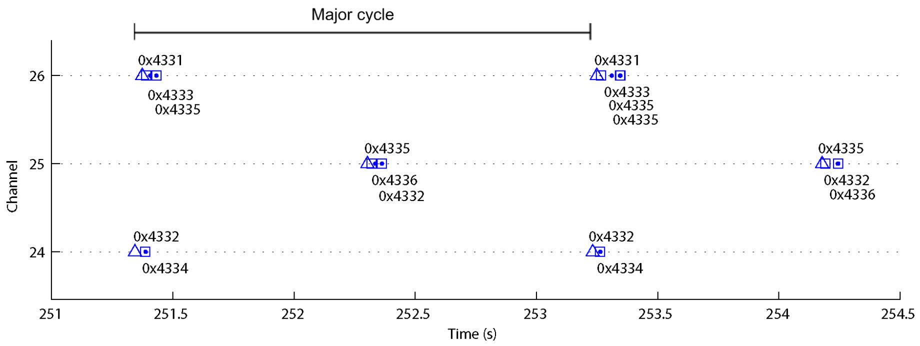


Figure 3.14: Temporal trace of the experiment: steady state network operations

schedule their beacons in the first timeslice, while C5 and C4 in the second. A similar behavior was obtained switching on the other transmitting coordinators, namely C6 and C3, thus they are not shown in the figures. Once all the nodes were active, the network reached a steady state, shown in Figure 3.14. In that figure the channels containing only beacon frames, i.e., channels 21-23, are omitted. Here it is possible to see that C2 receives data from C4 in the first timeslice (time 251.3), C5 then receives data from both C6 and C2 in the second timeslice (time 252.3) and finally, in the next first-timeslice (time 253.3), C1 receives two packets from C5 (containing data originated from C4 and C6, respectively) and a packet containing the data from C3. This pattern of transmissions repeated cyclically till the end of our experiment. Moreover we found that beacons on the different channels always remained perfectly aligned. In fact, as nodes keep tracking the beacons of their parents, they are able to maintain the synchronization of all the superframes.

This experiment, run on a real deployment, shows that nodes behaves exactly as they are supposed to do according to the MSS algorithm, thus providing evidence for the feasibility of the implementation on COTS hardware of the proposed approach.

3.10 Concluding remarks

This chapter presented a novel technique for collision-free superframe scheduling in cluster-tree IEEE 802.15.4/ZigBee networks and a novel scheduling algorithm called Multichannel Superframe Scheduling (MSS). This algorithm exploits multiple radio channels to allow contention-free scheduling of sets of superframes that could not be schedulable under single-channel superframe scheduling algorithms such as SDS, as the sum of their duty cycles exceeds one. The chapter provides a detailed description of the algorithm, together with some considerations on the schedulability and the frequency constraints. The chapter also addresses how to implement the proposed approach through only minor changes to the MAC layer and small add-ons to the upper layers. Finally, a working implementation based on the open source TinyOS is described and the outcome of an experiment run on a real testbed is shown, which proves both the feasibility and the proper functioning of the proposed approach on COTS hardware. Future work will deal with further enhancements of the superframe scheduling algorithm, such as the combination of time and frequency division superframe scheduling to further improve scalability and performance of large cluster-tree WSNs.

Chapter 4

A Topology Management protocol for RT-WSNs

Although in industrial WSNs the main concern is real-time performance, energy efficiency still plays an important role, because even in such networks there can be battery powered sensor nodes. However, the requirements for energy consumption and delivery speed clash with each other. Therefore, a big challenge in the design of industrial WSNs is how to increase energy efficiency without compromising real-time performance. In [64], the intuition of reducing energy consumption by scheduling activity and sleep periods through an Aggregation Layer in charge of creating and handling clusters of nodes was given. This chapter builds upon the idea sketched in [64] but focuses on the design and analysis of a cluster-based topology management mechanism. This mechanism decreases the duty cycle of nodes while providing bounded delays, thanks to a time division channel access strategy combined with a cellular radio architecture. This chapter provides three main contributions. First, a fully fledged topology management mechanism for WSNs, which is discussed and described in detail through a state machine. Second, analytical formulations for the energy efficiency and transmission rate, which enable us to estimate at design time the tradeoff between the power consumption and data delivery speed requirements. Finally, experimental results obtained using the ns-2 simulator, which confirm the analytical results on energy-consumption and assess the effect of the proposed topology management mechanism on the routing performance. In particular, a comparison between the performance of a well-know routing protocol, i.e., the SPEED protocol [12, 13], when it is used with or without our topology management mechanism, respectively, is provided.

The chapter is organized as follows. Section 4.1 summarizes related literature and gives the motivation for our work. Secttion 4.2 shortly introduces the WSN model here adopted, while an accurate description of the proposed topology management protocol is provided in Section 4.4. Performance of our approach in terms of energy consumption and delay is discussed using an approximated analytical model in Section 4.5, while Section 4.6 shows simulation results. Finally, Section 4.7 gives our conclusions and outlines directions for further work.

4.1 Approaches to improve WSN performance

Several approaches to achieve energy efficiency and/or delay bounds in WSNs have been proposed, which work at different levels of the protocol stack. Routing-level approaches typically take some cost parameter (e.g., energy and/or delay) into account explicitly when routing sensor data and target the optimization of relevant metrics [14, 65–69]. Energy-efficient MAC protocols typically implement some kind of coordination to decrease the duty cycle of nodes while regulating the medium access [70]. However, the combination of multiple protocols handling the same parameters (energy, delay or both) at different levels generates mutual interactions that are not easy to analyze. To overcome this problem, some protocols [9, 71, 72] use a cross-layer approach that spans from the physical (or the MAC) to the network layer. A notable example is the LEACH protocol [9], which proposes a clustered architecture in which a TDMA-based MAC is able to decrease the duty cycle of the nodes, while a CDMA-based PHY allows parallel transmissions between the cluster. However, LEACH does not take delay into account and suffers from scalability problems, as it assumes a direct connection between the cluster head and the base station. The DGRAM [71] routing and MAC protocol uses TDMA-based transmissions with slot re-utilization to reduce the latency between consecutive transmissions. The slot allocation strategy used is based on a distributed algorithm, that runs at the time of node deployment and then remains unchanged. As a result, DGRAM is not adaptive to varying WSN conditions. Moreover, it requires uniform node density and out-of-band network-wide clock synchronization, that is difficult to achieve in large WSNs. The SERAN [72] protocol suite for clustered sensor networks uses random routing between the clusters together with a hybrid TDMA/CSMA MAC to achieve energy-efficiency and robustness. A simplified analytical model based on Markov chains is used to select the protocol parameters so as to meet energy and average delay requirements in a given scenario. However, as the analysis is performed off-line, clusters are fixed and there is no dynamic adaptation. Moreover, as the duty cycle of nodes is fixed and depends on the topology, there is no energy balancing among nodes in different clusters.

Cross-layer approaches raise the complexity of WSN design. In fact, it may be difficult (or even impossible) to use COTS hardware or reuse well-known protocols, so cross-layer approaches require coding the whole protocol stack (including the basic low-level operations) and may require custom hardware as well. As a result, their implementation costs may be significantly higher than those encountered when deploying a WSN using COTS components and well-known protocols. In this chapter an approach is proposed that does not require specific hardware or low-level firmware and is based on the idea of separating the energy and delay requirements by addressing them at different levels of the protocol stack. This approach is based on the combination of an energy-efficient topology management protocol with a non-energy-aware routing protocol enforcing a real-time behaviour in data forwarding. In general, the role of the topology management protocols in WSNs is to coordinate the sleep transitions of the nodes in such a way that data can be forwarded to the data sink in an energy-efficient way. To achieve this goal, the SPAN protocol [19] elects in rotation some coordinators that stay (alert) awake and actively perform multi-hop data forwarding, while the other nodes remain asleep and check whether they should become coordinators at regular intervals. However, the election is based on non-deterministic local decisions, that are not able to guarantee routing fidelity. A more predictable approach is the Geographical Adaptive Fidelity (GAF) [17], where the whole area is divided into fixed virtual grids, small enough that each node in a cell can hear each node from an adjacent cell. Nodes belonging to the same cell coordinate active and sleep periods, so that at least one node per cell is active and routing fidelity is maintained. However in both GAF and SPAN traffic injection is not controlled. As a result, the delay such protocols may introduce is neither predictable or bounded. This makes them unsuitable for real-time WSNs. The topology management protocol described in this chapter schedules data transmissions as well as activity and sleep periods of the nodes, in such a way to reduce their energy consumption while introducing a bounded delay. On top of this, a routing protocol is run to enforce real-time behaviour in data forwarding. It has to be underlined that even if the routing protocol does not take energy efficiency into account, the desired property of achieving energy efficiency while maintaining both routing fidelity and delay bounds is globally met through the combination of the features provided separately by the topology management and the routing protocols. This way, both the design effort and the implementation cost for deploying a real-time WSN can be dramatically lowered, as it is possible to use COTS hardware featuring low-power capabilities (e.g., IEEE 802.15.4 modules) in combination with any known real-time routing protocol. In addition, the timing behaviour of the WSN is easy to analyze, as it is obtained by adding two separate delay components, i.e., the bounded delay introduced by the topology management protocol and the one enforced by the routing protocol, respectively. As the timing behaviour of the two protocols can be analyzed separately, it can be characterized by means of simple formulas.

4.2 Network model

The reference environment for both this and the next chapter reflects a typical monitoring application in which every node of a large and dense WSN mainly working in a proactive way periodically sends real-time data to a Sink node. As it will be discussed in Chapter 5, the dynamic approach is also able to deal with event-driven transmissions. In both chapters it is assumed that nodes are homogeneous, energy-constrained and stationary. A sensor node can be in one of the following four states: transmitting, receiving, idle, sleeping. Each state is characterized by a given power consumption, high for all the active states (i.e. transmitting, receiving and idle), low for the sleeping state [73]. The only non-energy constrained node is the Sink In our experiments, discussed in Sections 4.6 and 5.2, we have one Sink node, but this is not mandatory, as our protocol can also cope with multiple Sink nodes. All the nodes are supposed to be location-aware. This can be achieved through either a dedicated hardware (e.g., a low-power GPS receiver) or localization service protocols for wireless ad-hoc networks.

4.3 Design Principles

The main requirements of our topology management protocol are energy-efficiency, bounded delay and deterministic routing fidelity. Such requirements drove the design of our protocol, as is explained in the following.

Energy-efficiency: To reduce energy consumption, the typical redundancy of sensor nodes in a WSN can be exploited by putting nodes to sleep when they do not need to be active. This alternation between activity and sleep periods is here integrated into a two-level hierarchical network architecture, in which two network levels work in parallel and interact. The first level is made up of the clusters of sensor nodes, here called *Aggregated Units* (AUs), whereas the second level is a backbone of active nodes that performs real-time multi-hop forwarding. Sensed data is first collected within the first-level network, fully controlled by the topology management protocol, and then is forwarded towards the Sink node in aggregate form via the second-level network. This means that the real-time routing protocol runs only in the second-level network. Each AU node can be in one of the following states:

- *InitState*;
- *Cluster Head* (CH);
- *Relay Node* (RN);
- *Common Node* (CN).

The CH is the AU Master and is in charge of handling data transmission within the AU. The CH collects data from the sensor nodes (only CNs), performs data aggregation and periodically transmits it to the RN. The task of the RN is multi-hop data forwarding to other RNs or the Sink node, i.e., RNs form a QoS-enabled backbone of active nodes. Data transfer inside an AU, i.e., for CHs and CNs, follows a pre-established sequence which emulates a super-frame structure where each node has its own time-slot. Note that this protocol builds its own super-frame structure on a CSMA-based MAC. This has two advantages. First, it does not rely on a specific MAC protocol. Second, the superframe is tailored for the protocol needs, so as to overcome the limitations imposed by the specific protocols. As an example, the beacon-enabled IEEE 802.15.4 supports only 7 Guaranteed Time Slots, it does not support energy balancing techniques and does not allow a single network to span over multiple channels [22]. In this way, despite the adoption of a common CSMA/CA protocol at the MAC level, it is possible to avoid collisions and to control the duty cycles of nodes, thus also reducing overhearing and idle listening. As a result, energy consumption is significantly decreased. In the meantime, RNs stay active to perform real-time routing. As the CH and especially the RN states are more energy-consuming than the CN states, CH and RN nodes are elected in rotation to balance energy consumption and to increase the network lifetime.

Bounded delay: In addition to decreasing power consumption, contention avoidance within the AU allows for bounded delays. Given a transmission schedule, the delay introduced by the topology management protocol is the sum of the remaining time slots to reach the RN. To prevent collisions between packets from different AUs, a Frequency Division Multiple Access (FDMA) among nodes operating on different AUs is used. For this reason, when an AU is created, a private channel is selected, that should be different from the channels used by the neighbouring AUs. Nodes will transmit on the private channel for intra-AU communications, while a broadcast channel is used for all the other communications, such as elections and data forwarding towards the Sink. However, to forward data from CH to the RN, the CH has to temporarily switch to the broadcast radio channel, transmit data and then go back to the channel of its AU. To maintain the delay bounded, the CH-to-RN data packet must be transmitted necessarily during the synchronization slot, otherwise it is discarded. For this reason, this packet has also to be prioritized over the other traffic, i.e., there should be a high probability of receiving that packet even when collisions with other traffic occur. This is achieved by ensuring that for each RN the signal received from the relevant CH is much stronger than the one coming from the other RNs. This is obtained selecting very close CH-RN pairs. In addition it would also be possible to set a higher transmission power for CH-to-RN transmissions than for RN-to-RN forwarding.

Routing Fidelity: In our protocol, CNs always communicate to their CH, which is awake during the active part of the super-frame, while CH and RNs transmit to RN nodes, which are active in the broadcast radio channel at all times. As a result, in order to provide deterministic routing fidelity the protocol only has to guarantee that each AU always contains one CH and one RN, while the other nodes are CNs. This is achieved though a hybrid (distributed/centralized) election mechanism, which consists of a distributed algorithm used for the first election only, while the following elections are ruled in a centralized way as they are performed by the CH. Such a centralized election mechanism exploits the periodic beacons issued by the CH to communicate a CH or RN switch, so that there is no need to stop data transmissions. Moreover, this mechanism is robust to CN and RN failures, as the CH is able to detect them and, if an RN fails, the CH elects a new RN node. However, if a CH fails, the whole AU has to be re-built.

4.4 The proposed topology management protocol

As already mentioned in Sect. 4.4, according to the proposed topology management protocol protocol each node can be in one of four different states depcted in Figure 4.1.

With the exclusion of the InitState, which is entered by any node as soon as it is turned on, the decision on what the current state of a node should be is not local, but it is agreed between the nodes in the AU. In particular, in each AU at any time there is one CH, one RN and a varying number of CNs. The CH is the Master node of an AU. It manages and handles data transmission within the AU, collecting data from the sensor nodes (only CNs), performing data aggregation and periodically transmitting it to the RN. The task of the RN is to forward the data to other RNs or the Sink node, i.e., RN nodes form the QoS-enabled backbone of active nodes. In this architecture, therefore, the CH handles transmission within the AU, while the RN handles transmission outside the AU.

The normal functioning of the protocol is logically divided into three different phases, i.e., initialization, election and data transfer. The initialization phase is executed when a node is activated for the first time (i.e. during the InitState), while election and data transfer alternate, not necessarily at regular intervals. The main functions performed during the initialization phase are the definition of the cellular architecture (i.e. the channel selection based on the nodes position) and the first election, during which the CH is elected. Then the CH elects the RN (as described below) and sends the transmission schedule to all the nodes belonging to its AU.

In cluster-based protocols integrating a cluster head rotation mechanism whenever a CH is elected it is generally necessary to reconstruct the whole cluster. This provides the network with flexibility and adaptability to changes in environmental conditions. However, in the presence of tight deadlines, or when a continuous update of the variables being monitored is

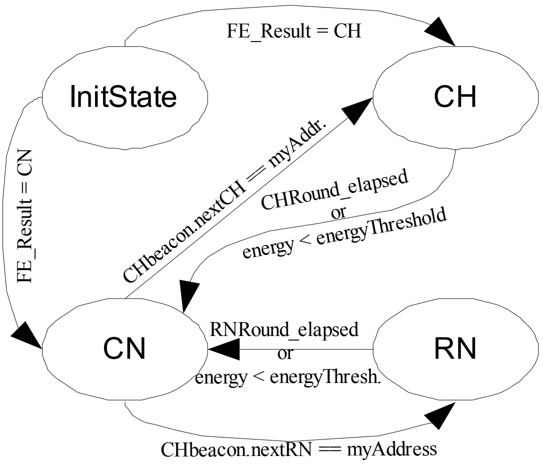


Figure 4.1: State diagram of the proposed topology management protocol.

needed, this may lead to unacceptable QoS degradation. For this reason we decided to separate the distributed algorithm used for the first election from the one used for the next ones, which is centralized. In the latter case, at a certain point (after a pre-established time or because its remaining power has dropped beneath a certain threshold), the CH autonomously decides which node will be its successor and notifies the nodes of its AU.

RN election is different and has to be as independent as possible from the election of the CH. This is because a node in the RN state never goes to sleep, as it remains active all the time to perform data forwarding between AUs, so it could run out of battery power more rapidly. However, an independent election would require complex management algorithms, so we propose a hybrid solution, as discussed in Sect. 4.4.2.

In the following we give an accurate description of the behaviour of a node into each of the four states.

4.4.1 InitState

InitState is the initial state of nodes on their first activation. This state performs two main tasks, i.e., initialization and first CH election. To prevent interferences between AU, our topology management protocol uses Frequency Division Multiple Access (FDMA) among nodes operating on different AUs. Each node for intra-AU communications transmits on a different channel from those of the neighbouring AUs. The transmission channel is automatically selected during the nodes' initialization. Such a selection is based on the node position. This way we can create the cellular radio architecture by setting the transmission channels to avoid interference among nodes on different AUs. In our scenario we assume that all the nodes know their own position and that they have been randomly arranged with a uniform density. Under these assumptions it is possible to create a homogeneous cellular structure in a simple and efficient way, with a virtual grid subdividing the area being monitored into a number of small uniform regions, each one hosting a cell. Channel selection is also based on the position of a node (and the grid it belongs to). In particular, the parameters calculated from the (x,y) position of the node are AU ID and AU coordinates (x_{AU}, y_{AU}). If the size of the monitored area is ($size\_x$) × ($size\_y$), and the predefined side of the AU is AU_side, these parameters can be calculated as follows:

$$x_{AU} = \lfloor x/AU\_side \rfloor \tag{4.1}$$

$$y_{AU} = \lfloor y/AU\_side \rfloor \tag{4.2}$$

$$ID_{AU} = x_{AU} \cdot \lceil side\_y/AU\_side \rceil + y_{AU}. \tag{4.3}$$

Another important parameter to be chosen during the initialization phase is the radio channel to be used for AU operations. The channel selection scheme here adopted is static, as the channel depends only on the AU coordinates. We used a table-based approach, where the radio channel C$_{AU}$ is selected as

$$C_{AU} = 11 + FTable[y_{AU} \bmod N_{rows}][x_{AU} \bmod N_{cols}] \tag{4.4}$$

where N_{rows} and N_{cols} represent the number of rows and columns of the virtual grid and $FTable$ is represented through a matrix that can be set off-line. The table has to be chosen so that the distance of AUs that use the same channel is greater than the radio range of the nodes. In the work described in this chapter we used the one shown below:

$$FTable = \begin{bmatrix} 1 & 2 & 5 & 6 & 9 & 10 \\ 3 & 4 & 7 & 8 & 11 & 12 \\ 6 & 9 & 10 & 1 & 2 & 5 \\ 8 & 11 & 12 & 3 & 4 & 7 \end{bmatrix} \tag{4.5}$$

The use of this frequency assignment permits to limit interferences between TDMA-based communication belonging to different AUs.

The next step is the first election, which decides the first CH. We remind that the first election is different from the next ones, as the first election is the only one that is based on a distributed algorithm. The next elections are centralized and performed directly by the current CH. We suppose that nodes are all homogeneous except the Sink node, which is not energy constrained. Each node sends an ECHO message containing its ID and its energy, then waits for other messages from the neighbours. Only nodes within the same virtual grid are considered, i.e., packets from other virtual grids are discarded. After a defined timeout, the node having the highest ID according to the collected ECHO messages takes the CH role and considers the other ones as members of its AU. This node then calculates the TDMA schedule for its AU and sends it to the AU members through an ADVISE message broadcasted on the AU. Then the CH waits for ACK messages from every node of the AU. If a timeout expires, the ADVISE message is broadcasted again, until either all the ACKs are received or a maximum number of retries is reached. At the end of this process nodes know their CH and their TDMA slot, so they can switch to the next state, that is, CH for the elected node and CN for the others. There is no need for reaching a consensus in the CH election. If, due to the loss of ECHO messages, two nodes think to have the highest ID, no inconsistency will raise, as the node sending the ADVISE first will be the CH.

4.4.2 Cluster Head.

When a node enters the CH state it first verifies whether an RN node is active on the network. This is always true except for the first election. In this case, a new RN is elected taking whichever of the nodes with the greatest amount of energy is the closest from the set of the CNs. Then, data acquisition can start. In each AU there is only one node in the CH state at any time. This node notifies the super-frame start and other information through periodical beacons, that are broadcasted to the AU at the beginning of each super-frame. The beacon is used by the CNs for synchronizing with the AU. In addition to the super-frame length and the AU ID, the beacon also informs the CNs about changes in the AU (e.g., that a new CH or a new RN were elected). After the beacon is sent, the CH remains active to collect the sensed data from the CNs. In addition to data, it also collects other useful information to manage the AU, such as the energy level of each CN, that is contained as a field of the data packets. After the duration of (N_{AU}-

2) data time slots, the collected data is packed into an aggregated data frame and is ready to be forwarded through the RN. Data fusion techniques may be used to reduce the size of aggregated data, however this aspect is not addressed in this chapter. The CH then synchronizes the AU data with its RN, e.g. switching from the AU radio channel to the broadcast radio channel (the one used by the RNs) and transmitting data to its RN during an appropriate time slot. The duration of this time slot has to be chosen so as to also allow the RN to inform the CH that the current RN turn is expired or that its energy is low. After this time slot, the CH goes back to the AU radio channel and enters the sleep state until the start of the next super-frame. At that time, a new beacon will be sent and the entire procedure will be repeated.

In addition to manage TDMA-based data transmissions, a CH node is also in charge of the election of the next CH and RN. Both elections can be either time-triggered or event-triggered. In particular, a node may decide to return to the CN state after a predefined time is elapsed or when its energy goes below a predefined threshold. The duration of CH and RN rounds may be different, e.g., the RN may have shorter rounds as its energy consumption is greater. The CH autonomously decides which node is going to be its successor and notifies all the nodes in the AU through the next beacon. From the next super-frame the new CH will start operating. The decision regarding the next CH is based on the residual energy of the nodes in the cluster, as signalled in the frame that nodes send during normal transmission phases. The RN election is also up to the CH, but is triggered by the RN. In fact, when a new RN election is requested, the RN notifies the CH during their synchronization phase. The CH consequently chooses as the next RN whichever of the nodes with the greatest amount of energy has the strongest signal (it is advisable for CH and RN to be close to each other to achieve a good QoS). The energy information can be obtained with a negligible overhead, inserting it in the packets that CN nodes send to their CH, while the signal strength can be derived directly by the hardware.

4.4.3 Relay Node.

A node entering the RN state switches to the active radio channel and starts forwarding data according to the routing protocol used An RN may forward data packets from other RNs or from its CH. It operates data forwarding on the broadcast radio channel. It can communicate with the

CH only during the synchronization time slot. This time slot is used also to request the election of a new RN to the CH because its turn is expired or its energy is low. After sending such a request, the RN has to wait until a new RN is elected by the CH in order to maintain connectivity and QoS of the routing protocol. This is implemented as follows. After the request from the RN through a NEWRN_REQ message, the CH elects a new RN and communicates its choice using the beacon of the next super-frame. As soon as the new RN receives the beacon, it enters the RN state and switches to the broadcast radio channel. At this point, the new RN sends a NEWRN_CONF message to the old one, so that this node can return to the CN state. If the routing protocol uses something like a routing table or a neighbouring table, it is useful to perform a table exchange between the old and the new RN. This requires minor modifications on the Routing Layer, but ensures that any RN switching will not affect the routing performance. We implemented this behaviour packing the table in a RTEXCH packet which is sent in response to the NEWRN_CONF message from the new RN.

In the RN state nodes never go to sleep, so they are the ones that feature the highest energy consumption. Notice that also the Sink node works as an RN. Moreover, in the AU containing the Sink node there is no need for rotating the RN. During the first election, if a CH finds that the Sink node falls into its cluster, it elects the Sink as the RN.

4.4.4 Common Node.

When a node is in the CN state, it saves more energy than in the other states, as the duty cycle is lower. This is because a CN is active only during the transmission of the CH beacon and during its own time slot, while it sleeps during the remaining time. It is important for the CN to receive every beacon, as any change of CH or RN will be notified at the beginning of the super-frame.

4.5 Discussion and Protocol Analysis

The proposed approach introduces a two-level hierarchical network architecture, in which sensed data is first collected within the first-level network, the AU level, and then forwarded toward the Sink node in aggregate form through a second-level network. The two levels of the WSN (AU and rout-

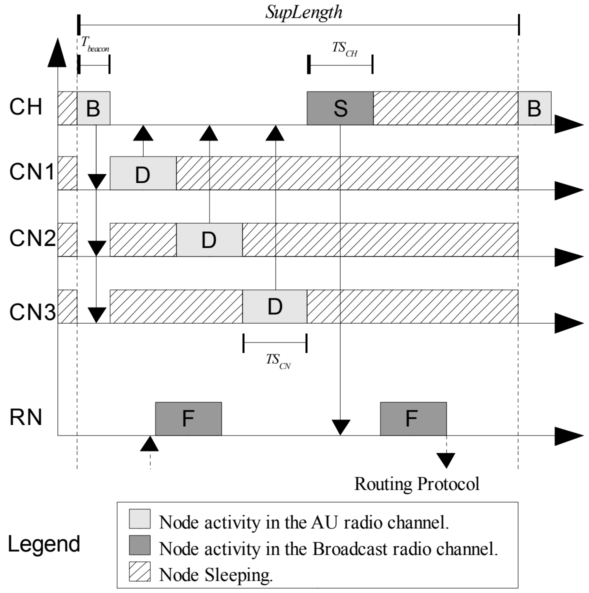


Figure 4.2: The proposed topology management protocol and data exchange with the routing protocol. B-labeled boxes: beacon transmissions from the CH node. D-labeled boxes: time-slots for data communication between CNs and the CH. S-labeled box: time slot for synchronization among the CH and the RN. F-labeled boxes: transmissions for data forwarding handled by the routing protocol running on the RN.

ing) run simultaneously for acquiring and forwarding data. The pair CH-RN acts as a gateway between the two levels. Although splitting the RN and CH roles might appear to complicate the AU management, it actually gives several benefits. Firstly, this improves routing performances, as the RN performs nearly full time packet forwarding. If RN and CH roles were unified in the CH, the CH would be able to perform packet forwarding only when there is no data from CNs. This would reduce the bandwidth utilization, as the CH would be a bottleneck. On the other hand, the parallelism between RN and CH operations achieved when the roles are separated provides a better bandwidth exploitation and reduces latencies and chances of congestion. Furthermore, splitting RN and CH roles combined with a separated channel for RN communication and different radio channels for nearby AUs, allows for isolation between contention-free intra-cluster communications and contention-based inter-cluster ones, using a single radio. Such a solution improves both performance and network scalability. This way, traffic within the AUs does not interfere with transmissions from other AUs or with data forwarding, thus TDMA can be used within the cluster, so that the delay is bounded as well as the energy consumption. Delay is bounded only for the operations of the topology management protocol, i.e., transmissions within an AU. However, this topology management protocol cannot turn a routing protocol for ad-hoc or sensor networks into a real-time routing protocol. So, in order to meet soft real-time constraints in multi-hop data forwarding, a QoS-enabled routing protocol has to be used. In order to assess the performance of our topology management protocol in this chapter we used the SPEED routing protocol. However, a different real-time routing protocol might be used as well. Furthermore, the two-level approach here adopted implies also some modification on the network view from the routing perspective. The aspects involved are node density and delay. As each AU is viewed as a single entity, the perceived node density is reduced by a factor equal to the number of nodes that make up the AU. As node density may impact on the performance of the routing protocol, the AU size should be carefully set. In addition, it is necessary to avoid the formation of "holes" between AUs. This could happen if two RNs of contiguous AUs are too distant to communicate with each other. However, as AU creation is based on node location and the AU shape is a square, this problem can be easily avoided when sensor nodes are homogeneous by setting an appropriate value for the AU side. In particular, the maximum distance between two nodes belonging to contiguous AUs is two times the diagonal of the AU, i.e., $AU\_side \cdot 2\sqrt{2}$, hence the absence of holes can be ensured by setting

$$AU\_side \leq \frac{R}{2\sqrt{2}}, \tag{4.6}$$

where R is the radius of the area covered by the radio transceivers.

The other aspect to consider during the forwarding phase is the delay introduced by intra-AU transmissions. As the proposed topology management protocol uses TDMA for data transmission within the AU, the delay for transmitting data from the CN to the CH is bounded by the size of the time slot. For the same reason, the delay due to the subsequent nodes of the TDMA schedule is bounded too. In order to limit the delay within the AU, we imposed a time slot also for the synchronization phase between CH and RN. However, during that phase also data forwarding may occur, thus lead-ing to collisions between the packet from the CH and the ones from other RNs. This problem may be solved by using the AU radio channel also for CH-to-RN traffic, thus making the RN temporarily switch to the channel used by the CH. This solution does not degrade routing performance when all the AUs have the same super-frame length, so the time slots may be synchronized along the network. A more flexible solution is the use of the broadcast radio channel also for this type of traffic, but some mechanism should be adopted in order to prioritize packets from CHs respect to the others. We used transmission power. In fact, as the CH elects RN the closest node among the ones with the highest energy, the distance between an RN and its CH is usually much smaller than the distance between RNs of different AUs. To further decrease the probability of losing synchronization packets, the transmission power of CHs can be set slightly higher than that of the RNs. We found that using these mechanisms the probability of losing synchronization packets is very low. This approach is viable unless data it is highly critical (crf. Sect. 4.6.2), otherwise synchronization of super-frames may be required. Anyway, the delay introduced on a packet by our topology management protocol is bounded and can be expressed as

$$d = (N_S + 1) \cdot TS_{CN} + TS_{CH}, \tag{4.7}$$

where NS is the number of subsequent CNs according to super-frame schedule, TS_{CN} is the data time slot duration for CN nodes and TS_{CH} is the duration of the synchronization phase between CH and RN.

As TDMA scheduling is used for data transmission, restrictions exist on the maximum number of nodes belonging to an AU, or, on the opposite side, on the minimum super-frame duration. Considering the super-frame structure shown in Figure 4.2, the following relation holds:

$$SupLength \geq T_{beacon} + N \cdot TS_{CN} + TS_{CH} \tag{4.8}$$

where *SupLength* is the super-frame duration, T_{beacon} is the time used by the CH for beacon frame transmission and N is the number of nodes in the AU. T_{beacon} and TS_{CN} parameters are constant values, while TS_{CH} depends on the number of nodes in the AU (N), on the data field length, (T_{data}) and on the adopted data fusion technique. However, in the worst case (without data aggregation), it can be set to (4.9)

$$TS_{CH} = T_{ov} + (N - 1) \cdot T_{data} \tag{4.9}$$

where T_{ov} is a constant overhead due to the (IEEE 802.15.4 and topology management protocol) packet header.

The proposed approach targets applications where sensed values have to be regularly updated. However, a refresh rate shorter than the super-frame length cannot be supported. Relation (4.8) can be rewritten to show the maximum number of nodes N in an AU as a function of the super-frame length. Given a super-frame with *SupLength* duration, for the number of AU nodes N, the following relation holds:

$$N \leq \max\left(0, \left\lfloor \frac{SupLength - T_{beacon} - TS_{CH}}{TS_{CN}} \right\rfloor\right) \tag{4.10}$$

This relation may be useful to a network designer for choosing the number of nodes in the various AUs, as the maximum super-frame length is often a requirement which has to be satisfied for the proper functioning of the monitoring system. The proposed protocol does not require equally large AUs, although such a choice improves the network lifetime.

During the normal network functioning each node has a deterministic duty cycle. Its value can be obtained by dividing the time interval in which the node is active by the whole super-frame duration. Relations (4.11) and (4.12) refer to CN and CH duty cycles, respectively:

$$DC_{CN} = \frac{T_{beacon} + TS_{CN}}{SupLength} \tag{4.11}$$

$$DC_{CH} = \frac{T_{beacon} + N \cdot TS_{CN} + TS_{CH}}{SupLength} \tag{4.12}$$

As RNs do not have a duty cycle, as they are always active, i.e., $DC_{RN} = 1$.

In the WSN modules we considered [48] the power consumption of the transmission and receive states are comparable (being the difference less than 10%). On the other hand, power consumption in the sleep state is much lower (several orders of magnitude). It is therefore possible to approximately estimate the average power consumption of nodes in a simple way, differentiating only between the mean power consumption in the active states (receive/idle and transmission) p_a, and the one during the sleep state p_s.

These parameters can be obtained from the datasheets of the WSN modules or through experimental evaluations. Starting from these data, the average power consumption for CNs, CHs and RNs, respectively, is obtained as follows:

$$\begin{aligned} P_{CN} &= DC_{CN} \cdot p_a + (1 - DC_{CN})p_s \\ P_{CH} &= DC_{CH} \cdot p_a + (1 - DC_{CH})p_s \\ P_{RN} &= p_a \end{aligned} \tag{4.13}$$

According to the rotating election mechanism, AU nodes eventually become RN and CH. As a result, considering that each AU has always one RN, one CH and (N – 2) CNs, and given formulas (4.13), the average power consumption of a node within the AU can be expressed as follows:

$$\begin{aligned} P_{AU} =& \frac{p_a}{N} + \frac{1}{N}\left(DC_{CH} \cdot p_a + (1 - DC_{CH})\,p_s\right) + \\ &+ \frac{N-2}{N}\left(DC_{CN} \cdot p_a + (1 - DC_{CN})p_s\right) \end{aligned} \tag{4.14}$$

where DC$_{CN}$ and DC$_{CH}$ values are those in (4.11) and (4.12).

As it will be shown in section 4.6.1, the average power consumption obtained through (4.14) agrees with the one obtained through our simulations.

By substituting in (4.14) the values from relations (4.11) and (4.12), it is possible to express the average power dissipation of a node as a function of the super-frame length and the number of nodes of the AU it belongs to. As the super-frame length is the reciprocal of the rate at which aggregated packets are forwarded, an interesting relation holds between data rate, number of nodes and average power dissipation within a generic AU. The plot of the resulting relation, together with equation (4.10), that identifies feasible and unfeasible regions, can be used to represent the architectural space for the design of an AU. This means that exploiting such relations it is possible to design the AUs so that the WSN requirements in terms of packet rate and energy consumption will be met. This is possible thanks to the deterministic behaviour of our cluster-based topology management mechanism. This process is also quite simple, as it is enough to follow a three dimensional chart. An example graph obtained with this method is shown in Figure 4.3, where the average AU power is expressed as the ratio between the average AU power P_{AU} and the p_a value. As it was expected, the average power consumption decreases as the number of node increases or data rate decreases. The reason for the former case is that an increase in the number of nodes without modifying the length of CH and RN rounds causes nodes to be elected CH or RN less frequently. As a result, nodes stay more time in the CN state, that is the state that features the lower power consumption. In the other case, i.e. decreasing data rate, the power consumption decreases because the super-frame gets bigger and, as long as data transmission is unchanged, the sleeping phase becomes longer. It can be also noticed from Figure 4.3 that the power consumptions is more sensitive to the number of AU nodes rather than the data rate, as it rapidly falls when the number of nodes increases. This is especially true for low N values, e.g., from 1 to 3 nodes the average power decreases by 50-66%. The same behaviour can be noticed observing the contour lines, each one identifying a locus of points with the same average power consumption. Notice also that there is a limit on the number of AU nodes given a maximum data rate or, vice versa, on the maximum packet rate given a defined number of AU nodes. In fact, when the number of nodes increases, also the number of time slots becomes higher, so the minimum super-frame length also grows. There is an unfeasible region, characterized by too high data rate and node number at the same time. This region is the set of all AU configurations that do not satisfy eq. (4.10), that is represented in Figure 4.3 with the grey area.

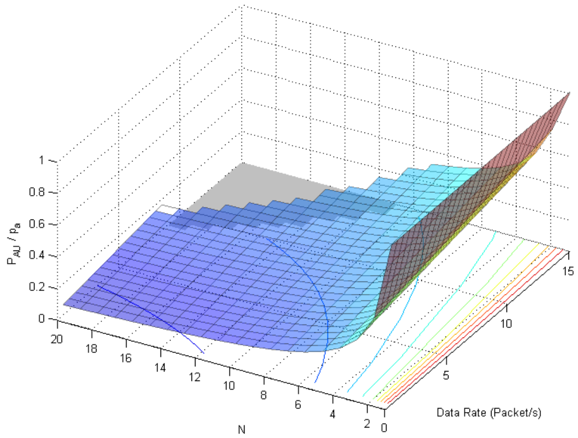


Figure 4.3: Design space of the proposed topology management protocol. Surface and contour lines: average power consumption of a node as a function of the required data rate and power consumption of its AU. Grey area: unfeasible combinations of N and Data Rate.

4.6 Performance Evaluation

In order to assess the advantages introduced by our topology management protocol, we simulated such a protocol by means of the *ns-2* [74] simulation tool. We extended the IEEE 802.15.4 model provided by the standard ns-2 distribution in order to directly control the radio channel assignment and sleep/wake-up schedules. On top of this we implemented our topology management protocol and a SPEED-inspired protocol. SPEED is a geographical real-time routing protocol for WSNs able to meet soft real-time deadline by imposing a minimum forwarding speed to data packets. The implementation of this protocol slightly differs from the original SPEED protocol, as presented in [12, 13]. In fact, as packets forwarding does not involve single nodes, but whole AUs (through the RNs), the address used here to route data packets is not made up of the real geographical coordinates of the current RNs, but of the *virtual coordinates* of the whole AU. Another difference is that hop-to-hop transmissions require ACKs and the per-hop delay is calculated according to the formula

$$delay = W_q + (T_{ack} - T_s)/2, \tag{4.15}$$

where W_q is the time elapsed waiting in the transmitting queue, T_s is the packet arrival time and T_{ack} is the time when the ACK is received. Finally, as the RNs periodically change, we need some way to keep the network in steady state even after the election of new RNs. Firstly, when a new RN is elected, the old one sends the new RN his neighbouring table. Consequently, as soon as an RN becomes active, it immediately sends a broadcast beacon, so that its neighbours can update their neighbouring table with the MAC address of the new RN. A second beacon is sent after a short time, in order to minimize the chance that any neighbour will fail to update its neighbouring table. Then the node can start sending periodic beacons, as described in [12, 13].

The physical parameters of the simulated nodes are taken from the datasheet of the MaxStream XBee modules we used [48].

4.6.1 Energy Efficiency of the proposed solution

Our first objective is to assess the accuracy of the approximated analytical model through accurate simulation. In this section, our basic scenario is constituted of 1500 sensor nodes grouped in 100 AUs, each with 15 nodes.

The monitoring area is set to 10000 m^2 (a square with 100 m sides), while the area covered by a single AU is 100 m^2 (a square with 10 $m$sides). The frame length for a CN data packet is 25 bytes. We assumed here the worst case in which there is no data aggregation, but all the values gathered from CNs are collected and used to fill the data field of a single bigger packet which is then sent to the RN. The setpoint speed of the SPEED protocol, i.e. the minimum delivery speed that has to be maintained in order to meet the packet deadline, is set to 1 *km/s.* Starting from this scenario, we also simulated several different other scenarios maintaining the same number of nodes per AU and super-frame length, but with varying network size and number of nodes in the whole WSN. The resulting power consumptions we obtained were very similar in both the average values and standard variations, so the graphs are not shown here. However, the fact that increasing the size of the WSN without modifying AU parameters does not affect energy efficiency shows that the proposed topology management mechanism is scalable in terms of energy efficiency versus network size. Furthermore, this result confirms that the only critical parameters for power dissipation are N and *SupLength* (or the data rate) so, despite the approximations, our model provides sufficiently reliable results.

The results obtained from the described scenario are shown in Figure 4.4. As the number of nodes per AU here is constant, the only factor that directly affects power consumption is the super-frame length. In fact, with a longer super-frame nodes can stay asleep for a longer time. On the other hand, as the super-frame becomes shorter, the CH and CN duty cycles increase, so we necessarily have an increase in power consumption. The power consumption we obtained through both analysis and simulations is much lower than the average values obtained from the SPEED protocol alone. In fact, using the same parameters for the simulation, but without lowering the nodes' duty cycles, we obtained an average power consumption of about 163 mW. We also notice that the plot in Figure 4.4 shows an asymptote slightly lower than 12 mW. This is due to the RN, which always stays awake, while all the other nodes reduce their power consumption by lowering their duty cycles. So, when the super-frame length significantly increases, the mean power consumption of a node inside an AU converges to a value that is the sum of the power consumption of the RN plus the power consumption of the other AU nodes in the sleep state, divided by the number of AU nodes. However the most important detail to notice is that the approximated analytical model given in Sect 4.5 and the accurate protocol simulation obtained through ns-2, provide very similar results. This happens thanks to the deterministic behaviour of nodes in terms of energy consumption. What mainly affects power consumption of the nodes is their duty cycles and with the proposed protocol duty cycles are imposed by the network parameters and do not change. This is valid only for average values. Obviously, when a node plays the RN role, it spends much more energy than a CN. However, the time it will be RN is limited and roles are assigned in rotation. Thanks to these features, energy consumption within each AU is balanced and the average power consumption in the long term is well approximated by the value obtained through relation (4.14).

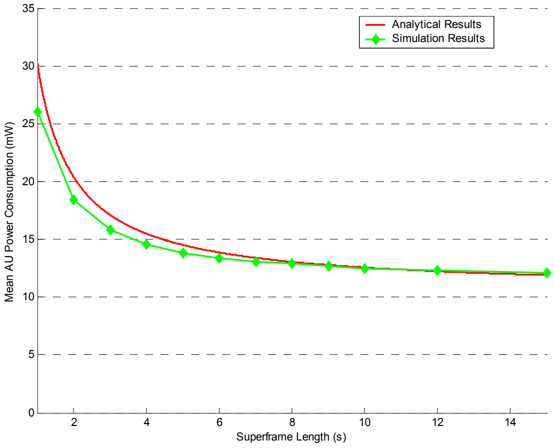


Figure 4.4: Mean AU power consumption vs. varying super-frame length.

4.6.2 Effect of the delay bound on CH to RN transmissions

One of the objectives of our topology management mechanism is to obtain a bounded delay for data transmission inside the AU. This behaviour is achieved by scheduling all data transmissions within the AU in a TDMA fashion and avoiding collisions among data from different AUs. While this behaviour is easy to obtain for CN-to-CH data transmissions through the cellular radio architecture, obtaining a similar behaviour for CH-to-RN data transmissions is not so simple. The reason is that RNs need to be always active on their radio channels in order to guarantee routing fidelity, while they should switch to the AU radio channel to make transmissions occur during the predefined time slot, thus obtaining a bounded delay. However, both bounded delay and guaranteed routing fidelity are essential requirements of our protocol. The only solution suitable for time-critical data would be to have a fully synchronized WSN, in which, instead of being always on the broadcast channel, all the RNs simultaneously switch to the radio channel of their AU at the same time, and simultaneously go back to the broadcast radio channel. However, this solution is not flexible, as synchronization should be maintained along the whole WSN and all the AUs should have the same super-frame length. Moreover, the main aim of our approach is not providing support for high-critical data transmission, but avoiding unbounded delays while routing soft real-time messages. For this reason we devised a different solution, in which the CH temporarily switches to the broadcast radio channel, transmits data and then goes back to the channel of its AU. The transmission of the aggregated data is asynchronous with the RN, although it remains synchronous with the AU. In fact, a time slot TS_{CH} is defined, so that transmission needs to be performed during the defined time. If the time slot elapses before data transmission has finished, the CH drops the packet and goes back to the channel of its AU. This way, bounded delay is maintained, but with a slightly increase on the packet loss rate. That increase can be limited by prioritizing CH frames. Our idea is to select very close CH-RN pairs, so that the signal received by the RN from its CH is the highest from all the other nodes. In addition it would also be possible to set for CH-to-RN transmissions a higher transmission power than the one used for RN-to-RN forwarding. We simulated the worst case in which the transmitting power is the same and only the lower distance prioritizes CH-to-RN packets. The CH-to-RN packet success ratio is shown in Figure 4.5. The chart plots the fraction of the synchronization packets that have been successfully received as a function of the super-frame length. We notice that up to an overall packet injection rate of 188 packets per second (obtained in our scenario with an 8s long super-frame), the success ratio is above 0.98. Above that value there is a slow increase of the packet loss ratio, but it is well controlled (less than 3%) by the IEEE 802.15.4 MAC+PHY layers. Only with the lowest super-frame duration we tested (1s, which provides an overall packet injection rate of 1500 packets/s) the loss rate is noticeable, with about an 8% packet loss ratio. These results are quite acceptable for non-critical data, while for critical data it may be convenient to implement a fully synchronous network.

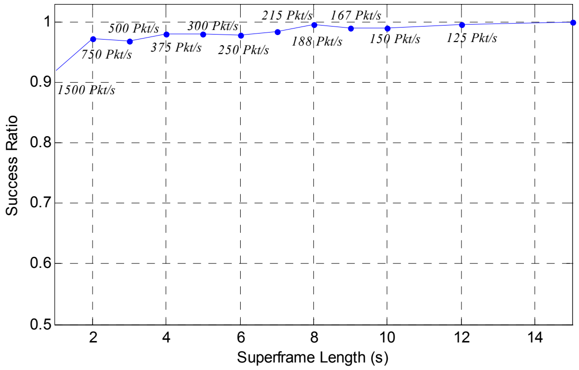


Figure 4.5: Success Ratio for CH-to-RN synchronous communications.

4.6.3 Effect of the proposed topology management mechanism on the SPEED routing protocol

In this section we analyze the effect of the proposed topology management mechanism on the data forwarding process. We decided to assess this aspect by comparing the results obtained simulating the SPEED routing protocol alone (i.e., on its own, without our topology management protocol) with the results obtained running SPEED on top of the proposed protocol. The two sets of simulations run in the same scenario, which is different from the one described in the previous section. Here the scenario comprises 240 nodes which in the simulation of SPEED combined with the topology management protocol, were grouped in 16 AUs of 15 nodes each. Each node has to periodically transmit its data towards the Sink node, that is located in the top left corner of the monitored area. The interval between consecutive transmissions is changed at each run in order to set the desired data rate, from a minimum of 50 packets per second to a maximum of 550 packets per second. As here we only want to assess the effect of the topology management mechanism, data aggregations is disabled, i.e. the CH sends to the RN all the values collected by the CNs, which are then forwarded to the Sink node. Notice however that all the values collected by the CH are packed into a single data frame. The RN in turn forwards all the data transmitted by the CH within a single, bigger data packet. So the use of data aggregation techniques only affects the size of these data packets. Here the pessimistic value is assumed, i.e. the sum of the payloads of all the AU nodes. The

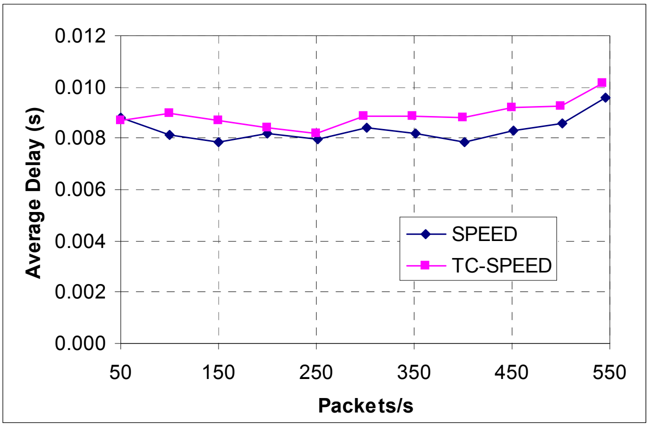


Figure 4.6: End-to-end delay.

results of our simulations are shown in Figure 4.6, 4.7 and 4.8. The graphs labelled as TC-SPEED in the Figures 4.6-4.8 refer to SPEED running on top of our topology management protocol. In particular, Figure 4.6 shows the end-to-end delays experienced with and without our topology management protocol. We notice that values are quite similar, although the SPEED protocol alone features generally slightly lower delay values. This is probably because of the larger packet size. However, while the packets are larger, there is actually a reduction of their number, as data from multiple CNs is packed into one packet. In these conditions the IEEE 802.15.4 CSMA/CA protocol achieves better collision avoidance capabilities. This behaviour is clearly shown in Figure 4.7, where the packet loss percentage is plotted.

While the packet loss of the standard SPEED protocol rapidly increases with the increased packet injection rate, the increase with the adoption of our topology management protocol is quite limited. The reason for this behaviour is that the SPEED protocol alone, as compared to SPEED running on top of our topology management protocol, uses a much higher number of smaller packets to carry the same amount of data. So in addition to the much lower energy consumption, another important result of our topology management protocol is the increased network capacity. Notice also that the network capacity could be further improved if data aggregation on CH and RN nodes would be performed. The effect of the increased network capacity is shown also in Figure 4.8, where we notice that, despite the fact

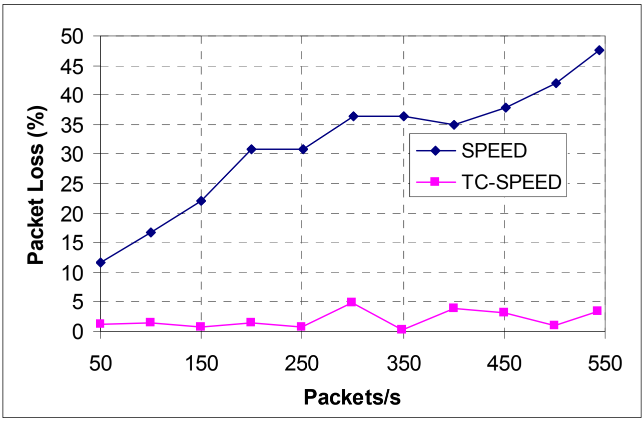


Figure 4.7: Packet loss ratio.

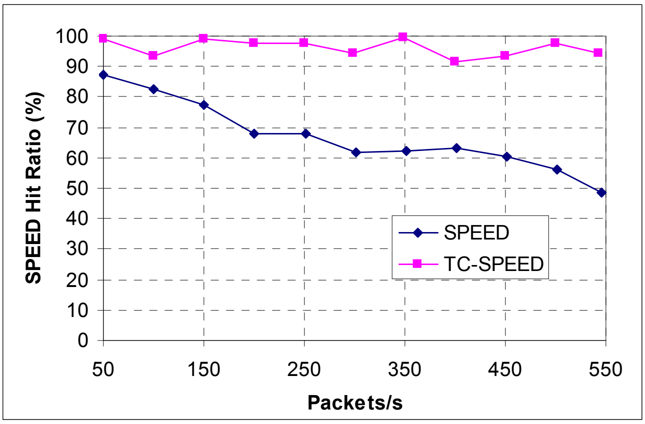


Figure 4.8: SPEED Hit ratio.

that average delays are often slightly higher, the SPEED hit ratio, i.e. the fraction of packets that meet the end-to-end forwarding speed requirements, is much higher with the adoption of our topology management protocol than with the SPEED protocol alone. The reason is that also lost packets are taken into account and while the SPEED protocol alone reaches very high values, up to 48%, the increase on packet loss ratio experienced with our topology management protocol is always under 5%.

4.7 Concluding remarks

This chapter described and analyzed a topology management mechanism with bounded delay for WSNs. This is a cluster-based protocol that has to work together with a real-time routing protocol to meet soft real-time constraints while achieving high energy efficiency. The proposed topology management protocol creates a super-frame structure where each node has an assigned time slot and data transmission is performed in a TDMA fashion. This access mechanism allows nodes to shut down their radio when no transmissions or receptions are needed, thus significantly decreasing their average energy consumption. In addition, the TDMA mechanism imposes a bound on the delay of intra-AU communications. Performance results obtained through simulations run under ns-2 showed the good behaviour of the topology management protocol in terms of energy consumption and also showed the increased performance of the routing protocol running on top of it. The reason is that, while the number of source nodes is the same, a smaller number of (bigger) packets are forwarded. This results in a much lower packet loss rate that highly increases the quality of the overall monitoring application. A slightly simplified version of this topology management mechanism has been implemented on the Maxstream XBee modules to show the feasibility of the proposed approach.

Chapter 5

An improved dynamic topology management protocol for RT-WSNs

The previous chapter described a novel approach to achieve real-time performance while prolonging network lifetime, based on the idea of separating the energy and delay requirements by addressing them at different levels of the protocol stack. This approach exploits the combination of an energy-efficient topology management protocol with a non-energy-aware routing protocol enforcing a real-time behaviour in data forwarding.

This chapter describes an improved topology management protocol which is based on the protocol described in Chapter 4, but introduces dynamic mechanisms that allow for both event-driven data transmission and dynamic network (re-)configuration. Moreover, the dynamic topology management protocol here proposed introduces a novel energy balancing feature that is able to significantly increase the overall network lifetime through a node exchange policy.

The chapter is organized as follows. Section 5.1 discusses the benefits and the limitations of the static approach, while Section 5.2 provides a detailed description of the dynamic topology management protocol. Section 5.3 provides simulation results on network lifetime and routing performance with comparative assessments. Finally, Section 5.4 gives some concluding remarks.

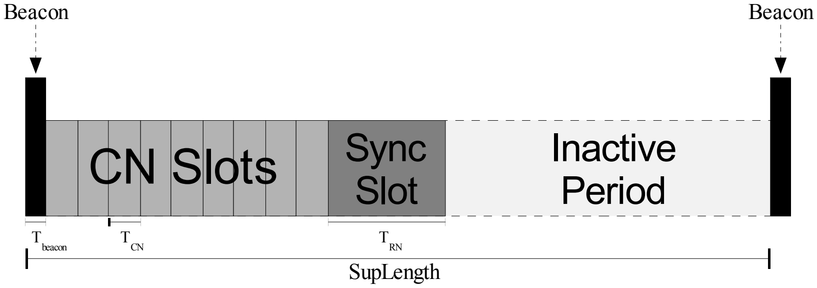


Figure 5.1: Super-frame structure of the static topology management protocol.

5.1 Benefits and limitations of the static approach

In Chapter 4 a static implementation of the concepts described in Sect. 4.3 is given, in which the size and composition of the AUs is fixed and has to be selected at design time. Such an implementation is quite simple, as both the AU creation and channel selection can be based solely on the location of nodes. A virtual grid subdivides the monitored area into a number of small uniform regions, each hosting a cell. To set-up the grid, nodes have to know only the dimensions of the AUs and of the monitored area, in addition to their own location. The transmission sequence making up the super-frame, shown in Figure 5.1, starts with a beacon frame sent by the cluster head to synchronize the transmissions inside the AU. All the CNs have to receive the beacon. Right after the beacon, there are the time slots reserved to the CNs for transmitting data. Then there is the Sync slot, used by the CH to transmit to the RN an aggregated data frame containing all the data collected from the CNs. Once the data is received, the RN performs data forwarding on the broadcast radio channel according to the adopted routing protocol. Finally, in the inactive part of the super-frame, all the nodes except the RN can sleep until the next beacon starts. The length of the inactive section is arbitrary and it can be selected to adapt the super-frame duration to meet the application requirements on the data acquisition rate.

Thanks to the fixed composition of the super-frame and to time-driven communications within the AU, the protocol is easy to analyze in terms of both capacity and energy consumption. In particular, the AU capacity, i.e., the maximum number of nodes that an AU may comprise, is bounded by the super-frame duration, as in formula (4.8).

Given the AU parameters, the duty cycles of nodes can be determined with simple formulas taking into account that: *a*) the CH sleeps only during the inactive part of the super-frame, *b*) the CNs stay asleep also during the slots of the other CNs and during the synchronization slot and *c*) the RNs have to remain active all the time. The duty cycles of CN and CH nodes can be therefore expressed as in formulas (4.11) and (4.12) respectively, while the duty cycle of the RN is 1. In COTS WSN nodes, the power consumption of the transmission and receiving states are usually comparable, while power consumption of the sleep state is significantly lower [48]. As a result, it is possible to approximately estimate the average power consumption of nodes differentiating only between the mean power consumption in the active states (receive/idle and transmission) p_a, and the one during the sleep state p_s. The values for such parameters can be found in the datasheets of WSN nodes or through direct measurements. Considering that each AU has always one RN, one CH and (N–2) CNs and that both RNs and CHs are elected in rotation, the average power consumption of a node within the AU can be approximated as in formula (4.14).

By substituting in (4.14) the values obtained from (4.11) and (4.12), it is possible to express the average power dissipation of a node as a function of the super-frame duration and the number of nodes in the AU it belongs to. As the super-frame duration is the reciprocal of the rate at which aggregated packets are forwarded, it is possible to relate the obtainable data rate to the number of AU nodes and the average power consumption. However, the AU parameters *SupLength*, T_{beacon}, T_{CN}, T_{CH} and N have to satisfy relation (4.8), otherwise their combination would lead to an unfeasible super-frame. An illustrative example of the design space given by the four relations was shown in Figure 4.3, which showed the normalized power consumption as a function of N and data rate. Given the application requirements in terms of time slot durations, such a figure can be used to choose off-line an N value in the feasible region so as to satisfy both energy and data-rate requirements. However, while the static AUs thus obtained allow us to find the trade-off between cost, performance and energy consumption at design time, the approach in Chapter 4 causes some disadvantages as well. The most noticeable is the lack of flexibility. In order to find the best AU parameters, the application requirements have to be known a priori and have to remain unchanged. Once an application has been deployed, it is not possible to extend or to reconfigure it. As an example, when many nodes run out of energy, a possible way of prolonging the network lifetime is to add new nodes full of energy. A reconfigurable protocol might use the new nodes for the energy-consuming tasks and decrease the duty cycles of low-energy nodes, thus prolonging the network lifetime. However, this is not possible using the static protocol. Another limitation of the static approach is that it requires that all the AUs have the same shape and size, whereas several scenarios exist where some areas require a higher node density than others. In addition, the wireless signal quality does not only depend on the distance, so AU selection based only on the location of nodes may not be the best choice. Finally, the super-frame structure proposed in Chapter 4 provides an effective support for periodic data transmission, but there is no direct support for event-driven communications. Even in WSNs that work mainly in a proactive fashion, the support for some kind of aperiodic transmission may be useful. For all of the above mentioned reasons we improved the protocol in Chapter 4 to include dynamic AU creation and reconfiguration. This is described in the next Section.

5.2 The dynamic approach

The dynamic topology management protocol maintains the same architecture of the static protocol. However, to overcome the limits of the static approach, three main changes have been made. Firstly, a slightly different super-frame structure is used to also support event-driven communications for both data transmission and service communications, such as node join requests. Secondly, a more flexible initialization phase has been designed, in which a dynamic clustering algorithm is used to build the AUs and the channel selection follows a distributed approach. Thirdly, as AUs are not fixed, an adaptive AU organization has been introduced, that is able to maximize network lifetime while balancing the energy between different AUs.

5.2.1 Super-frame structure

The super-frame structure of the dynamic approach is similar to the one of the static approach as far as the beacon, the CN data slots, the CH-to-RN Sync slot and the mechanism used to prioritize the CH-to-RN data packet over the other traffic are concerned. However, as node joining requests may come at run-time during data transmissions, to ensure that such requests do

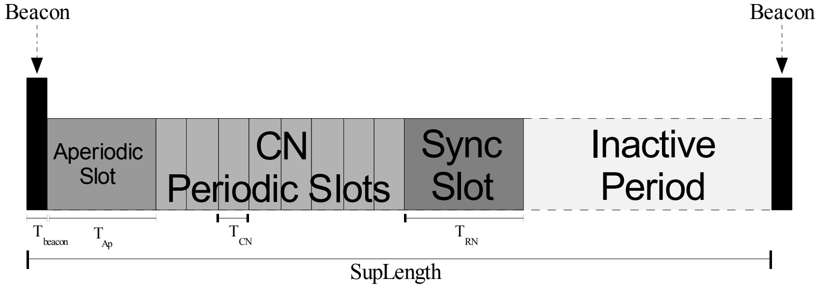


Figure 5.2: Super-frame structure of the dynamic topology management protocol.

not collide with data from the CNs, a proper time slot was introduced. Such a time-slot can also be used to transmit aperiodic data, so we will henceforward refer to it as the aperiodic slot. The novel super-frame structure is depicted in Figure 5.2. Given such a super-frame structure, the minimum super-frame duration in (4.8), duty cycle of CN in (4.11) and duty cycle of CH in (4.12) can be rewritten as

$$SupLength \geq T_{beacon} + T_{Ap} + N \cdot T_{CN} + TCH, \tag{5.1}$$

$$DC_{CN} = \frac{T_{beacon} + T_{Ap} + T_{CN}}{SupLength}, \tag{5.2}$$

$$DC_{CH} = \frac{T_{beacon} + T_{Ap} + N \cdot T_{CN} + T_{CH}}{SupLength}, \tag{5.3}$$

where T_{Ap} is the duration of the aperiodic slot. As in the static approach, the duty cycle of the RN is 1 and the average power consumption can be approximated with formula (4.14).

5.2.2 Node Initalization

The initialization mechanism for the dynamic topology management protocol has to accomplish three different tasks, i.e., Node Discovery, AU Creation and AU Join.

Node Discovery: The node sends a hello message and then collects information about the neighbours. In addition to discovering neighboring nodes, this task has also to assess the WSN state. In fact, a node can be activated either when the WSN is being deployed for the first time or when the network is already active but more nodes (or just *new* nodes) are needed. In the former case, the AUs have to be created, thus the node enters the AU Creation phase. In the latter case, a node enters the Join AU phase. The state of the WSN can be assessed by listening to the messages from other nodes. In particular, if a node recognizes on the broadcast channel messages sent by RNs, it assumes the AUs have already been created, so it can join one of the existing AUs, i.e., the one with the best link quality. On the other hand, if the node only hears other hello messages or election messages, then it realizes that the WSN is not active yet and therefore switches to the AU Creation task.

AU Join: This task is performed by switching to the radio channel of the chosen AU and sending an *AU_join_request* to the relevant CH. Such a request must not interfere with TDMA communications. For this reason, the node has to wait for the beacon and it will transmit during the aperiodic slot. Depending on the current AU conditions, the CH can accept or refuse the association. As a higher number of nodes per AU means a smaller average energy consumption, the association request is refused only when the maximum number of nodes per AU is reached for the desired superframe duration, i.e., when the addition of a time-slot would prevent the AU from reaching the desired data rate (given by the reciprocal of the superframe duration). In this case, the node can try to join the next best AU, and so on. If all the known AUs refuse the association, the node starts its own AU. In the case the association request is accepted, the CH assigns a new TDMA slot to the node and broadcasts the membership and the new TDMA schedule to all the AU members. Such a mechanism allows nodes to join or leave without the need for re-initializing the AUs and without affecting the TDMA transmissions of the operating CNs. However, all the nodes have to maintain the knowledge of the whole AU and the slot assignments, as they will eventually become CHs.

AU Creation: The objective of this task is to partition the WSN in different AUs, each with one CH, a unique ID and a radio channel that is different from the ones of the neighboring AUs. In order to simplify the problem of AU creation, it can be split into three different sub-problems, i.e., grouping the nodes into AUs, selecting AU addresses and selecting AU channels. The former is a typical clustering problem, in which the main objective is to find clusters with a balanced number of nodes. As

WSNs may comprise a very large number of nodes, node grouping should be accomplished in a distributed fashion and, as the nodes are energy-constrained, the distributed clustering algorithm should be fast and require a small number of messages. The distributed clustering problem has been widely studied in the literature and several algorithms have been specifically designed for ad-hoc and sensor networks, e.g., [75–79]. Such algorithms can be used to elect the first CHs and to partition the WSN into AUs. However, as the CHs are elected in rotation, it is not sufficient that all nodes can communicate directly with the CH, but all the nodes in the AU should be able to communicate with each other, i.e., they should belong to the same radio domain. The easiest way to achieve this is to limit the AU physical size, so that the AU area is contained in the radio coverage area of all its nodes. This can be achieved by discarding all the association requests coming from far nodes, or by imposing an association acceptance probability dependent on either the proximity or the signal strength of the requesting node. The latter solution is more effective in the long term, as it allows for a partial spatial overlapping between different AUs that can be exploited by the node exchange policy to maximize the network lifetime.

The second activity related to AU creation is assigning a unique ID to each AU. This is necessary as the routing protocol addresses packets on a per-AU basis. Assuming that each node is given a unique address, the AU ID can be set equal to the address of its first CH. In the case of geographic forwarding, the routing protocol uses the coordinates of nodes to address the data packets. In this case the coordinates of the RNs should be used. However, as RNs periodically change, neighbours tables could become inconsistent as soon as a change occurs. To avoid this problem it is possible to use, instead of the RNs coordinates, the AU centroids, calculated from the coordinates of all the nodes belonging to the AU.

The last activity concerning AU creation is the definition of the cellular radio architecture, i.e., the selection of a dedicated radio channel for each AU. Different AUs can use the same radio channel, but only when they do not interfere with each other. At the end of the clustering algorithm, the CH picks a random channel for intra-AU communication and announces it via a broadcast message. The broadcast channel is excluded by the selection, as well as all the radio channels used by nodes from other clusters that can be directly heard by the CH. However this is not sufficient to avoid interference between adjacent AUs, as a node X might be able to hear two different CHs that cannot communicate with each other. After joining one of them, communication between X and its CH may suffer from interference from the other CH. To avoid this situation, a procedure to detect channel conflicts and resolve them before the AU creation is needed. In general, a node that finds two AUs using the same channel should alert both the CH candidates. Then, only one of them will have to change the channel, picking another random channel among the unused ones. For this purpose, a fixed rule can be used, e.g., the node that maintains the channel might be the one with the highest address. However, to avoid the overhead due to a large number of channel conflict alerts, nodes should broadcast such messages after a small random period. If, in the meanwhile, the same alert is received from another node, there is no need for other nodes to send it.

5.2.3 AU organization

When using the static topology management protocol, the network designer can select the parameters of each AU in such a way to achieve a suitable trade off between energy-efficiency and timeliness, as in the example of Figure 4.3. However, when using the dynamic approach it is not possible to select in advance the exact number of nodes belonging to each AU, as it depends on the distributed clustering algorithm and on the exact topology. In this case it is possible to choose the super-frame duration so as to maintain the desired data rate. Then, it is up to the protocol itself to address the network lifetime optimization through a dynamic adaptation of the AUs. Such a dynamic mechanism exploits the fact that the duty cycle of nodes is a function of the AU parameters, so it is possible to estimate the average power consumption after a topology change through formula (4.14). Similarly to the static approach, here the average power consumption decreases when either the number of nodes or the super-frame duration increases. As the super-frame duration is set to meet the application requirements in terms of data rate, it cannot be dynamically modified by the protocol. However, the number of nodes of the AUs can be adapted by the topology management protocol through a suitable node exchange policy. This policy, which aims at balancing the expected lifetime of the nodes belonging to the AUs, follows two rules. The first rule is that an AU can request a node exchange only when its lifetime is smaller than the mean of the lifetimes of its neighbouring AUs. The second rule is that a node exchange request can be sent from an AU to another only if the difference between the lifetimes of the two AUs exceed a defined threshold (in percentage). These rules allow node exchanges only when a noticeable improvement in lifetime is obtained, because a large number of non-controlled node exchanges would instead lead to high overhead and low benefits.

5.2.4 Lifetime Estimation

One of the main objectives of this protocol is to maximize network lifetime. Here we make the conservative assumption that the network is properly working when all the AUs are active and provide the Sink with the information from their nodes. As a result, when even a single AU is no longer able to transmit its data, the network is not working properly. For this reason, we define the lifetime of the network (in properly working conditions) as

$$LT_{WSN} = \min_{i \leq N_{AU}} LT_i, \tag{5.4}$$

where LT_i is the i-th AU lifetime and N_{AU} is the AU number. According to this definition, to improve the network lifetime, the time at which the first AU dies has to be delayed. Nodes forming the AU have different energy consumption depending on their state, being RN the most energy-greedy state and CN the less energy-expensive one. State rotation balances energy among the AU nodes. However, in order to be appointed CH or RN, a node must have enough energy to accomplish the task. For this reason, nodes below a defined energy threshold E_{TH} cannot be elected CH or RN as, although they can still work in the CN state, they cannot guarantee the correct functioning of the AU. For this reason we consider an AU featuring an average energy below E_{TH} as non-working properly and we define the lifetime of an AU as the remaining time before the average energy of nodes drops below the E_{TH} threshold. Such a threshold has to be set by the network designer to a value that, at least, should allow a whole RN round to perform. A graphical representation of the AU lifetime is shown in Figure 5.3. According to the definition, the AU lifetime can be expressed as

$$LT_i = \big(\bar{E}_{AU} - E_{TH}\big)/P_{AU}, \tag{5.5}$$

where $\bar{E}_{AU}$ is the arithmetic mean of the residual energy of the AU nodes and P_{AU} is the average power consumption of AU nodes, calculated using formula (4.14). $\bar{E}_{AU}$ has to be computed gathering information about all the

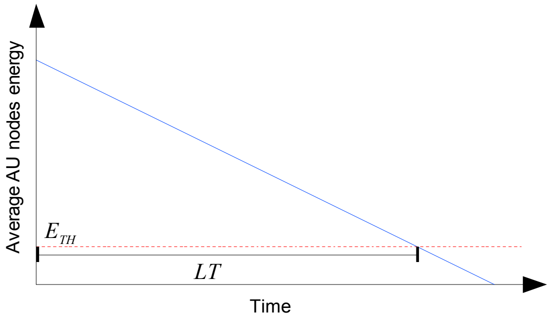


Figure 5.3: Graphical Representation of AU lifetime.

nodes. The most efficient way to obtain such an estimation is transmitting the residual energy value in the CN data packets and the aggregate value in the Sync packet. This way the RN is able to compute the average residual energy and the AU lifetime without any additional message exchange. For this reason, the expected AU lifetime is updated every super-frame.

5.2.5 Dynamic energy balancing

To achieve energy balancing among neighbouring AUs, the relevant RNs have to exchange their expected lifetimes. For this reason, the computed expected lifetime, decreased by the time elapsed from the last update, is piggybacked on data and service packets. In this way, it is possible to perform dynamic adaptations by moving nodes from an AU with a long lifetime to one with a shorter lifetime. As an RN knows the lifetime expectations of its AU as well as those of neighbouring AUs, it is able to assess when its AU lifetime is shorter than the average of the neighbours lifetimes and it can therefore send a help request message ($Help\_req$) to a neighbouring AU. If the RN gets a confirmation message (Yes_help), a CN node will leave the original AU to join to the short-lasting one, helping it to decrease its average power consumption and to increase the expected lifetime, as represented in Figure 5.4. Here, after the node exchange at T_{NE}, the average power consumption of AU 2 decreases, so that the expected lifetime increases from LT_a to LT_b. Otherwise, if a negative response (No_help) is received, the short-lasting AU can send the help request to another RN. The choice of the node that can leave its original AU and join the one re-

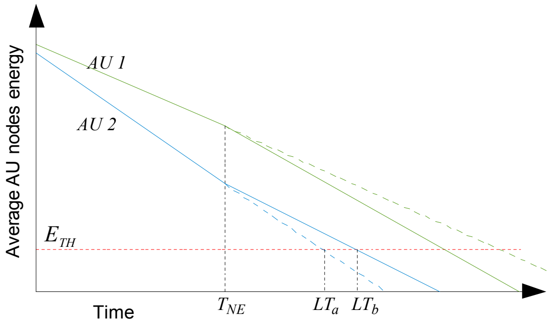


Figure 5.4: Example of node exchange between two AUs.

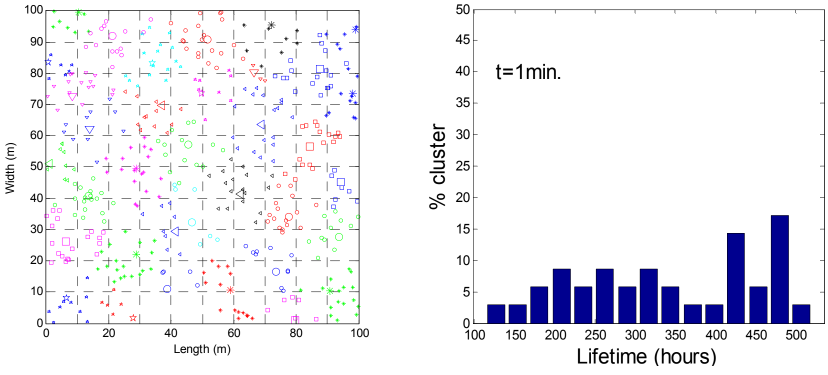
questing help is performed by the CH, taking the position of the AU nodes into account. So, the help request is forwarded to the CH, that searches for the CN closest to the requesting AU that has enough energy to leave (i.e., energy higher than the E_{TH} threshold). If such a node is sufficiently close to the requesting AU (e.g., less than half the transmitting range of nodes), the CH sends a confirmation message to the RN, otherwise it sends a negative response. The RN node must wait for the CH response before sending the *Yes_help* or *No_help* message, because a help request can be refused when no suitable nodes are found in the AU. There are also other situations in which a help request may be refused. Firstly, if another neighbouring AU is using the same radio channel of the requesting RN, as a node exchange could lead to collisions between intra-AU communications. As a result, unless the two AUs sharing the same radio channel are at least as distant as twice the optimal transmitting range of nodes, the help request has to be refused. Secondly, before allowing a node to leave the AU, the RN has to compute the expected AU lifetime after the node leaves. If the resulting lifetime is shorter than the one of the requesting AU, then the help request will be refused. In this way, unstable lifetime due to repeated node exchanges between two AUs is avoided. Thirdly, the *Help_req* is refused if the RN receiving the help request has sent in turn a *Help_req* to another neighbour, as it means that there are more energy-rich AUs in proximity, thus the requesting RN might find a better option. In the case the node exchange is confirmed, the CN is informed during the next super-frame. Then the CN will switch to the radio channel of the destination AU and perform the AU join task.

5.3 Performance evaluation

In order to assess the performance of our dynamic topology management protocol, we used the *ns-2* [74] simulation tool. We extended the IEEE 802.15.4 model provided by the standard ns-2 distribution in order to directly control the radio channel assignment and sleep/wake-up schedules. On top of this we implemented our topology management protocols and SPEED, a well-known, well-studied and easy-to-implement real-time routing protocol that performs geographic forwarding while enforcing a minimum delivery speed. The physical parameters of the simulated nodes are taken from the datasheet of the XBee modules [48]. The default ns-2 channel error model was used, where transmission errors are determined by the signal-to-noise ratio. The duration of the time slots was set to 20 ms for the CN data and to 100 ms for the synchronization slot as well as for the aperiodic slot. The duration of T_{beacon} is not fixed, but it is upper bounded by the duration of a CN data time-slot. The values used for the other parameters are presented in the description of the simulated scenarios.

5.3.1 Effect of the node exchange policy on network lifetime

We simulated a scenario in which 450 nodes were randomly deployed in a 100m × 100m terrain. The radio range of nodes was set to 30 m. We ran a distributed clustering algorithm and, after the AU initialization, we obtained the initial AU distribution shown in Figure 5.5a, where the same combination of shape and grey scale is used to denote nodes belonging to the same AU. Bigger shapes denote CHs. Figure 5.5b shows the distribution of lifetime among the obtained AUs after one minute of network functioning, i.e., before the occurrence of any change. Such a distribution spans over a wide range of values. As a result, if the AU composition did not change, some AUs would stop functioning while other AUs would still have energy remaining. The reason is the non-uniform number of nodes per AU. However, the dynamic adaptation performed in the next steps of the simulation makes the AU composition change thanks to the node exchange policy, that makes nodes leaving long-lasting AUs to join AUs with shorter lifetimes. This is shown in Figure 5.6, in terms of lifetime distribution taken at different times, i.e., at 11, 21, 31 and 41 minutes. It can be noticed that such a distribution becomes narrower and taller with the elapsed time. This means that the lifetime of the AUs tends to be balanced.



(a) (b)

Figure 5.5: AU configuration after the initialization phase (a) and AU lifetime distribution after one minute (b).

At the same time, the lifetime of the overall network according to formula (5.4), i.e., the minimum AU lifetime, significantly grows. However, after a period of time in which the lifetime distribution changes rapidly to balance energy consumption, i.e., 41 minutes in this scenario, the distributions tend to remain stable. This means that further changes are not possible because of physical constraints, (e.g., as nodes are too far from other AUs or can hear two AUs using the same radio channel) or simply because the difference between the expected lifetimes of near AUs does not exceed the minimum threshold (here set equal to 15% in all the simulations). In the latter case, it is possible that other node exchanges will happen after a while, as the expected lifetime decreases for all the AUs and the difference of lifetime expressed in percentage may exceed the threshold. In order to show this situation, we decreased the initial energy of nodes by a factor of 5 and re-run the simulation. The results obtained in terms of network lifetime, given in Figure 5.7, show that a further node exchange happens at about 140 min, as the minimum threshold is reached. The same figure also compares the network lifetime achieved by the dynamic topology management protocol with the one obtained without any topology management protocol for the same network. As expected, our topology management protocol provides the WSN with a significantly longer network lifetime. It should be also noticed that, for the configuration shown in Figure 5.5a, the lifetime obtained

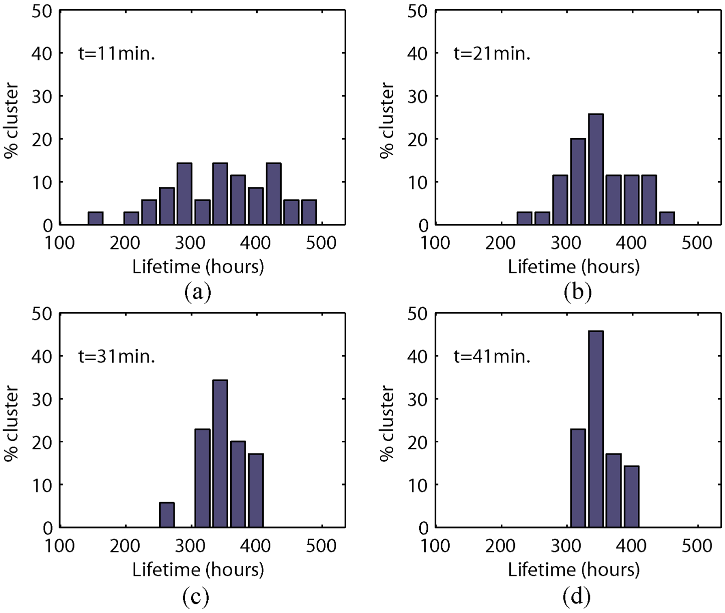


Figure 5.6: Dynamic evolution of the WSN in terms of AU lifetime.

with a topology management protocol dividing the area into fixed virtual grids, such as the static approach in Chapter 4 and GAF [17], is exactly the same obtained without any topology management protocol. In fact, using a transmission range of 30 m, the maximum AU size for the static topology management protocol is about 10m × 10m. As shown in Figure 5.5a, using this AU size with the same random topology there would be several single-node or two-node AUs. As node exchange is not possible in the static protocol, the AU with the lowest lifetime would have the same lifetime of a node in a WSN without any topology management protocol. The same conclusions hold for GAF, as the resulting virtual grids are the same as the ones obtained by the static topology management protocol. As a result, in such a random deployment the dynamic approach would provide much longer network lifetime than both GAF and the static approach.

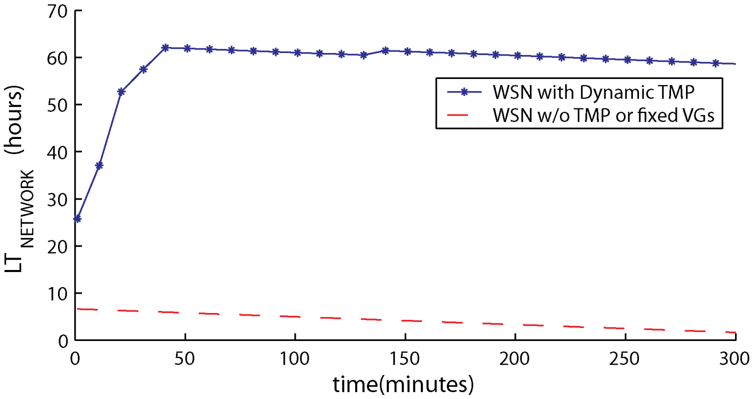


Figure 5.7: The effect of the dynamic AU adaptation implemented by the proposed Topology Management Protocol (TMP) in terms of network lifetime, compared to the network lifetime obtained without any TMP or a TMP featuring fixed Virtual Grids (VGs).

5.3.2 Effect of the topology management protocol on SPEED real-time performance

In order to show the effectiveness of the proposed dynamic topology management protocol, we compared the performance of the SPEED protocol [12,13] alone, i.e., without any topology management protocol, with the ones obtained by the following combinations: *a*) SPEED with the static topology management protocol proposed in Chapter 4, *b*) SPEED with the dynamic topology management protocol here proposed, *c*) SPEED with the GAF topology management protocol[1].

The four configurations will be henceforward referred as SPEED, sTM-SPEED, dTM-SPEED and GAF+SPEED, respectively. Simulations were run in the same scenario, comprising 240 nodes randomly deployed in a 40m × 40m area. In the case of the static approach, nodes were grouped in sixteen 10m × 10m AUs. Each AU was composed of exactly 15 nodes. A similar cluster composition was used for the GAF protocol [17]. In the

[1] Actually the results herein presented are obtained by an enhanced GAF, in which sleeping nodes temporarily turn on their radios and transmit to their leader whenever they produce data. Otherwise, we would obtain a miss ratio greater than 90% even under low workloads, due to messages waiting the pre-scheduled wake-up time.

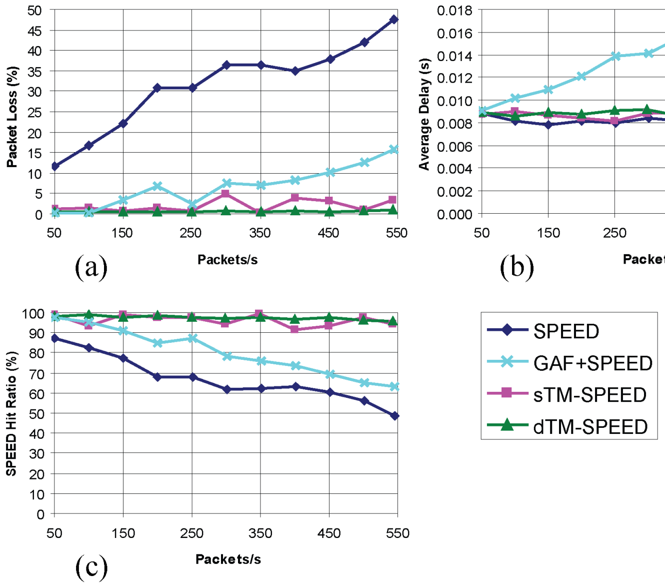


Figure 5.8: Performance comparison between the standalone routing protocol (SPEED), SPEED with the GAF protocol (GAF+SPEED), SPEED with the static topology management protocol (sTM-SPEED) and SPEED with the dynamic topology management protocol (dTM-SPEED).

case of the dynamic topology management protocol nodes built the AUs autonomously. In all the simulations each node had to periodically transmit data to the Sink located in the top left corner of the monitored area.

The interval between consecutive transmissions was changed every run in order to set the desired data rate, from a minimum of 50 to a maximum of 550 packets per second. No data fusion was performed, i.e., at each Sync slot the CH had to pack all the values collected by the CNs in a bigger data packet to be sent to the RN. This choice was made to stress the network injecting the worst-case network load in which the payload of an RN packet is the sum of the payloads of all the AU nodes. The results of these simulations are packet loss, average delay and SPEED hit ratio (i.e., the fraction of packets that meet the requirement on the end-to-end forwarding speed and thus the deadline, as defined in [12]) and are shown in Figure 5.8a, 5.8b and 5.8c, respectively. In Figure 5.8b it is possible to notice that there is not a big difference in the average delay obtained with the SPEED and either the sTM-SPEED or the dTM-SPEED configurations, although SPEED obtains slightly lower values. The reason for the last result is that in our topology management protocols data packets from the CNs are forwarded by the RNs embedded into a single data packet. As a result, the forwarded packets are bigger than the ones transmitted by the SPEED configuration, that only contain data from a single sensor. However, while the delay introduced by GAF+SPEED increases significantly when the workload is increased, the delay introduced by our topology management protocols does not increase. The reason is that, while GAF does not perform any control on traffic injection, the two-level network architecture of our approaches separates data collection from data forwarding through the use of multiple channels and concentrates all the traffic in the RNs. As only one packet is needed for a whole AU, both the sTM-SPEED and the dTM-SPEED configurations provide for a strong reduction on the number of packets to be forwarded and thus of the collision probability. The reduced number of collisions on the broadcast channel yields to the packet loss results in Figure 5.8a. While the packet loss for SPEED and SPEED+GAF configurations significantly grow with the increased packet injection rate, this effect is quite limited in sTM-SPEED and becomes almost negligible in dTM-SPEED. The latter result depends on the adaptive behaviour of AUs, that use a distributed clustering algorithm instead of a pre-defined mapping. The direct effect of bounded delay and reduced packet loss is that the SPEED hit ratio is much higher with our topology management protocols than with SPEED and SPEED+GAF configurations, being the performance obtained with the static and the dynamic approaches quite similar. This consideration means that the proposed dynamic approach is able to significantly improve flexibility and network lifetime of unbalanced WSNs without affecting real-time performance.

5.4 Concluding remarks

This chapter described and analyzed a dynamic topology management protocol for real-time WSNs that extends the one in Chapter 4 in several respects. Firstly, it provides support for both periodic and aperiodic transmissions. Secondly, it allows for dynamic clustering to effectively set-up the AUs when the density of nodes is non-uniform. Finally, it introduces a novel energy balancing feature that is able to significantly increase the overall network lifetime through a node exchange policy. Results obtained through ns-2 simulations showed the effectiveness of the energy balancing technique and its beneficial effect on the performance of the routing protocol running on top of it. Compared with the static approach, the dynamic one provides a significant improvement in the network lifetime for randomly deployed WSNs while maintaining the good real-time performance. On-going work is addressing the implementation of the proposed approach on COTS IEEE 802.15.4 modules and measurements in real scenarios.

Chapter 6

A chain-based routing protocol for industrial WSNs

While many routing protocols exist for traditional WSNs, currently only a few WSN protocols are tailored for industrial environments [80–82]. Moreover, none of them consider the integration with traditional wired networks, although this is recognized as one of the most significant challenges in industrial WSNs [2, 83].

A promising approach for industrial monitoring is the chain-based one, as it not only enables the integration with existing industrial networks, but also takes advantage of it to provide predictable latencies while limiting the power consumption. This chapter investigates the use of a chain-based protocol for industrial WSNs. After reviewing benefits and limitations of the existing protocols, a fully-fledged chain-based communication protocol tailored to industrial WSNs is presented. The proposed protocol, called CCDF (Circular Chain Data Forwarding), takes into account the architecture of industrial WSN deployments and exploits an existing real-time backbone to achieve real-time communication with limited power consumption. In the chapter, the CCDF protocol is discussed and thoroughly analyzed. Analytical relations are derived for the latency of a single hop and cycle times in the case of ideal channel. Then the analysis is extended to the case of noisy channels. An extensive simulation campaign has been performed to validate the analytical results and to show the effectiveness of our approach compared to the standard IEEE 802.11b protocol running a fixed routing.

This chapter is organized as follows. Section 6.1 outlines chain-based routing algorithms for WSNs. Section 6.2 addresses the case for chain-based communication protocols in industrial WSNs and the rationale behind the proposed CCDF protocl. Section 6.3 discusses the mechanisms needed to achieve fault tolerance in the CCDF protocol, while Section 6.4 introduces an algorithm to create the network chain in a distributed fashion. Section 6.5 presents the in-depth analysis of the CCDF protocol, and Section 6.6 validates the results obtained analytically through simulations and provides a comparative performance assessment to show the effectiveness of the proposed protocol. Finally, Section 6.7 provides some concluding remarks.

6.1 Chain-based routing in WSNs

In chain-based routing protocols, nodes form a chain which connects all the nodes and forward data packets along the chain in a sequential way. The chain-based communication paradigm was originally designed to achieve energy efficiency in WSNs running data gathering applications. Energy efficiency is obtained by evenly distributing the workload among all the WSN nodes and, in some chain-based protocol, by using data aggregation at every hop. Although data aggregation in not common in industrial automation, chain-based protocols can still bring considerable benefits to industrial WSNs. In fact, the ordered transmission and forwarding of chain-based communication resembles a token passing, in which the token is loaded with the payload of all the nodes that are preceding in the chain. As each node has to wait the reception of a data packet from the preceding node before it can access the medium, chain-based forwarding is able to avoid packet collisions. Moreover, as data forwarding follows a predefined chain, the path of each data packet is deterministically known. Therefore, a similar mechanism can be used to control both medium access and routing in a way that provides at least statistical guarantees on delivery delay.

Several chain-based schemes exist. The following subsections address the strengths and weaknesses of three widely-known ones, namely, the linear, binary-combining and multiple-chain schemes, respectively.

6.1.1 Linear scheme

In linear chain-based routing, proposed in [84], data packets are transmitted from one end of the chain to the other hand. When a node receives a data packet from the preceding node, it appends its own payload to the received packet and forwards the new packet to the next node in the chain. When the end of the chain is reached, data forwarding restarts from that end in the opposite sense. This approach has been extended in [84], where packets are transmitted along the chain until a special node, called *leader*, is reached. Once the leader has received the data, it forwards it to a base station. As nodes are supposed to be battery powered and the leader's transmissions consume more energy than the others, the leader changes at each round in a rotating fashion, so as to maximize the network lifetime. In addition, data aggregation is used to maintain the same size for all the packets traversing the chain.

Linear chains provide a good level of predictability, as only one node is allowed to access the channel at any time and the path from each node to the sink is deterministically known. However, the protocols in [84, 85] do not address the typical industrial scenarios and only aim at reducing energy consumption. Moreover, when dealing with a large number of nodes, the protocols in [84, 85] suffer from very low scalability, because they assume that all nodes can communicate with each other, and are affected by large delays.

6.1.2 Binary-combining scheme

The binary-combining scheme proposed in [85] divides each round into *log(n)* levels (where n is the number of nodes) and allows parallel communication of nodes. Each node transmits data to its neighbor at the current level. The receiving node raises its level, so it forwards data to its neighbor at the next level. This scheme improves energy efficiency and in some cases, also the average transmission delay as compared to linear chains. However, as nodes can transmit at the same time, collisions may occur, so it is not possible to provide some guarantees on delivery delay. So this scheme is not suitable for industrial WSNs.

6.1.3 Multiple-chain scheme

A multiple-chain scheme that divides the sensing area into multiple regions, each hosting a linear sub-chain, was proposed in [86]. In this approach each linear sub-chain is independent, so it is possible that the transmissions of nodes in different chains occur at the same time. As a result, there is no guarantee that transmissions are collision-free. For this reason, even this scheme is not suitable for industrial WSNs.

From these considerations it follows that, among the existing chain-based approaches, the most adequate for industrial WSNs is the linear scheme. This scheme has already been adopted by Bui et al. [87] to achieve soft real-time communication in multi-hop wireless ad-hoc networks. In that work, the limitations of classical linear schemes were overcome through the use of a different receiving channel for each adjacent node. Unfortunately, the small number of nonoverlapping channels offered by standard wireless protocols (e.g., IEEE 802.15.4) significantly limit the network topologies, that must be very sparse. On the contrary, industrial WSNs can be dense in proximity of an automation cell, thus it is not possible to use this approach to avoid contentions.

The following section investigates novel strategies to improve the linear chain-based scheme and make it suitable for industrial WSNs.

6.2 The Circular Chain Data Forwarding (CCDF) protocol

Most of the WSN protocols presented so far do not consider the architecture of typical industrial scenarios, where some of the sensor nodes are directly connected to a real-time backbone (as shown in Figure 6.1). In this scenarios, nodes can be classified into two categories, i.e., nodes directly connected to the wired backbone (henceforward called *sinks*) and nodes that are not directly connected to it (henceforward called simply *nodes*). In addition to nodes and sinks, an industrial WSN always comprises a base station that collects and analyzes sensor data. In a typical industrial scenario, the base station is connected to the wired backbone and therefore is directly reachable by many (or all) of the sinks. Usually, the performance of the wired backbone in terms of both throughput and latency are some order of magnitude better than those obtainable by the WSN. Therefore, a way to improve the performance of the WSN monitoring application is taking sinks as intermediate destinations, which in turn forward the received data to the final destination, i.e., the base station, over the wired backbone.

A communication protocol for industrial WSNs has to enable nodes to access the medium and forward data packets in a predictable way. Linear chain-based routing protocols such as [84] are able to disseminate data in

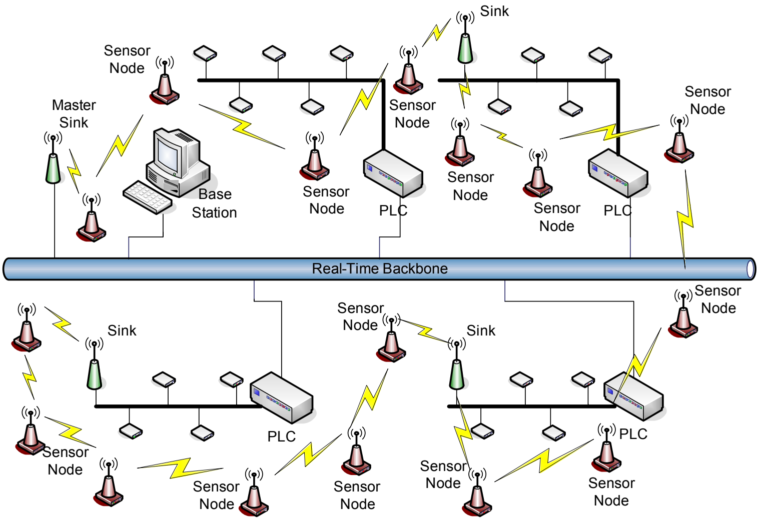


Figure 6.1: Network Architecture.

a predictable and distributed fashion, but, as discussed in Section 6.1, they are not tailored to the industrial communication. The typical transmission pattern of linear chains, that goes from one end of the linear chain to the other hand and then goes back in the opposite direction, is not suitable for monitoring applications in which most of the data transmissions are cyclic. A more application-oriented scheme is one in which data is transmitted from one end to the other end of the linear chain and then the communication restarts from the beginning. However, the problem of such a linear scheme is that the last network device of the chain has to trigger the start of a new cycle of communications. In a large industrial WSN it is likely that the last node of the chain is not under the coverage of the first node. A possible way to overcome this problem is to enforce that both the first and the last device of the linear chain are sinks, so that they can communicate via the wired backbone. However, this solution is not very efficient when sinks are used as intermediate destinations. As the communication between the first and the last sink occurs via the wired link, it is not possible for the nodes to exploit the first sink as an intermediate destination. This will result in longer sub-chains from one sink to the next.

We show an example to clarify the point. Consider the topology shown in Figure 6.2a, comprising 4 sinks (colored circles) and 12 nodes (white

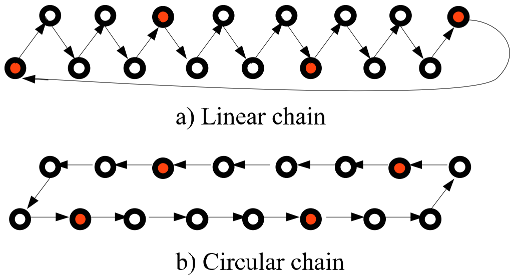


Figure 6.2: Chain-based schemes.

circles). Using the linear scheme, each sub-chain between two consecutive sinks is composed of 5 hops over the wireless link. To further improve the performance of the network a circular chain is proposed to replace the linear chain, as shown in Figure 6.2b. Such a circular chain is the union of multiple linear sub-chains going either from one sink to the next or from the last sink to the first. As it is possible that nodes in different sub-chains are on the same collision domain, data forwarding must be sequential and only one packet can be transmitted at any time, thus the circular chain acts as a logical ring. Data packets are forwarded by nodes across the chain, until a sink is reached, as data is forwarded to the final destination through the wired backbone. Consequently, a sink will not relay data through the wireless network, but it only forwards to the wireless successor a packet without payload, that simply gives to the recipient node the right to transmit. This solution fits well the requirements of industrial applications in which most of the traffic is cyclic. Moreover, using this scheme it is possible to fully exploit the wired forwarding performed by the sinks, as there is a sub-chain between any couple of consecutive sinks. This is clearly visible in the example of Figure 6.2b, where the number of wireless hops in each sub-chain is reduced from 5 to 4. As each node has to forward data from all the preceding nodes until the last sink, shorter sub-chains introduce shorter communication delays. The Circular-Chain Data Forwarding (CCDF) offers several advantages in industrial environments:

Predictability: When a node receives a data packet from the predecessor, it appends its own data (or a special padding, if there is no data to be sent) and forwards the resulting packet to the successor. This chain-based mechanism is used to control both data forwarding and channel access. This means that devices are not allowed to transmit if they have not received a data packet from a predecessor. In this way, contentions are avoided and the sequence of data transmission is predictable, thus the only unpredictability to take into account is that due to the wireless medium.[1]

As the circular chain works like a logical ring, if the traffic follows a known arrival pattern, e.g., periodic traffic, it is possible to calculate the minimum time needed by the network to complete the traversal of the overall chain. This can be used to calculate the minimum achievable update time for a given scenario on the basis of the number of nodes and the amount of data to be transmitted by each node. This feature can help a system designer in tuning both the networking infrastructure (e.g. the maximum length of the chains) and the industrial applications.

Reduced delay: As soon as sensor data reaches a sink, it is forwarded to the base station using the wired link. As the wired backbone provides both higher bandwidth and smaller transmission delays than the wireless network, the overall delay experienced by sensor readings to reach the base station decreases. Moreover, the workload over the wireless link is reduced, as once the data has reached a sink, it continues its path to the base station over the wired backbone.

Extended coverage: Nodes do not need to be within the coverage of any sink, they only need to have two neighbors, i.e., a *predecessor* and a *successor* in the transmission chain. The predecessor and the successor can be either nodes or sinks.

Energy Efficiency: Thanks to the predictable transmission and forwarding mechanism, it is possible to calculate the minimum interarrival time between data packets (or sink's tokens) as the time needed by the network to traverse the whole circular chain in the optimistic case of no packet losses. Nodes can save energy by going to sleep just after having forwarded a data packet and remaining asleep for the minimum interarrival time.

6.3 Fault Tolerance Mechanisms

The main problems of token-based protocols used over a wireless medium are fault tolerance and robustness to errors. One problem of classical token passing is that acknowledging the receipt of each token would be very

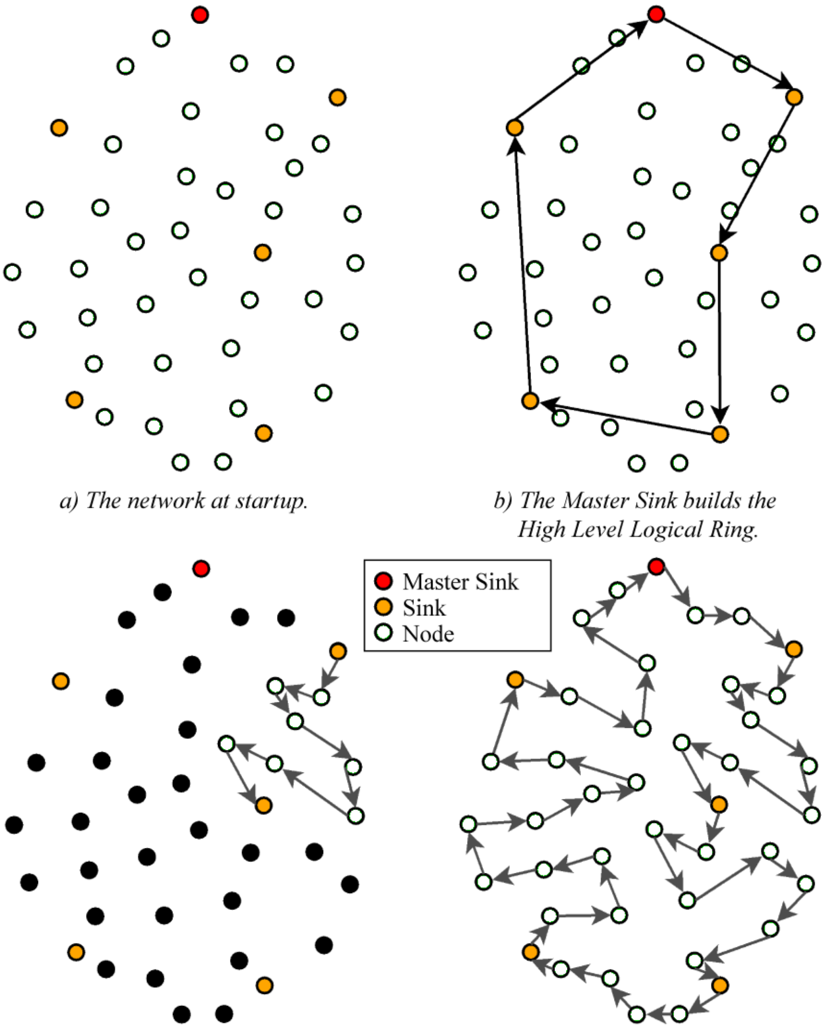
[1] In order to maintain the performance of the industrial WSN predictable, also the wired backbone has to provide a predictable behavior. This is why here a real-time backbone is assumed.

inefficient. However, the CCDF protocol solves this problem by using a single packet to both grant the medium access and send data to its successor. After a node receives a packet from the predecessor, it sends back an acknowledgement frame, which indicates that the node has successfully received data and has acquired the medium access. In the case where a data packet is lost, the sender will not receive any acknowledgement and, after a timeout, retries the transmission. To address the case for a loss of acknowledgement frames, a sequence counter is added to data packets, which is increased at each data transmission. In this way it is possible to recognize data packets originated by a missed Ack and avoid error propagation. Another issue which must be tackled is how to react to node failures. The solution adopted for this protocol is that if a node reaches the maximum number of retries for the direct successor and still does not receive an Ack, it sets the next node as the destination and attempts to transmit again. For this reason, each node keeps in memory not only the address of the direct successor, but also the next two nodes in the chain, which are considered two backup successors. Moreover, it is necessary to maintain the topology information of nodes updated in the case of topology changes, e.g., due to node failure. Therefore, in such cases a node has to send the updated addresses of the next two nodes that follow in the chain to its predecessor. An efficient way to do this is by piggybacking this information on Ack frames.

6.4 Distributed Chain Creation

To solve the circular chain creation problem in a distributed way, we divided the original problem into multiple sub-problems. The basic idea is the following; as a network comprises multiple sinks, the complete transmission chain can be divided into multiple sub-chains from one sink to the next. Each sub-chain is built independently of the other, and at the end all the sub-chains are joined together to form the circular chain containing all the nodes and all the sinks of the whole network. The phases of the chain creation are depicted in Figure 6.3.

The algorithm for the creation of the circular chain is divided into three steps and supposes that nodes are location aware. In the first step, a High Level Logical Ring (HLLR) is created that connects all the sinks. The second step is the association of nodes to the closest sink. The third step is the setup of the linear chains connecting each sink to the next in the HLLR.



a) The network at startup.

b) The Master Sink builds the High Level Logical Ring.

c) Each sink builds a sub-chain.

d) All the sub-chains are joined together to form the final chain.

Figure 6.3: Network Setup.

A summary of the operations occurring in these phases follows.

Logical Ring Creation: The sinks communicate through the wired backbone in order to establish the HLLR. All the sinks send a packet to the MS containing their position. The MS collects them and fills a Sink Table. After a timeout for the last reception expires, the MS builds the HLLR using the Nearest Neighbor algorithm, that at each iteration chooses the closest unvisited sink as the next move. This means that the first sink will be the MS, the second sink will be that closest to the MS, the third will be that closest to the second sink and so on. Once the HLLR has been built, the relevant information is transmitted to all the other sinks. Figure 6.3b depicts the network at the end of this stage.

Node/sink association: Each sink broadcasts packets into the wireless network, containing information such as its address, position, etc. Nodes collect the information on both neighboring sinks and other nodes. Moreover, they broadcast packets containing address and position of the known sinks (both those directly reachable from the node and those known by collecting packets from other nodes) as well as the number of hops to reach them. Additionally, nodes keep the address of the neighbor with the minimum distance (in terms of hop number) from each sink in memory. In this way, temporary paths are established to allow nodes that are not under the direct coverage of any sink to communicate with the closest sink. These paths are used by such nodes to send back the information about their own address, position and neighboring tables to the closest sink. After all nodes have communicated their data to the closest sink, each sink has a different Node Table, containing address, position and neighbors' list of the nodes, for which that is the closest sink.

Chain setup: This is the last phase of the network setup, in which the overall chain connecting all the nodes of the network is built. A possible way to proceed could be to collect, at the MS, all the information about nodes that at the end of the second step of the algorithm is distributed among the sinks. In that case, the MS could use a centralized algorithm to build the chain. However, as building a sub-chain is still a complex problem and a network may comprise a large number of nodes, it is convenient to use a distributed algorithm that allows parallel computations inside the sinks. This approach has two advantages over the centralized algorithm: it requires less memory on the sinks, so it better fits the resource-limited capabilities of sensor nodes, and speeds-up the chain creation. Moreover, it scales better with the number of nodes. The algorithm developed to reach the state depicted in Figure 6.3c is described in the following subsection.

6.4.1 Sub-chain creation algorithm

The algorithm run at each sink to build its sub-chain works as follows:

1) Each sink splits the list of associated nodes into two sets, namely, *outgoing* and i*ncoming nodes*. The former set contains all the nodes for which their distance from the succeeding sink is smaller than that from the preceding sink in the HLLR. The latter set, on the contrary, contains the nodes that are closer to the preceding sink.

2) Each sink sends the information about the incoming nodes to the preceding sink in the HLLR through the wired backbone.

3) After receiving the same data from the succeeding sink, each sink knows the information about all the nodes belonging to the path to the next sink. At this point the sink can compute the part of the chain that starts from it and ends to the succeeding sink in the HLLR. The algorithm used to build a path from a sink to the successor in the HLLR is based on a heuristic approach, that calculates the shortest path using the Dijkstra algorithm at the beginning and then iteratively adds to the chain nodes that are not present in the shortest path. In particular, at each iteration it substitutes a direct link with an indirect communication (a path) having the same source and destination nodes of the direct link, but passing through some unvisited nodes. Among the feasible paths, our heuristic approach selects that which increases the overall traversed distance by the minimum amount.

4) Once a sink has built its sub-chain, the relevant schedule is communicated to the relevant nodes. In particular, the sink creates a packet containing the ordered list of nodes which made up that part of the chain and sends it to the first node. Each node receiving that packet stores the information about the predecessor, the successor, and the two backup successors in its memory. Then, the node forwards such a packet to the successor. Figure 6.3c depicts the network at this point of the chain creation algorithm.

After all the sinks have set up their sub-chain, the MS sends a packet throughout the chain which is used to know the exact length of the whole chain as well as the expected duration of a complete cycle, i.e., the time needed to traverse the network chain. At this point (depicted in Figure 6.3d) the network becomes operational and the nodes start waiting for the data packet from the predecessor, to add their data and forward it to the successor.

6.5 Protocol Analysis

While the CCDF protocol exploits the wired infrastructure of industrial settings to improve the end-to-end performance, by forwarding data as soon as it reaches a sink, it does not actually depend on any particular technology. For this reason, to keep our discussion as generic as possible, in our analysis we only consider the wireless part of the network, e.g., by considering only node-to-sink delays. To obtain the end-to-end delays it is sufficient to add the delay from the sink to the final destination, which is specific to the wired backbone, although it is usually much smaller than that of the wireless part of the path, thanks to the higher data rate and lower packet error rates.

One of the main advantages of our chain-based protocol is that in normal operating conditions, i.e., in the absence of node faults, each data transmission follows the same path to the sink. This feature makes it possible to calculate the delay experienced by data packets in the case of no frame losses.

Consider a network chain comprising M nodes and N_{sink} sinks. The complete chain is made up of N_{sink} different sub-chains, having a length of $L_1, L_2, \cdots, L_{N_{sinks}}$nodes, respectively. If nodes transmit a fixed length payload, adopting the communication mechanism discussed in Section 6.2, it is possible to derive the following relations.

6.5.1 Node Traversal Time (NTT)

The first parameter that we estimate is the delay of a single hop, i.e., the time spent by a packet to traverse a generic node. We call this parameter a Node Traversal Time (NTT). To compute the value of this parameter, consider the activities performed by nodes at each hop, shown in Figure 6.4. When the node i receives a data packet from the predecessor, it has to process the data packet, send the acknowledgement to the predecessor, add its own data to the received data packet, and send the resulting packet to the successor. This means that the NTT can be expressed by

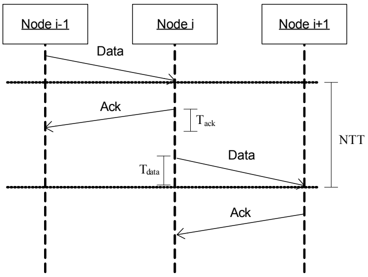


Figure 6.4: Node Traversal Time.

$$NTT = T_{ack} + T_{data} + T_{proc}, \tag{6.1}$$

where T_{ack} is the time needed by the transmission of the Ack frame, T_{data} is the time needed for the transmission of the data packet and T_{proc} is the overall processing time spent by node i. Both the terms T_{ack} and T_{proc} are not dependent on the position of a node in the chain. In fact, the Ack has always the same size, therefore its transmission time is constant when the data rate of nodes is fixed. Even the processing time can be considered constant in the NTT calculation. In fact, sensor nodes can be small embedded devices running either a single task or a lightweight real-time operating system, and so it is possible to estimate the worst case execution time and use it to obtain an accurate estimation of the NTT. On the contrary, the term T_{data} is not constant, as the amount of data that a node has to send to the successor is dependent on the node position in the chain. As each node appends its own data in the packet, the size of the data packets will vary from the minimum size of a packet containing no data (i.e., packets sent by the sinks) to the maximum size of the last node of the longest sub-chain. In particular, if we define p_i the position of node $i$from the beginning of its own sub-chain (while for all the sinks p_i=0), we can express the transmission time T_{data} for the node i as

$$T^i_{data} = T_{ov} + T^i_{payload} = T_{ov} + p_i \cdot \Delta T, \tag{6.2}$$

where T_{ov} is the constant overhead due to the protocol header and the lower layers encapsulation and$\Delta T$is the time contribution given by the data payload appended by each sensor node, i.e., the length of the payload divided by the data rate of communication. It must be noted that in the case of a sink, the packet is used only to grant the medium access to the successor, therefore there is no payload. As a result, $T_{data} = T_{ov}$ for all the sinks.

6.5.2 Chain Traversal Time

The Chain Traversal Time (CTT) is the time spent by the network to complete a cycle across the whole network chain in the optimistic case in which there are no packet losses and retransmissions. This value is important for two main reasons. Firstly, it gives an upper bound on the cycle times[2] of traffic that can be supported by the network, e.g., if CTT = 1 s, it will not be possible to support traffic requiring cycle times lower than 1 s using the given network topology. If there is traffic with such requirements the designer can either use a wired dedicated network or enhance the network topology so as to decrease the cycle times. As the CTT depends on the number of sinks, a possible operation to allow the support of traffic with higher rates is to add some new sinks to the network chain. Secondly, this parameter can be used to improve the energy efficiency of the network. In fact, a node that has transmitted its data packet at time t and has received the Ack from the successor knows that it will not receive any communication before a CTT from the transmission time t. As a result, it can sleep until time t + CTT – T_{sm}, where T_{sm} is a safety margin to account for possible clock drifts. In this way the duty cycle of nodes, and so energy consumption, can be drastically reduced.

Suppose that each sensor node transmits its data at each cycle. The CTT value can be calculated as the sum of the NTT values of all the nodes in the network, plus the sum of the NTT values of the sinks. In the most

[2] The cycle time is defined as the time between two consecutive packet transmissions from the same node.

general form, the CTT value can be expressed as

$$\begin{aligned} CTT =& N_{sink} \cdot NTT_{sink} + \sum_{k=1}^{N_{sink}} \sum_{i=1}^{L_k} NTT_i = \\ =& \left(N_{sink} + \sum_{k=1}^{N_{sink}} L_k \right) (T_{ack} + T_{proc} + T_{ov}) + \\ &+ \sum_{k=1}^{N_{sink}} \sum_{p=1}^{L_k} \Delta T \cdot p. \end{aligned} \tag{6.3}$$

This relation can be simplified in a particular scenario, i.e., when the chain is balanced. Under this assumption, each of the sub-chains of the network has the same length, i.e., $L_1 = L_2 = \cdots = L_{N_{sink}} = L$, and the number of sensor nodes is $L \cdot N_{sink}$. As a result, the CTT is N_{sink} times the delay of a single sub-chain with L nodes. As in a sub-chain there are L data packets with payload (sent by the sensor nodes) and one packet without payload (sent by the sink), the CTT can be expressed as

$$\begin{aligned} CTT =& N_{sink}(1 + L)(T_{ack} + T_{proc} + T_{ov}) + \\ &+ N_{sink} \cdot \Delta T \cdot \frac{L(L+1)}{2}. \end{aligned} \tag{6.4}$$

Now, under the same hypotheses, to analyze how the CTT value varies as a function of both the number of sensor nodes, M, and the number of sinks, N_{sink}, it is sufficient to substitute $\frac{M}{N_{sink}}$to the original variable L in formula (6.4). In this way we obtain formula (6.5), which can be used by a network designer to dimension a network in terms of both number of sensor nodes and number of sinks.

$$CTT = \frac{(M + N_{sink})(M \cdot \Delta T + 2N_{sink}(T_{ack} + T_{proc} + T_{ov}))}{2N_{sink}} \tag{6.5}$$

Figure 6.5 shows the design space of an example network configuration obtained through formula (6.5). It is clear to see that by increasing the number of nodes the CTT increases quadratically, while increasing the number of sinks the CTT can be noticeably reduced. However, it should be noted that such a trend does not hold for every network configuration. In fact, by adding a sink to the network, the average length of data packet is reduced, but the overall number of hops is increased, because the new sink has to

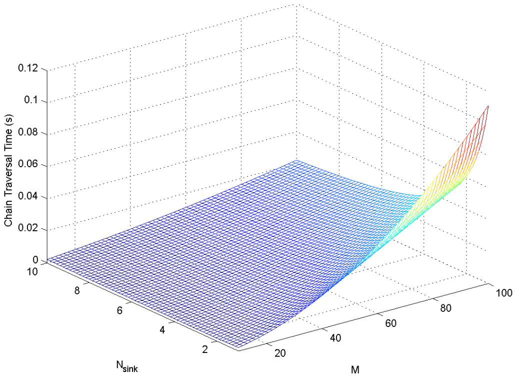


Figure 6.5: Chain Traversal Time.

forward a void data packet to grant medium access to the successor. As a result, for a given network setup there will always be a number of sinks over which the CTT will start increasing rather than decreasing. Analytically, this problem is shown by the fact that formula (6.5) diverges for indefinitely large values of N_{sink}. However, it is possible to calculate the optimal number of sinks N^*_{sink}as the number of sinks that minimizes the CTT for a given network configuration. Such a number can be found by analyzing the first derivative of formula (6.5) with respect to N_{sink}, and taking the *floor()* of the value that minimizes that function. In fact, differential calculus shows that formula (6.5) is monotonically decreasing for N_{sink} from 0 to N^*_{sink}. In particular, the optimal number of sinks obtained through this analysis is,

$$N^*_{sink} = \left\lfloor \frac{M \cdot \sqrt{\Delta T}}{\sqrt{2(T_{ack} + T_{ov} + T_{proc})}} \right\rfloor . \qquad (6.6)$$

6.5.3 Average Chain Trip Time

The Average Chain Trip Time (ACTT) is the average time spent by the network to complete a cycle across the whole network chain. This time is usually slightly larger than the CTT, due to possible packet loss and the relevant retransmissions. As the wireless medium is not deterministic, it is not possible to have an exact estimation of the duration of each cycle. However, under certain hypotheses on the packet loss probability, it is possible to compute an average value. In general, the ACTT is equal to the optimistic value of the CTT, plus the time lost for packet losses, i.e.,

$$ACTT = CTT + T_{recover}. \tag{6.7}$$

The term $T_{recover}$ is the time needed to recover the packets that were lost. In fact, every time a frame is lost, a procedure to recover the frame is needed. In the case of a data frame, shown in Figure 6.6a, the sender recognizes that its packet was lost after a timeout T_{to} and then it sends the packet again. As a result, the contribution of a data frame loss in the NTT is $T_{to} + T_{data}$. On the other hand, when an Ack frame is lost on node i-1, as in Figure 6.6b, node i sends its data frame to its successor, while the timeout expires on node i-1. After the end of the data packet, node i+1 sends an Ack frame, as acknowledgements are prioritized over data frames by using smaller interframe spaces. After such an Ack frame, node i-1 sends its data frame again, that will not, however, be forwarded again by node i, because it recognizes that it is a duplicate packet. Nevertheless, node i must send back a new Ack frame to $i-1$, to let $i-1$ know that the data packet has been received. To ensure that the retransmission from i-1 does not collide with the data packet from a successor, retransmissions should have a smaller interframe space than normal data transmissions (but higher than Ack frames). As a result, the contribution of an acknowledgement loss in the NTT is $T_{data} + T_{ack}$.

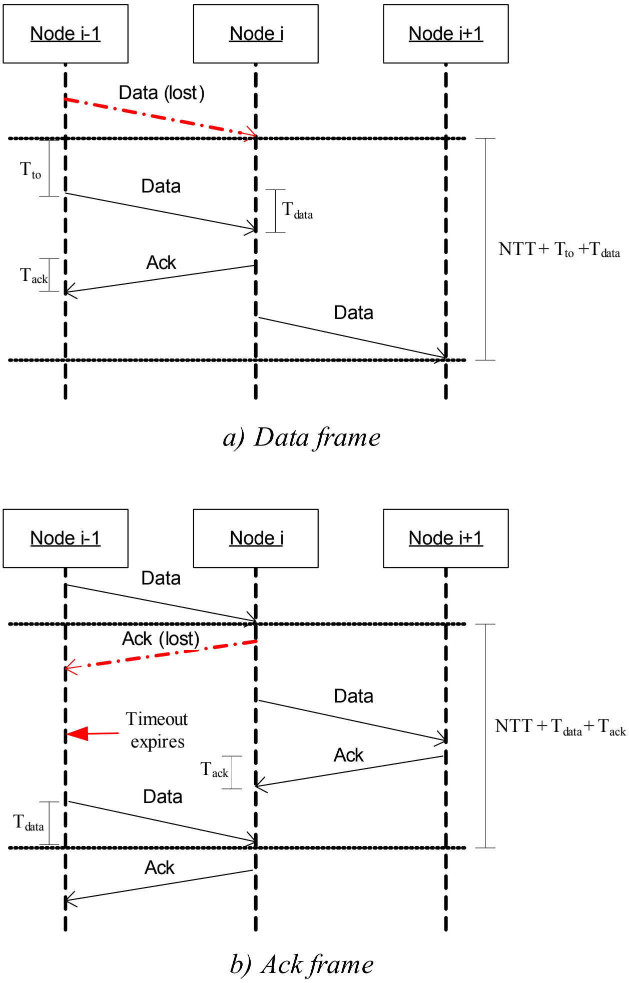
The time $T_{recover}$ can be subdivided in turn into three different contributions, i.e.,

$$T_{recover} = T_{recAck} + T_{recSink} + T_{recNode}, \tag{6.8}$$

where T_{recAck} is the time spent due to loss of acknowledgements, $T_{recSink}$ is the time spent due to loss of packets from the sink, and $T_{recSink}$ is the time spent due to loss of packets from nodes. Each contribution can be estimated independently to the others.

Suppose that E_{bit} is the bit error rate of the wireless technology used, in the particular environment where the network is deployed. In a case in which there is no error correction mechanism, a single bit error will cause a packet loss. This means that the packet error rate of a generic packet is, $E_{pkt} = E_{bit} \cdot l_{pkt}$,where l_{pkt} is the length of the packet expressed in bits.

As all the Ack frames have the same length, it is possible to calculate the error rate of acknowledgements as $E_{ack} = E_{bit} \cdot l_{ack}$.Moreover, as an Ack frame is sent on the reception of data packets from both nodes and



a) Data frame

b) Ack frame

Figure 6.6: Time to recover lost frame.

sinks, $(N_{sink} + M)$ Ack frames have to be sent to traverse the chain. At each acknowledgement loss, a time of $(T_{data} + T_{ack})$ is needed to recover from the error. However T_{data} is variable, as it depends on the position of the node that misses the Ack. Nevertheless, as the loss of Ack frames is independent to the data packets, it is reasonable that all the frames have the same probability to be retransmitted due to a missing Ack. As a result, it is possible to express T_{recAck} as

$$T_{recAck} = (N_{sink} + M)E_{ack}(\bar{T}_{data} + T_{ack}), \tag{6.9}$$

where $\bar{T}_{data}$is the average duration of data packets.

A similar reasoning can be used to calculate the second contribution of formula (6.8), i.e., that due to the loss of data packets from the sinks. In this case, the packet error rate is $E_{sink} = E_{bit} \cdot l_{ov}$,where l_{ov} is the size of a data packet without any payload (i.e., is the size of the void packet sent by the sink to grant the medium access to the successor sensor node), and the contribution given by these packets can be expressed as:

$$T_{recSink} = N_{sink} \cdot E_{sink}(T_{to} + T_{ov}). \tag{6.10}$$

Slightly more complex is the estimation of the last term of formula (6.8), as both the packet error rate and the recovery time depend on the size of data packets and, therefore, on the position of nodes in the chain. In particular, the packet error rate of a generic packet, whose source node i is located at the p_i-th position of its sub-chain, is given by$E^i_{node} = (l_{ov} + p_i \cdot l_{pl})E_{bit}$,where l_{pl} is the size of the payload added by each node, expressed in bits. The time to recover from a packet loss is given by $T_{to} + T^i_{data}$,where the last term is that in formula (6.2). As a result, it is possible to express $T_{recNode}$ as:

$$T_{recNode} = \sum_{k=1}^{N_{sink}} \sum_{i=1}^{L_k} (T_{to} + T_{ov} + i \cdot \Delta\mathrm{T})(\mathrm{l}_{ov} + i \cdot l_{pl})E_{bit}. \tag{6.11}$$

6.6 Performance evaluation

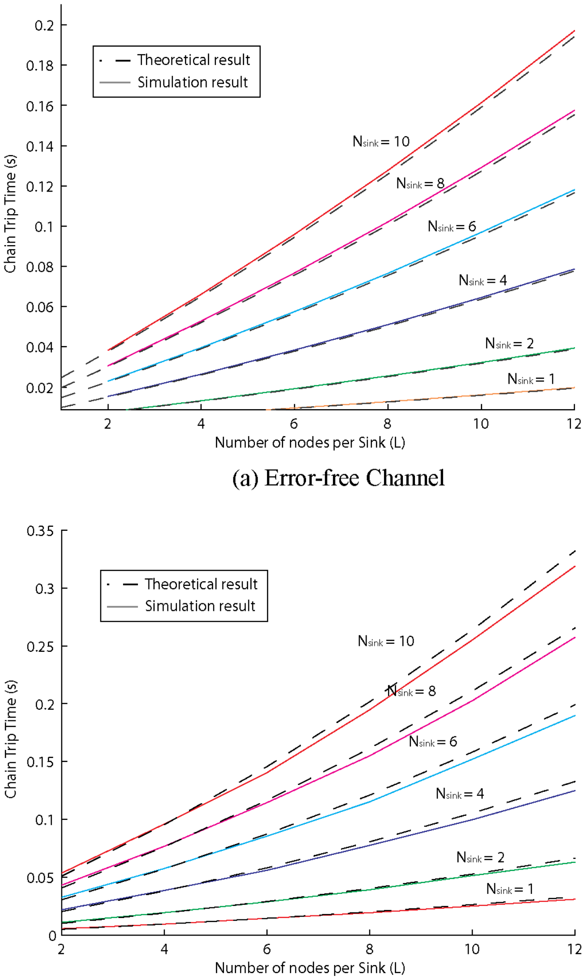
A simulation study was carried out to assess the effectiveness of the proposed protocol and to validate the analysis described in Section 6.5. To simulate the protocol we used the *ns-2* simulation tool [74], and we relied on the PHY and MAC models provided by *ns-2*, therefore implementing chain forwarding at the application level. However, to improve the efficiency of our implementation, we did not transmit Ack frames at the application layer too. Instead, we exploited MAC-level acknowledgements, which are smaller than data frames and also feature a smaller interframe space, which complies with our analysis in Section 6.5.3. Although our chain-based approach does not depend on any specific wireless technology, all the simulations shown in this chapter refer to the IEEE 802.11 protocol [88] at the Physical and MAC layers, used in ad-hoc mode. This protocol was preferred over the IEEE 802.15.4 [22], since the latter limits the maximum length of the MAC payload to 118 bytes, against the 2304 bytes supported by the IEEE 802.11. As in the chain-based approach the size of the data frame increases with the position of nodes in the sub-chain, IEEE 802.11 offers much higher scalability than IEEE 802.15.4, as it allows for longer sub-chains. However, as in industrial environment robustness and predictability are more important than throughput, we set the data rate to 1 Mbps, which provides the most resistant coding against noise and interference.

All the network configurations we simulated are generated using the same methodology to create comparable scenarios. In particular, at the beginning of the simulation we set up the number of sinks N_{sink} and the number of nodes per sink L. Each sink is placed at the center of a 15×15 m square region. These square regions, in turn, are placed side by side, so as to form a grid. Then, L nodes are put across the segments connecting two sinks, but with a random displacement from the ideal position. Thanks to this deployment mechanism, in all our simulations the chain creation algorithm produced a balanced chain, in which each sub-chain was made up exactly of L nodes. The choice of having balanced chains was made to make the comparison between different scenarios easier. In fact, if the deployment was completely random, the resulting topologies would be heterogeneous, and so difficult to compare (e.g., scenarios with a smaller number of nodes might still have longer chains).

6.6.1 Validation of theoretical results

We performed a set of simulations using the default settings of ns-2 for the channel model but varying the number of both sinks and nodes to assess the protocol performance in the case of no packet errors and to compare the CTT obtained analytically with that obtained through ns-2 simulation. In particular, we varied the number of nodes per cluster (L) from 2 to 12 and repeated the simulations for 1, 2, 4, 6, 8, and 10 sinks. In this way, the overall number of devices (nodes + sinks) ranged from a minimum of 3 to a maximum of 130. The data frame is made up of a 12-byte header, plus a 6-byte payload that represents the sensor reading from nodes. Data frames from the sinks contain only the 12-byte header. However, it should be noted that this format is relevant to the application, therefore the actual frame size is larger, because of the encapsulations at MAC and physical layer. In particular, a 58-byte overhead was observed in the ns-2 trace files for the data frames sent by the MAC layer, while the ACK size was 38 bytes according to the same files.

To compare the simulation results with those obtained though the theoretical analysis in Sect. 6, it is necessary to make some other considerations. In fact, both the T_{ov} and T_{ack} have to consider not only the time to transmit one frame over the air, but also other overheads due to the MAC and physical layers. Preamble is the same for all the frames and in particular ns-2 uses the long preamble of the IEEE 802.11 standard, which lasts for 192 μs. Then stations need 10.9 μs to synchronize the receivers before the actual frame transmission can start. These times should be added to both T_{ov} and T_{ack}. After each transmission stations have to wait for an interframe space before starting a new transmission. According to the IEEE 802.11 standard, a Short InterFrame Space (SIFS) of 10 μs is used for ACK frames, while a DCF Interframe Space (DIFS) of 50 μs is used for data frames. As a result, a DIFS and a SIFS have to be added to the T_{ov} and T_{ack}, respectively. Finally, half a minimum contention window (CWmin/2$\cdot$*SlotTime*) is added to the T_{ov} to account for backoff delays. In fact, as our transmission mechanism avoids collisions, all nodes maintain their contention window at the CWmin value. Moreover, as nodes delay their transmission for a random number of slots, uniformly distributed between 0 and CWmin, the average waiting time will be half a CWmin multiplied by the slot time. The processing time T_{proc} was extracted from the simulation results, by comparing the times in which the reception of data frames were completed by the PHY with the receiving times at the application layer. The theoretical CTTs obtained by formula (6.5) are plotted side by side with the ACTTs obtained from 100-second simulations, in Figure 6.7a. The figure shows that theoretical results (represented by dashed lines) closely match those obtained through the ns-2 simulator (represented by solid lines), in all the tested scenarios. This provides evidence for the effectiveness of the analysis in Section 6.5, in the case of good signal quality (i.e., with a negligible error rate). To assess the performance of the protocol in the case of a noisy environment,



(a) Error-free Channel

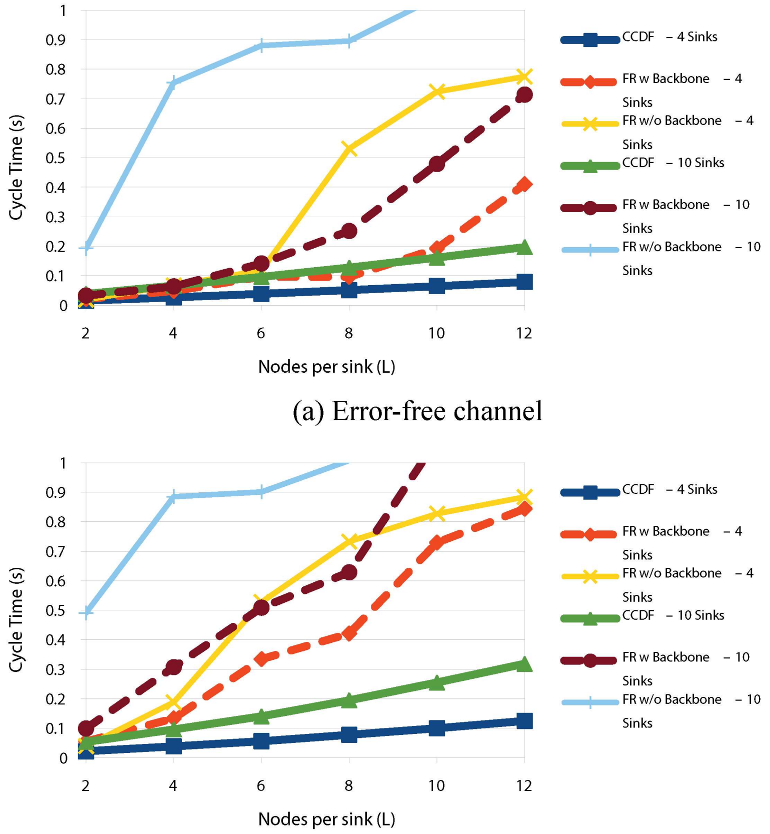
(b) Error-prone channel (BER = 3e-4)

Figure 6.7: Chain Trip Times.

we run the same simulation scenarios with an increased bit error rate of E_{bit}= 0.0003. Note that a similar value for the mean bit error rate of IEEE 802.11 was assessed in a harsh environment in [28]. However that result was obtained using the 2 Mbit data rate, therefore our simulation assumes even worse channel conditions, because we are setting the same bit error rate at 1 Mbit data rate. In Figure 6.7b, the average chain trip times obtained in such noisy conditions are compared with the theoretical results obtained by the analysis in Section 6.5 under the same conditions. Even in this case, the results obtained from the theoretical analysis closely match those obtained through ns-2 simulations. These results show that the model introduced in Section 6.5.3, to calculate the average time spent recovering lost frames, is able to produce accurate results even when the bit error rate is high.

6.6.2 Comparative assessments

To properly assess the effectiveness of CCDF, we compared the performance of the proposed protocol with that obtained under the same scenarios using the standard IEEE 802.11 MAC and a fixed routing protocol. Here we used the AODV protocol [89] at the beginning of the simulation to set up the routes, and maintained such routes for the whole simulation. For the sake of fairness, we discarded the results coming from the setup phase. To compare the protocols under the industrial perspective, we calculated the achievable cycle times, as defined in Section 6.5.2 under the different configurations. In the case of the CCDF protocol, the cycle time corresponds to the CTT, while to calculate the achievable cycle time in the case of CSMA/CA MAC with fixed routing, we released all data transmissions at the same time and took the time at which the last data frame was received. For each protocol, we repeated the measurement 100 times and plotted the average values. Figure 6.8 shows the results for three different configurations of nodes. In the first configuration, nodes run the CCDF algorithm on top of the IEEE 802.11 MAC, while in the second and third configurations nodes use fixed routing on top of the IEEE 802.11 MAC. The difference between these configurations is that the second uses the sinks as intermediate destinations to forward data through the backbone (and is labeled *FR w Backbone*) and the third (labeled *FR w/o Backbone*) does not. For every configuration we simulated the same scenarios addressed in Figure 6.7, but for the sake of clarity in Figure 6.8 we only show the scenarios featuring four and ten sinks. The results show that, although the fixed routing often produces shorter



(a) Error-free channel

(b) Error-prone channel (BER = 3e-4)

Figure 6.8: Comparative performance assessment.

node-to-sink paths, the CCDF protocol consistently outperforms the other configurations, achieving noticeably smaller cycle times in both the cases of error-free (Figure 6.8a) and error-prone (Figure 6.8b) channel. For example, in the case of error-free channel the CCDF can support 100 sensors with cycle times of about 160 ms or 32 sensors with cycle times down to 50 ms. Using the standard CSMA with fixed routing and no wired forwarding, these cycle times would be 1100 ms and 730 ms, respectively. As expected, the use of sinks as intermediate destinations to forward data through the wired backbone is also beneficial in the case of fixed routing, but even in this case the performance is far from that obtained by the CCDF protocol, e.g., in the two aforementioned scenarios the achievable cycle times are 1000 ms and 420 ms, respectively. The plots in Figure 6.8b show larger cycle times but analogous trends in the case of a noisy channel. Such results show that, although chain-based forwarding introduces a nearly linear delay at each hop and so one might think that this approach suffers from low scalability, actually the advantage over the standard IEEE 802.11 MAC with fixed routing increases with the increasing number of nodes and sinks. Moreover, in all our simulations we found that the standard deviations of the CCDF cycle times were one or two orders of magnitude smaller than those obtained by the other configurations. The reason for these results is that the standard CSMA/CA is not as effective as the chain-based communication protocol in collision avoidance.

6.7 Concluding remarks

This chapter proposed the Circular Chain Data Forwarding (CCDF) mechanism in the context of industrial WSNs. The chapter discusses the mechanisms used to build the chain and to achieve fault tolerance. Moreover an in-depth analysis of the CCDF performance has been provided for the case of error-free channels and then extended to the case of error-prone channels. A simulative assessment has been presented to validate the analytical results and to compare the performance of the proposed approach with that of the standard IEEE 802.11 MAC with a fixed routing protocol.

Future work will extend the theoretical analysis in the direction of providing statistical guarantees that consider not only the average values, but also the probability distribution of the performance metrics. Moreover, measurement campaigns on a test-bed will be run to assess the effectiveness of theoretical results when dealing with real deployments.

Chapter 7

Conclusions

Industrial Wireless Sensor Networks have peculiarities that distinguish them from typical WSNs. Although some requirements such as scalability and energy efficiency are in common with classical WSNs, in industrial deployments real-time performance is by far more critical than energy efficiency. Moreover, industrial WSNs have to be robust against interference and are usually integrated with wired industrial networks, because there are critical data flows that cannot be transmitted over the wireless medium. This thesis investigated novel techniques and communication protocols aimed at delivering real-time performance to power- and energy-constrained sensor nodes, even in large and dense deployments where nodes could not be directly covered by a sink.

In particular, Chapter 2 addressed the problem of robustness of IEEE 802.15.4 networks to cross-channel interference by providing a general methodology and a generic testbed devised for experimental on-site assessments of the impact of the interference in industrial networks. Moreover, a case study was presented, which explains how to set the testbed in order to assess the impact on cross-channel interference of one or multiple interferers and the effect of some MAC level parameters under cross-channel interference. Chapter 3 addressed the problem of scalability at the MAC layer by introducing a novel technique for collision-free superframe scheduling in cluster-tree IEEE 802.15.4 networks, which exploits multiple radio channels to enable scheduling sets of superframes that could not be feasible using a single radio channel. The chapter also addressed how to implement multichannel superframe scheduling through only minor changes to the MAC layer and small add-ons to the upper layers. The feasibility of this approach is demonstrated by a working implementation based on the open source TinyOS.

The problem of highly increasing energy efficiency while introducing only a predictable delay was addressed by means of two topology management protocols which run between the MAC and the Routing layer. In particular, Chapter 4 presented a static topology management mechanism with bounded delay, which works together with a real-time routing protocol to meet soft real-time constraints while achieving high energy efficiency. In this protocol, the combination of clustering and time driven communication not only allows nodes to shut down their radio when no transmissions or receptions are needed, thus significantly decreasing their average energy consumption, but also imposes a bound on the delay of intra-AU communications. The good behaviour of the topology management protocol in terms of energy consumption and real-time performance has been confirmed by simulations. A dynamic extension of this protocol was presented in Chapter 5. The dynamic approach introduces the support for both time-driven and event-driven communication and enables the use of dynamic clustering techniques, which are more effective when the density of nodes is non-uniform. Moreover, it also introduces a novel energy balancing feature which is able to increase the overall network lifetime thanks to a node exchange policy. The effectiveness of the protocol and the improvement in both network lifetime and real-time performance have been shown by a comparative assessment based on ns-2 simulations.

Finally, the problem of predictable end-to-end data delivery was addressed in Chapter 6 by providing a chain-based communication protocol, which not only supports integration with a wired industrial infrastructure, but also takes advantage of it to deliver real-time performance. The chapter provided an in-depth analysis of the protocol, at first for the case of error-free channels and then extended to the case of error-prone channels. A simulative assessment was also presented to validate the analytical results and to compare the performance of the proposed approach with that of the standard IEEE 802.11 MAC with a fixed routing protocol.

Bibliography

[1] K. Low, W. Win, and M. Er, “Wireless Sensor Networks for Industrial Environments”, in *International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce.*, vol. 2, 2005, pp. 271–276.

[2] J. Frey and T. Lennvall, “Wireless Sensor Networks for Automation,” in R. Zurawski Ed., *Embedded Systems Handbook, Second edition*, CRC Press, Taylor & Francis Group, 2009, pp. 27-1, 27-43.

[3] A. Manjeshwar, D. Agrawal, “TEEN: a Routing Protocol for Enhanced Efficient in Wireless Sensor Networks”, in Proc. of the 15th International Parallel and Distributed Processing Symposium, pp. 2009-2015, 2001.

[4] A. Manjeshwar, D. Agrawal, “APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks”, in Proc. of the 2^{nd} International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, Ft. Lauderdale, FL,April 2002.

[5] J.H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks”. In Proceedings of Advanced Telecommunications and Information Distribution Research Program, College Park, MD, 2000.

[6] R. C. Shah and J. M. Rabaey, “Energy aware routing for low energy ad hoc sensor networks”. In Proc. IEEE Wireless Communications and Networking Conference (WCNC),vol. 1, pp. 350- 355, Orlando, FL, March 2002.

[7] J. Kulik, W. Rabiner, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", Proc. of the 5th annualACM/IEEE int. conf. on Mobile computing and networking, pp. 174-185,1999.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", in Proc. of the 6th annual ACM/IEEE intl conf. on Mobile computing and networking, pp. 56-67, 2000.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", in Proc. of the 33rd Hawaii International Conference on Systems Science, Volume 8, pp. 3005-3014,2000.

[10] Yan Yu, Ramesh Govindan and Deborah Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks", UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.

[11] Brad Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks", In Proc. of ACM/IEEE MobiCom, pp. 243-254, Aug. 2000.

[12] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," in Proc. of the IEEE Int'l Conf. Distributed Computing Systems, pp. 46-55, 2003.

[13] T. He, J.A. Stankovic, T.F. Abdelzaher and C. Lu, "A spatiotemporal communication protocol for wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, vol.16, no.10,pp. 995-1006, Oct. 2005.

[14] O. Chipara, Z. He, Q. Chen, G. Xing, X. Wang, C. Lu, J. Stankovic and T. Abdelzaher, "Real-time Power-Aware Routing in Sensor Networks," Proc. of Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on, pp.83-92, June 2006.

[15] O. Kasten, "Energy consumption", 2001. [Online] Available at http://www.inf.ethz.ch/~kasten/research/bathtub/energy_consumption.html.

[16] M. Stemm and R. Katz “Measuring and reducing energy consumption of network interfaces in hand-held devices”, Institute of Electronics, Information, and Communication Engineers (IEICE) Transactions on Communications, vol.E80B (8), pp. 1125-1131, Aug. 1997.

[17] Ya Xu, John Heidemann, and Deborah Estrin, “Geography-informed energy conservation for ad hoc routing”. In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking,pp. 70-84, Rome, Italy, July 2001.

[18] Ya Xu, John Heidemann, and Deborah Estrin, “Adaptive energy-conserving routing for multihop adhoc networks” Tech. Rep. 527, USC/ISI, Oct. 2000.

[19] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. “Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks”.ACM Wireless Networks Journal, 8(5), September 2002.

[20] C. Schurgers, V. Tsiatsis, M.B. Srivastava, “STEM: Topology management for energy efficient sensor networks,” Aerospace Conference Proceedings. IEEE , vol.3, pp. 1099-1108, 2002.

[21] “IEEE 802.15.4 Standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard for Information Technology, IEEE-SA Standards Board”, 2003.

[22] “IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs),” *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, 2006.

[23] A. Willig, K. Matheus, e A. Wolisz, “Wireless Technology in Industrial Networks,” Proceedings of the IEEE, vol. 93, 2005, pp. 1130-1151.

[24] A. Willig, “Recent and Emerging Topics in Wireless Industrial Communications: A Selection,” *IEEE Transactions on Industrial Informatics*, vol.4, no.2, pp.102-124, May 2008.

[25] Korber, H.-J.; Wattar, H.; Scholl, G., "Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications," *IEEE Trans.s on Industrial Informatics*, vol.3, no.2, pp.111-119, May 2007.

[26] E. Toscano and L. Lo Bello, "Cross-channel interference in IEEE 802.15.4 networks", *Proc. of the IEEE Intl Workshop on Factory Communication Systems, WFCS 2008*, pp. 139–148, May 2008.

[27] "IEEE Recommended Practice for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands", *IEEE Std 802.15.2-2003*, 2003.

[28] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Transactions on Industrial Electronics*, vol. 49, 2002, pp. 1265-1282.

[29] U. Bilstrup and P. Wiberg, "Bluetooth in industrial environment", *Proc. IEEE Intl Workshop on Factory Communication Systems*, 2000.

[30] N. Amanquah and J. Dunlop, "Interference due to link management signalling in co-ordinated co-located Bluetooth networks", *Proc. of the IEEE 58th Vehicular Technology Conf.*, Oct. 2003.

[31] A. El-Hoiydi, J.D. Decotignie, "Soft deadline bounds for two-way transactions in Bluetooth piconets under co-channel interference", *Proc. IEEE Intl Conf. on Emerging Tech. and Factory Autom., ETFA'01*, 2001.

[32] F. Simonot-Lion, Y.Q. Song, "Safety Evaluation of Critical Applications Distributed on TDMA-Based Networks", *3rd Taiwanese-French Conf. on Inf. Technology*, 2006.

[33] I. Howitt and J. Gutierrez, "IEEE 802.15.4 low rate - wireless personal area network coexistence issues", *Wireless Comm. and Networking. WCNC 2003*, Mar. 2003.

[34] S. Y. Shin, H. S. Park, S. Choi, and W. H. Kwon, "Packet Error Rate Analysis of IEEE 802.15.4 under IEEE 802.11b Interference", *Proc. Wired / Wireless Internet Commun.*, pp. 279–288, May 2005.

[35] S. Y. Shin, H. S. Park, S. Choi and W. H. Kwon, "Packet Error Rate Analysis of ZigBee Under WLAN and Bluetooth Interferences", *IEEE Trans. on Wireless Comm.*, vol. 6, no. 8, pp. 2825–2830, Aug. 2007.

[36] A. Sikora and V. Groza, "Coexistence of IEEE 802.15.4 with other Systems in the 2.4 GHz-ISM-Band", *Proc. IEEE Conf. on Instrumentation and Measurement Technology, IMTC 2005*, vol. 3, May 2005.

[37] M. Bertocco, G. Gamba, and A. Sona, "Is CSMA/CA really efficient against interference in a wireless control system? An experimental answer", *Proc. of the IEEE Intl Conf. on Emerging Tech. and Factory Autom., ETFA 2008*, pp. 885–892, Sept. 2008.

[38] M. Bertocco, G. Gamba, A. Sona, and F. Tramarin, "Investigating wireless networks coexistence issues through an interference aware simulator", *Proc. of the IEEE Intl Conf. on Emerging Tech. and Factory Autom., ETFA 2008*, pp. 1153–1156, Sept. 2008.

[39] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy, "Experimenting with a Multi-Radio Mesh Networking Testbed", *1st Workshop on Wireless Networks Measurements*, 2005.

[40] C. M. Cheng, P. H. Hsiao, H. Kung, and D. Vlah, "Adjacent Channel Interference in Dual-radio 802.11a Nodes and Its Impact on Multi-hop Networking", *Proc. of the IEEE Global TeleComm Conf., GLOBECOM '06*, Nov. 2006.

[41] G. Cena, I. Cibrario Bertolotti, A. Valenzano and C. Zunino, "Reasoning about communication latencies in real WLANs", *Proc. of the 12th IEEE Conf. on Emerging Tech. and Factory Autom., ETFA'07*, Sep. 2007.

[42] O. Incel, S. Dulman, P. Jansen, and S. Mullender, "Multi-Channel Interference Measurements for Wireless Sensor Networks", *Proc. of the 31st IEEE Conf. on Local Computer Networks*, pp. 694–701, Nov. 2006.

[43] G. E. Jonsrud. (2006) "CC2420 Coexistence". Texas Instruments. [Online]. Available: http://www-s.ti.com/sc/techlit/swra094

[44] R. Rodriguez. (2005, July) “MC1319x Coexistence”. Freescale Semiconductor. [Online]. Available: www.freescale.com/files/rf_if/doc/app_note/ROD05.pdf

[45] F. De Pellegrini, D. Miorandi, S. Vitturi, A. Zanella, “On the use of wireless networks at low level of factory automation systems,” *IEEE Trans. on Industrial Informatics*, vol.2, no.2, pp. 129-143, May 2006.

[46] L. Tang, K.-C. Wang, Y. Huang, and F. Gu, “Channel Characterization and Link Quality Assessment of IEEE 802.15.4-Compliant Radio for Factory Environments”, *IEEE Trans. on Ind. Informat.*, vol. 3, no. 2, pp. 99–110, May 2007.

[47] J. R. Taylor, *“An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements”*, 2nd ed. University Science Books, 1997.

[48] Maxstream XBee datasheet. [Online]. http://www.maxstream.net.

[49] (2005, September) “XBee and XBee-PRO OEM RF Module Antenna Considerations”. MaxStream Inc. [Online]. Available: http://ftp1.digi.com/support/images/XST-Max05_XBeeAntennas.pdf

[50] D. G. Altman, D. Machin, T. N. Bryant and M. J. Gardner, *“Statistics with Confidence: Confidence Intervals and Statistical Guidelines”*, 2nd ed. BMJ Books, 2000.

[51] A. Koubâa, M. Alves, and E. Tovar, "Energy/Delay trade-off of the a GTS allocation mechanism in IEEE 802.15. 4 for wireless sensor networks." HURRAY-TR-060103, 2006.

[52] A. Koubâa, M. Alves, and E. Tovar, "Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks," Proc. of the 27th IEEE Real-Time Systems Symposium (RTSS’06), Brazil, 2006.

[53] J. Ha, W.H. Kwon, J..J. Kim, Y.H. Kim and Y.H. Shin, “Feasibility analysis and implementation of the IEEE 802.15.4 multi-hop beacon-enabled network.” In Proc. of the 15th Joint Conf. on Communic. & Info, Jun. 2005.

[54] A. Koubâa, A. Cunha, M. Alves, e E. Tovar, “TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks,” Real-Time Systems, vol. 40, Dic. 2008, pp. 321-354.

[55] E. Toscano, L. Lo Bello, “A topology management protocol with bounded delay for Wireless Sensor Networks,” Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008, pp.942-951, 15-18 Sept. 2008.

[56] http://grouper.ieee.org/groups/802/15/pub/TG4b.html.

[57] Zou You-Min, Sun Mao-Heng, e Ran Peng, “An Enhanced Scheme for the IEEE 802.15.4 Multi-Hop Network,” International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2006, pp. 1-4.

[58] P. Muthukumaran, R. Spinar, K. Murray, et al. “Enabling low power multi-hop personal area sensor networks.” In: Proc. of the 10th Intl Symposium on Wireless Personal Multimedia Communications. Jaipur, India, 2007.

[59] “ZigBee Alliance, ZigBee Specification”, Document 053474r17, January 2008.

[60] R. Diestel, “Graph theory” 3rd Ed. Springer, NY, 2005.

[61] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, e D. Culler, “Ti-nyOS: An Operating System for Sensor Networks,” Ambient Intelligence, 2005, pp. 115-148.

[62] http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/ TelosB_Datasheet.pdf.

[63] J. Hauer, TKN15.4: An IEEE 802.15.4 MAC Implementation for TinyOS 2, Telecommunication Networks Group, Technical University Berlin, 2009.

[64] E. Toscano, O. Mirabella, L. Lo Bello, “An Energy-efficient Real-Time Communication Framework for Wireless Sensor Networks”, Intl. Workshop on Real-Time Networks (RTN07), Pisa, Italy, July 2007.

[65] L. Lo Bello, E. Toscano, “Power-Efficient Routing in Wireless Sensor Networks”, in: Dr. Richard Zurawski (Ed.), Networked Embedded Systems Handbook, Chapter 7, pp. 1-40, CRC Press/Taylor & Francis, Boca Raton (US), 2009.

[66] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, “Protocols for Self-Organization of a Wireless Sensor Network”, IEEE Personal Communications, Vol. 7, No.5, pp. 16-27, 2000.

[67] K. Akkaya and M. Younis, “An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks”,in Proc. IEEE of the 23rd International Conference on Distributed Computing Systems, pp. 710–715, 2003.

[68] J. Trdlicka, Z. Hanzalek, M. Johansson, “Optimal flow routing in multi-hop sensor networks with real-time constraints through linear programming.,” Proc. of IEEE *Conf. on Emerging Technologies and Factory Automation ETFA 2007*, pp.924-931, 25-28 Sept. 2007.

[69] W. Lai, I. C. Paschalidis, “Sensor network minimal energy routing with latency guarantees”. In *Proc. of the 8th ACM international Symposium on Mobile Ad Hoc Networking and Computing* (Montreal, Canada, Sept. 2007). MobiHoc ’07. ACM, New York, NY, 199-208.

[70] L. Lo Bello, M. Collotta, E. Toscano. “Energy-Efficient MAC Protocols for Wireless Sensor Networks”, in: Dr. Richard Zurawski (Ed.), Networked Embedded Systems Handbook, Chapter 8, pp.1-24, CRC Press/Taylor& Francis, Boca Raton (US), 2009.

[71] C. Shanti and A. Sahoo, “DGRAM: A Delay Guaranteed Routing and MAC protocol for wireless sensor networks,” Proc. of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2008, Newport Beach, CA, USA, pp. 1-9.

[72] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, A. Sangiovanni-Vincentelli, “System Level Design for Clustered Wireless Sensor Networks,” *IEEE Transactions on Industrial Informatics*, vol.3, no.3, pp.202-214, Aug. 2007.

[73] A.A. Abidi, G.J. Pottie and W.J. Kaiser, “Power-conscious design of wireless circuits and systems”, Proceedings of the IEEE, vol.88, no.10,pp.1528-1545, Oct 2000.

[74] http://www.isi.edu/nsnam/ns/.

[75] S. Basagni, “Distributed Clustering Algorithm for Ad-hoc Networks”, Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks(ISPAN), 1999.

[76] M. Chatterjee, S. K. Das and D. Turgut, “WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks”, J. of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, pp.193-204, 2002.

[77] S. Banerjee, S. Khuller, “A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks”, in Proc. of IEEE INFOCOM’01.

[78] H. Alipour, M. Abbaspour, M. Esmaeili, H. Mousavi, H. Shahhoseini, “DACA: Dynamic Advanced Clustering Algorithm for Sensor Networks,” Proc. of 14th IEEE Intl Conf. on Electronics, Circuits and Systems, ICECS 2007, pp.518-525, Dec. 2007.

[79] O. Younis and S. Fahmy, “Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach”, in Proc. of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM, vol. 1, pp. 629-640, Mar. 2004.

[80] G. Merrett, N. Harris, B. Al-Hashimi, and N. White, “Energy Controlled Reporting for Industrial Monitoring Wireless Sensor Networks,” Proc. of *5th IEEE Conf. On Sensors,* 2006, pp. 892-895.

[81] J. Heo, J. Hong, and Y. Cho, “EARQ: Energy Aware Routing for Real-Time and Reliable Communication in Wireless Industrial Sensor Networks,” *IEEE Trans. Ind. Inf.*, vol. 5, 2009, pp. 3-11.

[82] G. Anastasi, M. Conti, M. Di Francesco, “Extending the Lifetime of Wireless Sensor Networks Through Adaptive Sleep,” *IEEE Trans. Ind. Inf,* 5, 2009, 351-365.

[83] V. Gungor and G. Hancke, “Industrial Wireless Sensor Networks: Challenges, Design Principles and Technical Approaches,” *IEEE Trans. Ind. Electron.,* vol. 56, 2009, pp. 4258-4265.

[84] S. Lindsey and C.S. Raghavendra, "Energy Efficient Broadcasting for Situation Awareness in Ad hoc Networks," *In Proc. Int. Conf. Parallel Proc. (ICPP'01)*, 2001, pp. 149-155.

[85] S. Lindsey, C. Raghavendra and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric", Proceedings 15th International Parallel and Distributed Processing Symposium, pp.2001-2008, Apr 2001.

[86] Kemei Du, J.Wu, D. Zhou, "Chain-based protocols for data broadcasting and gathering in the sensor networks," *Proc. of Intl. Symp on Par.and Distrib Processing,2003*.

[87] B. Bui, R. Pellizzoni, M. Caccamo, C. Cheah, and A. Tzakis, "Soft Real-Time Chains for Multi-Hop Wireless Ad-Hoc Networks," *Proc. IEEE Real Time and Embedded Tech. and Appl. Symposium*, 2007, pp. 69-80.

[88] "IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, 2007.

[89] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing", Proceedings of 2nd IEEE workshop on Mobile Computing Systems and Applications, pp.90-100, Feb. 1999.