UNIVERSITÀ degli STUDI di CATANIA

Department of Electrical, Electronics and Computer Engineering

PhD in Systems, Energy, Computer and Telecommunication Engineering

XXXII Cycle

Low-power Wireless Technologies for Real-Time Industrial Applications: Challenges, Novel Solutions and Future Directions

Ing. Luca Leonardi

Coordinator	Tutor
Prof. Paolo Arena	Prof. Lucia Lo Bello

to my love, Martina

Abstract

The use of wireless technologies in industrial environments has been significantly increasing in the last years, thus paving the way to novel applications, such as Industrial Internet of Things (IIoT), that plays a significant role in connecting machines, products, and humans in the ongoing industry digitalization process that is widely-known as Industry 4.0. Wireless communication systems are attractive for several applications in the context of Industry 4.0 as they offer low cost, easy deployment and mobility support. However, wireless communication systems pose several challenges in their implementation in industrial environments. In fact, despite a significant research effort in the area of wireless networks, there are several issues that have not been satisfactorily addressed yet. For example, typical industrial applications have stringent requirements, especially in terms of reliability, timeliness, and energy consumption. The wide range of IIoT applications exploits different wireless technologies that are targeted to fulfill the requirements of specific scenarios. A common aspect of these applications is the presence of energy-limited devices (e.g., mobile devices powered by batteries), therefore, the development of low-power wireless technologies is becoming increasingly important. The main issue is that some of these technologies are not immediately applicable to industrial use cases. For example, they do not provide support for real-time communications, that is instead a fundamental requirement of the typical industrial applications.

In this thesis work, innovative solutions for low-power wireless technologies are presented and described. Specifically, two specific low power wireless technologies are analyzed, i.e., Bluetooth Low Energy and LoRa. These protocols target short and long-range communication, respectively, thus the thesis provides an insightful exploration of a broad set of use cases.

The aim of the thesis is to determine the suitability of these technologies for IIoT applications and to identify the key elements that can be optimized/modified to improve their performance. In particular, novel mechanisms, algorithms and protocols built upon standard communication technologies are investigated, with the aim of meeting the typical requirements of applications in the Industry 4.0 scenario.

The thesis includes a broad range of assessments, such as evaluations obtained through simulations, analysis and also through experiments on proof-of-concept implementations, which prove the effectiveness and the suitability of the proposed solutions.

Acknowledgements

I would like to express my gratitude to my advisor Prof. Lucia Lo Bello for the continuous support in my PhD study and related research, for her patience, motivation, and immense knowledge. Her guidance helped me during all the time dedicated to the research and writing of this thesis.

My sincere gratitude also goes to my love Martina and to my family for supporting me throughout the PhD Course and also in every circumstance of life. Without they precious support it would not be possible to undertake a similar life journey.

Contents

List of Acronyms

ADR	Adaptive Data Rate
AFA	Adaptive Frequency Agility
ALOHA	Additive Link On-Line Hawaii
AP	Anchor Point
API	Application Program Interface
ATT	Attribute Protocol
ATT_MTU	Attribute Protocol Maximum Transfer Unit
BLE	Bluetooth Low Energy
BW	Bandwidth
CAP	Contention Access Period
CAPEX	CAPital EXpenditures
CCA	Clear Channel Assessment
CCCD	Client Characteristic Configuration Descriptor
CFP	Contention-Free Period
CE	Connection Event
CI	Connection Interval
COTS	Commercial-Off-The-Shelves
CSL	Connection Slave Latency
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance

CR	Coding Rate
CT	Cycle Time
CSS	Chirp Spread Spectrum
DC	Duty Cycle
DSME	Deterministic and Synchronous Multichannel Extension
E2ED	End-to-End Delay
FEC	Forward Error Correction
FIFO	First-In First-Out
GATT	Generic Attribute Profile
GB	Guard Band
GTS	Guaranteed Time Slots
HCI	Host Controller Interface
IFS	Inter Frame Space
IIoT	Industrial Internet of Things
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
IWN	Industrial Wireless Networks
IWSAN	Industrial Wireless Sensor and Actuator Networks
IWSN	Industrial Wireless Sensor Networks
LBT	Listen Before Talk
LHS	Left-Hand-Side
LL	Link Layer
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPLAN	Low Power Local Area Networks
LPWA	Low Power Wide Area
LPWAN	Low Power Wide Area Network

LR-WPAN	Low Rate Wireless Personal Area Network
LVAP	Light Virtual Access Point
MAC	Medium Access Control
MIC	Message Integrity Code
MN	Mobile Node
MRT-BLE	Multi-hop Real-time Bluetooth Low Energy
NFV	Network Functions Virtualization
OPEX	OPerating EXpense
PCA	Priority Channel Access
PDU	Protocol Data Unit
PER	Packet Error Rate
PLR	Packet Loss Ratio
QoS	Quality of Service
RAN	Radio Access Network
RHS	Right-Hand-Side
SDN	Software-Defined Networking
SF	Spreading Factor
SHM	Structural Health Monitoring
SIR	Signal-to Interference Ratio
SN	Stationary Node
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
TDMA	Time Division Multiple Access
ToA	Time on Air
TSCH	Time Slotted Channel Hopping
WCRT	Worst Case Response Time
WLAN	Wireless Local Area Network

WSN	Wireless Sensor Network
WTP	Wireless Termination Point

List of Publications

Included publications

- **Publication A.** L. Leonardi, G. Patti, L. Lo Bello, "Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks", *IEEE Access*, vol. 6, pp. 26505-26519, 2018.

- **Publication B.** L. Leonardi, F. Battaglia, G. Patti, L. Lo Bello, "Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications", *in the Proceedings of the 44th Annual Conference of IEEE Industrial Electronics Society (IECON)*, pp. 4141-4146, Washington, DC, Oct. 2018.

- **Publication C.** L. Leonardi, F. Battaglia, L. Lo Bello, "RT-LoRa: A Medium Access Strategy to support Real-time flows over LoRa-based networks for Industrial IoT applications", *IEEE Internet of Things Journal*, 2019.

- **Publication D.** L. Leonardi, M. Ashjaei, H. Fotouhi, L. Lo Bello, "A Proposal Towards Software-Defined Management of Heterogeneous Virtualized Industrial Networks", *in the Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN)*, Helsinki-Espoo, Finland, July 2019.

Publications not included

- F. Battaglia, M. Collotta, L. Leonardi, L. Lo Bello, G. Patti, “A scalable approach for periodic traffic scheduling in IEEE 802.15.4-DSME networks”, *in the Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN)*, Helsinki-Espoo, Finland, July 2019.

- S.C. Sciberras, R. Sinatra, P. Attard Cortis, L. Lo Bello, L. Leonardi, A. Cammarata, G. Laferla, G. Patti, “MEDIWARN: Implementation using virtual biosensors for medical alerts”, *10th Malta School Conference*, vol. 30 of the Malta Medical Journal, p. 252, Malta, Dec. 2018.

- L. Leonardi, G. Patti, F. Battaglia, L. Lo Bello, “Simulative assessments of the IEEE 802.15.4 CSMA/CA with Priority Channel Access in structural health monitoring scenarios”, *in the Proceedings of the IEEE 15th International Conference on Industrial Informatics (INDIN)*, Emden, Germany, July 2017.

- G. Patti, L. Leonardi, L. Lo Bello, “A Bluetooth Low Energy real-time protocol for industrial wireless mesh networks”, *in the Proceedings of the 42nd Annual Conference of IEEE Industrial Electronics Society (IECON)*, Florence, Italy, Oct. 2016.

Chapter 1

Introduction

During the last years, wireless technology has been increasingly utilized in the context of industrial communications [1], leading to the Industrial Internet of Things (IIoT), where Industrial Wireless Sensor Networks (IWSNs) and Industrial Wireless Sensor and Actuator Networks (IWSANs) play a central role [2, 3].

Traditionally, wireless networks consist of a number of spatially distributed nodes deployed to exchange information. Although wireless networks were initially designed for military and environmental monitoring applications [4, 5], recently, they have become popular for emerging applications in industrial domains [6]. For instance, an Industrial Wireless Network (IWN) can monitor critical parameters and control industrial processes and then provide this information in real-time to the control room [2].

Wireless technologies are able to provide great advantages over their wired counterparts, thanks to ease of deployment with lower cabling cost, higher flexibility and scalability, enhanced support for mobile devices, and reduced maintenance costs [7–9]. As a result, although the traditional industrial sensing and control systems are based on wired communication, a rapid development and standardization of diverse wireless technologies for different industrial applications is in progress [10].

The main benefits of industrial wireless networks are listed below:

- **Cost:** The main rationale behind the adoption of wireless networks in industrial environments is that they are cheaper and easier to install with respect to traditional wired communication systems. They also have a lower maintenance cost. As a result, wireless networks significantly help to reduce CAPital EXpenditures (CAPEX) and OPerating EXpense (OPEX)[1].

- **Flexibility and adaptivity:** Wireless networks allow to gather information from places either unreachable or too costly to reach through wired networks. In addition, wired networks are rigid and fixed in nature, and this hinders their ability to adapt to changes in the industrial environment. Conversely, wireless networks easily adapt to changes.

- **Mobility support:** Several industrial applications require mobile nodes (e.g., robots [13, 14], Automated Guided Vehicles [15], etc.). Wireless technologies allow to connect mobile devices, such as nodes mounted on mobile equipment, that could not be connected through wired networks.

Wireless technologies offer various advantages, but, at the same time, they pose several challenges that require significant efforts to be dealt with [16], such as:

- **Packet loss:** The intrinsic uncertainty of the wireless medium results in a non-negligible Packet Error Rate (PER) and introduces unpredictable packet delivery delays, thus affecting both timeliness and reliability [17, 18]. Wireless networks exhibit a

[1] The deployment cost consists of both CAPEX and OPEX [11]. CAPEX refers to installation costs, i.e., the total amounts required to roll out the network infrastructure. Conversely, OPEX refers to operation and maintenance cost, such as the electric bill, therefore it is highly influenced by the power consumption of the network [12].

higher packet loss than wired ones, as they are subject to multipath propagation and fading, and varying channel conditions. In addition, most of the low-power wireless standards operate in the crowded Industrial, Scientific and Medical (ISM) band, thus they are highly vulnerable to interference.

- **Variable communication delay:** The uncertain conditions of the wireless medium may determine random delays in packet delivery. Consequently, the end-to-end delay, i.e., the time needed for the data from the source to reach the destination is not bounded. The quality of the links between nodes also affects delay, as with poor link quality, retransmissions are more frequent, and both the delay and the energy consumption grow.

- **Data rate:** The number of devices that can share the wireless medium is limited by the channel bandwidth. In addition, since some wireless technologies only support low data rates, this may limit their use in specific industrial applications, such as high-speed control applications.

IIoT applications [19] feature diverse and stringent requirements [18, 20, 21], therefore, in order to meet such requirements, wireless industrial networks must fulfill the following objectives:

- **Reliability:** Industrial networks have to provide a low error probability. Techniques such as retransmissions (with or without acknowledgement) and relaying (i.e., the use of multiple paths for the transmissions from the source to the destination) can be adopted to cope with this requirement.

- **Fault tolerance:** Industrial network needs to be fault tolerant, so that a failure of one or a group of nodes shall not cause the failure of the entire network, that would entail high costs for the industry. For instance, low-power wireless networks have limited battery resources. Every time a device run out of battery, it can cause a network failure. For this reason, communication networks have to provide fault-tolerance through mechanisms such as node redundancy and replication of functions.

- **Scalability:** Some kinds of industrial networks are dense (for example, IWSN and networks for process control applications). Consequently, an efficient and fair distribution of the limited resources available in such networks is crucial to achieve a reasonable trade-off between different performance metrics. Scalability refers to the ability of a system to scale well, in terms of number of nodes and/or functionalities, without degrading the performance.

- **Interoperability:** Currently, wired networks are prevalent in industries. The adoption of novel wireless protocols has to be transparent to the existing solutions, i.e., the new protocols have to seamlessly integrate with the legacy systems already in use.

- **Mobility:** The introduction of mobile nodes in the industrial scenario entails additional requirements on the networks. For this reason, several mechanisms that modify the standard specification were proposed [22, 23]. However, the support for mobility should not clash with other requirements.

- **Multi-hop communication:** Industrial applications may demand large-scale communication network deployments, for example when a large number of sensor nodes are deployed over a wide area. In this case, sensor nodes need to transmit not only their own data, but they also have to relay the data of other nodes that are multiple hops away from the sink.

- **Low power consumption:** Many IIoT applications need to run on battery-powered nodes (often mobile). This calls for the design of low-power devices that either do not frequently need battery replacement or that never require it over their entire lifetime (e.g., in the case of sensor nodes located in inaccessible areas). This creates a demand for low-power communication design of battery-powered nodes. However, energy efficiency often clashed with reliability and latency. In fact, the latter requirements typically call for solutions that may cause significant energy consumption, therefore a trade-off among these contrasting applications needs has to be found.

- **Real-Time:** IIoT devices may be deployed in noisy environments to support mission-critical and safety-critical applications, having stringent timing requirements on message delivery. In such cases, industrial networks have to meet the timing constraints of the supported applications [24]. This means that the end-to-end (E2E) delays of messages, i.e., the delivery times from the source to the destination, have to be bounded and predictable.

- **Support for multiple traffic classes:** Most industrial communication tasks are periodic. When the interval between two consecutive transmissions on each node is relatively long [2], the deployment of a low power wireless network represents the best option. In that case the wireless nodes turn off during the idle time, while turn on when a transmission is required. This way, a significant energy saving is guaranteed. However, the industrial environment is heterogeneous, i.e., it involves different kinds of traffic as some applications demand the support for prioritized flows. The high priority (real-time) flows need to be delivered as soon as possible, while low priority flows are not subject to real-time constraints. In addition, both periodic and aperiodic data transmissions must often be supported. Consequently, low-power wireless networks that support multiple traffic classes are required.

This thesis mainly focuses on the design and development of efficient medium access control (MAC) protocols to improve the performance of industrial wireless networks in order to satisfy the IIoT applications requirements. The MAC protocol is the core of IWNs [4], as it determines when and how the nodes can access the communication medium. The MAC design directly affects a number of performance metrics such as reliability, scalability, latency, and low power consumption.

Each application may impose different requirements, therefore, in order to fulfill those requirements, several MAC protocols emerged. Basically, each protocol, with its specific design, defines a way to access the medium and achieve application-specific requirements. Generally, MAC schemes can be classified into two main categories, i.e., schedule-based and contention-based, that are described in the following.

- **Schedule-based MAC:** In a schedule-based MAC, each node is assigned a portion of the available bandwidth that is used exclusively by the node. The most used approach in this class is the Time Division Multiple Access (TDMA). In a TDMA-based protocol, the time is divided into timeslots. Each node is assigned at least one timeslot in which it has the right to access the channel following a predefined schedule. This way, each node can access the transmission medium in a collision-free manner. Most of these protocols adopts a superframe structure, made up of timeslots, that repeats over the time. Time synchronization among nodes is crucial in these approaches.

  Schedule-based schemes have several advantages, such as, collision-free communication and idle listening avoidance, that are very beneficial to energy saving. As a result, schedule-based MAC protocols are more energy-efficient than the contention-based ones. Schedule-based schemes also are more predictable and can offer deterministic end-to-end delays, with almost null jitter.

  However, as schedule-based protocols mainly follow a fixed predefined channel access scheme, they are unable to guarantee immediate channel access to high-priority event-driven time-critical data transmission (e.g., alarms or emergency data). In fact, most of schedule-based schemes do not support priority, so the transmission of emergency data has to wait for the dedicated timeslot, if any. Moreover, reserving resources, such as timeslots in a TDMA-based protocol, for event-driven transmissions whose occurrence is not predictable (in fact, they could even not occur at all) reduces the network bandwidth utilization. Generally, schedule-based MAC schemes suffer from low scalability, as a high number of nodes entails a high number of timeslots in the communication cycle. This entails higher message latency and may result in cycle times longer than the maximum value that some control applications can tolerate.

- **Contention-based MAC:** In these channel access schemes, each node competes with the others for the transmission medium access. Protocols under this class are the Additive Link On-Line Hawaii System (ALOHA), the Slotted ALOHA, and the Carrier Sense Multiple Access (CSMA) and its flavors. In the ALOHA protocol each node initiates transmission whenever it has data to transmit, regardless of the transmission activity of the other nodes. In the Slotted ALOHA each node can initiate transmission only at the beginning of a slot. This way, the collision probability decreases. In the CSMA approach a node, before accessing the channel, senses it to determine if any other transmission is already in progress (i.e., channel busy). If so, it waits a random amount of time and tries again later. If the channel is sensed free, the node starts its transmission.

  Although contention-based medium access schemes provide high flexibility to topology changes, low delays and sufficient throughput under low traffic conditions, they are not suitable for real-time applications [4], as they cannot guarantee deterministic communication due to their random collisions and delays. In addition, these MAC access schemes are not energy-efficient due to collisions, idle listening and retransmissions, especially when the node density (and, therefore, the probability of collision) increases.

This thesis proposes innovative mechanisms and protocols that build upon existing low power wireless communication technologies and add new features that enable such technologies to support the needs of real-time industrial applications. The main aim of this thesis work is to offer support for real-time constraints, scalability and different classes of traffic flows over low-power technologies, thus paving the way for their wide adoption in IIoT applications in the Industry 4.0 context. The thesis proposes a collection of innovative solutions, consisting of novel mechanisms, protocols and algorithms, for different low power technologies. For each solution proposed, the performance results are presented and discussed.

Chapter 2

Thesis outline

This thesis work focuses on low-power communication technologies and aims at improving them with the features that are needed to support IIoT applications. Among the available options, the focus of the thesis is on Bluetooth Low Energy (BLE) [25] and LoRa (Long Range) [26].

The selection of these low-power technologies derives from the observation that they are suitable candidates for different types of IIoT applications [9, 27–29]. In fact, on one hand, the Bluetooth Low Energy (BLE) protocol is an attractive solution for IIoT applications, as it is a low-cost technology that offers ultra-low power consumption [9, 30]. However, BLE is suitable for relatively small networks with a limited number of nodes.

On the other hand, IIoT applications, such as indoor industrial monitoring and intrusion detection, call for large and dense networks and therefore Low Power Wide Area Networks (LPWAN) are gaining an increasing interest in this area. For this reason, this thesis investigates the LoRaWAN protocol, that in the literature is considered more suitable for IIoT applications than other ones such as NB-IoT and Sigfox [31]. The thesis, in particular, focuses on the LoRaWAN Physical layer, LoRa.

As IIoT applications may also require to combine different protocols with very diverse features, their integration would result in a complex heterogeneous network, quite difficult to configure and manage. To cope with this, Software-Defined Networking (SDN) and network virtualization techniques can be exploited to improve the network management [32]. Unfortunately, the state-of-the-art network architectures support only a limited set of communication protocols, i.e., they are either software-defined WLAN (IEEE 802.11-based) or software-defined WSN (IEEE 802.15.4-based), such as [33] and [34]. For this reason, this thesis work addresses the design and an implementation roadmap of a network architecture that provides support for any communication technology.

In the following Sections, the contents of each Chapter are briefly sketched.

2.1 Introducing the support for real-time communications over Bluetooth Low Energy mesh networks.

As it was mentioned before, BLE is considered a promising solution for IIoT applications, thanks to its low-cost technology and ultra-low power consumption [9, 30]. However, BLE suffers from some limitations, as it does not provide real-time support for exchanged packets. In addition, BLE has range limitations, since it only supports star networks and a limited number of devices [35]. To overcome some of the BLE limitations, BLE mesh networking specifications [36] were recently released. Unfortunately, BLE mesh does not provide support for real-time communications over multi-hop mesh networks.

The work in [37] provides a configuration method for the BLE standard to guarantee bounded packet latency over star networks. It also outlines an approach to maintain bounded latency over mesh networks. This approach exploits Time Division Multiple Access (TDMA) to provide data packets with real-time support.

Chapter 3 of this thesis presents MRT-BLE, a real-time protocol for industrial wireless mesh networks that is developed on top of BLE and overcomes these limitations.

Besides describing the MRT-BLE protocol, Chapter 3 provides a timing analysis of multi-hop communications to compute the worst-case end-to-end delay of periodic packets, as well as a proof-of-concept implementation on STMicroelectronics BlueNRG-MS devices. The Chapter also presents and discusses analytical and experimental results.

2.2 Low Power Wide Area networks for long range applications in the context of Industry 4.0

Some of the widely-used wireless industrial standards, such as WirelessHART and ISA100.11a, are not designed to connect a large number of devices [9]. As this is the typical case for IIoT applications, such as indoor industrial monitoring and intrusion detection, Low Power Wide Area Network (LPWAN) are gaining an increasing interest for such applications. LPWANs were originally conceived for long range applications, such as smart metering [38] and smart cities [39], thanks to their wide coverage ranges and low power consumption. In particular, Low Power Wide Area (LPWA) technologies achieve low-power operation using several energy-efficient design approaches. In fact, LPWANs usually work on star topologies, thus avoiding the energy consumption due to packet routing in multi-hop networks. In addition, in LPWANs the end nodes maintain a simple design, while shifting the complexity to the gateway.

As a result, LPWANs can be considered for the IIoT, as their features nicely fit some IIoT application requirements. However, some aspects, such as the support for real-time flows, need to be investigated [40].

2.2.1 Introducing real-time communication in LoRa networks.

The most popular LPWAN technologies are LoRaWAN, NB-IoT and Sigfox. This thesis investigates LoRaWAN, as it is considered more appropriate for IIoT applications [31]. Comparing with the other two technologies, LoRaWAN operates in the license-free spectrum, whereas NB-IoT uses licensed frequency bands. Moreover, LoRaWAN supports a higher bit rate than SigFox. LoRaWAN also presents advantages over cellular technologies, since the latter are not suitable for low-power applications [9]. However, the LoRaWAN MAC protocol is ALOHA-based and therefore it is not able to support real-time communications. For this reason, in Chapters 4 and 5 alternative medium access protocols are proposed.

Chapter 4 presents the Industrial LoRa protocol and a simulative performance assessment in realistic scenarios.

2.2.2 Improving the suitability of LoRa for Industrial Internet of Things applications: the RT-LoRa protocol.

The Industrial LoRa protocol, described in Chapter 4, proposes the use of LoRa technology for industrial applications, offering support to both real-time and nonreal-time communications. However, there are still some aspects that can be improved, e.g., the reliability of aperiodic transmissions.

Consequently, this thesis also presents the RT-LoRa protocol, a LoRa-based MAC protocol that improves the suitability of LoRa for IIoT applications through a set of extensions and improvements over the Industrial LoRa approach.

Chapter 5 provides a description of the RT-LoRa protocol, guidelines for the configuration of a RT-LoRa network, a schedulability analysis for real-time periodic flows, and simulative assessments obtained through OMNeT++ simulations.

2.3 Software-defined management of industrial networks

Future industrial networks are expected to support a broad set of applications and services with diverse communication requirements that are difficult to meet adopting a single communication technology.

Software-Defined Networking (SDN) and network virtualization techniques represent two promising innovations and abstractions to improve the network management and to support scalable network control functions, flexible resource allocation, and changes in the network traffic. In addition, SDN with Network Functions Virtualization (NFV) may allow to simultaneously handle the physical infrastructure as separate logical network infrastructures, each one with specific functionalities and different performance requirements, in a centralized and more effective way [18]. These emerging technologies (along with big data and cloud computing) will bring great opportunities for promoting industrial upgrades towards Industry 4.0 [41].

In this context, Chapter 6 of this thesis draws a research direction towards software-defined management of heterogeneous Industry 4.0 communication networks. In particular, Chapter 6 outlines a software-defined architecture that is able to support a multitude of diverse wired and wireless communication technologies, while meeting the requirements of diverse applications. Chapter 6 also provides an implementation roadmap for the proposed architecture.

Chapter 3

Multi-hop Real-time Communications over Bluetooth Low Energy Industrial Wireless Mesh Networks

Bluetooth Low Energy [42] is a wireless technology conceived for low-power, low-cost, low-complexity short-range communications, that represents an interesting solution for implementing Industrial Wireless Sensor Networks (IWSN). In fact, BLE provides lower energy consumption compared to other wireless technologies adopted for industrial applications [43], [44]. Moreover, thanks to the low cost of devices and to their diffusion, BLE is a good candidate for several industrial applications [35]. However, the BLE specifications [42], [45], [25] do not include support for real-time traffic, thus the BLE standard is not suitable for IWSN, which require bounded packet delays. A recent work [37] has provided a configuration method for the BLE standard to guarantee bounded packet latencies over star networks. As BLE is a short-range technology, a multi-hop mesh network is needed to cover a large area. However, mesh topologies are not foreseen in the new Bluetooth version 5 specification [25]. The Bluetooth Smart Mesh Working Group has recently introduced the Bluetooth mesh networking specifications [36], which define the requirements to enable mesh networking solutions for BLE. Unfortunately, such specifications do not provide support for real-time communications over multi-hop mesh networks.

To overcome the above discussed limitations, thus enabling BLE to be introduced in industrial environments, this chapter proposes the Multi-hop Real-time BLE (MRT-BLE), that is a real-time protocol, developed on top of BLE, able to realize low-cost IWSN with mesh topologies.

The MRT-BLE protocol is connection-oriented, unlike the Bluetooth mesh networking specifications [36], which are defined as connectionless communications. There are two reasons for this choice. Firstly, the connection-oriented approach provides a higher throughput than the connection-less one, as packets can be transmitted more frequently. Secondly, in the connection-oriented approach, 37 channels can be exploited for channel hopping, instead of the 3 channels that can be used for connection-less advertising. This entails a lower channel collision probability and therefore a higher reliability. The MRT-BLE protocol adopts a Time Division Multiple Access (TDMA) approach that exploits a transmission allocation scheme able to support the timely exchange of multi-hop real-time data packets. This work builds upon the approach proposed in [37], that outlines a protocol working on top of BLE. Such a protocol allows to maintain bounded latencies over mesh networks. However, in the work in [37] the approach is merely sketched. The ability of taking advantage of the Client Characteristic Configuration Descriptor (CCCD) in order to enable/disable connection is mentioned, but nothing is said about the timing of this mechanism. Moreover, the work in [37] does not address the implementation of the MRT-BLE on Commercial-Off-The-Shelves (COTS) devices. Conversely, this work aims to describe in detail how the MRT-BLE works, address the timing of the connection enabling/disabling, introduce the support for packet priority, and offer two kinds of evaluation. The first one is based on measurements obtained in a realistic scenario, while the second one is a worst-case end-to-end delay analysis. The contribution of this chapter consists in the MRT-BLE mechanism itself and the relevant timing analysis that represents a useful tool for the network designers, since it allows them to assess whether a feasible schedule for a given flow set can be found or not. In the negative case, the timing analysis helps designers in the re-engineering process, allowing them to see the effects of changing the flow periods or paths along the network. To the best of my knowledge, this is the first work that proposes and assesses a mechanism to provide real-time multi-hop communications over BLE through assessment in a real scenario. The chapter is organized as follows. Section 3.1 overviews related work, while Section 3.2 summarizes BLE and the most relevant features exploited by the proposed protocol. Section 3.3 presents the MRT-BLE protocol design, while Section 3.4 provides a timing analysis of the protocol. Section 3.5 describes an implementation of the MRT-BLE protocol on real devices and presents some experimental results. Finally, Section 3.6 gives my conclusions and hints for future work.

3.1 Related Work

BLE is gaining ground in specific sectors, for instance, energy management for smart homes [46] [47], food manufacturing and traceability [48] [49]), and also in applications, such as machine monitoring and structural health monitoring, in which other technologies, e.g., the IEEE 802.15.4 standard, proved to be effective and efficient too [50] [51]. In [52] the IEEE 802.15.4 and BLE protocols are evaluated in terms of service ratio, delay, and energy efficiency under IPv6 traffic.

Some studies analyzed the BLE system behavior in wireless environments under realistic operating conditions. In particular, the work in [53] addresses the coexistence of IEEE 802.15.4, BLE and IEEE 802.11 through analysis and experiments to determine the effect of cross-technology interference on the wireless network reliability. The paper concludes that the MAC layer mechanisms of both IEEE 802.15.4 and BLE improve reliability and that cooperative solutions are required to achieve coexistence. In [54] an extension of the BLE channel access scheme in connection state that is based on the listen-before-talk mechanism is proposed. The numerical results in the paper show that the proposed scheme reduces the average transmission delay.

Several studies investigated the BLE discovery phase, which plays a critical role in the networks with dynamic topology. For instance, the paper [55] addresses the influence of parameter settings on the latency and energy performance of the discovery process. A mechanism to enhance the discovery phase for BLE devices is proposed in [56]. The mechanism aims at avoiding collisions during the advertisement process, so as to achieve lower latency and energy consumption.

Modern BLE radio transceivers allow to partition the network bandwidth between the BLE and another user-defined protocol. The paper [57] shows the feasibility of a dual-protocol approach and its capability to support a custom real-time protocol running on top of the raw radio layer. The proposed real-time protocol introduces a bounded overhead and can be implemented in specific devices able to support two different protocols.

In order to increase the communication coverage of BLE, some papers, as [58] [59] [60] [61], proposed different methods prior to the publication of the Bluetooth mesh networking specifications. For instance, the paper [58] proposes a method to implement a BLE network with a tree topology to extend the coverage to other wireless sensor networks. A cluster-based on-demand routing protocol to support multihop communication in Bluetooth low energy ad hoc networks is proposed in [59]. In [60] the applicability of BLE for mesh-enabled applications is addressed. The results, obtained using a prototype of the proprietary BLE-based CSRmesh protocol [62], show that BLE mesh is a promising technology for mesh applications, but that more studies are needed to fully exploit its potential. The paper [61] presents a new mesh proposal based on the proprietary BLE-based CSRmesh protocol, that exploits a connectionless network and tries to overcome some limitations of the CSRmesh network (e.g. the scalability, the high number of retransmissions). Finally, [63] surveys state-of-the-art BLE mesh network solutions and discusses their advantages and drawbacks.

The IP support has recently turned BLE into a potential candidate for a broad range of applications including healthcare, wearable devices, home automation, and Internet of Things (IoT) [64] [65] [66]. The delay performance of BLE for time-critical applications is addressed in [67]. The work analytically models the delay for connection-oriented applications under different bit error conditions. The works in [68] and [35] explored the BLE suitability for industrial and process automation communications and for time-critical IIoT applications, respectively. In particular, the last one [35] presents a retransmission scheme able to fulfill the reliability and timeliness requirements of IIoT applications.

The above mentioned literature indicates that it is worthwhile investigating novel extensions and configurations of BLE to make it more suitable for industrial environments. For instance, in [37], a configuration method for the BLE standard that guarantees bounded packet latencies with star topologies was proposed, together with a new protocol for BLE-based mesh topologies that is able to provide bounded latencies. However, the paper [37] leaves for future work both the development of the proposed solution on COTS devices and the protocol assessment in a realistic multi-hop scenario. Comparing with the work in [37], that simply outlines the MRT-BLE idea, this work turns the idea into a protocol, characterizing in detail the mechanisms to guarantee bounded latencies on multihop communications and their timing. This chapter also provides a timing analysis for multi-hop communications to compute the worst-case end-to-end delay of periodic packets. In addition, this work addresses the

MRT-BLE protocol implementation on COTS devices and assesses the end-to-end delay performance in a realistic multi-hop scenario. The measured delay values are compared with the computed worst-case end-to-end delays obtained through the analysis in Section 3.4. As it will be shown, the theoretical upper bounds are found compliant with the measured values.

The proposed MRT-BLE protocol differs from the Bluetooth mesh networking specifications [36] as the latter are defined as connectionless communications, based on the GAP Broadcaster and Observer roles, while the MRT-BLE protocol is connection-oriented and offers support for real-time communications over multi-hop mesh networks.

3.2 BLE Summary

The Bluetooth Low Energy core system architecture includes an RF transceiver and a protocol stack, shown in Fig. 3.1, that enable BLE devices to connect and exchange data.

The lowest system layers are grouped into a subsystem known as the Controller. This is a common implementation that uses a standard interface, called the Host Controller Interface (HCI), which enables two-way communications with the upper layers of the protocol stack that realize the so-called Host in Fig. 3.1. Applications are developed on top of the Host. The following subsections summarize the most important features of the BLE Link Layer, the Attribute Protocol and the Generic Attribute Profile. Such features are maintained in the BLE versions 4.1 [42], 4.2 [45] and 5.0 [25] .

3.2.1 Link Layer

The operation of the BLE Link Layer (LL) can be described through a state machine with the following states: Standby, Advertising, Scanning, Initiating, and Connection. The Link Layer may run multiple state machine instances and each Link Layer state machine allows only one active state at a time. When two devices are engaged in a

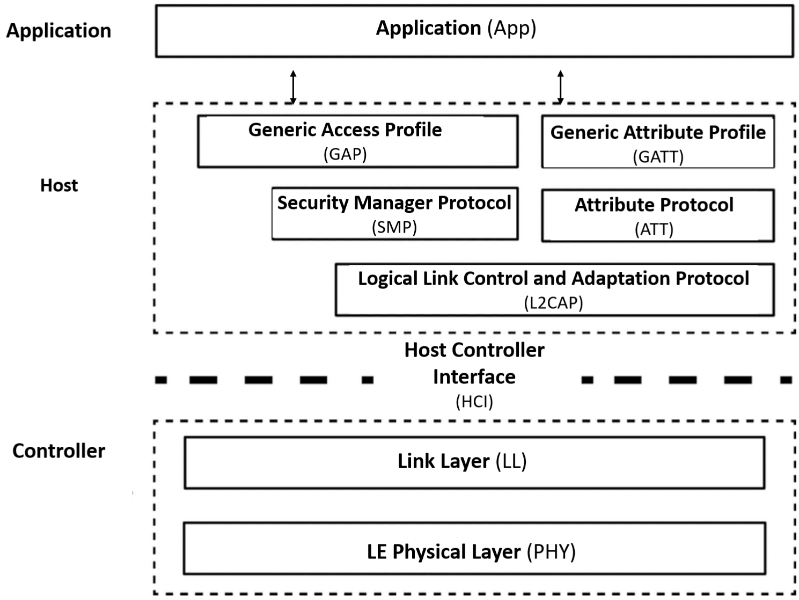
Application
Application (App)
Host
Generic Access Profile (GAP)
Generic Attribute Profile (GATT)
Security Manager Protocol (SMP)
Attribute Protocol (ATT)
Logical Link Control and Adaptation Protocol (L2CAP)
Host Controller Interface (HCI)
Controller
Link Layer (LL)
LE Physical Layer (PHY)

Figure 3.1: The BLE protocol stack

connection, they can play the master or the slave role. The master is the device that initiates the connection and coordinates the medium access, periodically polling the slaves in a Time Division Multiple Access (TDMA) way.

The master controls the timing of the connection events (CE), i.e., the synchronization points for the master and the slave. Since two devices must be tuned to the same RF channel to communicate and the Link Layer may use one physical channel at a given time, the master and slave shall also determine the data channel index for each connection event to reduce the collisions. A physical channel collision is possible because many BLE devices can independently operate within the same spatial and temporal area, and the number of physical channels (up to 37) is limited, thus two independent BLE devices may have their transceivers tuned to the same physical channel.

The start time of a connection event, called an Anchor Point (AP), is the instant at which the master shall start to transmit a packet to the slave. During a connection event, the master and the slave send and receive packets alternately. Consecutive packet transmissions are separated by an Inter Frame Space (IFS), whose duration is 150 μs. A connection event is considered open as long as both devices keep sending packets. If none of the devices has data to transmit, the slave switches to the sleep mode until the next AP.

The BLE specifications define two timing parameters for connection events, i.e., the Connection Interval (CI) and the Connection Slave Latency (CSL). The CI represents the time interval that regularly spaces the start times of connection events. The second parameter, the Connection Slave Latency, defines the number of consecutive connection events that a slave has to skip, while remaining in sleep mode, when it does not need to communicate with the master.

Several BLE implementations [69] require a guard band (GB) between the connection events to cope with the synchronization accuracy, and the master shall ensure that a connection event closes at least one guard band before the anchor point of the next connection event. Moreover, such BLE implementations provide configurable parameters to set the maximum duration of each connection event. The master has the ability to schedule the connection event anchor point at a time of its choice to efficiently schedule connection events for the multiple connections it is involved in. This way, the master basically splits the connection interval into as many connection events as the number of connections.

Fig. 3.2 shows an example in which there are two slaves (S1 and S2) and one master (M). In Fig. 3.2, the lower time axis gives the schedule of the connection events for the two connections, while the upper time axis shows the details of each connection event.

At the anchor point of the slave S1 ($\boldsymbol{AP}_{S1}$) the master starts the polling, transmitting a packet to S1, thus beginning the alternating transmission sequence (M to S1, S1 to M, etc.). The connection event for the slave S1 periodically repeats with a period equal to CI. At

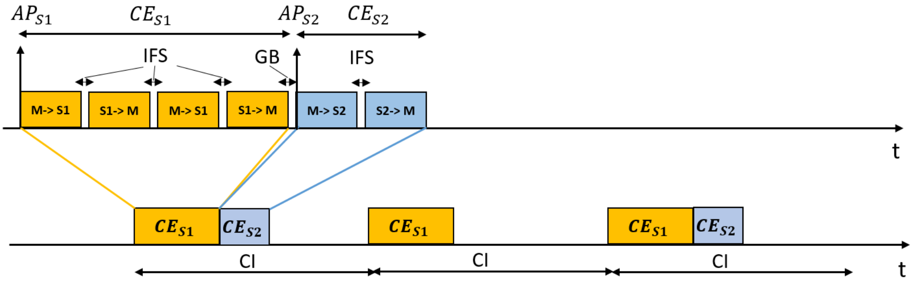

Figure 3.2: Example of the BLE master scheduling for multiple connections.

the end of the first CE for S1, here called CE_{S1}, after the guard band the master starts the connection event for the slave S2 (CE_{S2}), which periodically repeats with a period equal to 2*CI.

While the connection interval CI is the same for all the slaves connected to the same master, the anchor points of the relevant slaves are shifted, so the master polls one slave at a time. Every Link Layer connection uses a mechanism of acknowledgement and flow control. Consequently, the Link Layer guarantees the retransmission of the unacknowledged packets.

3.2.2 Attribute Protocol

The Attribute Protocol (ATT) [42] allows a device, named the server, to expose a set of attributes and their associated values to a peer device, named the client. An attribute is a value that has three properties associated with it, i.e., the attribute type, the attribute handle and a set of permissions. The client can discover, read, and write the attributes exposed by the server. The server can indicate and notify its attributes. For example an ATT server can expose three attributes that represent the data detected by three sensors (e.g., temperature, pressure and humidity) and periodically notifies the connected clients with new measured data. In another scenario, these attributes can be read by the clients when necessary. A client may send an Attribute Protocol request to a server and the server shall respond to all the received requests. A device can implement both the client and the server roles, which can concurrently run in the same device, but each device shall run only one instance of a server.

The Attribute Protocol procedures typically use a sequential request-response protocol. An Attribute Protocol {request, response} pair is considered a single transaction, i.e., once a client sends a request to a server (e.g., a Write request to write a server attribute), that client shall not send other requests to the same server until a response Protocol Data Unit (PDU) has been received (e.g., a Write response sent by the server to inform the client of the result of the Write request). Conversely, there are procedures, e.g., the notification (a server-initiated procedure) that do not have a response PDU. For example, a server notifies an attribute that contains the temperature detected by a sensor and updates its value. This value is received by the connected clients that have enabled the receiving of notification packets after the update of a specific attribute.

3.2.3 Generic Attribute Profile

The Generic Attribute Profile (GATT) [42] defines a service framework built on top of the Attribute Protocol. The GATT profile is designed to be used by an application or another profile so that a client can communicate with a server. The GATT defines the hierarchical data structure, shown in Fig. 3.3, that a device exposes to the connected devices and also defines the way two BLE devices exchange standard packets.

The server contains some attributes and the GATT Profile defines how to use the Attribute Protocol to discover, read, write, notify, indicate, and obtain indications about these attributes.

The top level of the hierarchy is the Profile, which is composed of one or more services that are needed to fulfill a use case. A ser-

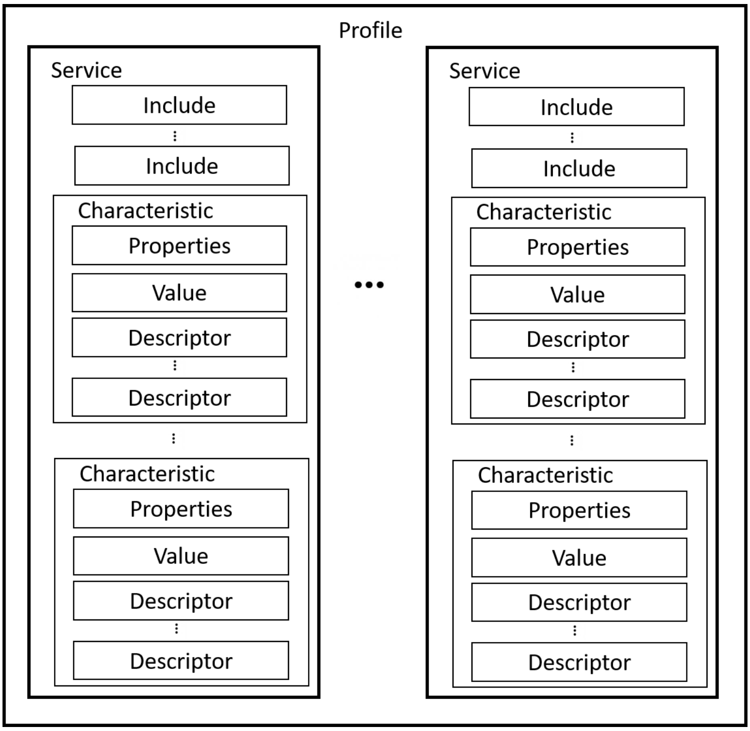

Figure 3.3: GATT profile hierarchy.

vice is generally composed of characteristics or references to other services. Each characteristic consists of various fields, i.e., a set of properties, a value, and one or more optional descriptors. An important descriptor is the Client Characteristic Configuration Descriptor (CCCD) that configures for the client a characteristic on the server (e.g., enabling/disabling of notifications and/or indications).

For example, as shown in [70], a profile may expose two services, i.e., the acceleration service and the environmental service. The acceleration service contains a characteristic called free-fall that cannot be read or written, but can be notified. The application will send a notification on this characteristic if a free-fall condition is detected by a sensor. Notifications can be enabled or disabled by a client writing on the related CCCD. The environmental service contains three characteristics (with read-only properties) that expose data from some environmental sensors, i.e., temperature, pressure and humidity. Each characteristic of the environmental service has a characteristic format descriptor that describes the type of data contained inside the characteristic.

The basic elements used in a profile, i.e., services and characteristics, are contained in the attributes used in the Attribute Protocol.

The Attribute Protocol Maximum Transfer Unit (ATT_MTU) is defined as the maximum size of any packet exchanged between a client and a server. The GATT client and server implementations shall support an ATT_MTU not smaller than the default value (23 bytes).

3.3 Protocol Design

The Multi-hop Real-time Bluetooth Low Energy (MRT-BLE) protocol for Industrial Wireless Sensor Networks proposed in this chapter allows the creation of BLE-based mesh networks able to provide bounded packet delays on multi-hop data transmissions, thus offering support for real-time communications. Fig. 3.4 shows the position of the MRT-BLE protocol in the BLE protocol stack.

The main idea behind the MRT-BLE protocol is to subdivide the network into a number of sub-networks, each one coordinated by a master. Two sub-networks are linked by a device that is either a shared slave or a master/slave device. These devices act as "bridges" among the sub-networks. As the masters of the two linked sub-networks are not synchronized, connection events may overlap at the bridge nodes, thus determining a collision that does not make it possible to guarantee transmission.

Fig. 3.5 shows an example of network topology in which two sub-networks are linked by a master/slave device.

In Fig. 3.5 the sub-networks are coordinated by the nodes MS1

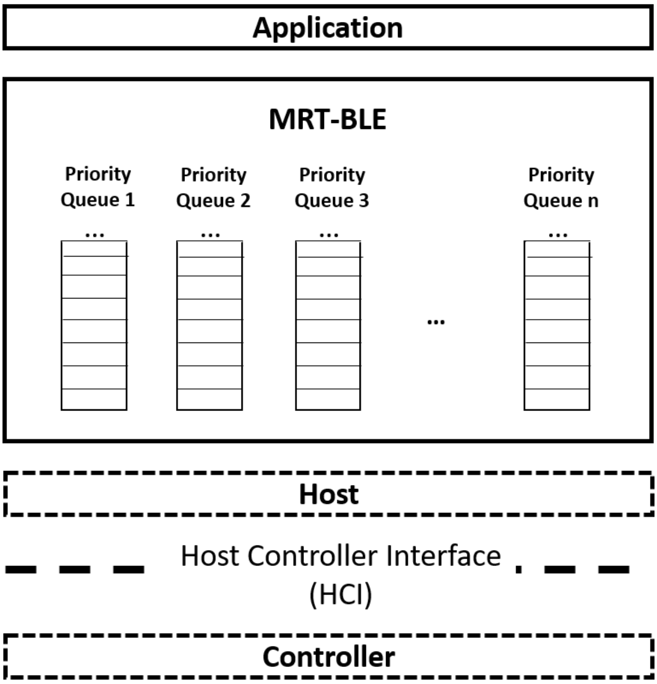

Figure 3.4: The BLE protocol stack including the MRT-BLE.

and M1, respectively. The node MS1 is a master/slave device, as it plays the role of master for the nodes S1, S2 and S3, and the role of slave for the node M1. Since M1 and MS1 are not synchronized, the communications in which MS1 acts as a slave may overlap with those in which it plays the master role. While MS1 is acting as a slave, its timing is defined by the master M1, therefore the timing of the connection event in which MS1 acts as a slave is not synchronized with the time base mechanism of the MS1 master connections.

However, a viable solution to avoid the above-mentioned problem is to give MS1 a proper schedule so that, while MS1 is communicating with the nodes of its subnetwork (i.e., S1, S2 and S3), M1 avoids transmitting to MS1. On the contrary, S4 and S5 may transmit simultaneously with M1, as MS1 is not intended to communicate directly

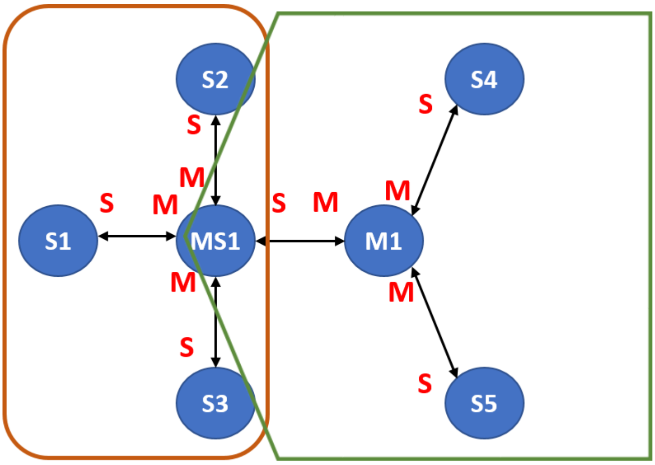

Figure 3.5: Example of topology with a master/slave device.

with M1. The approach here proposed is based on the latter consideration. In general, the problem of enabling adjacent subnetworks to communicate can be solved by scheduling adjacent subnetwork transmissions in two alternate timeslices, so that when a shared node (e.g., MS1) schedules the communication with the nodes of its subnetwork, its adjacent master is prevented from communicating with the shared node.

To achieve this result, thus avoiding communication overlap, in the proposed approach for each connection the duration of one CE is configured offline, so as to allow the transmission of one packet between the slave and the master and of one packet in the opposite direction. The connection intervals for all connections have to be set with the same duration. Moreover, CIs have to be sized so as to allow the communication between all slaves and the master of a sub-network once. This way, the duration of one timeslice is equal to the duration of one CI and the shared node alternates its connections with a sub-network at a time.

In the example shown in Fig. 3.5 the proposed solution consists in alternating the connections of the node MS1 with its master, with those with its slaves. This means that, in a given timeslice (according to the proposed solution a timeslice is a multiple of one CI) the node MS1 will disable the connection with the node M1 and enable the connections with the nodes S1, S2, and S3, while in the next timeslice it will operate in the opposite way. This solution can be easily realized enabling/disabling a connection.

Each device, before sending data, has to check that the connection is enabled. If this is not the case, the master will insert the packet in a queue managed by the MRT-BLE protocol and transmit it in the next connection event. This way the CE overlap is avoided.

Fig. 3.6 shows the timing of the masters MS1 and M1 relevant to the topology shown in Fig. 3.5. The semi-transparent rectangles represent the timeslices during which some connections are disabled.

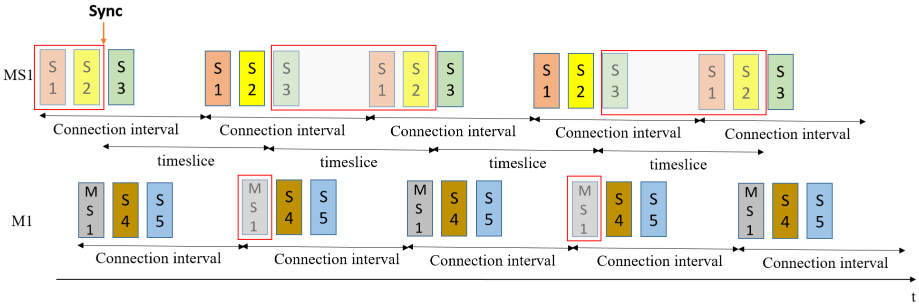

Figure 3.6: Solution to the overlapping CE problem with a master/slave shared device.

Fig. 3.6 shows as the node MS1, acting as a GATT client, can enable/disable the connections with the devices that act as a GATT server. In the initial phase, MS1 enables the connection with the master M1 (shown in the lower timeline), while keeping disabled the master connections. Once MS1 has received the first packet from M1 (considered a synchronization packet), it disables the connection with the master M1 and enables the master connections with the nodes S1, S2, and S3 (shown in the upper timeline). After a time equal to the connection interval (in the example, the timeslice duration is equal to one CI), the node MS1 disables the master connections and enables the connection with M1, for one timeslice. This mechanism repeats over time.

Fig. 3.7 shows an example of network topology in which there are two sub-networks linked by a shared slave device.

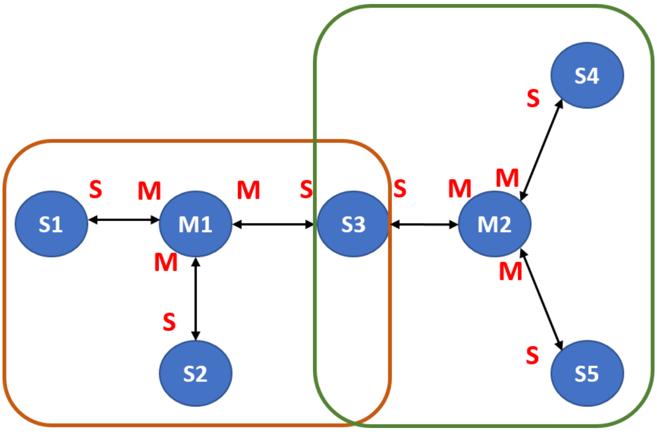

Figure 3.7: Example of topology with a shared slave device.

In Fig. 3.7 the sub-networks are coordinated by nodes M1 and M2, respectively.

As the node S3 is a shared slave it has to communicate with two masters that are not synchronized with each other, therefore the connection events of this node may overlap.

In the example shown in Fig. 3.7 the proposed solution consists in alternating the connections of the node S3 with the masters. This means that, in a given connection interval the node S3, acting as a GATT client, will disable the connection with the node M1 and enable the one with the node M2, while for the next connection interval the node S3 will enable the connection with the node M1 and disable the one with the node M2.

In this case, the shared slave S3 starts alternating its connections, for example, upon receiving a synchronization packet. Assuming that

S3 receives the synchronization packet from M1, the shared slave will start the mechanism at this very time instant. First, the node S3 disables the connection with the node M1 and enables the connection with the node M2. After a time equal to one timeslice, the node S3 will enable the connection with the node M1 and disable the connection with the node M2. The described mechanism repeats over time. This way the CE overlap problem is avoided.

Fig. 3.8 shows the timing of the masters M1 and M2 in the topology shown in Fig. 3.7. The semi-transparent rectangles represent the intervals during which the node S3 disables the connection with one of the two masters.

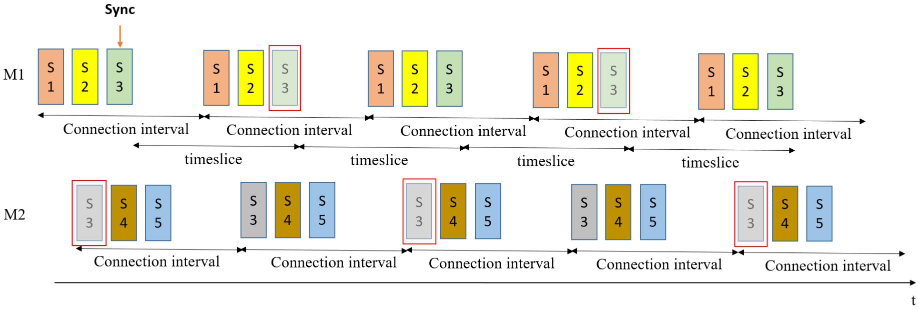

Figure 3.8: Solution to the overlapping CE problem in the shared slave topology.

Fig. 3.8 shows that the node S3 in the initial phase enables the notification only for the connection with the master M1. Upon receiving the sync packet, the node S3 disables the connection with the master M1 and enables the connection with the master M2. After one timeslice (in this case, equal to one connection interval) the node S3 makes the opposite action, i.e., disables the connection with M2 and enables the connection with M1, for another timeslice. This mechanism repeats over time.

Note that Fig. 3.6 and Fig. 3.8 show connections alternating every connection interval (i.e., in these examples, the timeslice duration is equal to one connection interval).

In order to achieve bounded delays, the routing is static and configured offline. To improve the fault-tolerance, it is possible to provide multiple paths to be enabled in case of faults.

In order to deploy a mesh network without connection event overlap, the MRT-BLE imposes the following configuration rules:

1. A node not acting as a master can establish a connection with up to two masters.
2. A node (A) acting as a master can establish a connection with at most another master (B). In this connection, the node A shall play the slave role.
3. The connection intervals of the different sub-networks must have the same duration.

According to the BLE specifications [42], a device can act both as a master and a slave for different connections and can also establish connections with multiple masters.

To realize the described mechanisms, the solution proposed in this chapter is based on the transmissions from the GATT servers that notify their Characteristic Values. Each device is both a GATT client and a GATT server and defines a profile, called the MRT-BLE Profile, shown in Fig. 3.9.

The MRT-BLE profile is composed of a service that includes a number of characteristics equal to the number of connections. Each characteristic is used to communicate on a specific connection (according to a programmer-defined static configuration) and consists of a set of properties, a value, and the client characteristic configuration descriptor (CCCD). In particular, the *value* field contains the data to be sent, the CCCD is a bit field where a set bit indicates an enabled action (e.g., notification), while a cleared bit indicates a disabled action. *Properties* is a bit field that determines how the *value* can be used and how the CCCD can be accessed (e.g., read, write without response, notify, etc.). The *characteristic of the connection* indicates

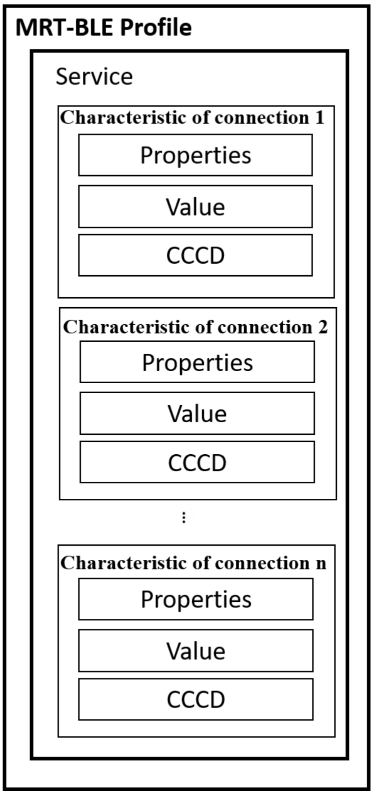
MRT-BLE Profile
Service
Characteristic of connection 1
Properties
Value
CCCD
Characteristic of connection 2
Properties
Value
CCCD
⋮
Characteristic of connection n
Properties
Value
CCCD

Figure 3.9: MRT-BLE Profile.

the specific characteristic that is used to send packets on a specific connection.

In the following of the chapter, when the *enabling/disabling of a connection* is mentioned, it means that the GATT client uses a {request, response} GATT procedure to write the CCCD of the characteristic of the connection on the server, so as to enable/disable the notification procedure of the GATT server. Consequently, a *disabled connection* indicates that the notifications of the characteristic of the connection are disabled, so the GATT servers involved in the connection cannot notify the value of this characteristic. An *enabled* *connection* will indicate the opposite case.

The proposed protocol does not implement a mechanism to dynamically configure the network topology, it makes the configuration offline. This aspect will be investigated in future works.

3.4 Timing Analysis

The analysis here proposed deals with the calculation of the worst-case end-to-end delay (hereinafter worst-case response time) of periodically transmitted packets belonging to a flow. The resulting worst-case end-to-end delay for each flow represents a lower bound for the minimum deadline that allows a feasible schedule for each flow. The analysis can be used to assess the feasibility, under a static priority assignment, of a flow set comparing the worst-case response time of each flow with the corresponding relative deadline. Table 5.1 summarizes the notation used in the analysis.

A connection between a master and a slave is defined as a link. A link is a bidirectional connection denoted by l_{xy} (where x is the index of the node acting as a master for the link and y the index of the slave). In the proposed approach for each link the duration of one CE is offline configured so as to allow the transmission of one packet between the slave and the master plus one packet between the master and the slave. The connection intervals for all links have to be set with the same duration and allow the communication between all slaves and the master of a sub-network once. A link in which at least one of the two connecting nodes is shared between two sub-networks is defined as a *shared link* (sl_{xy}). As discussed in Sect. 3.3, in the proposed approach, the time is divided into two timeslices. In each timeslice, a node with shared links alternates the connection with one sub-network at a time. Fig. 3.10 shows an example of the network timings. In the example the Node 2 acts as a master for the shared link sl_{21} and as a slave for sl_{32}. l_{34} is a link between two nodes that are connected to only one sub-network, so l_{34} is always active as no

Table 3.1: Summary of notation

Symbol	Definition
l_{xy}	The link between a node x acting as a master and a node y acting as a slave.
sl_{xy}	The shared link between a master node x and a slave node y, i.e., the link in which at least one of the nodes x and y is shared between two sub-networks.
i	The index of the i-th link.
Xci	The number of connection intervals within the two timeslices that can be used to transmit data packets over one shared link.
Tsw	The time that a node shared between two sub-networks takes to disable the shared links with a sub-network and to enable the other ones.
Tci	The duration of one connection interval.
$NL(i)$	The maximum between the number of shared links of the master and the slave, respectively, for the i-th shared link.
$CT(i)$	The cycle time, i.e., the time interval after which the entire schedule for the i-th link repeats.
$prio_f$	The priority of a packet of the f-th flow.
P_f	The period of a packet of the f-th flow.
R_f	The routing path of the f-th flow.
h^f	The index of the last link to the destination of the flow f.
RT_f	The maximum response time of a packet of the f-th flow.
$Q_t(l_i^{(f)})$	The maximum time that a packet of the f-th flow waits to be transmitted over the link l_i.
T_{tx}	The maximum time taken by the BLE controller to transmit a packet.
$s_i(t)$	The minimum number of CI start times dedicated to the transmission of data packets in the i-th link which may occur in any interval of length t.
$w_i(X)$	The longest time the i-th link has to wait to see X consecutive start times.
$I_f(t)$	The number of packets generated by the f-th flow in any interval of length t.
Tce	The duration of one connection event.

slot collisions can occur. The number of CIs in the two timeslices that a node can use to transmit data packets on a shared link is denoted by Xci and is equal for all the shared links in the network. In the example Xci is set to 2 and in fact sl_{21} and sl_{32} are active for only two CIs (sl_{21} in timeslice 1 and sl_{32} in timeslice 2).

To disable one link and enable the other one, the transmission of four packets is needed, i.e., two to disable one link (write_request and write_response) and two to enable the other one. According to the BLE specifications [42], request and response may be scheduled

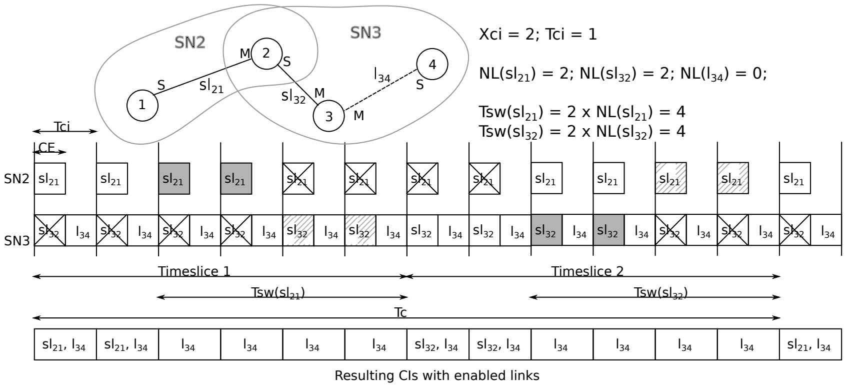

Figure 3.10: Network timings: an example.

within a single CE. Many implementations, such as the ST BlueNRG-MS devices [70], take four connection intervals to complete the entire procedure, as they transmit only one request or response in each CI. In the analysis such an interval is called Tsw. For example, in the ST BlueNRG-MS devices the Tsw of a node to enable/disable the connection of the i-th shared link is calculated as

$$Tsw(i) = 2 \times NL(i) \times Tci, \tag{3.1}$$

where $NL(i)$ for the i-th shared link is defined as the maximum between the number of shared links of the master and the number of shared links of the slave. The NL value for no shared links is 0. Equation (3.1) evaluates two times $NL(i)$, because the transmission of two packets, i.e., one write_request and one write_response packets, is required to disable/enable one link. Note that, the ST BlueNRG-MS devices [70] take two connection intervals to transmit one request and one response (i.e., only one request or response is transmitted in each CI).

The time interval in which the entire schedule, for the i-th link, repeats is called Cycle Time (CT), and it is calculated as

$$CT(i) = \begin{cases} NL(i) > 0 & \Rightarrow 2 \times Xci \times Tci + 2 \times Tsw(i) \\ NL(i) = 0 & \Rightarrow Tci \end{cases}. \tag{3.2}$$

In the MRT-BLE protocol, each node can generate multiple packets belonging to different flows (f) with different (or equal) priorities ($prio_f$). If two packets have the same priority, they are transmitted in *First-In First-Out* (FIFO) order. Packets are periodically transmitted with a period P_f. The routing path (R_f) for each flow is fixed and configured offline. To improve fault-tolerance, each node has to maintain a routing table with a backup path for each flow (this way the analysis can be repeated for each path). Each flow is therefore characterized by the tuple $(P_f, prio_f, R_f)$, where $R_f = (l_1^f, ..., l_h^f)$ is the vector of the links or shared links that a packet of the flow f has to traverse to reach the destination, while h is the index of the last link. The time RT_f taken by a packet of the f-th flow sent from a node to arrive to the destination node is calculated as

$$RT_f = \sum_{i=1}^{h} \left(Q_t(l_i^{(f)}) + T_{tx} \right) \tag{3.3}$$

where $Q_t(l_i^{(f)})$ is the queuing time that a packet of the f-th flow has to wait in the controller at each link to be transmitted and T_{tx} is the maximum time taken by the BLE controller to transmit a packet. T_{tx} has a fixed bound that is equal, in the worst case, to the duration of one connection interval, so in the worst case response-time analysis I assume that $T_{tx} = Tci$. In fact, the analysis here presented assumes that in each CI a node can transmit only one packet and the CE is sized so as to also accommodate retransmissions. So in one Tci the packet is transmitted and retransmitted as long as the CE for the node is not expired. In the following, a timing analysis for calculating the worst case $Q_t(l_i^{(f)})$ in each hop is presented. My aim as a future work is to deal with a stochastic analysis of flows characterized by a known distribution of the arrival times, as the one proposed in [71].

3.4.1 Response-time analysis

To determine the worst case $Q_t(l_i^{(f)})$ for each link, the worst case response-time (WCRT) analysis is made calculating both the resource availability and the worst case for the resource request. In MRT-BLE the resource availability for the i-th link is given by the number of CIs in the two timeslices (Fig. 3.10) that a node can use to transmit over the link i, while the resource request is the number of CIs needed to transmit the packets generated and/or forwarded by a node over the link i. The analysis here presented, which applies to fixed priority non-preemptive scheduling, is based on the worst case response-time (WCRT) analysis that was proposed in [72] and extended with the ”busy period” approach in [73], [74]. Consequently, the response times of all the packets of a flow within a busy period have to be examined. The busy period is defined as the maximum interval during which any packet of priority lower than the priority of the f-th flow is unable to start transmission. The busy period starts at the time t^s in which a packet with a priority higher than or equal to the one of the f-th flow is enqueued and there are no packets with priority higher than or equal to the f-th flow, enqueued strictly before t^s, that are waiting to be transmitted. The busy period ends at the earliest time t^e at which there are no packets of priority equal to or greater than the f-th flow, enqueued strictly before time t^e, that are waiting to be transmitted. Hence, all the packets with the same or higher priority than the f-th flow that were enqueued before the end of the busy period are transmitted during the busy period [74].

In the MRT-BLE protocol the worst case for the resource availability is determined by computing $s_i(t)$, i.e., the minimum number of CI start times dedicated to the transmission of data packets in the i-th link which may occur in any interval of length t. As illustrated in Fig. 3.11, the worst-case interval (that is the one containing the minimum number of start times) begins when the last CI, dedicated to data packet transmission, has just started. From this instant (please refer to Fig. 3.11 where the white dots denote the start time of data packets over the time), the time the i-th link has to wait before another data packet starts is $CT(i)-(Xci-1)Tci$. Then Xci data packet transmissions start, spaced at intervals of length Tci, and the pattern will repeat with period $CT(i)$. In the example in Fig. 3.10, the worst-case interval for the link sl_{21} begins when the second CI has just started. Then the time the link sl_{21} has to wait before another data packet starts is $CT(sl_{21}) - (Xci - 1)Tci = 12Tci - (2 - 1)Tci = 11Tci$.

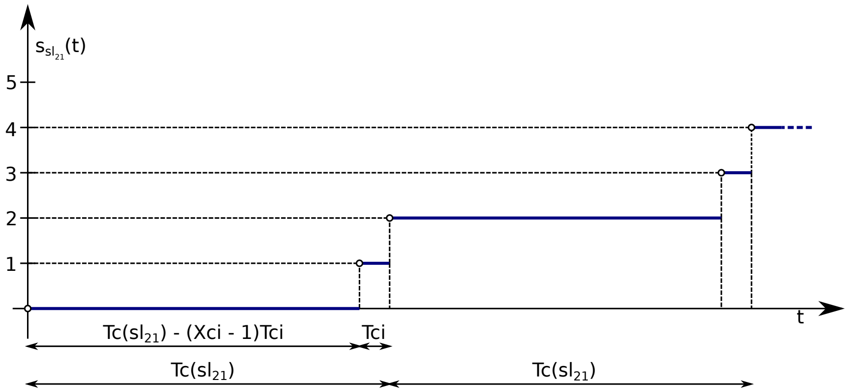

Figure 3.11: Example of the minimum number $s_{sl_{21}}(t)$ of CI start times dedicated to the transmission of data packets in the link sl_{21} for the network in Fig. 3.10.

The formal equation of $s_i(t)$ is

$$s_i(t) = \begin{cases} NL(i) > 0 & \Rightarrow \sum_{j=1}^{Xci} \left\lfloor \frac{t+(j-1)Tci}{CT(i)} \right\rfloor \\ NL(i) = 0 & \Rightarrow \left\lfloor \frac{t}{Tci} \right\rfloor \end{cases} . \tag{3.4}$$

where, if $NL(i) > 0$ (i.e., it is a shared link), the start times of Xci connection intervals dedicated to the transmission of data packets are separated by Tci, start times repeat with period CT and, in the worst case, the first start time occurs after $CT(i) - (Xci - 1)Tci$. If $NL(i) = 0$ (i.e., the link is not shared) data packets can be transmitted every CI, so the start times repeat with period Tci.

As the link transmissions, thanks to the mechanism here proposed, do not collide with each other, for the $Q_t(l_i^{(f)})$ calculation only the interference of the packets generated and forwarded within the same node has to be analyzed.

To calculate $Q_t(l_i^{(f)})$, the longest time $w_i(X)$ the i-th link has to wait to see X consecutive start times is calculated as

$$w_i(X) =$$
$$= \begin{cases} NL(i) > 0 & \Rightarrow (S+1)CT(i) - (Xci - 1 - O)Tci \\ NL(i) = 0 & \Rightarrow X \times Tci \end{cases} . \tag{3.5}$$

where S is quotient of the Euclidean division of $X-1$ by Xci and O is the remainder, i.e., $X-1 = SXci + O$. For instance, in the case of Figures 3.10 and 3.11, the link sl_{21} has $Xci = 2$ and $CT(sl_{21}) = 12Tci$, so for $X = 3$ we have $S = 1$ and $O = 0$. Therefore, the link sl_{21} has to wait for $w_{sl_{21}}(3) = 2CT - (2-1)Tci = 23Tci$. In fact, looking at the example in Fig. 3.10, if a packet to be transmitted in sl_{21} arrives when the second CI has just started, it will be transmitted after $11Tci$, i.e., $CT(sl_{21}) - (Xci - 1)Tci$.

To apply the busy period approach we first calculate the number of packets of the f-th flow generated in any interval of length t, i.e.,

$$I_f(t) = \left\lceil \frac{t}{P_f} \right\rceil. \tag{3.6}$$

Then, we calculate the largest number (X_j^f) of start times that are needed to transmit the i-th packet, which is given by the smallest value of X_j^f that satisfies the following equation

$$X_j^f = 1 + \sum_{\mathsf{prio}(k)>\mathsf{prio}(j)} I_k(w_i(X_j^f)) + \sum_{\substack{\mathsf{prio}(k)=\mathsf{prio}(j),\\ k\neq j}} 1. \tag{3.7}$$

In equation (3.7), 1 is the start time required to transmit the packet, while the parameter X_j encompasses the following sources of interference:

1. The k-th higher priority packets (i.e., $\mathsf{prio}(k) > \mathsf{prio}(j)$) that compete with the j-th packet for the transmission in the i-th link.

2. The packets with the same priority as the j-th packet that compete with it for the transmission in the i-th link (in this case a FIFO policy is adopted).

Equation (3.7) has not simple solution, as the X_j term appears both in the Left-hand-side (LHS) and in the Right-hand-side (RHS) of the equation under the ceiling operator. Thus, the calculation of X_j is performed by the following iterations

$$\begin{cases} X_j^{f^{(0)}} & = 1 \\ X_j^{f^{(v+1)}} & = 1 + \sum\limits_{\mathsf{prio}(k)>\mathsf{prio}(j)} I_k(w_i(X_j^{f^{(v)}})) \\ & \quad + \sum\limits_{\substack{\mathsf{prio}(k)=\mathsf{prio}(j),\\ k\neq j}} 1 \end{cases} . \tag{3.8}$$

Iteration starts with $X_j^{f^{(v)}} = 1$, with $v = 0$, as the packet to be transmitted requires one start time. Then $X_j^{f^{(v)}}$ is iteratively calculated until the LHS is equal to the RHS, i.e., until the interference stops growing. Equation (3.8) is proved to converge if the number of start times required to transmit every packet on the i-th link is lower than or equal to the number of start times available for the i-th link [72].

Once the X_j^f value has been found, the worst-case $Q_t(l_i^{(f)})$ is calculated as

$$Q_t(l_i^{(f)})_{\mathsf{worst}} = w_i(X_j^f). \tag{3.9}$$

3.5 Implementation of the MRT-BLE protocol on COTS devices

In order to show the feasibility of the proposed protocol on COTS hardware, I implemented it on the X-NUCLEO-IDB05A1 [75] devices, i.e., a Bluetooth Low Energy evaluation board produced by STMicroelectronics. These devices are equipped with communication modules that are compliant with the Bluetooth Specifications v4.1, i.e., the SPBTLE-RF BlueNRG-MS ones. The X-NUCLEO-IDB05A1 allows expansion of the STM32 Nucleo boards and interfaces with the STM32 microcontroller via the SPI pin.

The testbed is composed of eight devices, as shown in Fig. 3.12.

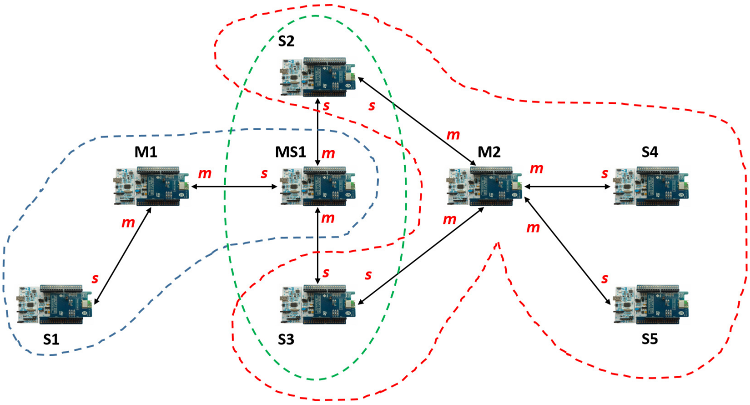

Figure 3.12: The implemented topology.

Each device acts as both a GATT client and a GATT server. Bidirectional communication consists in transmissions initiated by the GATT servers, which update their exposed attributes and send them through notifications. In the implemented topology, there are three linked sub-networks and the devices S2, S3 and MS1 are shared nodes. As the masters of the linked sub-networks are not synchronized, connection events may overlap at the shared nodes. The mechanism described in Section 3.3 starts after the node MS1 has received a synchronization packet from node M1.

Any time the shared slaves, i.e., S2 and S3, note (following a GATT event) that MS1 has modified one of their exposed attributes (the descriptor of one of their characteristics), they will enable or disable the connections with M2. In particular, if MS1 has enabled the connections with S2 and S3, both of them will disable the connections with M2. Conversely, if MS1 has disabled the connections with S2 and S3, both of them will enable the connections with M2.

Fig. 3.13 shows the timing of the masters M1, MS1 and M2 in the topology in Fig. 3.12. The semi-transparent rectangles represent the timeslices during which the depicted connections are disabled.

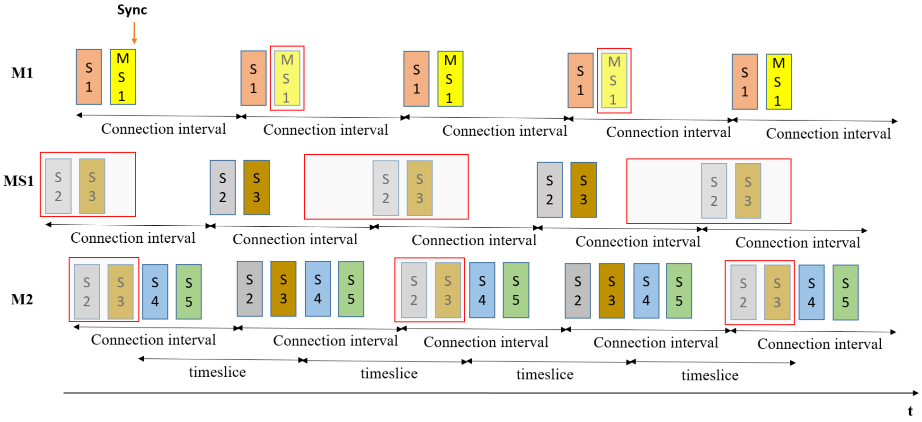

Figure 3.13: Timing of the masters.

Fig. 3.13 shows that the node MS1 in the initial phase enables only the connection with the master M1. Once the sync packet, sent from node M1, is received by the node MS1, the mechanism of enabling/disabling the connections starts. Each device implements as many priority queues as the number of established connections managed by the MRT-BLE sublayer. Each priority queue contains the outbound packets for a specific connection. The insertion into a given queue occurs, using a static routing function, when a packet is generated or when a node receives a packet to forward. A backup path can be provided for each flow. The priority of packets can be assigned according to a configurable criterion. In this scenario I dynamically assigned the priority of packets according to the hop count, i.e., the number of hops that the packet has traversed. This way, I can favor the packets that have to traverse more links to reach the destination, thus avoiding to excessively increase their end-to-end delay. For example, on the node MS1, a packet with source S4 (two hops) has a higher priority than a packet with source M2 (one hop). When, on a device acting as a GATT server, a GATT event reports that a connection has been enabled, the device is allowed to send one packet for each enabled connection every connection interval. Before sending a packet, each device has to check which connection is enabled and then transmits the highest priority packet from the queue of the enabled connection.

The realized testbed, shown in Fig. 3.14, consists of seven sensors (M1, MS1, S2, S3, M2, S4 and S5) and one sink (S1) that collects the data from all the sensors. My implementation does not include the application layer of the sink that receives and processes the received data.

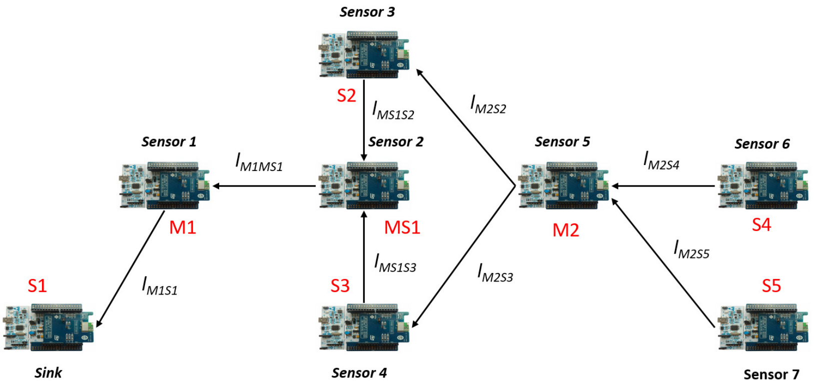

Figure 3.14: The considered scenario.

The arrows in Fig. 3.14 show the direction of the data transmission flows. As it can be seen, the nodes M1 and MS1 are a potential bottleneck in the realized mesh network.

In the described scenario, the sensors generate a new packet every second. Furthermore, the used devices permit to set the maximum duration of each connection event (Tce). For each connection, the duration of the connection interval Tci was set to 30ms and the Tce was set to 5ms to allow up to five master connections and one advertising or scanning interval (they are mutually exclusive in a CI). In each connection interval, the connected devices exchange only one packet in their CE. The Tce is sized so as to take into account the time to transmit one packet and any possible Link Layer retransmission.

On the final destination (i.e., the actuator S1) a specific check is implemented to verify that no packet is lost.

The following subsections present the results of the measurements performed to assess the feasibility of the MRT-BLE approach on COTS devices and to analyze the application level end-to-end delay experienced by multi-hop real-time data exchanges.

3.5.1 Packet end-to-end delay

This subsection presents the results of experimental measurements of the packet end-to-end (e2e) delay as a function of the number of hops from the source to the final destination. Note that, in the following, the hop count is obtained summing up the number of traversed links and the node S1 is assumed to be the final destination for all the packets.

The relevant network parameters are summarized in Table 3.2. The Tce parameter was set to accommodate the transmission of a packet plus the potential retransmissions, while Tci is equal to the duration of five connection events plus the scanning interval. As the calculated $Tsw(i)$ for enabling/disabling the connection is really high (e.g., the Tsw for the link sl_{MS1S2} in Fig. 3.14 is 180ms according Eq. (3.1)), Xci was set to 4 to reduce the network end-to-end delay.

Table 3.2: Testbed parameters

Parameter	**Value**
Tci	30 ms
Tce	5 ms
Xci	4

In fact, as S2, S3, S4, S5, M2, and MS1 generate one packet with destination S1 every second, the node MS1 should forward six packets per second to M1 (the only device connected with S1). Consequently, an overhead of 180ms for each packet would cause a large end-to-end delay increase. Setting Xci equal to 4 in this case significantly reduces the overhead of the proposed mechanism, i.e., 180ms every four packet transmissions.

I defined seven flows (one from each node, except the sink) generated with a period of 1s and with a 30-byte payload. The parameters of the flows are shown in Table 3.3. The routing paths determine which packets actually interfere with the packets originated by a given node. The interfence is due to the waiting times in the outbound queues of the intermediate nodes that the packets traverse to reach the final destination.

Table 3.3: Parameters of the flows

Flow ID	**Source**	**Period**	**Routing Path**
0	S5	1s	$l_{M2S5}, sl_{M2S3}, sl_{MS1S3}, sl_{M1MS1}, l_{M1S1}$
1	S4	1s	$l_{M2S4}, sl_{M2S2}, sl_{MS1S2}, sl_{M1MS1}, l_{M1S1}$
2	M2	1s	$sl_{M2S3}, sl_{MS1S3}, sl_{M1MS1}, l_{M1S1}$
3	S3	1s	$sl_{MS1S3}, sl_{M1MS1}, l_{M1S1}$
4	S2	1s	$sl_{MS1S2}, sl_{M1MS1}, l_{M1S1}$
5	MS1	1s	sl_{M1MS1}, l_{M1S1}
6	M1	1s	l_{M1S1}

The assessed metric here adopted is the packet end-to-end delay at the application layer, defined as the time difference between the packet generation time at the source and the reception time at the receiver, measured at the application layer. The end-to-end delay is measured by a third device, hard-wired to both the source and the destination nodes, as the difference between the packet reception time (i.e. when the receiver node sets up a pin) and the packet generation time (i.e., when the source node sets up a pin) .

According to the timing analysis in Section 3.4, the worst case response-time (WCRT) values are shown in Table 3.4. These values represent the upper bound of the expected end-to-end delay for each flow.

Table 3.4: WCRT values

Source	**WCRT value**
$M1$	240 ms
$MS1$	2010 ms
$S2$	2460 ms
$S3$	2490 ms
$M2$	1770 ms
$S4$	1710 ms
$S5$	1710 ms

Fig. 3.15 gives the end-to-end delay distribution for the packets generated by node M1 (i.e., single-hop), that shows the percentage of packets that experienced the plotted end-to-end delay values.

As it is shown in Fig. 3.15, every packet generated by the node M1 was delivered to the destination (S1) within 120ms. The measured delay is only due to the waiting time that a packet experiences in the queue before transmission. Note that the node M1 can transmit a packet every connection interval, as the connection with the node S1 is always enabled. The delay experienced by the packet with source M1 is a multiple of Tci (depending on the number of higher priority packets enqueued in the priority queue) plus the waiting time for the assigned Link Layer slot. If both the application queue and BLE queue are empty, the packet will be delivered to destination within 30ms (Tci).

The experimental results show that the path of these packets in the considered scenario (i.e., the one in Fig. 3.14) and the priority policy bound to three connection intervals the waiting time for the Link Layer transmission of the packets originating from M1. Conse-

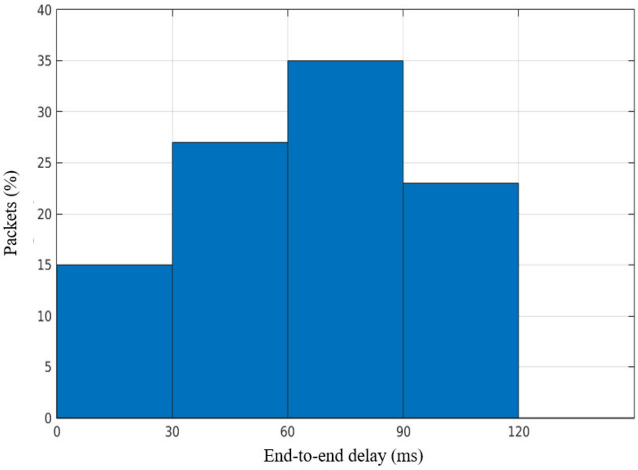

Figure 3.15: End-to-end delay distribution - node M1 (single hop).

quently, the end-to-end delay for these packets is lower than 120ms (i.e., 4*Tci).

Fig. 3.16 shows the end-to-end delay distribution for the packets originating from the node MS1 (i.e., two-hop packets).

Almost all these packets reach the final destination in less than 390ms. Fig. 3.16 shows that a low percentage of packets (lower than 4%) with source node MS1 reach the final destination within 420ms. This delay mainly consists of the waiting times in the queues due to the interference of higher priority packets, i.e., the packets originating from the other source nodes (i.e., all of them but M1), that accumulated more hops on their way.

Fig. 3.17 shows the end-to-end delay distribution for the packets with source node S3 (i.e., three-hop packets).

Almost all these packets (i.e., 95% of them) reach the final destination within 810ms. A very limited percentage of packets (about 1%) arrived to destination within about 1170ms.

Fig. 3.18 shows the end-to-end delay distribution for the packets with source node M2 (i.e., four-hop packets).

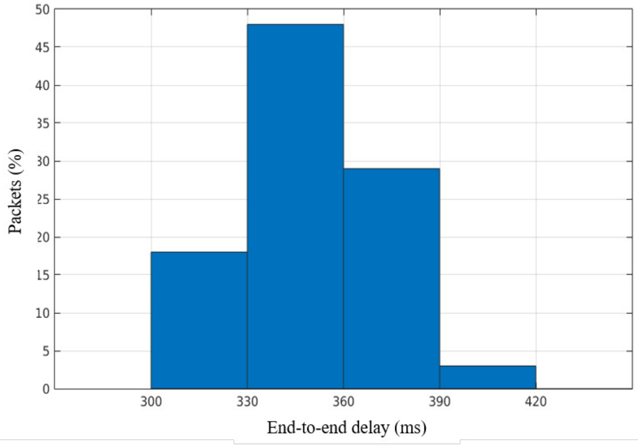

Figure 3.16: End-to-end delay distribution - node MS1 (two hops).

Almost all of the packets (about 95%) were delivered to the final destination within 1050ms. An end-to-end delay of 1580ms was experienced by a limited percentage of packets (lower than than 2%).

The end-to-end delay peak values shown in Figures 17 and 18 are due to the longer queuing delays that the packets generated by the nodes S3 and M2 experience due to the interference of higher priority packets. More specifically, the interfering packets for the ones generated by the node S3 are those originating from the nodes S5 and M2, while the interfering packets for the node M2 are the ones generated by node S5.

Fig. 3.19 shows the end-to-end delay distribution for the packets with source node S4. These packets experience a five-hop delay.

These packets arrived at destination within 1400ms. They experienced a waiting time in their source node lower than 30ms, as the node S4 is not a forwarding node in the considered scenario and the node M2 always keeps enabled the connections with node S4. Consequently, packet transmission occurs within a connection interval since the packet generation. Similar results were found for the packets gen-

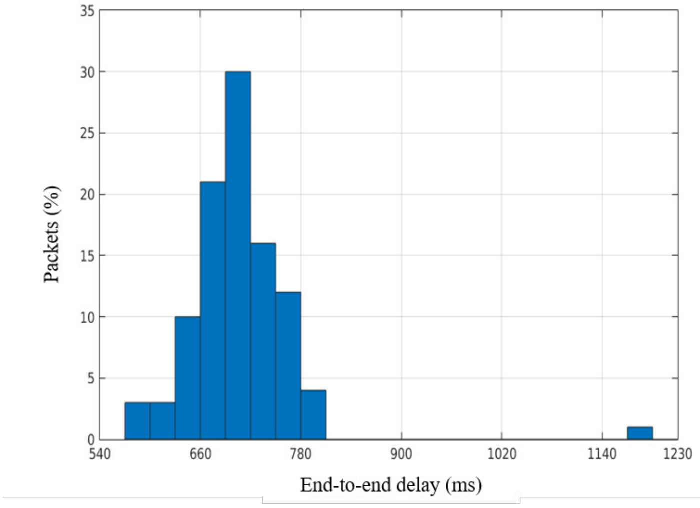

Figure 3.17: End-to-end delay distribution - node S3 (three hops).

erated by the node S5, that have the same hop count and thus the same priority (i.e., the highest in the network) as the packets generated by S4. This is in accordance with the analysis, that obtained the same worst-case response times for the packets originated by S4 and S5.

In all the considered cases, the measured end-to-end delays are always lower than the worst case response-time values shown in Table 3.4, as the analysis is safe.

Fig. 3.20 summarizes the cumulative distribution of the packet end-to-end delay versus the source node and, therefore, versus the number of hops between the source and the final destination.

Fig. 3.20 shows that the end-to-end delays of the packets that have the same hop count to the final destination, for example, the nodes S2 and S3, are centered around a given range of values. In fact, almost all the packets generated by the nodes S2 and S3 reach the destination within about 700 ms. Consequently, the cumulative distributions grow fast to reach a specific range of end-to-end delay val-

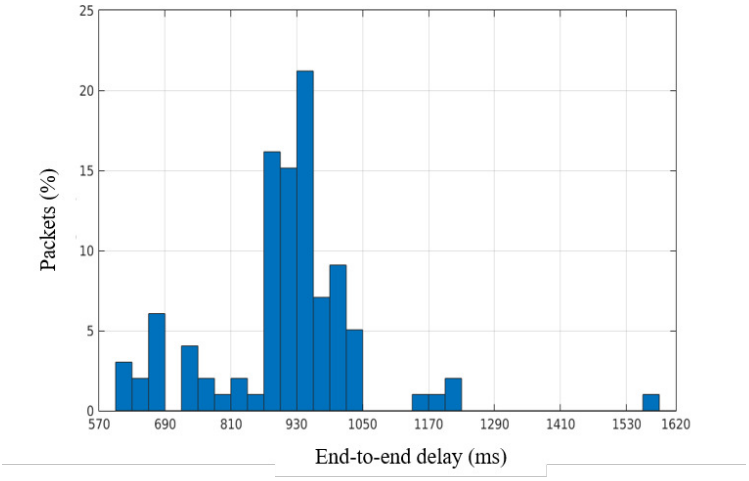

Figure 3.18: End-to-end delay distribution - node M2 (four hops).

ues. This consideration is especially true for the packets that at most experience a three-hop delay. For example, the cumulative distribution of end-to-end delay for the packets generated by MS1 grow very fast around 330-360 ms. When the number of hops from the source to the final destination increases, more factors affect the end-to-end delay, therefore some packets will experience a higher variable delay and the cumulative distributions grow more slowly. For instance, the packets generated by the node S4 experience an end-to-end delays distributed between 510 ms and 1410 ms.

3.6 Conclusions

This chapter proposes the MRT-BLE, a protocol working on top of BLE that provides for bounded delays over IWSN with mesh topologies. The chapter addressed in detail the MRT-BLE protocol and its configuration and implementation on COTS devices. To assess the performance of the protocol, a worst-case timing analysis is also

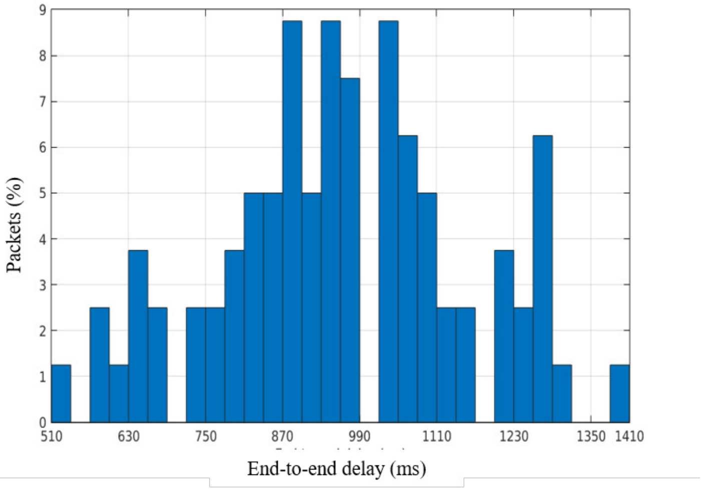

Figure 3.19: End-to-end delay distribution - node S4 (five hops).

presented and its outcomes are compared with the packet end-to-end delays obtained in experiments on a real testbed. To the best of my knowledge, in the literature there are no other mechanisms supporting real-time communication over BLE mesh networks. Other approaches addressing mesh topologies over BLE (e.g., Mesh BLE [36]) work in a best-effort way, therefore, they can be more bandwidth-efficient and may also provide better average performance than MRT-BLE, but they cannot temporally isolate time-critical flows and cannot guarantee end-to-end delay bounds to individual flows. As a result, these approaches cannot cope with the real-time requirements of industrial communications. For this reason, no meaningful performance comparison between MRT-BLE and such approaches can be made for the purposes of this work. The MRT-BLE protocol is a connection-oriented protocol, as the main aim here was on ensuring a bounded packet delivery delay over multihop mesh IWSN. The protocol therefore foresees that the network topology is offline configured. This feature limits the applicability of MRT-BLE to IWSN

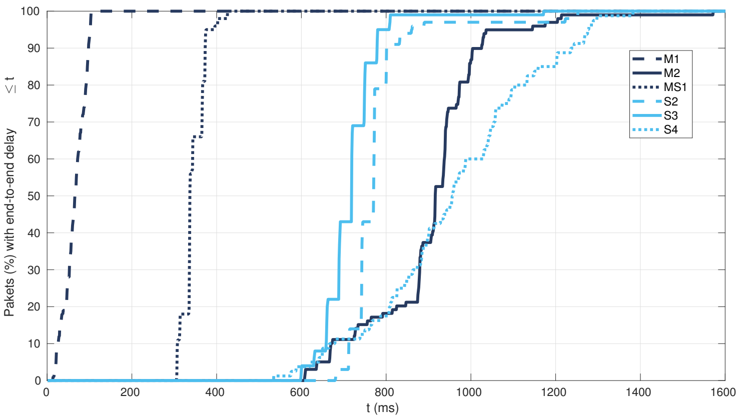

Figure 3.20: Cumulative distribution of the packet end-to-end delay.

that include only fixed nodes. Future works will therefore investigate a dynamic configuration mechanism to allow for both more flexibility and the free movement of nodes. Another line of investigation includes a dynamic topology management mechanism, as the one proposed in [76], and a stochastic response-time analysis able to take into account also aperiodic traffic flows.

Chapter 4

Industrial LoRa: a Novel Medium Access Strategy for LoRa in Industry 4.0 Applications

Industry 4.0 involves several enabling technologies (e.g., machine-to-machine communications, real-time industrial communications, big data analytics, etc.) to design, monitor and improve industrial processes for a more efficient and more sustainable production. In this context, wireless technologies play an important role. Industry 4.0 wireless applications typically require to cover large areas and transmit a small amount of data per node. Consequently, while bandwidth is not the main concern [77], reliability, bounded latency for real-time flows, and energy efficiency are the key performance indicators [78].

Recently, Low-Power Wide-Area (LPWA) networks are attracting attention because they offer connectivity to low-power devices distributed over very large geographical areas. LPWA networks represent a novel communication paradigm that will complement traditional cellular and short-range wireless technologies in several applications, e.g., smart cities, personal IoT applications, agriculture, and vehicular communications [79], [80], [81], in which LPWA networks would remove the need for multi-hop communications protocols, thus reducing the transmission latency and the network overhead.

There are several competing LPWA technologies today, which adopt various techniques to achieve long-range, low-power operation and high scalability. In this chapter I investigate LoRa, a long-range wireless communications system promoted by the LoRa Alliance. LoRa stands for "Long Range" and is intended for long-lived battery-powered devices, where the energy consumption is of paramount importance. The medium control access for LoRa is currently defined by LoRaWAN, a standardized MAC layer over LoRa. As LoRaWAN was designed for sporadic nontime-constrained communications among a relatively large number of nodes, it cannot support real-time industrial applications.

Several industrial applications can benefit from a centralized approach, in which transmission scheduling is up to a central node that determines the medium access according to a predefined order that provides real-time flows with bounded latencies.

This work therefore proposes Industrial LoRa, a LoRa-based centralized MAC scheme that allows the use of LoRa technology in Industry 4.0 real-time applications.

The chapter is organized as follows. Section 4.1 summarizes related works, while Section 4.2 outlines the LoRa Physical layer, the LoRaWAN MAC layer, and the constraints for a LoRa device operating in the EU863-870MHz ISM Band. Section 4.3 presents the Industrial LoRa design, while Section 4.4 addresses a simulative assessment of Industrial LoRa and discusses the results obtained. Finally, Sec. 4.5 gives my conclusions and hints for future works.

4.1 Related Work

LoRaWAN [82] is the most adopted among LPWA technologies [83] thanks to its low-cost and low-power properties. The paper [83] provides an overview of the capabilities and the limitations of LoRaWAN. The work in [84] analyses the LoRaWAN functional components and evaluates the physical and data link layer performance through field tests and simulations. The work in [31] addresses a simulative performance assessment to evaluate the behavior of LoRaWAN industrial monitoring applications. Simulations are performed with multiple SF assignment strategies and different networks sizes. The results showed the LoRaWAN suitability for this kind of applications.

LoRa is gaining ground in specific sectors, such as, smart cities [85] [86], smart metering [38], health monitoring [87], fleet/goods tracking, energy management systems [88] and also in other monitoring applications in which other technologies, e.g., the IEEE 802.15.4 standard, proved to be effective and efficient too [51].

Recently, low-power low-cost communication technologies that were not originally intended for industrial environments, such as, LoRa [77] and Bluetooth Low Energy [37] [89], have been successfully adapted for being used in industrial applications. In particular, the work in [77] investigates the use of LoRaWAN in wireless sensors and actuators network for Industry 4.0 applications and proposes a light modification of the LoRaWAN upper layer that applies a TSCH-like approach [90]. The proposed Industrial LoRa protocol defines a MAC strategy to support real-time and nonreal-time traffic, unlike the work in [77], which provides only the support for real-time flows. Moreover the approach proposed in this thesis schedules a period for downlink communications. Furthermore, in Industrial LoRa the confirmed messages are globally acknowledged with a single ack message while in [77] these messages are confirmed in their uplink reserved slot after the transmission, thus a longer slot must be set.

4.2 LoRa Overview

The LoRa Alliance specification [82] [91] defines two distinct layers, i.e., the LoRa Physical layer and the LoRaWAN MAC layer protocol, which are briefly summarized in the following Subsections.

4.2.1 LoRa Physical Layer

The LoRa is a physical layer radio modulation technique patented by Semtech [26] [92]. LoRa operates in the unlicensed sub-GHz ISM band and utilizes Chirp Spread Spectrum (CSS) modulation to encode an input signal into chirp pulses spread over a wide spectrum. This technique offers robustness to long-range transmissions.

Each LoRa transmission is characterized by several customizable parameters [93], such as the Spreading Factor (SF), the Coding Rate (CR) and the Bandwidth (BW). Tuning these parameters has an impact on the communication range, the bit rate, the robustness to interference or noise, the error-correction capability, and the ease of decoding. The available values for each parameter depend on the region where LoRa devices are deployed [91], as will be described in Subsection 4.2.3.

LoRa defines the spreading factor as the base-2 logarithm of the number of chirps per symbol [84]. The SF configuration allows to tune the bit rate and the covered distance. The higher the SF (i.e., the lower the bit rate), the longer the communication range, therefore a trade-off between data-rate and communication range is needed. As the choice of different SFs also enables orthogonal signals, a receiver can successfully receive distinct signals sent over a given channel at the same time.

LoRa adopts cyclic error coding to perform forward error detection and correction. The code rate is the forward error correction rate and it affects the Time on Air (ToA) of packet transmissions. Error coding causes a transmission overhead (i.e., extra bits in the LoRa PHY layer payload) that depends on the CR parameter. The latter can be set to one of the following values, 4/5, 4/6, 4/7, and 4/8.

Taking this into account, as well as the fact that SF bits of information are transmitted per symbol, Equation (4.1) allows to compute the useful bit rate (R_b).

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR \tag{4.1}$$

Although the LoRa modulation can be used to transmit arbitrary frames, a PHY frame format, shown in Fig.4.1, is specified and implemented in Semtech's transmitters and receivers.

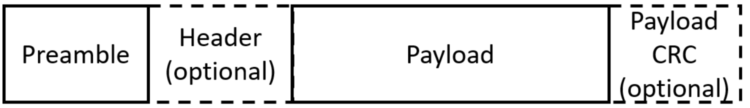

Figure 4.1: The LoRa physical frame structure.

After a preamble, there is an optional header that indicates the payload size (in bytes), the code rate used to mark the end of the transmission and the presence/absence of a 16-bit payload CRC. The payload size is limited to 255 bytes.

The work in [94] provides a mathematical model for calculating the Time on Air for transmitting data in a LoRa network.

4.2.2 LoRaWAN

LoRaWAN is standardized by the LoRa Alliance and defines a MAC layer protocol and a system architecture for networks using the LoRa physical layer. LoRaWAN provides a medium access control mechanism that enables multiple end-devices to communicate with a gateway using the LoRa modulation.

The LoRaWAN networks work over a *star-of-stars* topology, as shown in Fig. 4.2, in which communications between end-devices and a central network server occurs through gateway nodes that transparently relay messages. End-devices send data to gateways over a single wireless hop and gateways are connected to the network server through a non-LoRaWAN network. Communication is bidirectional. Uplink communication from end-devices to the network server is strongly favored.

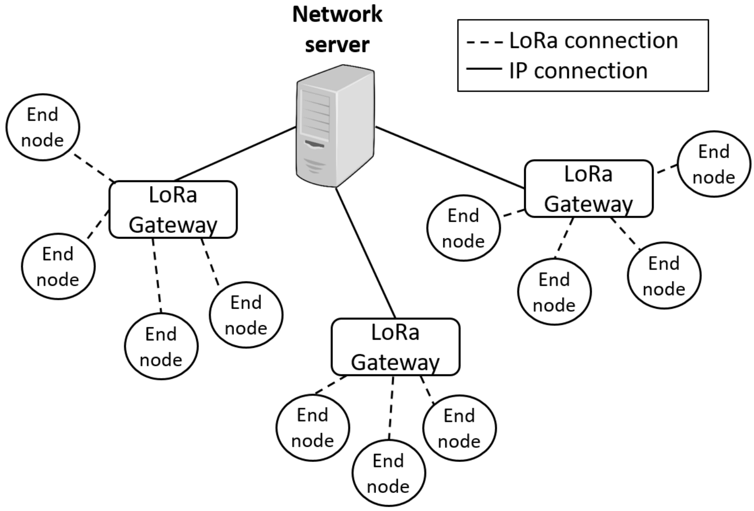

Figure 4.2: The LoRaWAN star-of-star topology.

LoRaWAN defines three device classes, i.e., Class A, B and C, with different capabilities [91]. Class-A nodes have the lowest power consumption and use Pure ALOHA access for the uplink. Downlink transmission is only allowed after a successful uplink transmission. Class-B nodes allow the schedule of additional receive windows for downlink traffic without prior successful uplink transmissions. Finally, Class C devices continually listen to the channel except when they are transmitting.

All LoRa messages carry a PHY payload (i.e, the MAC frame) starting with a single-octet MAC header (MHDR), followed by a MAC payload (MACPayload), and ending with a 4-octet message integrity code (MIC). The MAC payload contains a frame header (FHDR), followed by an optional port field (FPort) and an optional frame payload field (FRMPayload). The Fig. 4.3 shows the MAC frame format (the number under the field name indicates the field size in byte).

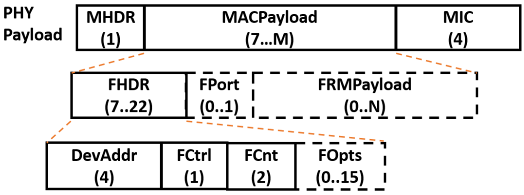

Figure 4.3: The LoRaWAN MAC frame structure.

The maximum length of the MACPayload and FRMPayload fields (i.e., respectively, M and N) is region-specific and is specified in the Subsection 4.2.3.

The LoRaWAN specification [82] provides more detail about the MAC frame.

4.2.3 Regional Parameters EU863-870MHz ISM Band

Due to the relatively low bit rate and operating in unlicensed bands, the LoRa behavior is mainly limited by the duty-cycle regulations. In Europe, operation in the 868 MHz region is normed by the ETSI regulations [EN300.220] [95], in the bands and sub-bands specified in the ERC Recommendation 70-03. The ETSI regulations impose some restrictions such as, the maximum time the transmitter can be on or the maximum time a transmitter can transmit per hour, and they allow to choose of using either a duty-cycle limitation or a so-called Listen Before Talk Adaptive Frequency Agility (LBT AFA) transmissions management. As described in the next Section, the Industrial LoRa approach proposed in this work, following the Lo-

RaWAN specifications, [91], uses duty-cycled limited transmissions to comply with the ETSI regulations.

The limitations on duty-cycle and end-device EIRP (TXPower) expressed in dBi in the EU863-870MHz ISM Band are shown in Table 5.3.

Table 4.1: EU863-870 duty cycle limitations

Sub-band	**Frequency Band**	***Maximum EIRP***	***Duty Cycle***
h1.4	868.00 - 868.60	14	1%
h1.5	868.70 - 869.20	14	0,1%
h1.6	869.40 - 869.65	27	10%
h1.7	869.70 - 870.00	14	1%

Note that the four sub-bands do not correspond to four channels, e.g., the sub-band h1.4 is divided into three channels, each one with a bandwidth of 125 KHz.

The devices working in the EU863-870Mhz ISM band must use a preamble of 8 symbols. The size (in bytes) of the maximum MAC payload (M) and the maximum application payload (N) in the absence of the optional FOpt control field derives from limitations of the PHY layer depending on the effective modulation rate used. Table 4.2 shows the physical bit rate, and the M and N values that are relevant to the LoRa configuration.

Table 4.2: EU863-870 Physical bit rate

LoRa configuration	**Indicative physical bit rate (bit/s)**	***M***	***N***
SF12 / 125KHz	250	59	51
SF11 / 125KHz	440	59	51
SF10 / 125KHz	980	59	51
SF9 / 125KHz	1760	123	115
SF8 / 125KHz	3125	250	242
SF7 / 125KHz	5470	250	242
SF7 / 250KHz	11000	250	242

4.3 Industrial LoRa design

The proposed MAC scheme works with a star topology, as shown in Fig. 4.4, where a sink node collects data from end-devices and acts as the network coordinator synchronizing the nodes with a periodic beacon. The sink node can also communicate data/commands to the end nodes (ENs).

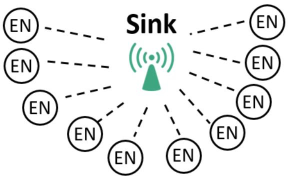

Figure 4.4: The network topology.

The Industrial LoRa protocol, here proposed, provides a medium access mechanism able to support periodic real-time traffic and aperiodic traffic. In Industrial LoRa the network time is organized in cyclically scheduled superframes. The superframe is composed of five sections, as shown in Fig. 5.1: a)The beacon section; b)The Contention Access Period (CAP); c)The Contention-Free Period (CFP); d)The Downlink Period; e)The CFP Ack section.

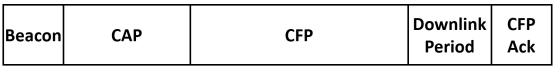

Figure 4.5: The structure of superframe.

Each superframe starts with a beacon section. During the beacon section a beacon is transmitted in broadcast by the sink. The beacon synchronizes all end nodes in the network and communicates the start of the superframe. After the beacon section, the CAP starts. During the CAP, non-periodic unconfirmed data can be sent from the end nodes to the sink. The end nodes compete for channel access using the Pure ALOHA mechanism. When an end node has a message to transmit, it generates the transmission parameter pair (channel, spreading factor). The message is transmitted on the selected channel and with the selected spreading factor after a random time delay. Each CAP messages transmission can start as long as it can be completed within the CAP. To meet the duty cycle restrictions, the generated transmission parameter pair is considered valid only after a local duty cycle check function verifies the transmission feasibility according to the duty cycle restrictions in each used sub-band (Table 5.3). The duty cycle check function takes also into account the periodic guaranteed communications, as a portion of the duty cycle is reserved for CFP transmissions. In the mechanism here proposed,during the CAP the channel and the spreading factor are randomly chosen to reduce the collision probability. However, I am aware that the distance between a specific node and the sink can be used as an indicator to choose the the best SF values when a node generates the transmission parameter pair (channel, spreading factor). Such an aspect is out of the scope of this chapter and will be investigated in future works.

When the CAP finishes, the CFP starts. The CFP is devised for periodic real-time confirmed flows and exploits a Multi-Channel and Multi-SF Time Division Multiple Access (TDMA) to access the channel. This superframe section is made up of a set of timeslots, which are used for unidirectional communications from the end nodes to the sink. Timeslots are characterized by the assignments of the triplet (channel, spreading factor, timeslot) to an end node. This way, once fixed the set of used channels and the set of used spreading factors, it is possible to assign the duration of the CFP so as to provide the required number of guaranteed communications within a superframe. Note that, the transmissions with different spreading factors are made with orthogonal signals, so the sink is able to receive multiple messages transmitted over the same channel at the same time.

In each CFP timeslot a node is allowed to transmit messages with the maximum-sized application payload, i.e. an application parameter compliant with the regional parameters of a specific ISM band (Table 4.2). The timeslot duration is fixed depending on the maximum size of the physical layer payload (closely related to the fixed maximum size of the application layer payload) and the physical bit rate. As a consequence, timeslots that schedule transmissions with different SF will have a different duration. In fact, the SF affects the bit rate and thus the time required to transmit a maximum-sized message. The CFP timeslots have to be assigned so as to provide guaranteed communications while maintaining the compliance with the duty cycle limitations. For the sake of simplicity, the proposed protocol does not define any mechanism to dynamically assign the timeslots to the network devices. Hence, the timeslots in the CFP are assigned offline. However, the proposed approach does not exclude the possibility to adopt a dynamic CFP timeslot assignment.

At the end of the CFP, the Downlink Period starts. The Downlink Period is devised for both confirmed and unconfirmed communications from the sink to the end nodes. A downlink confirmed transmission requires that an end node transmits an ack message immediately after receiving the message.

The CFP Ack section starts after the Downlink period. In the CFP Ack section one ack message (containing acknowledgments in the form of ack-bit array) is transmitted in broadcast by the sink to confirm the messages it received in the CFP.

The duration of the sections in the proposed superframe are configurable so as to achieve a trade-off between the resources allocated to periodic and aperiodic transmissions and the energy consumption. Moreover, the configuration of the superframe has to be made compliant with the duty cycle restrictions.

Industrial LoRa uses a frame format compliant with the LoRaWAN specifications. Consequently x bytes of the PHY payload correspond to (x-*12*) bytes of the application layer payload in the absence of the optional FPort and FOpts fields.

The contribution of this chapter consists in the Industrial LoRa protocol itself and the relevant simulative assessment to show the feasibility of the proposed approach. To the best of my knowledge, this is the first LoRa-based MAC scheme that provides support to different types of flows.

4.4 Simulative Assessment

In this section the simulative assessment of Industrial LoRa is described. The used simulator was developed using the OMNeT++ simulation environment. The FLoRa [96] (Framework for LoRa) framework was adopted for the wireless channel and the LoRa physical layer, while the MAC layer was developed from scratch. FLoRa is based on the OMNeT++ simulation environment and uses components from the INET framework.

The performance metrics used to evaluate Industrial LoRa are the Packet Loss Ratio (PLR) and the End-to-End Delay (E2E delay). The PLR is defined as the ratio between the number of lost messages and the overall number of messages transmitted by the sensor nodes. The PLR is measured at the application layer and is expressed in percentage. The E2E delay is defined as the time difference between the packet generation time at the source and the reception time at the receiver, measured at the application layer.

The simulations do not consider processing delays, as they are implementation-dependent.

4.4.1 Simulated scenario

The assessment refers to an industrial use case similar to the one proposed in [97], where a large number of nodes need to communicate with a sink.

Two different kinds of nodes are considered, i.e. stationary nodes (S_nodes) and mobile nodes (M_nodes). I took into account that 75% of the end nodes are M_nodes and 25% of end nodes are S_nodes.

The simulated network topology consists in a network of 101 nodes, i.e. a sink, 25 stationary nodes and 75 mobile nodes. The sink is located in the center of the sensing area to ensure the maximum coverage range. In order to have acceptable bit rate, in the simulations I only use values of spreading factor from 7 to 9 (i.e., a bit rate from 1760 bit/s to 5470 bit/s if bandwidth is equal to 125 kHz).

All the nodes are placed in the radio coverage of the sink, i.e. a range of 250 meters with spreading factor equal to 9. Table 4.3 shows an indicative maximum coverage range for messages sent with a specific SF.

Table 4.3: Simulation parameters

SF	***Indicative maximum coverage range***
7	125m
8	180m
9	250m

In the simulated scenario 50 nodes (i.e., 40 M_nodes and 10 S_nodes) are located within a distance of 125m from the sink with an SF equal to 7 for periodic transmissions, while 30 nodes (i.e., 20 M_nodes and 10 S_nodes) are located within a distance of 180m with an SF equal to 8 for periodic transmissions. Finally, 20 nodes (15 M_nodes and 5 S_nodes) are located within a distance of 250m with an SF equal to 9 for periodic transmissions.

All nodes (with the exception of the sink) simulate sensor nodes with the same communication requirements. The application of each sensor node generates a periodic message every 35s (e.g. the requirements of distributed measurement systems for industrial automation include an update rate, i.e. a cycle time, up to 60s [98]). The periodic messages are transmitted in reserved slots in the CFP. The workload of each node is compliant with the ETSI regulations [EN300.220] [95] without any restrictions on the choice of considered sub-bands. In fact, every device maintains the duty cycle less than or equal to the 1% (i.e., the minimum allowed duty cycle among the considered sub-bands).

Furthermore, the nodes generate aperiodic messages with a period that varies with an exponential distribution with a mean of 70s. These messages will be sent in the CAP using random physical parameters (among a set of allowed values) after a duty cycle check function that validates the chosen parameters.

The considered application payload is set to 38 bytes, as in industrial application a typical physical layer payload is 50-byte long [77], [97].

The superframe is set so as to assign at least one slot to each node and to provide a superframe duration shorter than the application message generation period.

The configured superframe includes:

- one beacon;
- one Contention Access Period (CAP) of 4.040s;
- a Contention-Free Period (CFP) of 6.060s with 525 slots. In the CFP there are 300 slots with SF equal to 7. 150 slots with SF equal to 8 and 75 slots with SF equal to 9;
- one downlink period of 3.232s in which there are 8 slots with SF equal to 9;
- one CFP Ack.

Note that, the sink uses SF equal to 9 for the transmission of both the beacon and the downlink period messages to guarantee the maximum coverage range. The duration of the slots in CFP is calculated according to the equation provided in [94], that suggests a lower bound for the slots duration. In particular, the slots with SF equal to 7 are 101ms long, while the slots with SF equal to 8 are 202ms long and the slots with SF equal to 9 are 404ms long.

The relevant simulation parameters are summarized in Table 5.8.

Table 4.4: Simulation parameters

Parameter	***Value***
Coding Rate	4/5
Bandwidth	125 KHz
Transmission Power	14 dBi
Spreading Factor	7 - 8 - 9
Sub-bands	h1.4 - h1.6 - h1.7
Physical Payload	50 bytes
Periodic messages generation period	35s
Superframe duration	17.5s

4.4.2 Simulation results

The simulation run was set to 3600s (i.e., an hour) in order to collect a significant number of data. In the simulations the LoRa log-normal shadowing propagation model (provided by the FLoRa framework) was adopted.

In the simulations no periodic messages are lost in the CFP. In fact, all the 102 messages (i.e., the amount of messages generated in a hour) that every node sends to the sink arrived to the destination, and the nodes received all the 102 acknowledgements for the messages transmitted in the CFP.

Fig. 4.6 shows the average PLR for the aperiodic messages transmitted in the CAP, grouped by their type (stationary or mobile) and their maximum distance from the sink.

The results in Fig. 4.6 show that most of the aperiodic messages are lost due to the Pure ALOHA medium access strategy used in the CAP. For the CAP transmissions, nodes uses random spreading factors (from 7 to 9 in the simulated scenario), thus the nodes closest to the sink present an average PLR slightly better than the other nodes, also if they choose low spreading factor. Conversely, the spreading factor has a high impact on the nodes that are located far away from the sink, as a low spreading factor entails low reliability on long distances. Mobile nodes obtained a better PLR than stationary nodes (250m) as they move, and on average their position is closer to the

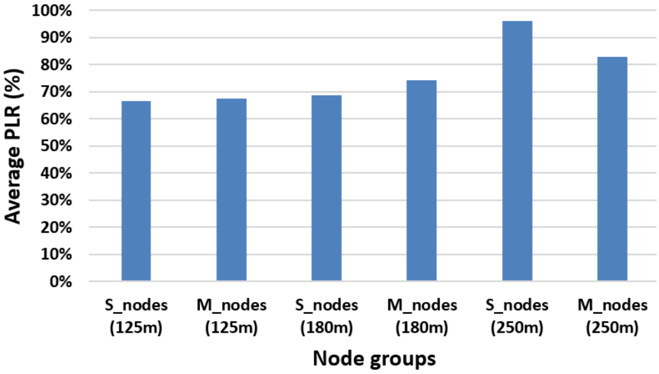

Figure 4.6: Average PLR for aperiodic messages.

sink than the stationary nodes located 250m far away from the sink.

The results relevant to the minimum and maximum end-to-end delays of the periodic messages transmitted in the CFP (grouped by spreading factor) are shown in Table 4.5.

Table 4.5: E2E delay

SF	*Min*	*Max*	*Avg*	*Std*
7	0.176s	17.575s	11.117s	7.109
8	0.256s	17.554s	9.668s	7.350
9	0.618s	17.714s	11.555	6.732

The end-to-end delay results confirmed the effectiveness of the proposed approach. In fact, as it was expected, the maximum application end-to-end delay that was obtained is always lower than 17.9s, that corresponds to the superframe duration plus the largest duration of one CFP timeslot (i.e., the one with SF equal to 9). The high variability of the end-to-end delay (i.e., from Min E2E delay to Max E2E delay) is due to the position of the CFP slots in the superframe. In fact, the application layer is not synchronized with the MAC layer. Hence for instance, a periodic message may be generated just after its transmission timeslot elapsed (worst case delay) or just before its transmission timeslot starts (best case delay). Anyway, this result proves that the latency of periodic communications is bounded, which represents the most important feature of the proposed approach.

4.5 Conclusions

This chapter proposed Industrial LoRa, a novel medium access strategy working on top of LoRa that provides support for both real-time and nonreal-time communications. Simulation results showed that Industrial LoRa guarantees periodic transmissions in the CFP with no collisions, high reliability (thanks to an accurate planning of the spreading factor parameters), and bounded end-to-end delays. Moreover, Industrial LoRa provides high scalability thanks to the support for nonreal-time aperiodic transmissions. However, the Pure ALOHA medium access strategy, that is used in the CAP, does not provide high reliability for aperiodic message transmissions. The reason is twofold. a) The ALOHA mechanism does not avoid collisions; b) the Spreading Factor selection algorithm does not take into account the signal quality. The CAP reliability can be improved defining a suitable spreading factor selection strategy. Future works will investigate strategies to increase the reliability for nonreal-time transmissions and other medium access strategies in the CAP. Furthermore, dynamic configuration mechanisms (e.g., a dynamical timeslot assignment mechanism) to allow more flexibility and scalability will be proposed.

Chapter 5

RT-LoRa: A Medium Access Strategy to support Real-time flows over LoRa-based networks for Industrial IoT applications

Low Power Wide Area (LPWA) networks, such as the ones based on the LoRa (Long Range) technology represent a novel communication paradigm that will replace or complement traditional cellular and short-range wireless technologies in several applications.

In the Internet of Things (IoT) field, LPWA networks are expected to offer energy-efficient connectivity to a high number of low power devices, distributed over very large geographical areas, that do not require to transmit a large amount of traffic [31]. LoRa offers notable properties, such as long range, low data rates and low energy consumption. Several IoT applications requiring those properties are found in smart cities, smart metering, fleet/goods tracking, security, and health monitoring [48] [99]. For this reason, industry and academia see LoRa as one of the "rising stars" of LPWAN-based IoT technologies [86] [100]. Moreover, LoRa is suitable for distributed measurement systems [98], i.e., typical IoT applications based on millions of sensors that collect data from the real world. LoRa is an attractive solution also for industrial IoT, thanks to its high robustness [101]. However, the LoRaWAN [82] [91] medium access control (MAC) protocol for LoRa-based networks[1] is intended for sporadic nontime-constrained communications between low-cost long-lived battery-powered devices. As a result, LoRaWAN adopts an ALOHA-based medium access protocol, that cannot provide bounded latency to the real-time flows typical of industrial applications. Conversely, a centralized approach, in which a central node manages the medium access according to a predefined order, would be more appropriate for the real-time flows generated from industrial IoT applications.

For this reason, Chapter 4 proposed Industrial LoRa [102], a centralized MAC protocol for star topologies working over LoRa that is able to support both real-time and nonreal-time communications for Industrial IoT applications. Chapter 4 (see Section 4.4) showed interesting simulation results obtained by Industrial LoRa in a realistic industrial scenario.

Motivation. However, Chapter 4 does not explore several aspects, i.e., the way to configure a generic network and to analytically derive the upper bounds on the message latency. Moreover, the Pure ALOHA-based medium access strategy for aperiodic messages does not provide sufficient reliability. The RT-LoRa protocol proposed in this chapter builds upon the Industrial LoRa [102] protocol proposed in Chapter 4, but it overcomes the limitations of Industrial LoRa through a number of extensions and improvements that also increase the communication reliability.

Contributions. This chapter contribution consists of the de-

[1] In this work, the term LoRa refers to the physical layer, that is based on a proprietary spread spectrum modulation scheme patented by Semtech, while the term LoRaWAN refers to the MAC layer, an open standard promoted by the LoRa Alliance.

tailed design of the RT-LoRa MAC scheme itself and the guidelines on setting the network parameters. The guidelines enable the network designer to correctly configure the superframe structure and assess whether a feasible schedule for a given set of flows (i.e., a schedule compliant with the application constraints) can be found. The chapter also provides a simulative performance assessment in terms of packet loss ratio and end-to-end delay.

Chapter overview. The chapter is organized as follows. Section 5.1 deals with background and related works. Section 5.2 outlines the LoRa technology and constraints. Section 5.3 presents the RT-LoRa design, while Section 5.4 provides guidelines for setting the superframe in a RT-LoRa network complying with the protocol restrictions. Section 5.5 addresses a simulative assessment of RT-LoRa and discusses the results obtained. Finally, Sec. 5.6 gives conclusions and hints for future works.

5.1 Background and Related Work

5.1.1 Background

Among the wireless communication technologies for low power IoT communication, two main categories can be identified, i.e., Low Power Local Area Networks (LPLANs) and Low Power Wide Area Networks (LPWANs) technologies, as discussed in [84].

The LPLANs include several technologies, such as IEEE 802.15.4 and Bluetooth Low Energy (BLE). They are suitable for short-range personal area networks, body area networks or for larger areas when a mesh topology is used [35, 37, 63, 89]. In particular, the IEEE 802.15.4 and BLE standards are widely used for wireless sensor networks. The IEEE 802.15.4 defines a physical (PHY) and a MAC layer used in several wireless protocol stacks (such as ZigBee and 6LoWPAN [44, 103, 104]), while BLE defines a complete protocol stack. BLE is mainly used in indoor human-oriented applications (home entertainment, health monitoring, personal security) [105] [106], while

IEEE 802.15.4 is adopted in industrial and environmental monitoring, security, and process automation [51, 90, 107].

The LPWANs can be seen as low power competitors of cellular networks, so they are suitable for long range applications where each "cell" covers thousands of end-devices. LPWANs include several technologies, such as LoRaWAN, SigFox and NB-IoT [108]. They provide very low data rates, but support large areas (up to several kilometers), e.g., a smart city. The power consumption of a long-range transceiver is similar to the one of an IEEE 802.15.4 transceiver [109]. Typical applications for long-range technologies are smart metering, smart grid and environmental monitoring.

LoRaWAN is considered one of the most successful LPWAN technologies [110] [83]. Comparing with other technologies, such as NB-IoT and Sigfox, LoRaWAN operates in the license-free spectrum, whereas NB-IoT uses licensed frequency bands. Moreover, LoRaWAN supports a higher bit rate than SigFox. LoRaWAN networks work over a star-of-stars topology and exploit a mechanism that enables multiple end devices (EDs) to communicate with a central network server through gateway nodes. The EDs send messages to gateways through a single-hop LoRa communication using an ALOHA-based MAC mechanism. In turn, the gateways relay data to the central network server. Such a relay operation raises the developer from the need to use a peer-to-peer protocol for IoT that performs routing over subnetworks in the application layer [111]. The LoRa physical layer allows multiple EDs to communicate simultaneously with the same gateway, using different spreading factors (SFs) and channels. The communication is bidirectional, but uplink transmissions from the end devices to the network server are strongly favoured. As it was previously mentioned, the LoRaWAN ALOHA-based MAC protocol is not able to support real-time communications. For this reason, this work proposes an alternative medium access protocol on top of LoRa. Further details about LoRaWAN can be found in [82].

5.1.2 Related work

Recent works have dealt with LoRaWAN [82] capabilities [83] [93], performance [84], and parameter setting [31] [112] for indoor industrial monitoring applications. For example, RS-LoRa [113] exploits a *lightweight scheduling* upon LoRaWAN, in which the gateway periodically sends some messages to the EDs, specifying communication parameters, such as the allowed transmission power on each channel. Next, each ED independently determines the channel and *time offset* to use for the deferred transmission. Although the probability of collision between the transmitting EDs is reduced exploiting different channels and other features, RS-LoRa is not able to support timeslots and time-bounded communications, as the time-offset is randomly determined.

In [114] a Slotted ALOHA approach for LoRaWAN is proposed, as an alternative to the classic ALOHA, to improve the packet loss and throughput performance. While the approach in [114] proposes a different way to rule transmissions in LoRaWAN, RT-LoRa is not LoRaWAN, but a different MAC protocol that, unlike LoRaWAN, is able to guarantee bandwidth to support real-time and non-real-time flows over LoRa. Conversely, the approach in [114] cannot provide bounded latency to real-time flows, as it does not allow to reserve bandwidth.

In [115] the use of a network synchronization and scheduling entity (NSSE) integrated in the LoRaWAN network server is proposed. Each ED synchronizes to the NSSE, sending a request that contains the traffic periodicity. In turn, the NSSE replies with a data structure that contains the timeslot assigned to the node for communicating with the gateway. However, in [115] the ED communication with the gateway uses the transaction model defined by LoRaWAN for Class A devices [82]. Consequently, the gateway is unable to start downlink communications before a message from the ED is received. In fact, class A devices open two receive windows (for downlink) after an uplink transmission.

On-Demand LoRa [109] is a MAC-layer, alternative to LoRaWAN, that uses two different TDMA strategies, called Unicast and Broadcast TDMA, respectively. Each ED is equipped with a low power transceiver, compliant to the Wake-up radio standard [116], that is normally maintained in deep listening state. With Unicast TDMA, the gateway sends a wake-up beacon to one specific ED that starts uplink communications. With Broadcast TDMA, the gateway sends a wake-up beacon to multiple EDs that start deferred uplink communications using scheduled timeslots. Although On-Demand LoRa is energy-efficient, it requires a non-standard LoRa transceiver.

In [77] a slightly modified version of the LoRaWAN upper layer that implements a TSCH-like approach was proposed to make the protocol suitable for industrial wireless networks. There are several differences between RT-LoRa and the approach in [77]. First, the TSCH-like approach works on top of LoRaWAN, while RT-LoRa works on top of LoRa as an alternative option to LoRaWAN. Second, the TSCH-like approach supports real-time flows only, while RT-LoRa both real-time and nonreal-time traffic. Third, the approach in [77] only supports uplink unconfirmed communications, whereas RT-LoRa supports uplink/downlink confirmed/unconfirmed communications, thus being more flexible. Moreover, in [77] each flow transmits only in one sub-band, as each flow is assigned one specific slot with a fixed channel. Conversely, RT-LoRa adopts a frequency rotation mechanism, according to which the slot assigned to each flow has a varying channel, so the flow transmissions occur on different sub-bands. This improves bandwidth exploitation while complying with the duty cycle constraints that are imposed on each sub-band by the regulations.

5.1.3 LoRa for industrial applications

Several recent works have addressed LoRa properties for industrial applications. For instance, in [117] the LoRa performance and noise robustness for a specific industrial application was assessed and the best configuration and trade-off between data rate and packet loss were determined. The work in [101] presents an accurate timing synchronization system for TDMA scheduling implemented on LoRa and shows some experimental results. The main idea consists of using the LoRa physical layer, but without the LoRaWAN MAC layer, in order to obtain a new protocol stack that maximizes the radio performance and complies with all the requirements of industrial IoT. Using this approach LoRa may become a candidate technology for low-bandwidth industrial IoT applications [97].

In particular, to the best of my knowledge, Industrial LoRa [102], described in Chapter 4, is the first work that proposed a mechanism to provide support for both real-time and nonreal-time communications over LoRa. In Industrial LoRa, the network access is organized in cyclically repeated superframes, consisting of five sections (see Section 4.3), i.e., the beacon section, the Contention Access Period (CAP), the Contention-Free Period (CFP), the Downlink Period, and the CFP Ack section. In the beacon section, a beacon synchronizes all the network nodes and indicates the start of the superframe. During the CAP, that is intended for nonreal-time communications, the end nodes compete for channel access using Pure ALOHA. The CFP, that is intended for periodic real-time flows, consists in a set of timeslots and exploits a Multi-Channel and Multi-Spreading Factor Time Division Multiple Access protocol. The Downlink Period is for the communications from the network sink to the end nodes. In the CFP Ack section, the sink broadcasts one acknowledgement message to confirm the messages received during the CFP. Compared to Industrial LoRa, the RT-LoRa protocol here proposed adds the following features:

- *Free mobility.* All mobile nodes can move within the coverage range of the sink without any restriction;
- *Smart mobile nodes.* An innovative mechanism based on multiple transmissions of the beacon frames by the sink allows the mobile nodes to know the Spreading Factor values that are rec-

ommended for transmission at a given time;

- *Smart Contention Access Period.* A novel MAC strategy, based on Slotted ALOHA with spreading factor selection strategy for aperiodic nonreal-time transmissions, enables more reliable transmissions in the CAP;

- *QoS classes for the flows of mobile nodes.* Three classes of Quality of Service (QoS) for the flows of mobile nodes allow to choose the desired trade-off between reliability and energy consumption on the basis of the flow requirements;

- *Frequency rotation for CFP slots.* Each node that schedules a periodic real-time flow and has a timeslot assigned in the Contention-Free Period performs a frequency rotation at every superframe. This mechanism improves the communication robustness and bandwidth efficiency, while complying with the duty cycle restrictions.

5.2 LoRa Overview

5.2.1 LoRa Physical Layer

The LoRa physical layer [26] enables long-range, low power communications in the unlicensed sub-GHz ISM band and exploits the Chirp Spread Spectrum (CSS) technique in order to offer robustness to the transmissions. A typical LoRa radio is characterized by some customizable parameters [93], such as Spreading Factor (SF), Coding Rate (CR), and Bandwidth (BW). The values allowed for these parameters depend on the region where the LoRa devices are deployed [91].

The *spreading factor* is the base-2 logarithm of the number of chirps per symbol [84], therefore a LoRa symbol, composed of 2^{SF} chirps, can encode SF bits of information. The SF configuration allows to tune the bit rate, covered distance, and energy consumption. A higher spreading factor increases the Signal to Noise Ratio (SNR) and therefore the sensitivity and coverage range, but it also increases the Time on Air (ToA), i.e., the transmission duration of a packet. In particular, each SF increase approximately halves the transmission rate and, hence, it approximately doubles the ToA and energy consumption. Consequently, a trade-off between bit rate and communication range is needed. According to several works in the literature, such as [26, 118] and [100], radio communications with different SFs are orthogonal to each other, so a receiver can successfully receive distinct signals that are sent over a given channel at the same time using different SFs. Recently, the work in [119] demonstrated the quasi-orthogonality of the SFs through both simulations and an implementation based on the universal software radio peripheral (USRP). Consequently, transmissions with different SFs are not completely immune to the adjacent SFs. However, the messages simultaneously transmitted on the same channel with different SFs can be correctly decoded when the Signal-to Interference Ratio (SIR) of the received packet is above the isolation threshold (see Tables I and II in [119]).

The *coding rate* is the forward error correction (FEC) rate used by the LoRa receiver to improve the robustness against interference. It can be set to $4/(4+Z)$, with $Z \in \{1, 2, 3, 4\}$. A higher value of Z offers more robustness, but increases the ToA, as a better error coding introduces a higher transmission overhead (i.e., extra bits in the payload of the LoRa frame).

The *bandwidth* represents the range of frequencies in the transmission band. A higher bandwidth gives a higher bit rate (thus, a shorter ToA), but a lower sensitivity, due to additional noise. LoRa networks typically operate at 500 kHz, 250 kHz or 125 kHz.

LoRa modulation can transmit arbitrary frames. Semtech's transmitters and receivers use a physical frame format that includes a preamble, an optional header, a payload (limited to 255 bytes), and an optional payload CRC.

The work in [94] provides Eq. (5.1) to calculate the Time on Air for the transmission of messages using the LoRa modulation. In Eq. (5.1) SF, BW and CR are the Spreading Factor, Bandwidth and Coding Rate, respectively. The other notation used in Eq. (5.1) is summarized in Table 5.1.

$$
\begin{aligned}
ToA &= \frac{2^{SF}}{BW}(NP + 4.25 + SW + max(\mathcal{H}, 0)) \qquad (5.1)\\
\mathcal{H} &= \left\lceil \frac{8PL - 4SF + 28 + 16CRC - 20IH}{4(SF - 2DE)} \right\rceil (Z + 4)
\end{aligned}
$$

Table 5.1: Notations used for the ToA calculation in Eq. (1)

Symbol	**Definition**
Z	Value of the parameter Z being $CR = \frac{4}{4+Z}$.
NP	Number of preamble symbols.
SW	Length of synchronization word.
PL	Number of PHY payload bytes.
CRC	CRC Presence (1=yes ; 0=no).
IH	PHY header Presence (1=no ; 0=yes).
DE	Use of data rate optimization (1=enabled ; 0=disabled).

For example, the transmission of a message with a physical payload of 50 bytes, SF=7 and BW=125kHz, takes about $97 \cdot 10^{-3}$ s, using the parameter values shown in Table 5.2.

Table 5.2: Parameters for the ToA calculation

Parameter	Z	NP	SW	PL	CRC	IH	DE
Value	1	8	8	50	1	0	0

5.2.2 Regional Parameters EU863-870MHz ISM Band

ETSI regulations impose to limit the LoRa transmitter activities using either a duty cycle limitation or the so-called Listen Before Talk Adaptive Frequency Agility (LBT AFA) transmission management. The proposed RT-LoRa uses duty-cycle limited transmissions to comply with the ETSI regulations. The maximum duty cycle is defined as the maximum percentage of time during which a transmitter node can occupy a sub-band per hour. Table 5.3 shows the limitations on duty cycle (DC) and maximum transmission power (expressed in dBm) in the EU863-870MHz ISM Band. The number of available LoRa channels (n_{CH}) that are reported in Table 5.3 refers to channels with bandwidth of 125kHz (every channel needs at least 200kHz considering some guard band [120]).

Table 5.3: EU863-870MHz ISM Band limitations relevant to this chapter

Sub-band	***Frequency Band (MHz)***	***Available Channels***	***Maximum TX Power (dBm)***	***Duty Cycle***
h1.4	868.00 - 868.60	3	14	1%
h1.5	868.70 - 869.20	2	14	0.1%
h1.6	869.40 - 869.65	1	27	10%
h1.7	869.70 - 870.00	1	14	1%

The four sub-bands do not correspond to four channels, e.g., the sub-band h1.4 is divided into three channels, each one with a bandwidth of 125 KHz.

Table 5.4 shows the physical bit rates that are relevant to the configurations used in the assessment.

5.3 RT-LoRa design

RT-LoRa uses a star topology in which the end nodes and the sink communicate through bidirectional links. The sink synchronizes the

Table 5.4: EU863-870 Physical bit rates relevant to this chapter

LoRa configuration	**Indicative physical bit rate (bit/s)**
SF9 / 125kHz	1760
SF8 / 125kHz	3125
SF7 / 125kHz	5470

end nodes with periodic beacons, collects data from the entire network and sends data/commands to the end nodes when needed. Two different types of end nodes are supported, i.e, stationary nodes (SN), that are fixed in the sensing area, and mobile nodes (MN), that move in the sensing area. The sink is a stationary node located approximately in the center of the sensing area. Each node can send data belonging to one or more periodic real-time flows.

The centralized medium access mechanism proposed in this chapter supports both periodic real-time and aperiodic nonreal-time traffic. In RT-LoRa, the network time is organized into superframes that cyclically repeat. Each superframe is composed of five main sections: *Beacon*, *Contention Access Period (CAP)*, *Contention-Free Period (CFP)*, *Downlink*, and *CFP Ack*. Fig. 5.1 shows the superframe structure (the size of the sections is not in scale for graphic reasons).

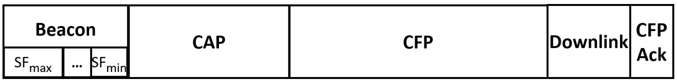

Figure 5.1: The superframe structure.

The CAP and CFP contain several *timeslot sets*, where each timeslot is a conventionally defined time interval in the schedule. The timeslots are scheduled over different channels and spreading factors. Messages that are sent on the same channel with different SFs do not collide, as in my design I make the assumption commonly found in the relevant literature (e.g., in [77]) that the transmissions performed using different spreading factors are orthogonal.

To easily describe RT-LoRa, I assume that the maximum size of the messages is set by the application. The timeslot duration is set as a function of the maximum message size and of the physical bit rate, so as to ensure that a maximum-sized message can be transmitted in each timeslot. Consequently, the timeslots that schedule transmissions with different SF values have a different duration, as the SF affects both the bit rate and the ToA. The notation τ_{SF_x} indicates a timeslot for transmissions with the SF value set to x, while τ_{s_x} is a set that comprises all the timeslots in which transmissions take place with $SF = x$. The duration of the timeslots is calculated according to Eq. (5.1), that imposes a lower bound on the timeslot duration. For the sake of simplicity, we start calculating the duration of the timeslots belonging to the first set (τ_{s_0}), assuming that the transmissions are performed with the minimum SF value that is allowed by the application ($SF_{\tau_{s_0}} = SF_{min}$). Then, we set the duration of a second set of timeslots (τ_{s_1}) for transmissions with the next SF value ($SF_{\tau_{s_1}} = 1 + SF_{min}$). The operation is repeated for each of the n allowed SF values, thus defining n timeslot sets, where the following condition holds: ($SF_{\tau_{s_{i+1}}} = 1 + SF_{\tau_{s_i}}$). As each SF increase approximately doubles the ToA, the duration of $\tau_{s_{i+1}}$ is twice the duration of τ_{s_i}, and so on. This way, the timeslots durations are multiple of each other.

The RT-LoRa design assumes that the number n of SF values allowed by the application is less than or equal to the number of used sub-bands. I suggest to set the duration of the CAP and CFP equal to a multiple of the maximum-sized timeslot duration. For example, let us consider an application that allows the SF values $s \in \{7; 8; 9\}$ and transmits messages using the parameters shown in Table 5.2 on channels with bandwidth of 125kHz. In this case, the duration of τ_{s_0} should be set to 0.101s, i.e., the 0.097s value obtained by Eq. (5.1) plus 0.004s, as suggested in [120]. Then, the duration of τ_{s_1} and τ_{s_2} should be equal to 0.202s and 0.404s, respectively.

Fig. 5.2 shows the structure of CFP/CAP for an application scenario that allows 3 SF values and uses 3 channels. In Fig. 5.2, τ_{SF_x}

indicates the timeslots belonging to the the first set (i.e., τ_{s_0}), therefore, $\tau_{SF_{x+1}}$ corresponds to τ_{s_1}, while $\tau_{SF_{x+2}}$ corresponds to τ_{s_2}.

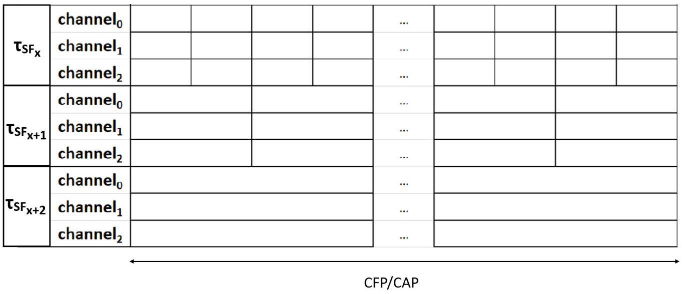

Figure 5.2: The CFP/CAP structure.

The following subsections provide details about each superframe section.

5.3.1 Beacon section

The first section of a superframe is reserved to beacon transmissions. Differently from Industrial LoRa [102] (see Section 4.3), in which the Beacon section is only for synchronization purposes and only one beacon is broadcast by the sink (using the highest SF among the values allowed by the application), RT-LoRa proposes a novel approach based on multiple beacon transmissions. Given a set of n SF values allowed by the application requirements, these are used by the sink to broadcast n beacons at the beginning of each superframe. Each node updates a dynamic list of recommended SF values according to the SFs of the beacons received during the current superframe. The list is called the $l_{SF}^{(a)}$ list. The update period of the list is equal to the superframe duration. If no beacons are received during the Beacon section of a superframe, the list will be automatically updated with the highest available SF value allowed by the application. The proposed multiple beacon strategy provides several advantages to mobile nodes. In fact, the nodes are synchronized to the network time and they are able to determine the SF values currently recommended for transmission to the sink. However, this strategy results in a larger energy consumption, as it requires the transmission of multiple beacon messages. The beacon format is implementation-specific. The lower bound for the Beacon section duration corresponds to the sum of the ToA of the beacons.

5.3.2 Contention Access Period

In the CAP, aperiodic unconfirmed messages are sent from the end nodes to the sink. The end nodes compete for the channel access using a Slotted ALOHA-based mechanism. RT-LoRa randomly generates the 3-tuple of parameters (channel, spreading factor, timeslot) that will be used for transmission. The spreading factor is randomly selected among the currently recommended values, i.e., those in the $l_{SF}^{(a)}$ list described in the previous Section. The set of available channels is prepared using a duty cycle check function, that verifies the transmission feasibility on the basis of the expected ToA of the message according to the duty cycle restrictions [2]. If the list of available channels is not empty, the message is transmitted using the selected channel and spreading factor, so as to reduce the collision probability. Otherwise, the message is queued. The CAP is made up of a set of timeslots on different channels and spreading factors, as shown in Fig. 5.2. I suggest to set the duration of the CAP as a multiple of the maximum-sized timeslot duration. A longer CAP duration gives a lower collision probability, thus achieving better performance (i.e., lower packet loss ratio) for aperiodic transmissions. During the CAP, each end node remains in sleep mode, except when it has an aperiodic message to transmit. In that case, the end node changes its state from sleep to transmit mode rigth before the message transmission. After the transmission, the end node returns to the sleep mode.

[2] The duty cycle check function takes into account both the bandwidth reserved for the communications in CFP per hour and that consumed in the CAP in the last hour

5.3.3 Contention-Free Period

The CFP is devised for the real-time unidirectional communications from the end nodes to the sink (uplink). It consists of n sets (one for each SF value allowed by the application) of timeslots (see Fig. 5.2). The timeslots are used to schedule the confirmed transmission of messages belonging to periodic flows, using a Multi-CH (channel) and Multi-SF (spreading factor) Time Division Multiple Access (TDMA) strategy. Each i-th timeslot in the τ_{s_k} set is characterized by three items: the *identifier* $\mathcal{I}_{(i,k)}$, that determines the start time of the scheduled transmission within the CFP, the *channelset* $\mathcal{C}_{(i,k)}$, that contains the set of frequencies used, and the *SF value* $\mathcal{S}_{(i,k)}$. All the timeslots in the same set τ_{s_k} are characterized by the same SF value, i.e., $\mathcal{S}_{(i,k)} = SF_{\tau_{s_k}}$. During the timeslot $\mathcal{I}_{(i,k)}$, the transmission is performed using the spreading factor $\mathcal{S}_{(i,k)}$ and the channel $\mathcal{C}_{(i,k)}(0)$ (i.e., the first frequency in the channelset). Next, the frequencies in the channelset rotate. It is advisable that the frequencies in the channelset belong to different sub-bands. This mechanism, that runs every superframe, improves communication robustness and bandwidth utilization, while remaining compliant with the duty cycle restrictions of the selected sub-bands. For the sake of simplicity, in RT-LoRa the timeslots in the CFP are assigned offline.

In the CFP, each real-time flow generated by a stationary node is assigned to one timeslot. The latter is chosen so that the related SF value ensures a reliable communication between the stationary source node and the sink. Conversely, the real-time flows generated by a mobile node may need multiple timeslots to transmit one message. RT-LoRa supports three QoS-classes for these flows, each associated to a different strategy for the transmission over one or multiple timeslots. The available QoS classes are:

- *Normal (N).* These flows are scheduled in y non-overlapping timeslots (with different SF values) in the CFP, where y is the number of spreading factors allowed by the application. The channelsets of the timeslots go through the frequency rotation mechanism. Every message is transmitted in the timeslot that, among the ones assigned to the N-flow, is characterized by the lowest recommended SF value. A SF-value is “recommended” when it belongs to the $l_{SF}^{(a)}$ list. This flow class optimizes energy efficiency, as it takes the shortest ToA among the available options.

- *Reliable (R)*. These flows are scheduled in one timeslot in the CFP, with the highest SF value allowed by the application. While representing the best option in terms of bandwidth utilization, as it needs one transmission per message, this class results in higher energy consumption (due to the longer ToA).
- *Most Reliable (R+)*. These flows are scheduled in y non-overlapping timeslots (with different SF values) in the CFP, where y is the number of spreading factors allowed by the application. Every message is transmitted multiple times (*replicas*) during the scheduled timeslots whose spreading factor is in the $l_{SF}^{(a)}$ list. The flows in the R+ class achieve the highest reliability, thanks to the redundant transmissions, but the highest energy consumption.

The QoS-class is assigned to each flow offline, thus realizing a trade-off between reliability in transmission and energy consumption. Note that the application layer may not be synchronized with the MAC layer, so every message will be sent within the superframe that follows the superframe in which the message has been generated.

Each end node is in transmit mode during the timeslots used for the transmission of periodic real-time flows, while it remains in sleep mode during the other timeslots in the CFP.

5.3.4 Downlink section

The Downlink section is used by the sink for transmitting to the end nodes either *Unicast* or *Broadcast* messages. The first ones are sent by the sink to a specific end node. A bit in the payload indicates if the message is confirmed. If so, the end node acknowledges immediately after receiving the message. The sink transmits to the end node using the same SF of the last successful transmission from the end node, thus increasing the success probability comparing with a random choice. Conversely, Broadcast messages are sent by the sink to all the end nodes in the network, using the maximum SF value among the ones allowed by the application. Broadcast messages are unconfirmed. The end nodes must be on for listening to the downlink channel (i.e., the established channel for downlink communications) during this section. The Downlink section is optional and its duration is fixed by the network designer so as to find a trade-off between the sink downlink throughput and the energy consumption of the end nodes.

5.3.5 CFP Ack section

The CFP Ack section is used for *global-Ack* (GACK) frames that the sink broadcasts to confirm the messages received from the end nodes in the CFP of the same superframe. The GACK message contains a data structure of z bits per each of the r end nodes in the network. The format of such a data structure is implementation-specific. If one GACK message of $r * z$ bits cannot be broadcast, as it is larger than the maximum allowed physical payload, it can be split into multiple frames, consecutively transmitted in the CFP Ack section. The end nodes that send messages in the CFP section of the last superframe, if they are interested in the ack, must be on and listening during this section. The duration of this section is implementation-dependent.

5.4 RT-LoRa network configuration

This section proposes guidelines for the superframe configuration in a RT-LoRa network for an application with a specific cycle time (CT), here defined as the time interval in which all nodes can transmit once for each locally generated flow, highlighting the protocol limits due to the duty cycle restrictions. The equations provided in this section were used to set simulation parameters in Sect. 5.5.2 and to verify the simulation results in Sect. 5.5.3 and 5.5.4.

In the following (Sect. 5.4.1), a schedulability assessment of RT-LoRa is provided in order to verify the schedule feasibility for the network flows, with a specific set of configuration parameters. If no feasible schedule is found, a change of the application parameters is required, such as a new set of SFs to be used for the application and/or a different QoS class for the flows (N, R, R+).

For the schedulability assessment, we need to calculate the minimum superframe duration (Sect. 5.4.1) for a given set of periodic flows. To accomplish this, we calculate the minimum CFP duration needed to schedule all the periodic flows in the network (Sect. 5.4.2) and the timing constraint, in terms of the minimum superframe duration, that derives from the duty cycle restriction (Sect. 5.4.3).

The equations provided in the following are relevant to the end nodes, so we do not consider beacons, downlink and CFP Ack communications. However, a duty cycle check function, implemented in the sink, takes into account all the sink transmissions. Consequently, in the simulations all the needed checks are in place.

Table 5.5 summarizes the notations used in this Section.

5.4.1 Schedulability analysis

This section presents a schedulability assessment of RT-LoRa. In particular, a methodology to calculate the minimum superframe duration for a given set of periodic flows is provided. Then, the superframe duration is compared with the timing constraint deriving from the duty cycle restriction, that defines the longest time interval (in percentage) during which a transmitter node can occupy a specific sub-band per hour. The value of the maximum end-to-end delay for a message is calculated as the sum of the superframe duration and the application-defined value $\sigma^{(k)}_{(X,i)}$ corresponding to the time interval,

Table 5.5: Summary of notations

Symbol	Definition
n_{SB}	Number of used sub-bands.
$\{\mathcal{S}\}$	Set of SF values allowed by the application.
s_{min}	Minimum SF value allowed by the application.
s_{max}	Maximum SF value allowed by the application.
$ToA_{(SF=x)}$	Time on Air of a maximum-sized messages transmitted with SF value equal to x.
$\{\mathcal{N}_S\}$	Set of stationary nodes (SNs).
$\{\mathcal{N}_M\}$	Set of mobile nodes (MNs).
$\mathcal{T}_{sl(SF=x)}$	Duration of a timeslot with transmissions performed with SF value x.
$\mathcal{T}^{(k)}_{sl(X,i,SF=x)}$	Duration of the timeslot assigned to the i-th flow of the k-th node characterized by SF value equal to x. X can be N, R or R+ for MNs and Q for SNs.
$\{\mathcal{S}\}^{(k)}_{(X,i)}$	Set of SFs in the timeslots assigned to the i-th flow generated by the k-th node. X can be N, R or R+ for MNs and Q for SNs.
$\sigma^{(k)}_{(X,i)}$	Time interval within which the first and the last timeslot for the N (or R+) i-th flow generated by the k-th node are allocated.
$D^{(k)}_{(X,i)}$	Deadline of the i-th flow generated by the k-th node. X can be N, R or R+ for MNs and Q for SNs.
DC_{min}	Minimum duty cycle value among those allowed by all the used sub-bands.
n_N	Overall number of N-flows generated by the MNs.
n_R	Overall number of R-flows generated by the MNs.
n_{R^+}	Overall number of R+-flows generated by the MNs.
$n_{(SF=x)}$	Overall number of flows that need a timeslot with SF value equal to x generated by the SNs.
$n^{(j)}_X$	Number of X-flows required by the j-th MN (X can be N, R or R+).
$n^{(h)}_{(SF=x)}$	Number of flows generated by the h-th SN with SF value equal to x.
$\Delta_S(h)$	Overall time needed by the h-th SN to send one maximum-sized message for each of the $n^{(h)}_{(SF=x)}$ locally generated flows.
$\Delta_M(k)$	Overall time needed by the k-th MN to send one maximum-sized message for each of the $n^{(k)}_X$ locally generated flows.
η_{tx}	Minimum value within the set containing the number of times that each node can transmit the real-time flows in the CFP per hour, according to the duty cycle restrictions.
$\mathcal{T}_{supfrm}$	Superframe duration.
$\mathcal{T}_{CFP}$	CFP duration.
$\mathcal{T}^{ALL}_{CFP}(s)$	Time needed to schedule the transmission of the real-time flows generated by all the SNs and MNs in the CFP for the SF s.
$\mathcal{T}_{id}$	Implementation-dependent time duration.

within a superframe, between the start of the first scheduled timeslot for a specific flow and the end of the last timeslot scheduled for the same flow (if only one timeslot is scheduled for the flow, $\sigma^{(k)}_{(X,i)}$ is equal to the timeslot duration). If the maximum end-to-end delay value is lower than or equal to the application timing constraint, then the schedule is feasible, as the flows will meet their deadline while complying with the duty cycle constraint. Otherwise, a change of the application parameters is required, for instance, a new set of SFs to be used for the application and/or a different QoS class for the flows (N, R, R+), etc., so as to find a feasible schedule with RT-LoRa.

It is mandatory that $\mathcal{T}_{supfrm} \geq \mathcal{T}^{MIN}_{supfrm}$ i.e., the superframe duration $\mathcal{T}_{supfrm}$ must be larger than or equal to the minimum admissible value for a superframe duration $\mathcal{T}^{MIN}_{supfrm}$, calculated as:

$$\mathcal{T}^{MIN}_{supfrm} = max\left[\mathcal{T}_{CFP}, \mathcal{T}^{(DC)}_{supfrm}\right] + \mathcal{T}_{id} \tag{5.2}$$

In Eq. (5.2):

- $\mathcal{T}_{CFP}$ is the CFP duration needed to schedule the transmission to the sink of one maximum-sized message for each real-time flow in the network (Sect. 5.4.2).

- $\mathcal{T}^{(DC)}_{supfrm}$ is the superframe duration ($\mathcal{T}_{supfrm}$) according to the duty cycle (DC) limitation that considers the maximum number of times that each node can transmit the real-time flows per hour (Sect. 5.4.3).

- $\mathcal{T}_{id}$ is an implementation-dependent time duration that includes the duration of the Beacon, CAP, Downlink and CFP Ack sections.

The $\mathcal{T}_{CFP}$ is a "network limitation", as it considers all the flows in the network, while $\mathcal{T}^{(DC)}_{supfrm}$ is a "local limitation", as it is relevant to the node schedule. Consequently, the $\mathcal{T}^{MIN}_{supfrm}$ is calculated using the maximum between these values.

In detail, the flow set can be scheduled when $\mathcal{T}_{supfrm}$ is lower than the required cycle time.

$$\mathcal{T}_{supfrm} \leq CT \tag{5.3}$$

I assume that the message generation period is equal to the cycle time in all the network end nodes. As discussed in Sect. 5.3.3, the number of non-overlapping timeslots for the normal flow (N) or the most reliable (R+) i-th flow generated by the k-th node, i.e., w, is equal to the number of values in the $\{\mathcal{S}\}$ set. These w timeslots are allocated in the CFP within an application-defined time interval $\sigma_{(X,i)}^{(k)}$ between the sum of the durations of the timeslots scheduled for the flow and the CFP duration:

$$\sum_{s \in \{\mathcal{S}\}_{(X,i)}^{(k)}} \mathcal{T}_{sl(X,i,SF=s)}^{(k)} \leq \sigma_{(X,i)}^{(k)} \leq \mathcal{T}_{CFP} \tag{5.4}$$

where $X \in \{N, R^+\}$. In fact, $\sigma_{(X,i)}^{(k)}$ will be equal to the sum of the durations of the timeslots scheduled for the flow if these timeslots are scheduled consecutively (i.e., if these timeslots are adjacent to each other). On the other hand, $\sigma_{(X,i)}^{(k)}$ will be equal to $\mathcal{T}_{CFP}$ if the first and the last timeslots of the CFP are scheduled for the flow. The message belonging to the i-th flow is delivered within the deadline $D_{(X,i)}^{(k)}$ if:

$$D_{(X,i)}^{(k)} \geq \mathcal{T}_{supfrm} + \sigma_{(X,i)}^{(k)} \Rightarrow \mathcal{T}_{supfrm} \leq D_{(X,i)}^{(k)} - \sigma_{(X,i)}^{(k)} \tag{5.5}$$

Eq. (5.5) can be used also for an R-flow or a flow generated by a stationary node, with $\sigma_{(X,i)}^{(k)}$ equal to the duration of the timeslot scheduled for that flow.

The worst case in terms of end-to-end delay is represented by specific conditions on an N-flow due to the features of this type of flow. In fact, every message of an N-flow is transmitted in the scheduled timeslot that is characterized by the lowest recommended SF value.

Then, it is possible that in a superframe the timeslot recommended for the transmission is the first among the ones assigned to the N-flow, while in the next superframe the timeslot recommended for the transmission is the last one. Such a case can be described as follows. I assume that the first scheduled timeslot for that N-flow is selected for the transmission in a specific superframe. I also assume, to take the worst case, that the application layer of the nodes is not synchronized with the MAC layer and, for this reason, the flow message is generated right after the assigned timeslot. Consequently, the message cannot be sent in the current superframe. As I am looking for the worst case, I assume that the message is generated right after the end of the timeslot it is assigned. Then, the message trasmission is postponed to the next superframe. In the worst case, the timeslot assigned for the trasmission of the message in the next superframe will be the last one.

5.4.2 CFP duration

In this subsection I deal with the minimum CFP duration that is needed to schedule the transmission to the sink of one maximum-sized message for each real-time flow in the network.

The overall time $\mathcal{T}_{CFP}^{ALL}(s)$ needed to schedule the transmission of the real-time flows generated by all the the stationary and mobile nodes in the CFP, calculated for a given SF value s in $\{\mathcal{S}\}$, is:

$$\mathcal{T}_{CFP}^{ALL}(s) = \left\lceil \frac{n_R^{\diamond}(s) + n_N + n_{R^+} + n_{(SF=s)}}{n_{SB}} \right\rceil * \mathcal{T}_{sl(SF=s)} \tag{5.6}$$

where $n_R^{\diamond}(s) = \{n_R \text{ if } (s = s_{max}); 0 \text{ otherwise } \}$

Note that $\mathcal{T}_{CFP}^{ALL}(s)$ takes into account the number of used subbands (i.e., n_{SB}). In fact, the number of CFP timeslots required by all the nodes for each SF allowed by the application is divided by n_{SB}. The CFP duration $\mathcal{T}_{CFP}$ is the maximum value among the times $\mathcal{T}_{CFP}^{ALL}(s)$, calculated for each SF s in $\{\mathcal{S}\}$, i.e.:

$$\mathcal{T}_{CFP} = max \left[\mathcal{T}_{CFP}^{ALL}(s)\right]_{\forall s \in \{\mathcal{S}\}} \tag{5.7}$$

5.4.3 Superframe duration according to the duty cycle restrictions

In this subsection I address the minimum superframe duration according to the duty cycle limitation.

Eq. (5.8) and (5.9) provide $\Delta_S(h)$ and $\Delta_M(k)$, i.e., the overall time needed by the h-th stationary node and the k-th mobile node, respectively, to send one maximum-sized message for each of its real-time flows, according to the transmission strategies described in Sect. 5.3.3.

$$\Delta_S(h) = \left\{ \sum_{i=s_{min}}^{s_{max}} \left[(n_{(SF=i)}^{(h)}) \cdot ToA_{(SF=i)} \right] \right\} \tag{5.8}$$

$$\Delta_M(k) = \left\{ \sum_{i=s_{min}}^{s_{max}} \left[(n_N^{(k)} + n_{R+}^{(k)}) \cdot ToA_{(SF=i)} \right] \right\} + n_R^{(k)} ToA_{(SF=s_{max})} \tag{5.9}$$

In Eq. (5.8) (5.9), the aperiodic communications in the CAP and the ack messages of confirmed downlink communications are not considered, as they are unpredictable. However, a duty cycle check function, implemented in each end node, takes into account all the transmissions.

In Eq. (5.10), I calculate η_{tx} as the minimum value within the set containing the number of times that each node can transmit the real-time flows in the CFP per hour in the sub-band with $DC = DC_{min}$, i.e., I also consider the worst case in terms of duty cycle. If the DC limit is met in the sub-band in which $DC = DC_{min}$, it is also met in all the other sub-bands in which $DC > DC_{min}$. In the Eq. (5.10), $\Delta_S(h)$ and $\Delta_M(k)$ are calculated $\forall h \in \{N_S\}$ and $\forall k \in \{N_M\}$, respectively. This way, Eq. (5.10) takes into account all the possible nodes in the network and chooses the minimum value, i.e., the worst case in terms of the number of times that each node can transmit the real-time flows in the CPF per hour. Note that the worst case is the one in which $\Delta_S(h)$ or $\Delta_M(k)$ takes the highest value among the ones calculated for all the nodes in the network. Furthermore, Eq. (5.10) takes into account the frequency rotation for CFP timeslots introduced by RT-LoRa through the parameter n_{SB}, that makes the number of allowed transmissions increase n_{SB} times.

$$\eta_{tx} = min \begin{cases} \left[\left\lfloor \frac{3600 \cdot DC_{min} \cdot n_{SB}}{\Delta_S(h)} \right\rfloor \right]_{\forall h \in \{N_S\}} \\ \left[\left\lfloor \frac{3600 \cdot DC_{min} \cdot n_{SB}}{\Delta_M(k)} \right\rfloor \right]_{\forall k \in \{N_M\}} \end{cases} \tag{5.10}$$

Eq. (5.11) provides the superframe duration $\mathcal{T}^{(DC)}_{supfrm}$, calculated according to the worst case in terms of the number of times a node can transmit real-time flows per hour (i.e., every 3600 seconds).

$$\mathcal{T}^{(DC)}_{supfrm} = \frac{3600}{\eta_{tx}} \tag{5.11}$$

5.5 Simulative Assessment

In this section, a performance assessment of RT-LoRa obtained using the OMNeT++ environment is presented. The module that simulates the RT-LoRa MAC layer was developed as an extension of the Industrial LoRa simulator [102], while the LoRa physical layer and the wireless channel were simulated using FLoRa [96]. Consequently, as discussed in [96], simultaneous transmissions on different SFs are considered orthogonal using the FLoRa physical layer. A duty cycle check function was implemented in each node (both sink and end nodes) in order to be compliant with the duty cycle limitations. Two performance metrics were used: the Packet Loss Ratio (PLR) and the End-to-End (e2e) delay.

The PLR, measured at the application layer, is expressed as a percentage according to Eq. (5.12),

$$PLR = \left(\frac{n_{lostMsg}}{n_{txMsg}}\right)*100 = \left(1 - \frac{n_{rxMsg}}{n_{txMsg}}\right)*100 \tag{5.12}$$

where n_{txMsg}, $n_{lostMsg}$ and n_{rxMsg} are the number of transmitted, lost or correctly received messages, respectively, measured over all the end nodes in the network. In Eq. (5.12) a message is counted once even if it is transmitted or received in multiple replicas.

The e2e delay is the time that a message takes since its generation at the application layer of the source node and its reception at the application layer of the sink, calculated according to Eq. (5.13):

$$e2eDelay = ArrivalTime - GenTime \tag{5.13}$$

In the simulation the processing delays were not considered, as they are implementation-dependent.

5.5.1 Simulated scenario

I performed the simulative assessment in a realistic industrial use case similar to the ones presented in [97] [102]. The considered scenario involves a large number of end nodes that communicate with one sink. An end node can be either stationary (SN) or mobile (MN).

The simulated network topology consists of a network with 101 nodes, i.e., a sink (located in the center of the sensing area to ensure the maximum coverage range), 25 stationary nodes and 75 mobile nodes. All the nodes are placed within an area, here called sensing area, with a range of 250m around the sink.

The simulated scenario consists of sensor nodes that generate a periodic message every 30s. The deadline D by which the message must be delivered is 30s, too. This is a realistic use case of a distributed measurement system for industrial IoT applications (e.g., automation ones) in which cycle times up to 60s are required [98] [77]. RT-LoRa, due to the LoRa duty cycle limitation, is devised for applications with no very tight time requirements. For example, RT-LoRa may fit well process control applications, as they feature long cycle times. Conversely, as in discrete manufacturing the cycle times can be quite short, it could happen that, with a given set of flows and configuration parameters, no feasible RT-LoRa schedule is found. In this case, the network designer needs to try different configuration parameters, such as a new set of SFs to be used for the application and/or a different QoS class for the flows (N, R, R+), and so on.

The considered physical payload is 50 bytes long, a typical value for industrial applications [77] [97].

Every end node has one periodic real-time flow scheduled in the CFP. Furthermore, aperiodic nonreal-time messages are generated by the end nodes every t seconds, where t is a random variable that varies according to an exponential distribution with mean 70s. The nonreal-time messages are sent in the shared bandwidth of the CAP. The application layer of the end nodes is not synchronized with the MAC layer.

5.5.2 Simulation settings

The transmissions were performed on three sub-bands (h1.4, h1.6 and h1.7, see Table 5.3) in order to achieve a duty-cycle limitation of 12%. As the h1.5 sub-band is not used, in Eq. (5.10) $DC_{min} = 1\%$. CFP frequency rotation involves only one channel for each of the three sub-bands. The h1.4 sub-band contains three available channels, but only one of these (randomly chosen) was used. In fact, as the h1.4 sub-band is subject to a DC limit of 1%, transmitting one message for each of its 3 channels (i.e., using all the 5 channels available in all the three sub-bands at the same time) would not be advisable. The use of 5 channels would be suitable only with a high cycle time and a very large number of end nodes, as it allows to schedule more simultaneous transmissions in the CFP.

The RT-LoRa protocol was set to use values of spreading factor $s \in \{7; 8; 9\}$ in order to work with a bit rate complying with the application requirements (i.e., from 1760 bit/s to 5470 bit/s, with a bandwidth of 125 kHz).

The configuration settings of the end nodes in the simulated scenario are summarized in Table 5.6.

Table 5.6: End nodes configuration settings

Number of nodes	***Type of nodes***	***Distance from the sink***	***Allocated CFP timeslots***
10	SN	0-125 m	τ_{SF_7}
10	SN	125-180 m	τ_{SF_8}
5	SN	180-250 m	τ_{SF_9}
75	MN	0-250 m	According to the QoS classes of the flows

The 75 MNs moved within the sensing area with speed between 0.5 m/s and 1 m/s according to the ChiangMobility model, available in the INET framework.

In the simulations it was assumed that 25 MNs schedule a *Normal flow (N)*, 25 MNs schedule a *Reliable flow (R)* and 25 MNs schedule a *Most Reliable (R+)* flow. Since the number of SFs allowed by the application is three (i.e., 7, 8 and 9), every message of $R+$ flows is transmitted three times. Eq. (5.1) was used to calculate the lower bound for time slot duration with the parameter values in Table 5.2. About 0.004s were added to that value, as suggested in [120]. In particular, the duration $\mathcal{T}_{sl(SF=7)}$ of the τ_{SF_7} timeslots, the duration $\mathcal{T}_{sl(SF=8)}$ of the τ_{SF_8} timeslots and the duration $\mathcal{T}_{sl(SF=9)}$ of the τ_{SF_9} timeslots were set to 0.101s, 0.202s and 0.404s, respectively. The three non-overlapping slots allocated in the CFP for a N-flow or a R+-flow were scheduled within a time interval of ($\sigma^{(k)} = 3 * \mathcal{T}_{sl(SF=9)}$ for each k-th node (see Eq. (5.4)).

The lower bound for the superframe duration was calculated using Eq. (5.2). The calculated value for $\mathcal{T}_{supfrm}^{MIN}$ is 20.112s, lower than the required deadline D (30s). Moreover, such a value is also lower than $D - \sigma^{(k)}$ for each k-th node, therefore the system was able to schedule the given periodic real-time flows in the CFP (see Eq. (5.5)). The $\mathcal{T}_{CFP}$ duration was set to 10.908s, according to Eq. (5.7).

I considered two case studies corresponding to different configurations in the same scenario in order to show how different superframe configurations affect the performance of aperiodic messages in terms of PLR and the e2e delays of periodic messages. In particular, in the configuration (A) the $\mathcal{T}_{supfrm}$ is set at a value close to $\mathcal{T}_{supfrm}^{MIN}$ ($\mathcal{T}_{supfrm} \geq \mathcal{T}_{supfrm}^{MIN}$), i.e., the minimum allowed value of the superframe duration. Conversely, in the configuration (B) the $\mathcal{T}_{supfrm}$ is set at a value close to $D - \sigma^{(k)}$ ($\mathcal{T}_{supfrm} \leq D - \sigma^{(k)}$), i.e., the maximum allowed value of the superframe duration. The larger $\mathcal{T}_{supfrm}$ in the configuration (B) than the configuration (A) allows to extend the CAP. This way, I expect to obtain a lower PLR for aperiodic messages and a higher (but always less than the deadline) maximum e2e delay for periodic messages. The configurations setting are shown in Table 5.7.

Table 5.7: Configurations setting

Parameter	***Configuration (A)***	***Configuration (B)***
Beacon section	0.707s	0.707s
CAP	6.060s	14.140s
CFP	10.908s	10.908s
Downlink section	0.808s	0.808s
CFP Ack	2.0s	2.0s
$\mathcal{T}_{supfrm}$	$\mathcal{T}_{supfrm}^{cfgA}$= 20.483s	$\mathcal{T}_{supfrm}^{cfgB}$= 28.563s

Note that the duration of the beacon section was set to $\mathcal{T}_{sl(SF=9)} + \mathcal{T}_{sl(SF=8)} + \mathcal{T}_{sl(SF=7)}$, but the sink only transmits for a time interval equal to $ToA_{(SF=9)} + ToA_{(SF=8)} + ToA_{(SF=7)}$, according to Eq. (5.1), where the PL parameter is the physical payload of a beacon message (implementation-dependent).

The simulation time was set to $\mathcal{T}_{sim} = 36000s$ (i.e., ten hours) in order to collect a significant amount of data. The LoRa log-normal shadowing model [96], a propagation model provided by the FLoRa framework, was adopted for the simulations.

The relevant parameters are summarized in Table 5.8.

Table 5.8: Simulation parameters

Parameter	***Value***
Coding Rate	4/5
Bandwidth	125 kHz
Transmission Power	14 dBm
$\{\mathcal{S}\}$	$\{7; 8; 9\}$
Sub-bands	$\{h1.4; h1.6; h1.7\}$
Physical payload	50 bytes
Generation period of the periodic messages	30s
$\mathcal{T}_{sl(SF=7)}$	0.101s
$\mathcal{T}_{sl(SF=8)}$	0.202s
$\mathcal{T}_{sl(SF=9)}$	0.404s
$\mathcal{T}_{sim}$	36000s
Propagation model	LoRa log-normal shadowing model

5.5.3 Simulation results - Aperiodic transmissions

Fig. 5.3 shows the average PLR measured for the configurations (A) and (B) for the aperiodic messages transmitted in the CAP. The flows of stationary nodes (SN) are grouped by the maximum distance between the end node and the sink. The flows of mobile nodes (MN) are grouped by their QoS class.

Fig. 5.3 also shows that the messages sent by the stationary nodes that are furthest from the sink experienced the highest PLR. This can be explained as follows. For a CAP transmission, each end node randomly selects a spreading factor among the "recommended" values in the list that is updated according to the multiple beacons received during the current superframe. The stationary nodes closest to the sink can choose among multiple SF values, whereas it is likely that the furthest ones can only select SF9. As a consequence, the probability of collision, i.e., that two nodes randomly select the same set of transmission parameters in the CAP (i.e., the same timeslot identifier, channel and spreading factor), is lower for the devices closer to the sink than for the ones that are further away. The lower probability of collision determines the lower PLR measured for the closest

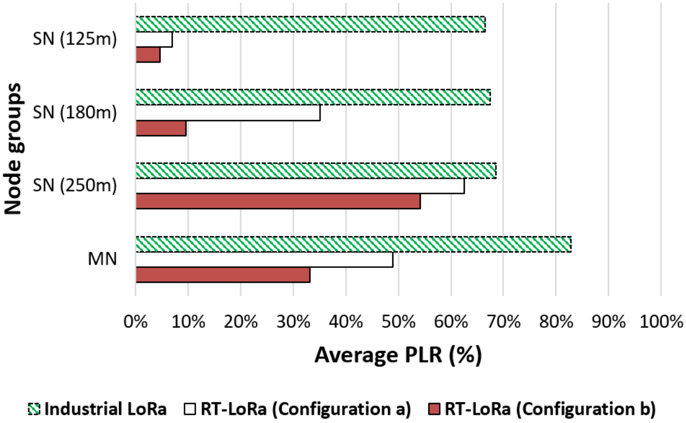

Figure 5.3: Average PLR for aperiodic messages. Comparison between RT-LoRa (configuration A and B) and IndustrialLoRa.

stationary nodes. About the MNs, the measured PLR depends on their distance from the sink, which changes according to the Chiang-Mobility model.

The aperiodic messages in configuration (B) experienced a lower PLR than in configuration (A). This is due to the longer duration of the CAP in configuration (B). In fact, using a larger CAP the end nodes can choose the timeslot to use for transmission among a larger set of timeslot identifiers, thus lowering the collision probability.

Finally, Fig. 5.3 shows that the Slotted ALOHA-based MAC layer (used for transmissions in the CAP) of RT-LoRa outperforms the Pure ALOHA-based MAC layer used by Industrial LoRa [102] (see Section 4.4).

5.5.4 Simulation results - Periodic transmissions

The measured PLR values for the periodic N-flow messages using the configurations (A) and (B) were 1.44% and 2.28%, respectively, and are tolerable for the considered scenario. All the other periodic messages were received by the sink and confirmed in the CFP Ack

Table 5.9: Maximum e2e delay of real-time flows

Nodes **(QoS class of the flows)**	**Max e2e delay** ***Configuration (A)***	**Max e2e delay** ***Configuration (B)***
Stationary nodes	20.884s	28.966s
Mobile nodes (N-flows)	21.695s	29.764s
Mobile nodes (R-flows)	20.874s	28.964s
Mobile nodes (R+-flows)	20.887s	28.960s

section. The loss of some N-flow messages is because each message of an N-flow is transmitted in the timeslot with the lowest "recommended" SF value among those in the dynamic list maintained by the source node. If the source node is mobile, sometimes this list can be no longer reliable, because it is updated on the basis of the beacons received every superframe. As a consequence, a message might be sent using an SF that is no longer reliable, thus being lost or corrupted.

The maximum e2e delays measured for the periodic messages transmitted in the CFP section are shown in Table 5.9. As expected, the e2e delays (measured at the application layer for all the periodic real-time messages) are always lower than the deadline $\boldsymbol{D} = 30s$. Hence, they are upper-bounded. Moreover, as discussed in Sect. 5.4 and according to Eq. (5.5), the measured e2e delays are always lower than (or equal to) $(\mathcal{T}_{supfrm}^{cfgA} + \sigma^{(k)}) = 21.695s$ for configuration (A) and than $(\mathcal{T}_{supfrm}^{cfgB} + \sigma^{(k)}) = 29.775s$ for configuration (B), thus demonstrating the effectiveness of the proposed approach.

5.5.5 Summary and Discussion

Table 5.10 recaps the features of the LoRaWAN [82], Industrial LoRa [102] (see Section 4.3) and RT-LoRa MAC protocols.

Both Industrial LoRa and RT-LoRa organize the network time in superframes that cyclically repeat. Morever, both protocols support a mechanism for superframe synchronization that is based on beacon messages sent broadcast over the network. Conversely, Lo-

Table 5.10: Main Characteristics of LoRaWAN, Industrial LoRa and RT-LoRa

	LoRaWAN	**Industrial LoRa**	**RT-LoRa**
Topology	Star (*)	Star	Star
Synchronization	Not supported	Beacon-based	Multiple beacons based
Transmission of real-time periodic flows	Not supported	Supported	Supported
Smart selection of Spreading Factors	Supported through ADR	Not supported	Supported through multiple beacons
MAC strategy for aperiodic transmission	Pure ALOHA	Pure ALOHA	Slotted ALOHA
MAC strategy for periodic transmission	Pure ALOHA	Multi-CH and Multi-SF TDMA	Multi-CH and Multi-SF TDMA
QoS classes	Downlink only (3 device classes provided)	Not supported	Uplink only (3 QoS classes provided)
Support for retransmission	Uplink only	Not supported	Not supported
Frequency rotation	Pseudo-random channel hopping	Not supported	Supported

(*) star-of-stars when multiple gateways are used

RaWAN does not support any superframe structure or synchronization mechanism. Both industrial LoRa and RT-LoRa are tailored for industrial applications. In fact, they support the transmission of periodic real-time flows in the CFP of the superframe, whereas LoRaWAN does not provide any support for real-time messages, due to the non-determinism of its ALOHA-based medium access strategy.

Both LoRaWAN and RT-LoRa, for different purposes, use smart mechanisms to select the Spreading Factor values eligible for transmission. LoRaWAN supports the so-called Adaptive Data Rate (ADR) scheme to manage data rate (i.e., the spreading factor) and RF output in order to maximize the battery life of the node. RT-LoRa uses a novel mechanism based on the transmission of multiple beacon frames that allow the end nodes to be aware of the spreading factor values currently recommended for a transmission, thus ensuring higher reliability. I recall that the communication reliability is, together with timeliness, very important in industrial IoT scenarios. RT-LoRa performs aperiodic transmissions in the CAP of the superframe using a custom Slotted ALOHA strategy. Conversely, LoRaWAN and Industrial LoRa use a Pure ALOHA strategy for the transmission of aperiodic flows. As a consequence, RT-LoRa outperforms both LoRaWAN and Industrial LoRa, as the smart version of Slotted ALOHA determines a lower probability of collision between two transmitting nodes.

RT-LoRa supports three QoS classes (named *N*, ***R*** and ***R*+**) that allow to achieve a trade-off between reliability of uplink communications and energy consumption. LoRaWAN instead provides three device classes (named A, B and C) in order to achieve a trade-off between downlink communication latency and battery lifetime. Industrial LoRa does not provide any QoS support.

In LoRaWAN, Industrial LoRa, and RT-LoRa the transmissions sent by nodes (uplink) can be confirmed using the ACK messages. However, RT-LoRa and Industrial LoRa do not support the retransmission of the unconfirmed messages. Conversely, LoRaWAN supports retransmissions.

The RT-LoRa frequency-rotation procedure allows for a better utilization of the available bandwidth comparing with Industrial LoRa, while remaining compliant with the duty-cycle restrictions stated by the ETSI regulations. LoRaWAN, instead, implements only a pseudo-random channel hopping.

5.6 Conclusions

This chapter proposed RT-LoRa, a medium access strategy working over LoRa that guarantees bounded end-to-end transmission delays to real-time flows in LPWAN-based industrial IoT applications. The focus of this chapter is on enabling real-time communications on LoRa-based networks and, for this reason, here I do not explicitly deal with power consumption aspects. This chapter discussed RT-LoRa performance through simulation and analysis. The implementation of RT-LoRa on COTS devices has already started and it will addressed in future work. I am using the SX1272 LoRa transceiver [121] for the end nodes and the SX1302 LoRa transceiver [122] for the sink. The SX1302 can detect at any time, any packet in a combination of 8 different spreading factors (SF5 to SF12) and 10 channels, and demodulate up to 16 packets at any time. To implement RT-LoRa, the LoRaWAN protocol of the transceiver has to be disabled, thus I am implementing the RT-LoRa MAC layer from scratch. RT-LoRa uses a star topology and assumes that the transmissions performed using different spreading factors are orthogonal. Future work will go further in both directions. First, suitable approaches to enable the coexistence of multiple star networks running RT-LoRa operating in the same frequency bands, such as synchronization mechanisms for the sinks, will be addressed. Second, the quasi-orthogonality will be implemented in the simulator to assess how much it affects the performance of RT-LoRa.

Chapter 6

A Proposal Towards Software-Defined Management of Heterogeneous Virtualized Industrial Networks

The Industry 4.0, also referred as the Factory of the Future, refers to the fourth industrial revolution that introduces novel communication and computation technologies, such as, the Internet of Things (IoT) and cloud computing, in industrial manufacturing systems [123]. In this scenario, several devices, machines, and applications with diverse communication requirements in terms of data rates, latency and reliability, need to be connected to interoperate. Consequently, these industrial networks are highly heterogeneous, as they need to rely on different communication technologies. Typically, the communication networks in industrial systems are based on wired technologies, as they can provide high communications reliability. However, wired technologies are unable to fully meet the required flexibility and adaptivity of future manufacturing processes in terms of mobility support and network reconfiguration. Conversely, wireless communication technologies can provide connectivity to mobile nodes (e.g., robots or workers) and offer the deployment flexibility without the need for cable installation. Unfortunately, they are not as reliable as wired technologies. As discussed in [124], industrial wireless communication systems play a crucial role in several Industry 4.0 applications, e.g., in those relevant to equipment status monitoring.

Consequently, the Industry 4.0 needs to use multiple communication technologies (wired and wireless) to meet the wide range of requirements of industrial applications. Therefore, a flexible network architecture is required in order to manage the network through an abstraction level, which is decoupled from the underlying technologies. This can be achieved through Software-Defined Networking (SDN) and network virtualization, which are promising techniques for handling the complexity of heterogeneous networks and improving Quality of Service (QoS) provisioning [123] [125].

Recently, SDN has received significant interest from the industrial and the academic communities, as it promises to achieve flexible and easily manageable networks by providing a clear decoupling between the control plane and the data plane. This way, SDN allows to implement logically centralized intelligence in the control plane and generalized network hardware in the data plane. Thanks to the network virtualization, the physical infrastructure of a virtualized network can be partitioned into dedicated logical networks with specific functionalities. However, the current SDN ecosystem provides a rich support for wired packet-switched networks, but not for wireless technologies that need specific data-plane abstractions, controllers, and programming primitives to be established [126].

Software-Defined architectures for Wireless Local Area Networks (WLANs) and Wireless Sensor Networks (WSNs) have been recently presented in order to provide easier wireless networks control mechanisms [33, 34, 126–128].

Motivation. To the best of my knowledge, there is no state-of-the-art global network architecture that has been proposed for the management of Industry 4.0 scenarios, i.e., able to support any combination of communication technologies. Consequently, a network architecture that provides support for both software-defined WLANs and software-defined WSNs is required, as the modern industrial networks require to deal with a multitude of communication technologies in order to fulfill the application requirements.

Contributions. The main contribution of this chapter is a software-defined architecture for the management of communication networks in the Industry 4.0 context, i.e., in a heterogeneous scenario that involves different communication technologies suitable for satisfying the different requirements of diverse industrial applications. The proposed architecture addresses the use of SDN and network virtualization as enabling technologies to meet a broad range of application requirements. I also discuss an implementation plan for the proposed architecture.

Organization. The rest of the chapter is structured as follows. The next section provides a background on Software-Defined Networking and network virtualization. Section 6.2 summarizes related works. Section 6.3 presents the network architecture design, while Section 6.4 describes the proposed architecture from a practical point of view and briefly outlines the architecture of EmPOWER, which is an open Mobile Network Operating System. Section 6.5 provides a description of an implementation roadmap based on the EmPOWER architecture. Finally, Sec. 6.6 draws my conclusions and hints for future works.

6.1 Background

In this section, I provide a basic background on the SDN and network virtualization technologies.

6.1.1 Software-Defined Networking technology

SDN is an emerging network architecture that has been proposed to simplify and improve the design and development of complex wired network control mechanisms. Furthermore, SDN is also gaining popularity for the design of WLANs, WSNs and cellular networks [129]. SDN is characterized by the following main features:

- *Separation of the control plane from the data plane.* The control plane provides the logic of the network appliances, including QoS-aware forwarding strategies, while the data plane provides the actual packet processing and forwarding hardware.

- *Centralized control and global view of the network.* The SDN architecture is logically centralized and the control plane has a global view of the network. This way, SDN enables the development of highly intelligent and optimized forwarding rules and QoS algorithms. Conversely, traditional networks integrate control and data plane into each device, and then control methods are distributed with limited awareness of the network state.

- *Abstraction of the network.* Services and applications running on the SDN technology interact with the network through APIs. This way, they are abstracted from the underlying technologies and hardware that provide physical connectivity.

- *Network programmability.* The network behavior can be controlled by software that resides beyond the networking devices.

The SDN architecture, shown in Fig. 6.1a, consists of three layers:

- *Infrastructure layer*, that consists of hardware equipment and merely focuses on transmission and forwarding of data. The switches are basic forwarding elements.

- *Control layer*, that consists of the SDN controller with several software functions that remotely control the forwarding rules of the network. The controller can flexibly and dynamically allocate resources on demand of different users and based on the global network topology.

- *Application layer*, that allows the programming of the bottom-layer equipment through the interfaces provided by the control layer. Network operators can write high-level control programs that specify the behavior of the entire network. Conversely, in a conventional network, operators must code functionalities in terms of low-level device configurations.

The communication between the infrastructure layer and the control layer is realized through the so-called southbound interface, whereas the communication between the application layer and the control layer through the so-called northbound interface. OpenFlow [130] is the de facto standard for the southbound interface. It provides basic functions to set the forwarding rules in the forwarding plane of network devices. On the other hand, the forwarding abstraction provided by OpenFlow does not account for the stochastic nature of wireless links, the resource allocation granularity and the heterogeneity in the link and radio layer technologies.

6.1.2 Network virtualization technology

Network virtualization divides a physical network infrastructure into multiple virtual networks, also known as slices, i.e., multiple logical/virtual networks on top of a common network infrastructure. A slice must contain the resources required to meet the communication requirements of the application or service that such a slice supports. Each service or application can use a slice of the physical network without knowing the details of the underlying network.

As shown in Fig. 6.1b, an SDN network can be virtualized by inserting a network hypervisor between the network infrastructure and the SDN controller. Therefore, each network slice has a separate SDN controller to manage the forwarding rules within the slice. Each SDN controller only ”sees” a slice of the physical network. This way, the slices are controlled independently from each other, thus simplifying the network management. In fact, one of the main objectives of network virtualization is to ensure isolation, i.e., independent management of each slice as a separated network, thus representing a key technology to deploy flexible communication requirements for industrial applications.

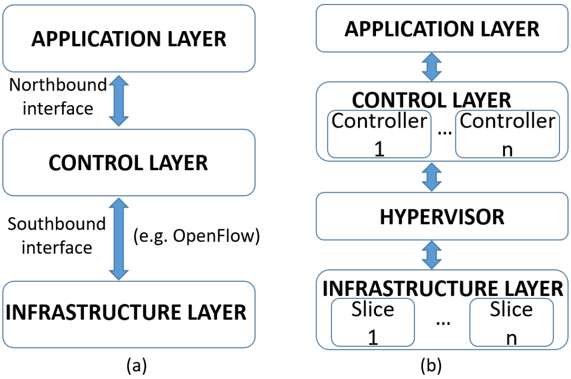

Figure 6.1: (a) The SDN architecture. (b) A virtualized SDN architecture.

6.2 Related Work

Industrial applications demand a wide range of different communication requirements that can be efficiently satisfied through the use of wired and wireless technologies. While some wired technologies are obviously suitable for industrial networks thanks to their reliability, wireless technologies are attractive for their flexibility. The most important wireless technologies developed to support industrial automation and control are based on the IEEE 802.15.4 (e.g., WirelessHART, ISA100.11a) [51] [131]. Other wireless technologies have been adapted for industrial wireless networks, e.g., Bluetooth Low Energy [89] [37], IEEE 802.11 [132–134] and LoRa [102].

A novel approach to manage wireless networks is represented by the SDWLANs, i.e., the SDN-based architectures for WLANs. In [135], the authors presented an overview of SDWLAN architectures, such as Odin [127] [136] and EmPOWER [126] [137], and a qualitative comparison of their features, such as programmability and virtualization. As discussed in [135], EmPOWER [126] is one of the most interesting SDWLAN architecture. For instance, EmPOWER enables a seamless handoff [138] in IEEE 802.11 networks that is completely managed by the control plane, thanks to the use of the Light Virtual Access Point (LVAP) abstraction. However, EmPOWER does not offer support for real-time communication for mission-critical applications, which is essential for industrial process control and factory automation.

The work in [34] proposes SDN-WISE for the realization of programmable IEEE 802.15.4-based WSNs. In [139] SDN-WISE is used to implement the Forwarding and Time Slotted Channel Hopping (TSCH) Scheduling over SDN (FTS-SDN), an SDN-based approach to handle node mobility and scheduling in Industrial WSN running the TSCH protocol of the IEEE 802.15.4 standard. SDN-WISE targets efficient sensor resources utilization and provides low data rate. However, SDN-WISE considers only one communication technology and ignores technologies with high data rates, so it offers a limited support for Industry 4.0 applications.

The Soft-WSN architecture proposed in [128] focuses on both device management and topology management to meet run-time application-specific requirements of IoT, while enhancing flexibility and simplicity of WSN management. Soft-WSN is based on three main entities, i.e., a sensor node, an access point and a controller. The sensor nodes exchange data/commands with the controller through an access point. In particular, Soft-WSN uses an IEEE 802.15.4-based architecture for the communications between sensor nodes and the IEEE 802.11 protocol for the communications between the access points and the controller. Soft-WSN is not currently able to include all types of sensor devices in the network, as it provides support only for IEEE 802.15.4-based sensor nodes. Furthermore, the architecture in [128] is devised for IoT applications, therefore it uses an unpredictable protocol, e.g., the IEEE 802.11, for the communication between the access points and the controller, and it cannot cope with the requirements of industrial applications.

The work in [32] proposes the use of heterogeneous communication technologies integrated in a hierarchical communication and data management architecture consisting of several cells (i.e., subnetworks). Each network node is connected to the cell that is able to efficiently satisfy its communication requirements. Each cell is connected to a central orchestrator through a local manager. The central orchestrator coordinates the network management decisions and ensures the efficient control of the system. However, the approach in [32] does not present an implementation of the proposed architecture, and gives insufficient information about the design of the orchestrator and local manager.

6.3 The proposed network architecture

This chapter proposes a network architecture that aims at providing flexible and efficient connectivity in Industry 4.0 scenarios. The proposed architecture supports the use of heterogeneous wired/wireless communication technologies to efficiently meet the diverse communication requirements of industrial applications. To efficiently integrate different communication technologies in a unique network, I adopt a software-defined networking approach that supports network virtualization, thus providing a flexible and scalable network architecture that is able to support several applications, while managing the application requirements in isolation and the technology-dependent details in a transparent way. In fact, using SDN, the system designer does not need to deal with technology-dependent details, but only with network state information in order to define the desired network behavior. This way, features and services can be seamlessly ported across different technologies. Furthermore, a global view of the network resources is available and bandwidth allocation models, ranging from random access to scheduled access, can be easily defined.

Fig. 6.2 sketches the proposed network architecture, which presents three layers, i.e., the Resource Layer, the Control Layer and the Application Layer. The left side of Fig. 6.2 shows the wireless part of the architecture, while the right side shows the wired part. Below I describe the layers in detail.

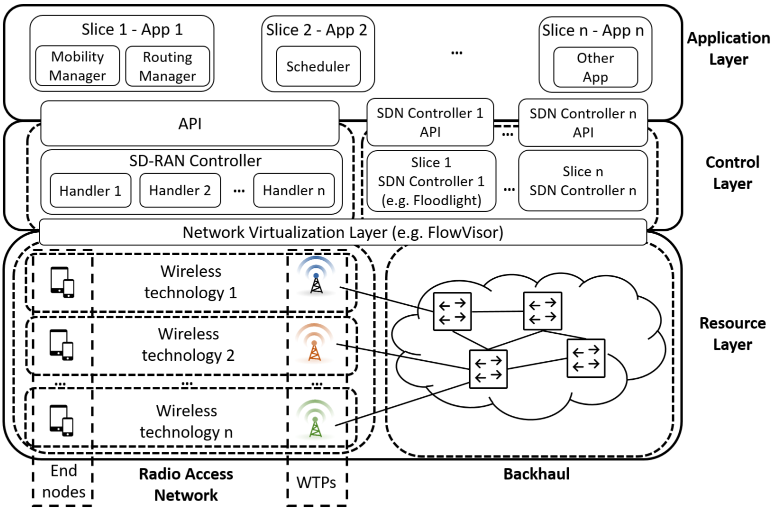

Figure 6.2: The proposed network architecture.

Resource layer. The resource layer includes the end nodes, the Wireless Termination Points (WTPs) and the wired backhaul network. The end nodes are sensors and/or actuators. The WTPs are the physical devices that provide wireless connectivity to end nodes and form the Radio Access Network (RAN). The end nodes are wirelessly connected to the core network through the WTP, that coordinates the management of its local resources across the radio sites based on the controller commands. Consequently, the WTP acts as a sink whose control logic is managed by the SD-RAN controller. For instance, in an IEEE 802.11 network, a WTP coincides with an Access Point (AP). A WTP and its locally managed end nodes realize a subnetwork.

The WTP and the SD-RAN Controller communicate through a wired technology (alternatively, it is also possible to use a wireless technology), i.e., through the backhaul network, using a customized protocol. Conversely, the message exchanges between the WTP and the end nodes are based on a wireless protocol. A customized agent, made up software modules, is implemented within the WTP and acts as a relay between the end nodes and the SD-RAN Controller every time a centralized decision is required. The WTPs are interconnected through the wired backhaul, that consists of wired switches that can be controlled through a standard protocol, such as OpenFlow.

Control layer. The control layer includes two types of controllers, i.e., the SDN Controller (e.g., Floodlight[1]) for the wired part of the network, and the SD-RAN Controller (e.g., a custom version of EmPOWER Runtime[2]) for the wireless part. A Network Virtualization Layer can be inserted between the network infrastructure and the Control Layer in order to abstract the physical network resources, so the latter can be monitored and allocated to each subnetwork, which is also called a slice. This way, each network slice has a separate SDN Controller and the network devices communicate with the SDN Controllers through the network hypervisor (e.g., FlowVisor [140]), again with a standard protocol, e.g., OpenFlow. The SD-RAN Controller has the control of the nodes connected to it and performs several management functions, e.g., radio resource management, mobility management, etc. Although in the proposed network architecture end nodes can belong to multiple network slices, the mechanisms for dealing with such a specific case are not dealt with here and are left for future work.

A custom protocol is used for the communication between the SD-RAN Controller and a WTP. Consequently, the SD-RAN Controller needs to define a protocol and run a handler for each different wireless technology. The Control Layer provides APIs to the Application Layer.

[1] http://www.projectfloodlight.org/floodlight

[2] https://github.com/5g-empower/empower-runtime

Application layer. Using the interface provided by the controller, various applications can be developed in the application layer to coordinate the behavior of a heterogeneous network, improve the network performance and implement functions that are difficult to realize in a traditional architecture. Each Network App runs in its own slice of resources, on top of the SD-RAN Controller, which only presents the network view that corresponds to the specific slice.

6.4 Sample network

This section addresses the feasibility of the proposed architecture in practice. Fig. 6.3 shows an example of a network partitioned into three slices. In each slice, multiple WTP and, therefore, different wireless technologies can coexist. A WTP and its locally managed end nodes realize a subnetwork (SN) within a slice. The coverage range of a subnetwork can overlap both with the other subnetworks within its slice and with the subnetworks belonging to the other slices. Thus, mechanisms to manage interference within the network are required.

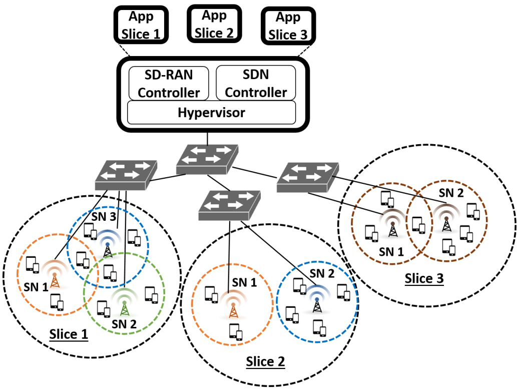

Figure 6.3: A network setup example.

The network setup example, shown in Fig. 6.3, presents a simplified network topology. In order to implement an initial version of the proposed approach, I selected EmPOWER [126] as the reference architecture, for its interesting features [135]. EmPOWER is an open and free code framework that is constantly updated by its developers. It includes a set of programming abstractions, that model the main aspects of a wireless network, i.e., state management, resource provisioning, network monitoring, and network reconfiguration. The provided abstractions hide the implementation details of the underlying wireless technologies, thus the programmers can easily control the network state.

In the following subsection, an overview of EmPOWER is provided. More details can be found online[3].

6.4.1 EmPOWER overview

EmPOWER is an open Mobile Network Operating System provided with a flexible architecture and high-level programming APIs that allow fast prototyping of novel applications.

The key abstractions provided by EmPOWER for wireless networks are listed below.

- *Resource Pool*, that exposes the collective resources available in the network. The allocation unit in the Resource Pool is the Resource Block, i.e., the minimum chunk of wireless resources that can be assigned to a wireless node.

- *Light Virtual Access Point (LVAP)*, that was originally introduced in [127] and provides a high-level interface for wireless node state management. The implementation of this interface handles the technology-dependent details. A LVAP represents the state of a wireless client scheduled on a set of Resource Blocks. A new LVAP is created every time a new client attempts to join the network. Thus, each WTP hosts a number

[3] https://5g-empower.io/

of LVAPs equal to the number of wireless clients that are currently under its control. For example, in a WiFi network the LVAP can be thought as a Virtual AP and handover consists in removing an LVAP from a WTP and instantiating it on another WTP.

- *Channel Quality and Interference Map*, that allows the control logic to assign resources according to the channel quality and interference experienced by the end nodes.

- *Port abstraction*, that models the dynamic and reconfigurable characteristics of the link between the WTP and the end nodes on a set of Resource Blocks. This is because links in a wireless network are stochastic, so the physical layer parameters that characterize the radio link between an end node and a WTP (e.g., transmission power, modulation and coding schemes) must be adapted to the actual channel conditions.

6.5 Implementation roadmap

This section outlines how to implement the proposed architecture as an extension of EmPOWER, discussing some challenges and the relevant solutions.

6.5.1 Introducing support for other technologies

As EmPOWER currently provides support for IEEE 802.11, cellular and wired software-defined networking only, here I address the main steps that are needed to add support for a generic communication technology. The proposed roadmap includes the following steps:

- Definition of a customized version of WTP;

- Development of a handler in the SD-RAN Controller;

- Definition of a customized protocol for the communication between WTP and SD-RAN.

Fig. 6.4 shows a modular concept of WTP and SD-RAN Controller in order to highlight the implementation steps needed to add support for a new technology. In particular, the left side of Fig. 6.4 shows a general modular architecture for WTP that is independent of the wireless technology that the WTP locally manages. A WTP consists of a device with two network interfaces, i.e., a wireless interface and a wired one. The wireless interface can refer to any type of radio interface (e.g., IEEE 802.11, Bluetooth, IEEE 802.15.4, LoRa, etc). The WTP uses the wireless interface to exchange messages with the local wireless network managed through a standard or customized wireless protocol. The wired interface is used to exchange data/commands between the WTP and the SD-RAN Controller through a customized protocol that defines the semantics of the exchanged messages. The core of the WTP architecture consists of a customized agent made up of various modules, e.g., a module that works on TX/RX statistics, a module in charge of forwarding the requests from the end nodes to the SD-RAN Controller, and a module that implements the medium access strategy for the local wireless network based on the commands of the SD-RAN controller. The latter must be customized for supporting the new communication technology, thus several modules must be updated, e.g., the LVAP manager and the bandwidth manager. Moreover, the SD-RAN Controller must provide further APIs to the network programmers and support additional data/commands exchange over the customized protocol used for the communications with the new WTP.

6.5.2 Time-based scheduling

Industrial networks need to meet the real-time requirements imposed by the controlled processes. Several applications require deterministic medium access methods that can be realized by time-based scheduling.

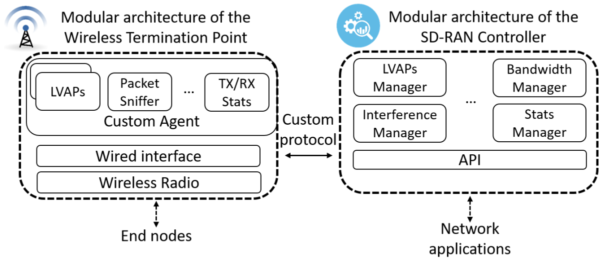

Figure 6.4: The modular architectures of WTP and SD-RAN Controller.

EmPOWER does not address any time-based mechanism, e.g., a Time Division Multiple Access (TDMA) approach, to access the transmission medium. For example, EmPOWER-based IEEE 802.11 networks support only a random channel access scheme, as the Resource Blocks are allocated only in the frequency domain, i.e., no time dimension is provided [141]. I propose to update the Resource Block abstraction in order to add the time dimension on the chunk of wireless resources that an end node can use for its transmissions. Moreover, the SD-RAN Controller must update the bandwidth manager module and provide APIs for the management of the time-based channel access. This way, TDMA-based scheduling algorithms can be easily defined as network applications. The WTP must add a module for the management of the medium access of the locally managed end nodes. The latter must implement time-based transmissions using timers. For example, in an IEEE 802.11 network the WTP can manage access to the medium through the beacon frames, which can contain the time-based schedule information [132]. This way, the end nodes synchronize to the WTP through the beacon frames and use the schedule information contained therein.

6.5.3 Management of mobile nodes

The management and control of wireless networks often involve handling mobile nodes. The LVAP abstraction simplifies network management and introduces seamless mobility support. For instance, in an IEEE 802.11-based network the handover (i.e., when a mobile node moves from one AP to another) is a time-consuming process resulting in network performance degradation, thus reducing the QoS.

EmPOWER currently supports seamless handoff in IEEE 802.11 networks [138] and customized algorithms can be proposed for predictive handover and load balancing, taking advantage of the LVAP abstractions. The LVAP manager must be extended to support the mobility management of each communication technology, thus the SD-RAN Controller must be able to deal with the network state of each LVAP, i.e., each of the wireless clients associated to the network.

6.5.4 Interference management

In a realistic Industry 4.0 scenario, interference between different wireless technologies operating in the same frequency band need to be monitored and controlled. Mechanisms to detect interference from other slices are needed, and slices need to be coordinated to guarantee interworking and coexistence between concurrently operating technologies.

The work in [141] presents a set of high level programming abstractions for channel quality, interference and network reconfiguration for effective interference management in an EmPOWER-based 802.11 network. However, for interference management in generic heterogeneous networks in Industry 4.0 scenarios, an extension of the interference manager must be provided.

6.6 Conclusions

This chapter paves the way to a software-defined networking architecture for virtualized communication networks that is suitable for providing flexible and efficient connectivity in Industry 4.0 scenarios. This chapter also discusses an implementation roadmap of the proposed architecture, based on the EmPOWER mobile network operating system, highlighting design challenges and solutions. Implementation is in progress and the performance of a working prototype of the proposed architecture will be dealt with in future work. In particular, I am using PC Engines apu2 system boards as Wireless Termination Points (WTP) for the first IEEE 802.11-based prototype that will introduce a time-based scheduling mechanism over IEEE 802.11 in order to provide support for real-time communications.

Chapter 7

Conclusions and future works

Currently, low-power wireless networks are not able to satisfactorily cope with all the different requirements imposed by the whole set of industrial applications. The main challenge to face is providing support for real-time communications. Moreover, since in the Industry 4.0 context multiple communication technologies, each one tailored for a specific scenario, are needed, innovative solutions need to target both short-range and long-range technologies, so as to meet the requirements of a broad set of IIoT applications.

As far as short-range communications are concerned, this thesis addressed the challenge of supporting real-time communications on mesh topologies using Bluetooth Low Energy.

The solution proposed in the thesis, called MRT-BLE, is a protocol working on top of Bluetooth Low Energy that provides support for real-time traffic over mesh topologies. A proof-of-concept implementation of the MRT-BLE protocol was realized on STMicroelectronics X-NUCLEO-IDB05A1 devices, and the protocol was tested to validate and demonstrate its effectiveness.

The thesis also proposed two low LoRa-based wireless solutions to support real-time applications for long-range low-power IIoT applications. Starting from an analysis of the current standardized LoRaWAN MAC protocols over LoRa, that highlighted the lack of support for real-time transmission, the first solution proposed in this thesis, called Industrial LoRa, was specifically devised to overcome this limitation.

Thanks to the introduction of a superframe structure made up of five sections to regulate access to the wireless medium, Industrial LoRa supports both real-time and nonreal-time communications. Industrial LoRa exploits a Multi-Channel and Multi-Spreading Factor TDMA to access the channel during the Contention-Free Period (CFP), that is devised for periodic real-time confirmed flows. Moreover, it adopts a Pure ALOHA mechanism for channel access during the Contention Access Period (CAP), that is reserved for the transmission of non-periodic unconfirmed data.

Simulative assessments in a realistic scenario showed that the Pure ALOHA-based medium access strategy for aperiodic messages does not provide them with sufficient reliability. For this reason, the thesis proposed a novel protocol, called RT-LoRa that, compared with Industrial LoRa, introduces a number of extensions and improvements that increase the communication reliability.

In particular, RT-LoRa adopts as the MAC protocol for the CAP a smart version of Slotted ALOHA and it also supports three QoS classes that allow to achieve a trade-off between the reliability of uplink communications and the energy consumption. The results of simulative assessments showed the performance improvements introduced by RT-LoRa. The implementation of RT-LoRa on COTS devices that use the SX1272 and SX1302 LoRa transceivers has already started and it will be addressed in future work.

Finally, the thesis drew a research direction towards software-defined management of heterogeneous Industry 4.0 communication networks, in order to enable flexible network management. The proposal aims to design a software-defined architecture able to support multiple diverse wired and wireless communication technologies.

Future works will deal with novel approaches to improve the network flexibility, such as mechanisms for self-adaptation to topology changes. Moreover, the management of mobile nodes will be addressed with the aim of providing seamless mobility support [138]. In order to provide long range communication to Bluetooth Low Energy clustered nodes, the integration of both Bluetooth Low Energy and LoRa technologies will be addressed. Currently, a device capable of supporting both technologies has already been designed and implemented. The next step will be the proposal of an innovative protocol for a dual-stack network.

Future work will also target software-defined management of heterogeneous real-time virtualized networks. In particular, time-based bandwidth reservation mechanisms for software-defined networks will be proposed. The implementation of a first version of the architecture proposed in Chapter 6 is in progress.

References

[1] L. Seno, F. Tramarin, and S. Vitturi. Performance of Industrial Communication Systems: Real Application Contexts. *IEEE Industrial Electronics Magazine*, 6(2):27–37, June 2012.

[2] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli. Low-Power Wireless for the Internet of Things: Standards and Applications. *IEEE Access*, 6:67893–67926, 2018.

[3] S. Gaglio, G. L. Re, G. Martorella, and D. Peri. A fast and interactive approach to application development on Wireless Sensor and Actuator Networks. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–8, Sep. 2014.

[4] H. Farag, M. Gidlund, and P. Österberg. A Delay-Bounded MAC Protocol for Mission- and Time-Critical Applications in Industrial Wireless Sensor Networks. *IEEE Sensors Journal*, 18(6):2607–2616, March 2018.

[5] S. Gaglio, G. Lo Re, D. Peri, S. D. Vassallo, and G. Martorella. Development of an IoT Environmental Monitoring Application with a Novel Middleware for Resource Constrained Devices. In *Proceedings of the 2nd Conference on Mobile and Information Technologies in Medicine (MobileMed 2014).*

[6] M. Wollschlaeger, T. Sauter, and J. Jasperneite. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, March 2017.

[7] J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *2011 9th IEEE International Conference on Industrial Informatics*, pages 410–415, July 2011.

[8] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen. Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems. *Proceedings of the IEEE*, 104(5):1013–1024, May 2016.

[9] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Transactions on Industrial Informatics*, 14(11):4724–4734, Nov 2018.

[10] F. Tramarin, A. K. Mok, and S. Han. Real-Time and Reliable Industrial Control Over Wireless LANs: Algorithms, Protocols, and Future Directions. *Proceedings of the IEEE*, 107(6):1027–1052, June 2019.

[11] Y. Chen, S. Zhang, S. Xu, and G. Y. Li. Fundamental trade-offs on green wireless networks. *IEEE Communications Magazine*, 49(6):30–37, June 2011.

[12] M. M. Rahman, C. Despins, and S. Affes. Analysis of CAPEX and OPEX benefits of wireless access virtualization. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 436–440, June 2013.

[13] A. Chella, G. Lo Re, I. Macaluso, M. Ortolani, and D. Peri. Multi-robot Interacting Through Wireless Sensor Networks. In R. Basili and M. T. Pazienza, editors, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, pages 789–796, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[14] A. Chella, G. Lo Re, I. Macaluso, M. Ortolani, and D. Peri. A Networking Framework for Multi-Robot Coordination. In Aleksandar Lazinica, editor, *Recent Advances in Multi Robot Systems*, chapter 1. IntechOpen, Rijeka, 2008.

[15] H. Martínez-Barberá and D. Herrero-Pérez. Autonomous navigation of an automated guided vehicle in industrial environments. *Robotics and Computer-Integrated Manufacturing*, 26(4):296 – 311, 2010.

[16] S. Raza, M. Faheem, and M. Günes. Industrial wireless sensor and actuator networks in industry 4.0: Exploring requirements, protocols, and challenges-A MAC survey. *International Journal of Communication Systems*, page e4074, 08 2019.

[17] S. Vitturi, F. Tramarin, and L. Seno. Industrial Wireless Networks: The Significance of Timeliness in Communication Systems. *IEEE Industrial Electronics Magazine*, 7(2):40–51, June 2013.

[18] S. Vitturi, C. Zunino, and T. Sauter. Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G. *Proceedings of the IEEE*, 107(6):944–961, June 2019.

[19] Y. Liu, M. Kashef, K. B. Lee, L. Benmohamed, and R. Candell. Wireless Network Design for Emerging IIoT Applications: Reference Framework and Use Cases. *Proceedings of the IEEE*, 107(6):1166–1192, June 2019.

[20] P. Bartolomeu, M. Alam, J. Ferreira, and J. A. Fonseca. Supporting Deterministic Wireless Communications in Industrial IoT. *IEEE Transactions on Industrial Informatics*, 14(9):4045–4054, Sep. 2018.

[21] R. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert, and N. Montavont. From Best Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks. *IEEE Transactions on Industrial Informatics*, 14(10):4468–4480, Oct 2018.

[22] Jin-Woo Kim and Jae-Wan Kim. Energy efficient clustering algorithm for the mobility support in an IEEE 802.15.4 based wireless sensor network. *Wireless Networks*, 25(6):3441–3452, Aug 2019.

[23] Y. Al-Nidawi, H. Yahya, and A.H. Kemp. Tackling Mobility in Low Latency Deterministic Multihop IEEE 802.15.4e Sensor Network. *IEEE Sensors Journal*, 16(5):1412–1427, Mar. 2016.

[24] S. Vitturi, T. Sauter, and Z. Pang. Real-time networks and protocols for factory automation and process control systems [scanning the issue]. *Proceedings of the IEEE*, 107(6):939–943, June 2019.

[25] Bluetooth SIG. *Bluetooth Core Specification Version 5.0*, Dec. 2016.

[26] Semtech Corporation Wireless Sensing and Timing Products Division. *LoRa^{TM} Modulation Basics*. Semtech, May 2015.

[27] D. Hortelano, T. Olivares, M. C. Ruiz, C. Garrido-Hidalgo, and V. López. From Sensor Networks to Internet of Things. Bluetooth Low Energy, a Standard for This Evolution. *Sensors*, 17(2), 2017.

[28] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things. *IEEE Communications Magazine*, 56(2):60–67, Feb 2018.

[29] E. Sisinni, D. F. Carvalho, P. Ferrari, A. Flammini, D. R. C. Silva, and I. M. D. Da Silva. Enhanced flexible LoRaWAN node for industrial IoT. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, June 2018.

[30] J. Yang, C. Poellabauer, P. Mitra, and C. Neubecker. Beyond beaconing: Emerging applications and challenges of BLE. *Ad Hoc Networks*, 97:102015, 2020.

[31] M. Luvisotto, F. Tramarin, L. Vangelista, and S. Vitturi. On the Use of LoRaWAN for Indoor Industrial IoT Applications. *Wireless Communications and Mobile Computing*, 2018:1–11, 05 2018.

[32] M. C. Lucas-Estañ, T. P. Raptis, M. Sepulcre, A. Passarella, C. Regueiro, and O. Lazaro. A software defined hierarchical communication and data management architecture for industry 4.0. In *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 37–44, Feb 2018.

[33] D. Zhao, M. Zhu, and M. Xu. SDWLAN: A flexible architecture of enterprise WLAN for client-unaware fast AP handoff. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–6, July 2014.

[34] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 513–521, April 2015.

[35] R. Rondón, M. Gidlund, and Krister L. Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications. *International Journal of Wireless Information Networks*, 24(3):278–290, Sep 2017.

[36] Bluetooth SIG. *Mesh Profile Specifications 1.0*, July 2017.

[37] G. Patti, L. Leonardi, and L. Lo Bello. A Bluetooth Low Energy real-time protocol for Industrial Wireless mesh Networks. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 4627–4632, Oct 2016.

[38] N. Varsier and J. Schwoerer. Capacity limits of LoRaWAN technology for smart metering applications. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[39] D. Magrin, M. Centenaro, and L. Vangelista. Performance evaluation of LoRa networks in a smart city scenario. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2017.

[40] R. Sanchez-Iborra and M.D. Cano. State of the Art in LP-WAN Solutions for Industrial IoT Services. *Sensors*, 16(5), 2016.

[41] J. Wan, S. Tang, Z. Shu, D. Li, S. Wang, M. Imran, and A. V. Vasilakos. Software-Defined Industrial Internet of Things in the Context of Industry 4.0. *IEEE Sensors Journal*, 16(20):7373–7380, Oct 2016.

[42] Bluetooth SIG. *Bluetooth Core Specification Version 4.1*, Dec. 2013.

[43] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen. How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4. In *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 232–237, April 2012.

[44] E. Toscano and L. Lo Bello. Comparative assessments of IEEE 802.15.4/ZigBee and 6LoWPAN for low-power industrial WSNs in realistic scenarios. In *2012 9th IEEE International Workshop on Factory Communication Systems*, pages 115–124, Lemgo, Germany, May 2012.

[45] Bluetooth SIG. *Bluetooth Core Specification Version 4.2*, Dec. 2014.

[46] M. Collotta and G. Pau. A Solution Based on Bluetooth Low Energy for Smart Home Energy Management. *Energies*, 8:11916–11938, 10 2015.

[47] M. Collotta and G. Pau. A Novel Energy Management Approach for Smart Homes Using Bluetooth Low Energy. *IEEE Journal on Selected Areas in Communications*, 33(12):2988–2996, Dec 2015.

[48] L. Lo Bello, O. Mirabella, and N. Torrisi. Modelling and evaluating traceability systems in food manufacturing chains. In *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 173–179, June 2004.

[49] B. Bordel Sánchez, R. Alcarria, D. Martín, and T. Robles. TF4SM: A Framework for Developing Traceability Solutions in Small Manufacturing Companies. *Sensors*, 15(11):29478–29510, 11 2015.

[50] J. Medina-García, T. Sánchez-Rodríguez, J. A. G. Galán, A. Delgado, F. Gómez-Bravo, and R. Jiménez. A Wireless Sensor System for Real-Time Monitoring and Fault Detection of Motor Arrays. *Sensors*, 17(3):469, 02 2017.

[51] L. Leonardi, G. Patti, F. Battaglia, and L. Lo Bello. Simulative assessments of the IEEE 802.15.4 CSMA/CA with Priority Channel Access in structural health monitoring scenarios. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 375–380, July 2017.

[52] P. Trelsmo, P. Di Marco, P. Skillermark, R. Chirikov, and J. Ostman. Evaluating IPv6 Connectivity for IEEE 802.15.4 and Bluetooth Low Energy. In *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6, March 2017.

[53] R. Natarajan, P. Zand, and M. Nabi. Analysis of coexistence between IEEE 802.15.4, BLE and IEEE 802.11 in the 2.4 GHz ISM band. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6025–6032, Oct 2016.

[54] M. H. Dwijaksara, W. S. Jeon, and D. G. Jeong. A channel access scheme for Bluetooth low energy to support delay-sensitive applications. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, Sep. 2016.

[55] K. Cho, G. Park, W. Cho, J. Seo, and K. Han. Performance analysis of device discovery of Bluetooth Low Energy (BLE) networks. *Computer Communications*, 81:72 – 85, 2016.

[56] J. Seo, K. Cho, W. Cho, G. Park, and K. Han. A discovery scheme based on carrier sensing in self-organizing Bluetooth Low Energy networks. *Journal of Network and Computer Applications*, 65:72 – 83, 2016.

[57] M. Marinoni, A. Biondi, P. Buonocunto, G. Franchino, D. Cesarini, and G. Buttazzo. Real-Time Analysis and Design of a Dual Protocol Support for Bluetooth LE Devices. *IEEE Transactions on Industrial Informatics*, 13(1):80–91, Feb 2017.

[58] J. Kim, S. Kang, and J. Park. Bluetooth-based tree topology network for wireless industrial applications. In *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, pages 1305–1308, Oct 2015.

[59] C. Jung, K. Kim, J. Seo, B. N. Silva, and K. Han. Topology Configuration and Multihop Routing Protocol for Bluetooth Low Energy Networks. *IEEE Access*, 5:9587–9598, May 2017.

[60] P. Zenker, S. Krug, M. Binhack, and J. Seitz. Evaluation of BLE Mesh capabilities: A case study based on CSRMesh. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 790–795, July 2016.

[61] D. Hortelano, T. Olivares, M. C. Ruiz, C. Garrido-Hidalgo, and V. López. From Sensor Networks to Internet of Things. Bluetooth Low Energy, a Standard for This Evolution. *Sensors*, 17(2), 2017.

[62] Qualcomm Technologies International. *Qualcomm Bluetooth Low Energy Solutions, CSRmesh and CSR1010 (Datasheet 87-CE855-1 Rev A)*, Sept. 2016.

[63] S. M. Darroudi and C. Gomez. Bluetooth Low Energy Mesh Networks: A Survey. *Sensors*, 17(7), 2017.

[64] S. Raza, P. Misra, Z. He, and T. Voigt. Bluetooth smart: An enabling technology for the Internet of Things. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 155–162, Oct 2015.

[65] T. Zhang, J. Lu, F. Hu, and Q. Hao. Bluetooth low energy for wearable sensor-based healthcare systems. In *2014 IEEE Healthcare Innovation Conference (HIC)*, pages 251–254, Seattle, USA, October 2014.

[66] G. Pau, M. Collotta, and V. Maniscalco. Bluetooth 5 Energy Management through a Fuzzy-PSO Solution for Mobile Devices of Internet of Things. *Energies*, 10(7), 2017.

[67] R. Rondón, K. Landernäs, and M. Gidlund. An analytical model of the effective delay performance for Bluetooth low energy. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, Sep. 2016.

[68] M. Grover, S. K. Pardeshi, N. Singh, and S. Kumar. Bluetooth low energy for industrial automation. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pages 512–515, Feb 2015.

[69] STMicroelectronics. *BlueNRG, BlueNRG-MS stacks programming guidelines (Programming manual PM0237)*, Dec. 2016.

[70] STMicroelectronics. *BlueNRG-MS development kits (User manual UM1870)*, May 2016.

[71] G. A. Kaczynski, L. Lo Bello, and T. Nolte. Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems. In *2007 IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2007)*, pages 101–110, Sep. 2007.

[72] M. Joseph and P. Pandya. Finding Response Times in a Real-Time System. *The Computer Journal*, 29(5):390–395, 01 1986.

[73] J. P. Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *[1990] Proceedings 11th Real-Time Systems Symposium*, pages 201–209, Dec 1990.

[74] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, Apr 2007.

[75] STMicroelectronics. *X-NUCLEO-IDB05A1 - Bluetooth Low Energy expansion board based on the SPBTLERF module for STM32 Nucleo*, July 2015.

[76] E. Toscano and L. Lo Bello. A topology management protocol with bounded delay for Wireless Sensor Networks. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 942–951, Sep. 2008.

[77] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund. Using LoRa for industrial wireless networks. In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, May 2017.

[78] A. Khan and K. Turowski. A Survey of Current Challenges in Manufacturing Industry and Preparation for Industry 4.0. In Ajith Abraham, Sergey Kovalev, Valery Tarassov, and Václav Snášel, editors, *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'16)*, pages 15–26, Cham, 2016. Springer International Publishing.

[79] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, and L. Lo Bello. Simulative assessments of IEEE 802.1 Ethernet AVB and Time-Triggered Ethernet for Advanced Driver Assistance Systems and in-car infotainment. In *2012 IEEE Vehicular Networking Conference (VNC)*, pages 187–194, Nov 2012.

[80] R. Sanchez-Iborra, J. Sánchez-Gómez, J. Santa, P. J. Fernández, and A. Skarmeta. Integrating LP-WAN communications within the vehicular ecosystem. *J. Internet Serv. Inf. Secur*, 7:45–56, 2017.

[81] G. Alderisi, G. Iannizzotto, and L. Lo Bello. Towards IEEE 802.1 Ethernet AVB for Advanced Driver Assistance Systems: A preliminary assessment. In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012)*, pages 1–4, Sep. 2012.

[82] LoRa Alliance Technical Committee. *LoRaWAN^{TM} 1.1 Specification.* LoRa Alliance, October 2017.

[83] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, 55(9):34–40, Sep. 2017.

[84] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley. A Study of LoRa: Long Range amp; Low Power Networks for the Internet of Things. *Sensors*, 16(9), 2016.

[85] G. Pasolini, C. Buratti, L. Feltrin, F. Zabini, C. De Castro, R. Verdone, and O. Andrisano. Smart city pilot projects using lora and ieee802.15.4 technologies. *Sensors*, 18:1118, 04 2018.

[86] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi. Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, October 2016.

[87] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen, and J. Iinatti. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. In *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–5, March 2016.

[88] G. Capizzi and G. Tina. Long-term operation optimization of integrated generation systems by fuzzy logic-based management. *Energy*, 32(7):1047 – 1054, 2007.

[89] L. Leonardi, G. Patti, and L. Lo Bello. Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks. *IEEE Access*, 6:26505–26519, 2018.

[90] G. Alderisi, G. Patti, O. Mirabella, and L. Lo Bello. Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic process automation scenarios. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 948–955, July 2015.

[91] LoRa Alliance Technical Committee Regional Parameters Workgroup. *LoRaWAN Regional Parameters*. LoRa Alliance, January 2018.

[92] O.B.A. Seller and N. Sornin. Low power long range transmitter, August 6 2014. EP Patent App. EP20,130,154,071.

[93] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso. Do LoRa Low-Power Wide-Area Networks Scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '16, pages 59–67, New York, NY, USA, 2016. ACM.

[94] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen. Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology. In *European Wireless 2016; 22th European Wireless Conference*, pages 1–6, May 2016.

[95] ETSI. *Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment* , Sept 2017.

[96] M. Slabicki, G. Premsankar, and M. Di Francesco. Adaptive configuration of lora networks for dense IoT deployments. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, April 2018.

[97] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke. LoRa indoor coverage and performance in an industrial environment: Case study. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, Sep. 2017.

[98] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni. Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications. *IEEE Transactions on Instrumentation and Measurement*, 66(12):3340–3349, Dec 2017.

[99] L. Beltramelli, A. Mahmood, M. Gidlund, P. Österberg, and U. Jennehag. Interference Modelling in a Multi-Cell LoRa System. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, Oct 2018.

[100] M. Bor, J. Vidler, and U. Roedig. LoRa for the Internet of Things. In *EWSN 2016 - International Conference on Embedded Wireless Systems and Networks*, pages 361–366, USA, 2016. Junction Publishing.

[101] L. Tessaro, C. Raffaldi, M. Rossi, and D. Brunelli. Lightweight Synchronization Algorithm with Self-Calibration for Industrial LORA Sensor Networks. In *2018 Workshop on Metrology for Industry 4.0 and IoT*, pages 259–263, April 2018.

[102] L. Leonardi, F. Battaglia, G. Patti, and L. Lo Bello. Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 4141–4146, Oct 2018.

[103] I. A. Ismaili, A. Azyat, N. Raissouni, N. Ben Achhab, A. Chahboun, and M. Lahraoua. Comparative Study of ZigBee and 6LoWPAN Protocols: Review. In *Third International Conference on Computing and Wireless Communication Systems, ICCWCS*. EAI, 5 2019.

[104] L. F. Schrickte, C. Montez, R. d. Oliveira, and A. R. Pinto. Integration of Wireless Sensor Networks to the Internet of Things Using a 6LoWPAN Gateway. In *2013 III Brazilian Symposium on Computing Systems Engineering*, pages 119–124, Dec 2013.

[105] L. F. Del Carpio, P. Di Marco, P. Skillermark, R. Chirikov, and K. Lagergren. Comparison of 802.11ah, BLE and 802.15.4 for a Home Automation Use Case. *International Journal of Wireless Information Networks*, 24(3):243–253, Sep 2017.

[106] M. Syafrudin, K. Lee, G. Alfian, J. Lee, and J. Rhee. Application of Bluetooth Low Energy-Based Real-Time Location System for Indoor Environments. In *Proceedings of the 2018 2Nd International Conference on*

Big Data and Internet of Things, BDIOT 2018, pages 167–171, New York, NY, USA, 2018. ACM.

[107] W. Du, D. Navarro, and F. Mieyeville. Performance Evaluation of IEEE 802.15.4 Sensor Networks in Industrial Applications. *Int. J. Commun. Syst.*, 28(10):1657–1674, July 2015.

[108] U. Raza, P. Kulkarni, and M. Sooriyabandara. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys Tutorials*, 19(2):855–873, Secondquarter 2017.

[109] R. Piyare, A. Murphy, M. Magno, and L. Benini. On-Demand LoRa: Asynchronous TDMA for Energy Efficient and Low Latency Communication in IoT. *Sensors*, 18:3718, 11 2018.

[110] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J. Prévotet. Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility. *IEEE Communications Surveys Tutorials*, 21(2):1561–1581, Secondquarter 2019.

[111] F. Battaglia and L. Lo Bello. A novel jxta-based architecture for implementing heterogenous networks of things. *Computer Communications*, 116:35–62, Jan 2018.

[112] M. Bor and U. Roedig. LoRa Transmission Parameter Selection. In *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 27–34, June 2017.

[113] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, June 2018.

[114] T. Polonelli, D. Brunelli, and L. Benini. Slotted ALOHA Overlay on LoRaWAN - A Distributed Synchronization Approach. In *2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 129–132, Oct 2018.

[115] J. Haxhibeqiri, I. Moerman, and J. Hoebeke. Low Overhead Scheduling of LoRa Transmissions for Improved Scalability. *IEEE Internet of Things Journal*, 6(2):3097–3109, April 2019.

[116] M. Magno, F. A. Aoudia, M. Gautier, O. Berder, and L. Benini. WULoRa: An energy efficient IoT end-node for energy harvesting and heterogeneous communication. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 1528–1533, March 2017.

[117] L. Tessaro, C. Raffaldi, M. Rossi, and D. Brunelli. LoRa Performance in Short Range Industrial Applications. In *2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, pages 1089–1094, June 2018.

[118] O. Georgiou and U. Raza. Low Power Wide Area Network Analysis: Can LoRa Scale? *IEEE Wireless Communications Letters*, 6(2):162–165, April 2017.

[119] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance. *IEEE Communications Letters*, 22(4):796–799, April 2018.

[120] P. Ferrari, A. Flammini, M. Rizzi, E. Sisinni, and M. Gidlund. On the evaluation of LoRaWAN virtual channels orthogonality for dense distributed systems. In *2017 IEEE International Workshop on Measurement and Networking (M N)*, pages 1–6, Sep. 2017.

[121] Semtech. *SX1272/73 - 860 MHz to 1020 MHz low power long range transceiver*, Janaury 2019.

[122] Semtech. *SX1302 – LoRa Gateway Baseband Transceiver*, June 2019.

[123] A. E. Kalør, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski. Network Slicing in Industry 4.0 Applications: Abstraction Methods and End-to-End Analysis. *IEEE Transactions on Industrial Informatics*, 14(12):5419–5427, Dec 2018.

[124] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.F. Lai, and S. Wang. A review of industrial wireless networks in the context of Industry 4.0. *Wireless Networks*, 23(1):23–41, Jan 2017.

[125] S. Aglianò, M. Ashjaei, M. Behnam, and L. Lo Bello. Resource management and control in virtualized SDN networks. In *2018 Real-Time and Embedded Systems and Technologies (RTEST)*, pages 47–53, May 2018.

[126] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed. Programming Abstractions for Software-Defined Wireless Networks. *IEEE Transactions on Network and Service Management*, 12(2):146–162, June 2015.

[127] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao. Towards Programmable Enterprise WLANS with Odin. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, pages 115–120, New York, NY, USA, 2012. ACM.

[128] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat. Soft-WSN: Software-Defined WSN Management System for IoT Applications. *IEEE Systems Journal*, 12(3):2074–2081, Sep. 2018.

[129] T. Luo, H. Tan, and T. Q. S. Quek. Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks. *IEEE Communications Letters*, 16(11):1896–1899, November 2012.

[130] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[131] G. Alderisi, S. Girs, L. Lo Bello, E. Uhlemann, and M. Björkman. Probabilistic scheduling and Adaptive Relaying for WirelessHART networks. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–4, Sep. 2015.

[132] Y. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka. RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications. In *2013 IEEE 34th Real-Time Systems Symposium*, pages 140–149, Dec 2013.

[133] G. Patti, G. Alderisi, and L. Lo Bello. SchedWiFi: An innovative approach to support scheduled traffic in ad-hoc industrial IEEE 802.11 networks. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–9, Sep. 2015.

[134] Z. Yang, J. Zhang, K. Tan, Q. Zhang, and Y. Zhang. Enabling TDMA for today's wireless LANs. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1436–1444, April 2015.

[135] B. Dezfouli, V. Esmaeelzadeh, J. Sheth, and M. Radi. A Review of Software-Defined WLANs: Architectures and Central Control Mechanisms. *IEEE Communications Surveys Tutorials*, pages 1–1, 2018.

[136] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz. Programmatic Orchestration of WiFi Networks. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, pages 347–358, Berkeley, CA, USA, 2014. USENIX Association.

[137] R. Riggio, T. Rasheed, and F. Granelli. EmPOWER: A Testbed for Network Function Virtualization Research and Experimentation. In *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, pages 1–5, Nov 2013.

[138] S. Martinez, N. Cardona, and J. F. Botero. Seamless Handoff Management in IEEE 802.11 Networks Using SDN. In *2018 IEEE Colombian Conference on Communications and Computing (COLCOM)*, pages 1–6, May 2018.

[139] L. Lo Bello, A. Lombardo, S. Milardo, G. Patti, and M. Reno. Software-Defined Networking for Dynamic Control of Mobile Industrial Wireless Sensor Networks. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 290–296, Sep. 2018.

[140] R. Sherwood, G. Gibb, K.k. Yap, M. Casado, N. Mckeown, and G. Parulkar. FlowVisor: A Network Virtualization Layer. Technical report, 2009.

[141] R. Riggio, M. K. Marina, and T. Rasheed. Interference management in software-defined mobile networks. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 626–632, May 2015.

Biography

Luca Leonardi received the M.S. degree (*summa cum laude*) in Computer Engineering at the University of Catania, Italy, in March 2016. At the end of 2016, he was awarded a PhD scholarship at the University of Catania, that is funded by Operational Programme European Social Fund Sicily 2014-2020.

Since the end of 2015, Luca Leonardi has been working with Prof. Lucia Lo Bello in the field of real-time industrial networks, Wireless Sensor and Actuators Networks (WSANs), and Industrial Internet of Things. He was a Visiting PhD student at Mälardalen University, Västerås, Sweden, from September 2018 to March 2019, where he started to work on the application of Software-Defined Networking (SDN) and network virtualization for the management of communication networks in Industry 4.0 scenarios.

Luca Leonardi attended some IEEE international conferences, where he presented his works.

Luca Leonardi serves as reviewer for several international conferences (e.g., ETFA 2019, IECON 2018, GIIS 2018, etc.), and journals (e.g., the IEEE Transactions on Industrial Informatics, IEEE Access, the IEEE Industrial Electronics Magazine, etc.).