Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Elettronica, Elettrotecnica e delle Telecomunicazioni

---

Optimization of communication protocols for low-power Wireless Sensor Networks

Ph.D. Dissertation of:
Antonio Del Campo

Advisor:
Prof. Ennio Gambi

Coadvisor:
Prof. Susanna Spinsante

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Elettronica, Elettrotecnica e delle Telecomunicazioni

---

Optimization of communication protocols for low-power Wireless Sensor Networks

Ph.D. Dissertation of:
Antonio Del Campo

Advisor:
Prof. Ennio Gambi

Coadvisor:
Prof. Susanna Spinsante

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

To my Grandparents.

Acknowledgments

I would like to thank my academic tutors Prof. Ennio Gambi and Dr. Susanna Spinsante for their trust, support and insights during the Ph.D. school. Thanks to Adelmo De Santis, for the consistent support in all technical issues. With them and the whole "Telecommunication Systems Team" I grew up, both socially and professionally. I am grateful to all the collegues of "q165-Ph.D. Hall" (i.e. Sala dottorandi di quota 165) and the whole "pausers" (i.e. pauseggiatori) for the brain-refreshing coffee breaks and lunches, for the empathy, and sympathy. Finally, I would like to thank Sabrina, my family, my nephews Arianna and Diego, and, my angel Grandparents. The love of each of them gave and will give me the strength to build the future.

Ancona, Gennaio 2017

Antonio Del Campo

Abstract

With the introduction of low-power wireless technologies, new applications are approaching the contexts of home-, building- and industrial- automation, health-care monitoring, automotive, consumer electronic markets and, generally, Internet of Things (IoT). In these scenarios, heterogeneous Wireless Sensor Networks (WSN) can be deployed and integrated, matching design requirements with several system features. Nowadays, the devices communicating within the last 100 meters are typically not interoperable and, rarely, connected to Internet, but more than the 90% of IoT market size concerns the BAN, PAN, LAN design. These classes of WSNs constitute the natural solution for supporting user-environment interactions in smart as well as in assisted spaces. This Thesis is mainly focused on the WSN communication modeling: among them, Bluetooth Low Energy (BLE) is firstly taken into account, being a protocol suitable for short range applications, due to its low-cost and low-energy features, as well as the widespread integration in smartphone and many other mobile devices, that provide an easily available tunnel to Internet. Together with the connection-oriented scheme of BLE, the semplicity of its connectionless mode makes it a good choice for long-term monitoring in a residential as well as in assisted environments. Regarding the first topic, the goal is to model and properly design the BLE at both the application- and link- layer in a peculiar streaming scenario, characterized by low throughput, but strict reliability requirement. The result of this activity is the development of a wearable, consisting of sensorized shoes connected via BLE to a smartphone, forming a classic connection-oriented star-topology. This study shows that BLE fulfills the requirements, opening the way to further investigations on power management analysis, and the impact of increasing throughput on the reliability requirements. The second work deals with the neighbor discovery procedure, implemented by BLE, and models the energy-efficiency of the Periodic-Random-Access-to-Link-Layer (PRALL) protocol, that is modeled through a deterministic approach. The comparison with the simulation results, shows that the developed model has the capability to evaluate the discovery-latency for the most of parameterizations, and is a useful tool to point out and better understand the drawbacks of PRALL protocol for some parameter configurations. The considerations derived by the results discussion are the basis for the development of a modified version of PRALL, denoted by

mPRALL. The comparison of the performed simulations, clearly, shows the improved performance of mPRALL against the standard PRALL, expecially for the worst parameterizations. The third work studies BLE in scenarios where the connectionless paradigm is recommended. The use of the PRALL also for data delivery leads to the characterization of the medium access management of the central node in the star topology, focusing on the Medium Access Control (MAC) scheduling procedure in both connectionless and connection-oriented communication. The developed models are merged into a single tool, intended to properly design device parameters, maintaining, for example, useful timing for discovery process of event-driven sensors and avoiding undesired overlaps between the scheduled scan and connection duty-cycles, due to inaccurate parameters configuration. In the fourth and last topic, the adoption of data-centric communication paradigm by one of the most promising light-weight alternative to HTTP (Hyper-Text Transfer Protocol) for IoT, i.e. Message Queue Telemetry Transport (MQTT), is investigated, firstly, in an Ambient Assisted Living (AAL) scenario and, secondly, in a building automation context. In the first work three possible architectures featuring different degrees of integration of the MQTT protocol into the technology platform are proposed. The obtained results, and the power consumption estimation provided for each case, show that MQTT may be effectively adopted for a quick and reliable distribution of notification messages among the different actors involved by the platform. In the second work, the MQTT data-structure is adopted over LoRa physical communication link to support real-time building automation services. Extensive experimental measurement campaigns proves that LoRa is well suited to ensure adequate radio coverage in indoor scenarios, even in big and bulky buildings, resulting in a received signal strength well over the receiver sensitivity.

Sommario

L'introduzione delle tecnologie low-power sul mercato ha portato allo sviluppo di nuove applicazioni nei settori della domotica, dell'automazione industriale, dell'healthcare, dell'automotive, dell'elettronica di consumo, e più in generale dell'Internet delle cose (IoT). In ognuno dei suddetti contesti le reti di sensori wireless (WSN) devono essere progettate, affinché rispondano alle particolari esigenze e carattetistiche richieste dal contesto. Ad oggi, i dispositivi che comunicano entro un raggio di 100 metri sono tipicamente non interoperabili, e raramente dispongono di connessione ad Internet; eppure le applicazioni a corto-raggio (BAN,PAN,LAN) rappesentano il 90% della domanda del mercato. Le reti di sensori a corto raggio costituiscono la naturale soluzione allo sviluppo di applicazioni per l'interazione con l'uomo in ambienti domotici e assistivi (AAL). In ragione di quanto detto, la presente Tesi è principalmente focalizzata sulla valutazione e la modellazione delle WSN con particolare enfasi su quelle che implementano il protocollo Bluetooth Low Energy (BLE), il quale rappresenta una valida soluzione per la comunicazione a corto-raggio, offrendo basso costo dei sensori e bassi consumi, anche se usati in connessione permanente. Altra fondamentale caratteristica è la sua integrazione in gran parte degli smartphone e dispositivi portatili, i quali forniscono la connessione remota (verso Internet) a sensori che nascono per uso locale. Un altro punto di forza del BLE deriva dalla semplicità del protocollo di communicazione senza connessione, che riduce notevolmente i consumi, rendendo i sensori BLE particolarmente adatti ad applicazioni di monitoraggio continuo in ambienti residenziali e assistiti. Il primo argomento affrontato riguarda la progettazione di un sistema di sensori indossabili, le scarpe intelligenti (smart shoes), connesse tramite BLE ad uno smartphone in una classica rete a stella. Sulla base del suddetto sistema, si è valulata l'appropriatezza del protocollo BLE per applicazioni come questa, in cui è richiesto da un lato un basso throughput, implementabile anche con un'architettura senza connessione, ma dall'altro un'affidabilità del dato, ottenibile, invece, col paradigma di connessione persistente. In particolare, si è utilizzato il secondo approccio, ottimizzando la configurazione dei parametri del protocollo nello scenario descritto. Il secondo lavoro riguarda lo studio della procedura di scansione, implementata dal BLE per la ricerca dei nodi nelle vicinanze o per la comunicazione dei dati senza connessione. Si è quindi sviluppato un modello deterministico del protocollo di scansione

(Periodic-Random-Access-to-Link-Layer - PRALL). Tale modello, rapportato ai dati del simulatore, si è dimostrato sufficientemente accurato nella stima della latenza media di scansione, ed ha messo in evidenza alcuni limiti del PRALL, sui quali è stata progettata una proposta di modifica al protocollo, indicata con l'acronimo mPRALL. Tale modifica, implementate al simulatore, si è dimostrata molto performante in termini di abbassamento dei massimi di latenza. Il terzo argomento inquadra il BLE in uno scenario dove è richiesto che entrambe le architetture, con connessione o senza, coesistano e condividano lo stesso mezzo trasmissivo. Il problema di schedulazione dell'accesso al mezzo è affrontato modellando il PRALL con un modello probabilistico, e la TDMA (Time Division Multiple Access) con un modello deterministico. I due modelli sono stati integrati, ottenendo un valido strumento per la fase di progettazione delle reti miste BLE, in cui il nodo centrale deve gestire entrambe le architetture, il cui fine è quello di trovare un compromesso tra un'appropriata latenza di scansione e una perdita contenuta di pacchetti per i nodi sensore in connessione. Il quarto e ultimo argomento analizza l'adozione del paradigma di comunicazione data-centrico da parte di una delle più interessanti e valide alternative all'HTTP (Hyper-Text Transfer Protocol), ovvero il protocollo MQTT (Message Queue Telemetry Transport). In una prima applicazione per ambienti assistivi (Ambient Assisted Living - AAL) è stata valutata la sua integrazione a vari livelli dell'architettura, come gateway verso Internet o integrato nativamente sui sensori. Nella seconda applicazione, per automazione degli edifici, la struttura dei dati del protocollo MQTT è stata integrata per la comunicazione su mezzo fisico LoRa. Le campagne di misura realizzate hanno dimostrato che LoRa è in grado di assicurare un'adeguata copertura di segnale negli scenari indoor, compresi i grandi edifici.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

With the introduction of low-power wireless technologies, new applications are approaching the context of home-, building- and industrial- automation, healthcare monitoring, automotive, consumer electronic markets and, generally, Internet of Things (IoT). In these scenarios, heterogeneous Wireless Sensor Networks (WSN) can be deployed and integrated, matching design requirements with system features, such as coverage area, nodes number and topology, energy efficiency, quality of service (QoS), security, data aggregation and others. The Figure 1.1, borrowed from [1], distinguishes five classes of WSNs, basing on the transmission range: body (BAN), personal (PAN), local (LAN), metropolitan (MAN), and wide (WAN) area networks. Among these, BAN, PAN and LAN wireless networks constitute the natural solution for supporting user-environment interactions in a smart or assisted space, within the last 100 meters. Nowadays, the devices that communicate in this range are typically not interoperable and, rarely, connected to Internet, but more than the 90% of IoT market size falls within the field of the short-range technologies.

For this reason, the research activity discussed within this Thesis will be focused on the technologies for implementing these architectures. Among the variety of communication technologies, the Bluetooth in its low energy declination, i.e. BLE, emerged as a de facto standard, to interface wearable devices in BANs, i.e. within the "last 10 feet" [2]. While meshing is foreseen in the latest version of BLE, it usually pairs devices to mobile phone in a star topology network. Generally, the mobile phone acts as a gateway from body/personal/local area to Internet [3]. Together with the connection-oriented scheme of BLE, the semplicity of its connectionless scheme makes it a good choice for long-term monitoring in a residential as well as in assisted environments (PAN/LAN), which neither require wide signal-coverage nor dense or ultra-dense networks [4]. However, the main key for popularity of BLE is its own low power capabilities, despite the small sizes of the devices which deploy it. In fact, the wireless nature, the environmental constraints and the scalability features of WSNs

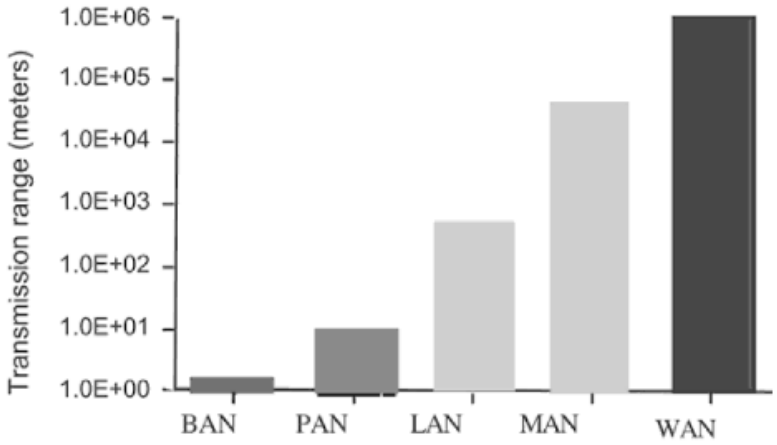


Figure 1.1: Networks taxonomy.

usually require the use of low-power, low-cost and battery powered devices. Advances in microelectronics speeded up the development of low-cost tiny sensor nodes that are equipped with sensing, processing, and communication units. A major power consuming component of a sensor node is the communication unit (radio), which is controlled by the MAC protocol, hence the energy efficiency of MAC scheduling results as one of the fundamental criteria in the WSN protocols design. The reduction of the radio power consumption can be reached through mainly three approaches [5]:

(i) using power-aware routing protocols [6];

(ii) duty-cycling [7];

(iii) in-network/in-node data-reduction [8].

Duty-cycling is one of the techniques for saving power which is consumed by the peripheral when it is turned on during its operating period [9]. The duty-cycling works at three different layers: MAC layer, network layer and application layer. The process of switching nodes can be controlled by two techniques: external wake-up signal and self-switching. When the switching of the node is controlled by a controlling node, the former must stay awake for receiving the control signal from the latter. It is the case of the actuator nodes, which, usually, have to wait for asynchronous commands, spending the most of the energy budget in radio operations. A way to reduce the wake-up time is provided by synchronizing the medium access, as for the sensors that must periodically send data and, hence, require an end-to-end communication with the sink node, in order to optimize the duty-cycle operations, avoiding collisions. But, synchronized

schemes, as Time Division Multiple Access (TDMA), lead to overhead for keeping time synchronization, and, this deteriorate the energy-efficiency. Whenever the wakeup time is aperiodic, the interrupts from the peripherals connected to the node may act as itself-wake-up signal. This operating paradigm is adopted by the event-driven sensors, which turn on the communication-unit in response to changes in sensing-unit state or specific processing-unit tasks. The event-driven paradigm is only one of the data delivery models exploited by nodes communicating with the sink ones. The other forms are: continuous mode, in which data is periodically sent by the sensor, query-driven mode, in which data is transmitted only when a query takes place, and hybrid mode, that takes features among the other modes. The data delivery models [10] influence the routing protocol [11] as well as the upper layer protocols. As opposed to the address-centric routing, in data-centric routing, queries are sent to specific physical or logical areas, and the reached sensors send the data back to the sink. As suggested in [12], the data-centric communication seems either more efficient and more appropriate for wireless sensor networks (WSNs). Message Queue Telemetry Transport (MQTT) exploits the data-centric paradigm [13], playing the role of communication middleware, to hide the complexity of the low-level networking protocols, and facilitate the development of applications and their interoperability [14].

In the aforementioned scenario, the objectives of this thesis include:

1. the design optimization for the BLE-based wearable system [15, 16, 17], mainly through the in-node data-reduction and the modeling of MAC-scheduling;

2. the study and the modeling of the Periodic Random Access to Link Layer Protocol (PRALL) [18], that is operated by BLE devices in connectionless network architecture;

3. the improvement of PRALL in terms of discovery procedure efficiency;

4. the modeling of the medium access management, operated by the central node, communicating with connected and not-connected devices and the development of a useful tool supporting the configuration design of BLE devices [18].

5. the implementation of MQTT communication in two different use cases: (i) a monitoring system in Ambient Assisted Living (AAL) [19], (ii) a building-automation system exploiting MQTT data structure over LoRa (Long Range) physical layer for Low Power Wide Area Network (LP-WAN) [20].

1.2 Thesis Structure

The thesis is organized in seven chapters.

Chapter 2 reviews the state-of-art of WSN MAC protocols with particular emphasis on duty-cycled MAC protocol and introduces one of the most popular protocol for Personal Area Networks (PAN), i.e. the Bluetooth Low Energy. Finally, a brief description of the physical layer of the emerging technology for LPWAN LoRa is provided.

Chapter 3 describes the BLE-based application developed to enable data communications between wearable sensors (a pair of smart shoes) and a smartphone. It outlines the designed efficient data transfer, implemented to ensure adequate throughput and reliability, while preserving the low-power behavior of the node, and, finally, provides the test results about the performances evaluation.

Chapter 4 details the study and the modeling of the discovery procedure, deployed by BLE and here denoted as Periodic Random Access to Link Layer (PRALL). Firstly, it describes the proposed deterministic-model, and finally, provides a discussion about the results, ending with the proposal of an improved version of PRALL, aimed at limiting the discovery-latency for some critical parameterizations.

Chapter 5 describes the proposed models of connectionless and connection-oriented BLE communication, and, introduces the developed tool for parameters, aimed at balancing and optimizing the BLE-network setting.

Chapter 6 provides an overview of middleware solutions designed for Internet of Things (IoT) in health and wellness domains, and presents how the MQTT, can be effectively applied in assistive scenarios too, with different architectural options and communication technologies. Furthermore, it provides the evaluation of the MQTT payload structure usage over LoRa physical communication link, to jointly exploit the lightweight pub-sub paradigm offered by MQTT and the long range, low power wireless capabilities offered by LoRa, to deliver building automation services.

Chapter 2

Background

The Wireless Sensor Network (WSN) is built of nodes, from a few to several hundreds, where each node is connected to one or more sensors according to the topology design. Each sensor node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, one or more sensing units, an electronic circuit for interfacing with the transducers and an energy source, usually a battery or an embedded form of harvester to gather energy from the environment. Usually these nodes rely on small and inexpensive hardware sizes, and are in most cases spread in harsh environments without any predetermined infrastructure. As a result, the network must be able to self-adjust its own configuration, reacting to internal and external events, as failures, battery exhaustion and environment changes. Energy-efficient communication is required for power-constrained devices. Since the radio operation dominates the energy consumption of the sensor node [21], energy efficiency is mainly affected by the scheduled duty-cycle, that is the fraction of time the radio spends in active state. Numerous protocols for WSNs have been proposed in recent years. A brief overview of these protocols is provided in Section 2.1, focusing on low duty-cycle based solutions for the medium access. In Section 2.2, a classification of the neighbor discovery protocols is introduced, in Section 2.3 the discussion concentrates on BLE protocol, and, finally, a brief introduction of LoRa (Long Range) Physical Layer is provided in Section 2.4.

2.1 Duty-cycle based MAC Protocols

MAC protocols for WSNs are aimed to guarantee efficient access to the communication media, while carefully spending energy budget allotted to the node. Idle listening is often the largest source of energy waste [22] and duty-cycle-based MAC protocols are considered as necessary to reduce energy consumption at MAC layer. Such protocols implement sleep/wake cycles to save energy by periodically turning off the radio. According to channel access policies, most of the existing MAC protocols are categorized into two categories [23]: contention-

based and scheduled-based MAC protocols. In the following, a brief overview of these classes of protocols is provided.

2.1.1 Contention-based MAC protocols

In contention-based MAC protocol, all the nodes share a common medium and compete for the same medium for transmission. Thus, the question arises of how collisions can be avoided. One early energy efficient and most cited duty-cycle MAC protocol is S-MAC [24], that stands for Sensor-MAC. S-MAC is a complex protocol that applies periodic sleep-wake cycles to IEEE 802.11 for WSNs to reduce energy consumptions due to overhearing, idle listening, collisions and support self-configuration [25]. The design of S-MAC assumes that applications will have long idle periods and can tolerate some latency. The listen and sleep periods in the S-MAC are fixed intervals. It assumes that the nodes do not need to be in wake/standby mode all time; therefore, it groups all nodes in flat manner, arranging them by synchronising the sleep/wake schedules of neighboring nodes. In the synchronization phase, the neighbors synchronize their listen periods and a table is maintained to update neighbors schedules. DATA message is used for data transmission using the handshake methods of Request-To-Send (RTS)/ Clear-To-Send (CTS). S-MAC uses a combined contention scheme and scheduling for collision avoidance. In S-MAC, long messages will be divided into small fragments in order to be sent as burst. This method creates more messages to send, which requires longer access to the medium. In conclusion, while S-MAC was designed mainly to reduce energy consumption, it ignores other important performance factors, such as fairness, throughput, bandwidth utilisation, and latency [26]. Timeout-MAC (T-MAC) [27] was introduced to outperform S-MAC by using a dynamic duty-cycle instead of a fixed one. The idea is to transmit all messages from one node to another in bursts of variable length, and to sleep between bursts for further energy saving. It also determines the length of variable load by maintaining an optimal time. T-MAC applies RTS and CTS method as well as S-MAC. When RTS did not get CTS response it would try again before giving up. As in S-MAC, T-MAC can only send the message to one hop every duty-cycle, which results in high latency. In addition, T-MAC has an early sleep problem, as a node switches to sleep even when a neighbor has some messages waiting to be sent. As a result, the throughput is decreased in nodes to sink transmission. RMAC [28] (Routing enhanced MAC protocol) is similar to S-MAC, as sensor nodes have three ways in each cycle (SYNC, DATA, and SLEEP). It differs from S-MAC by sending a pioneer frame (PION) during the DATA mode to reserve the channel in the SLEEP cycle to send the message through many nodes in one duty cycle. PION is doing RTC and CTS respectively, and continues through the

network until the end of DATA cycle, or the PION reached its target. Building on RMAC, Pattern-MAC (P-MAC) [29] proposed to send multiple messages per duty cycle. That has given better traffic handling advantage over RMAC.

2.1.2 Scheduled-based MAC protocols

In TDMA MAC protocol, the total time duration of a communication is partitioned into a fixed number of time-slots. TDMA configures these time-slots into time-frames that repeat periodically. Each node is allowed to transmit only in the allocated time-slots for each frame. Based on the nature of the algorithms, the TDMA protocols are classified in two groups [30]: centralized protocols and distributed protocols. In the latter, the cluster-head (in case of Hierarchical WSN) centrally schedules different slots to different nodes in the network. Every node uses these time-slots for data communication, while in the former, scheduling is performed by each node, basing on the local information that it owns. There is no need of any central node, and, thus, the message communication is less reduced compared to the centralized algorithms and, hence, it is more energy efficient. A key advantage of TDMA protocols is the fact that their transmissions operate in a completely predictable way. While the transmission operation of TDMA allows to reduce power consumption, it fails to take advantage of this on the receiving end-side as it must use methods such as low power listening for the reception of messages since it does not know when for sure it is not going to receive messages. Many-to-One-Sensor-to-Sink (MOSS) [31] MAC layer protocol addresses this concern by retaining knowledge of when it could possibly receive a message, allowing the radio to be completely shut down when it will be certain to receive no message. In general time-slotted protocols tend to lack adaptability. Moreover, a node joining the network can lead to global or local reconstruction of the schedules, which may cost time and energy. On the other hand, time-slotted protocols aim at providing deterministic end-to-end delays and collision-free communication, and show good performance under high traffic loads [32]. Centralized TDMA scheme is adopted by protocols as legacy Bluetooth, Bluetooth Low Energy (BLE) and ANT/ANT+. In fact their medium is accessed with TDMA where slots are repeated on configured message intervals. While in ANT the central device assigns a combination channel-timeslot to the paired devices, in Bluetooth and BLE the master partitions the communication timeline, assigning a timeslot and a frequency-hopping channel map. Further details on BLE will be given in the following.

2.1.3 Asynchronous Low Duty-Cycle MAC protocols

Berkeley MAC or B-MAC [33] is an asynchronous duty cycle MAC protocol. In B-MAC, each node has its independent duty cycle scheduling. Node can transmit by sending a preamble along with the data packet, which must be longer than the receiver's sleeping time, to make sure that the receiver will be in wake up mode. If a node is in a wake cycle, it samples the medium only when a preamble has been detected. Power consumption, throughput and latency are improved in B-MAC, however, overhearing and the long preamble are major drawbacks. X-MAC [34, 35] was proposed to overcome the drawbacks of B-MAC. It uses short preambles to avoid the overhearing problem. The preamble contains the target address to help untargeted nodes to sleep and allows the targeted node to send early ACK. This does not avoid only overhearing but also reduces the latency by half. The lack of flexibility is the main drawback of this protocol as it is very hard to reconfigure it after deployment. Another problem with this approach is that it fails to operate when the traffic builds up, because of the preamble transmissions, that occupies the wireless medium. Receiver-Initiated asynchronous MAC protocol (RI-MAC) [36] uses the receiver initiated mechanism to achieve lower power consumption, higher throughput and packet delivery ratio. It is similar to B-MAC, but differs from it for the independent duty cycle scheduling of each node in the network. Compared to B-MAC and X-MAC, the strength is that the sender in RI-MAC stays in active mode until the targeted receiver is ready and the message starts to be delivered. Receiver will inform the sender by sending a beacon frame.

2.2 Neighbor discovery MAC Protocols

In WSNs the neighbor discovery problem is critical, mainly, because of nodes, which are awake infrequently and perhaps rarely co-located, must discover each other as quickly as possible, without any prior knowledge or synchronization. Neighbor discovery algorithms can be mainly classified into two categories: deterministic and probabilistic.

2.2.1 Deterministic Protocols

The key concept of deterministic neighbor discovery algorithms is that each node transmits according to a predetermined transmission schedule that allows it to discover all its neighbors by a given time with full probability. The down-side of these algorithms is that usually they need increased running time and often the apriori knowledge of the number of neighbors [37]. Some examples of deterministic protocols are: Disco, U-Connect, quorum-based protocols and Searchlight.

Disco [38] and U-Connect [39] belong to the family of slotted-protocols, where time is partitioned into fixed-size windows or slots. In each slot, a device can either sleep or be active. Disco selects a pair of prime numbers p and q such that the sum of their reciprocals is close to the application's desired radio duty cycle. The coprimality-based approach assumes that active slots occur with a certain repetition period. The Chinese Remainder Theorem (CRT) guarantees an overlap within a bounded number of slots, even if nodes independently set their own duty cycle. One drawback of this class of protocols is that its parameters are fundamentally limited by prime selections.
The concept of quorum is at the basis of others deterministic protocols, classified as quorum-based protocols [40, 41]. They assume a period of m slots, dividing time into groups of m^2 continuous intervals. Within each group, the m^2 intervals are organized as a 2-dimensional $m \times m$ array in a row-major manner. A node can arbitrarily pick one column and one row of entries as active intervals. For any two nodes, this paradigm ensures successful discovery within m^2 intervals. Given two nodes, which are perfectly time-synchronised, it results that their quorum intervals have at least two intersecting beacon intervals. However, the assumption that nodes are time-synchronised is not exploitable. The initial quorum design [40] requires that m is a global parameter, while the following design [41] allows to support two different schedules, which both underperform in dealing with duty-cycle asymmetry.
In order to outperform quorum- and CRT -based protocols, a protocol called Searchlight has been proposed in [42]. Searchlight presents a beacon scheduling mechanism. In each period of t slots, each node sends two beacons at the beginning and the end of the slot, performing two active slots. The latter, denoted as anchor slot and indexed from 0, is always an active slot and has a fixed position in each period, while the former active slot, called the probe slot, traverses from position 1 to $\lfloor \frac{t}{2} \rfloor$ across to $\lfloor \frac{t}{2} \rfloor$ periods. This scheme ensures a successful discovery within a period under symmetric scenarios. In order to maintain the constant offset under asymmetric scenarios, Searchlight restricts duty cycles to power-multiples of the smallest one, significantly limiting its parameter t and, then, the set of duty-cycles that can be realized.

2.2.2 Probabilistic Protocols

The key concept of probabilistic neighbor discovery algorithms is that each node transmits at randomly chosen time, discovering all its neighbors within a given time-interval, with a certain probability. Examples of probabilistic protocols are the Aloha OMS protocol [43], the Coupon Collector's Protocol [37], but the best representative among probabilistic approaches remains the family of Birthday protocols [44]. The Birthday protocol [44] has been proposed for

asynchronous neighbor discovery in static ad-hoc networks. The inspiration for this protocol is the birthday paradox, in which it is reckoned the probability that at least two individuals in a room have the same birthday. This approach foresees a node will decide to transmit, listen, or sleep with probabilities. Due to their probabilistic nature, they support asymmetric duty cycles, but suffer from aperiodic and unpredictable discovery, leading to long tails in discovery probabilities. In fact, this probabilistic approach cannot guarantee bounded worst-case latency, hence each node does not always meet the other ones within a certain amount of time.
Among the probabilistic protocols, we further classified the family of asymmetric and periodic interval-based protocols. One of the first periodic interval-based protocols, STEM (Sparse Topology and Energy Management), has been proposed in [45]. This protocol does not provide a complete MAC protocol, however another MAC protocol can be used along with it to complete the implementation. The STEM protocol has two flavors: STEM-B, STEM-T. STEM-B (B stands for beacon) foresees that one device, called the initiator or advertiser, periodically broadcasts advertising packets on wake-up channel, independently from another device, called scanner, that periodically listens for incoming packets during a scan window, at least equal to the scan interval. If the scanner receives a packet on wake-up channel, it switches on the data channel and sends a response on a different wireless channel. Once the initiator receives this packet, both devices are synchronized. In STEM-T (T stands for Tone), the transmitter sends out a simple busy tone on the wake-up channel for a time long enough to hit the receiver scan window. Since the tone carries no information about the address of the receiver, all neighboring nodes, which hear busy hop to data channel to receive the data, but only the node for which the data was intended will reply, while all the others go back to sleep. STEM-B scheme is adopted by BLE and ANT/ANT+ with slight variations. The ANT/ANT+ [46] protocol, which is used in over 100 million devices [47], implements this scheme, but using a single channel to perform the complete discovery procedure [48]. The latter has been implemented in a slightly modified version for BLE, where the advertiser periodically sends a burst of three packets, each on a different wake-up channel, called advertising channel [49]. This period consists of the sum of a set interval with the addition of a small random delay. On the remote side, the scanner periodically senses these three channels for a set window, once per scan period. More details about BLE protocol are provided in the following section.

2.3 Bluetooth Low Energy

Bluetooth Low Energy has been designed to be low cost, easy to implement, and optimized to transmit small chunks of data. The advantages in the use of BLE include: (i) low power functionalities, operating for months or years on a coin-cell, (ii) small size and low cost, (iii) full compatibility with commonly-used devices such as mobile phones, tablets and computers. These properties make the BLE more favorable for many short-range communication applications. There are two ways for BLE communication: connectionless and connection-oriented. The latter is aimed at discovery operation and generic broadcasting tasks, for example the transmission of state data, while the former performs a scheduled and reliable data exchange.

2.3.1 Overview on BLE protocol

The BLE radio operates in the 2.4 GHz to 2.4835 GHz frequency Industrial, Scientific and Medical (ISM) band, with Gaussian Frequency-Shift Keying (GFSK) modulation. It splits the spectrum into 40 channels, each 2 MHz wide, and runs Frequency-Hopping Spread Spectrum (FHSS) to avoid interference. Out of the fourty BLE PHY channels in the ISM band, three are used for device discovery, connection establishment and broadcast transmission (advertising), the remaining ones enable bidirectional communications among connected devices. The BLE protocol stack [49] is partitioned in Controller and Host. The Controller runs the Physical (PHY) and Link layers. The Host portion runs on application processor and contains: Logical Link Control and Adaption Protocol (L2CAP), the Attribute Protocol (ATT), the Generic Attribute Profile (GATT), the Generic Access Profile (GAP) and the Security Manager Protocol (SMP). The Link Layer (LL) controller handles low level communication over a PHY interface, scheduling the discovery and connection operations. In BLE, each advertising device runs an advertising-event once at the beginning of each advertising-interval. This can be set per device in the range of values from 20 ms to $10,240$ ms in steps of 0.625 ms. The BLE protocol does not implement carrier sensing on the used channels, and, because of this, the advertiser sends the same advertising packet to all the dedicated three channels 37 (2402 MHz), 38 (2426 MHz), 39 (2480 MHz) at each advertising-event in order to avoid repeated collisions. Figure 2.1 depicts the advertising-event timeline. Moreover, each advertiser adds a random delay up to 10 ms to avoid a massive contention. The advertising channels are between or outside the main frequencies used by IEEE 802.11 protocols, to prevent interference with the channels used by WiFi. In addition, the scanning devices run periodic operations, listening to advertisers during a scan-window, once at the beginning of every scan-interval. The scan-interval and scan-window can be set per scanner device. The scanner sub-

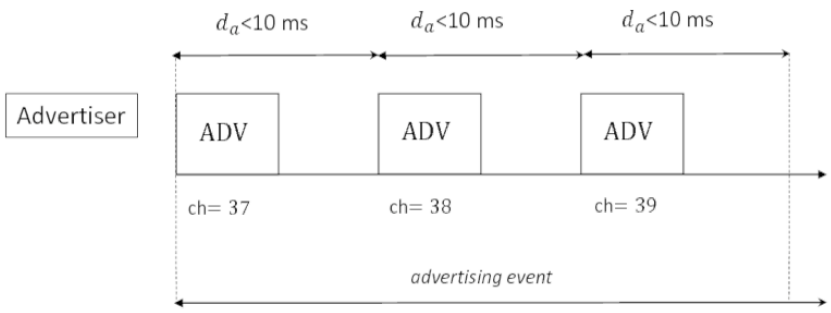


Figure 2.1: Timeline of an advertising event. It consists of three advertisements over channels 37, 38, and 39. After sending the advertisement message, the device remains on the channel for a random time up to 10 ms, listening to a connection request or a scan request for more data.

sequently changes listening-channel at each scan-window. The advertisers can operate in passive or active scanning. In passive mode, each node periodically broadcasts an advertisement message on the three channels. After sending, it can wait up to 10 ms before sending the same packet on the subsequent channel. The maximum payload per packet is 31 bytes long. The scanner does not reply to passive advertisers. On the contrary, in active mode, the scanner must reply immediately, unicasting the scan request message to the advertiser. The latter stops the messages exchange by broadcasting a scan response message (SCAN_RSP). Since the maximum payload of SCAN_RSP is 31 bytes, the advertiser doubles its transmission data capacity per advertising-event. When a central device intends to initiate a connection with the advertiser, it sends a request (CONN_REQ) packet on the same channel used by the advertiser 150 µs after the reception of the advertisement message. The CONN_REQ payload contains the hopping map of channels that the connecting device must use in sequence during the connection. Thus, the advertiser immediately stops its advertising-event, and jumps to the requested data channel to continue the connection sequence. Figure 2.2 depicts the connection procedure timeline. Firstly, the two devices exchange information about the connection configuration. The master decides how often the slave has to wake up for incoming transmission (connection-interval and slave-latency). The choice depends on the slave and master availability and on the throughput requested by the particular application. After the initial configuration is completed, the devices can start the data transfer. In general, the slave can have multiple data transfers

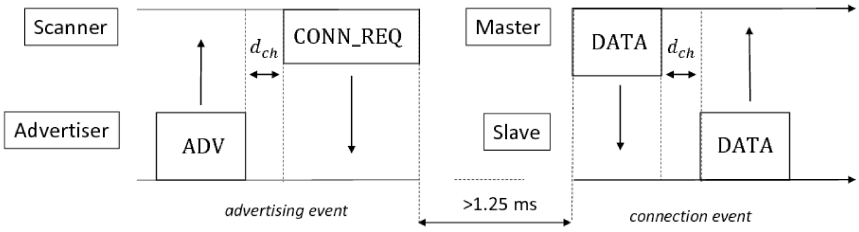


Figure 2.2: Timeline of connection procedure. After receiving the advertisement message, the central device, that intends to connect the advertiser, waits the inter-frame time 150 µs and, then, sends the connection request packet (CONN_REQ) on the same channel. After the CONN_REQ, the central, acting now as master, and the advertiser, acting now as slave, wait for a minimum of 1.25 ms before continuing at a data channel.

in a single connection event, with a maximum payload of 20 bytes. A random 4 byte Access Code identifies the messages exchanged by a paired master and slave. Bit error detection is carried out by including a 24-bits Cyclic Redundancy Check (CRC) code in each packet. A single master can manage multiple simultaneous connections with different slaves, setting up a star topology network called piconet. Each slave can be part of a single piconet only. To save energy, slaves are in low energy mode by default and wake up periodically to listen for master poll packets. The master uses TDMA to coordinate medium access by the slaves.

2.3.2 Overview on Discovery Process Issues

Many works and several statistical and/or simulative approaches have been formalized to estimate the discovery latency in BLE networks.

In [50], the authors proposed an analytical model for a 3-channel-based neighbor discovery. Specifically, the model derives the discovery latency, not taking into account the collision amongst packets from homogeneous BLE devices as well as the interference with one or more channels. The model outcomes seem to match the simulation results for a scanning duty ratio greater than a certain value (0.3).

In [51] the authors propose an energy model for all the operating modes foreseen by the BLE protocol. Specifically, they provide a solution that is not closed-form, which estimates the mean discovery latency. The algorithm developed by the authors is at the basis of the model proposed herein and, for this reason some details will be provided below and a comparison among them

discussed in Chapter 5.

In [52], an analytical model is developed to investigate the discovery probability. The authors have analyzed both the continuous and discontinuous scanning performed in active mode (i.e., with active exchange of scan request and scan response). According to the authors' conclusion, the results, obtained via the proposed closed-form solution match the performed simulation experiments. The analysis of theoretical results appears to exhibit an exponential growth of device discovery delays, related to the increasing tags number. Despite this, the use of three advertising channels and tiny-sized frames has weak effects on the discovery latency.

In [53], starting from the model developed in [52], the authors present intensive simulations to investigate discovery probability and provide a quantitative examination of the influence of parameter settings on the discovery latency and the energy performance metric of the discovery process.

In [54], the authors propose a general model for analyzing the performance of neighbor discovery process in BLE networks. According to the authors' conclusion, the numerical results, produced by their model, meet the simulation outcomes for some parameter values specified by the standard.

2.3.3 Overview on Connection-Oriented Issues

In [55] the author analyzes how the protocols BLE and ANT may coexist on a single chip. Given that both ANT and BLE are low duty-cycle protocols, the cited work defines the general scheduling principles.

In [52] the authors investigate the impact of various critical parameters on the performances of BLE devices operating in connection mode. The paper provides experimental results that complement the theoretical and simulation findings, and indicates implementation constraints that may reduce the BLE performance.

In [17] the authors analyze a proof of concept of critical parameters setting in connection-oriented networks. The proposed system consists of two wearable BLE devices (smart shoes) that need to remain connected to a central node in order to stream data to it. The smartphone, often designated to be the central node in a *piconet* (i.e., connection-oriented network), is a commonly used device, that has sufficient capacity to jointly manage different radios (such as mobile radio, Bluetooth, WiFi, Near Field Communication (NFC), etc.) and/or a single radio serving sub-networks with different topologies and communication strategies. Starting from this key point, the following section will discuss how the central node can jointly manage BLE connected devices, and event-driven tags without a persisting connection.

2.4 LoRa Physical Layer

With the introduction of the IoT paradigm, a new approach of connectivity [56], characterized by low-rate, long-range transmission technologies in the ISM frequency bands, has challenged the legacy models, based on multihop mesh networks using short-range communication in the unlicensed spectrum, or the standard long-range cellular technologies, mainly 2G/GSM/GPRS, to ensure data access to things. The new approach foresees a star topology, in which devices communicate directly to a sink node, generally referred to as gateway, which removes the need of constructing and maintaining a complex multi-hop network and provides the bridging to the IP world. The architecture of these networks, referred to as Low-Power Wide-Area Network (LPWANs), is designed to supply wide area coverage and also ensure connectivity to nodes that are deployed in very harsh environments. New LPWAN technologies such as LoRa [57], Sigfox [58] and Weightless [59] are emerging because of their capability to enable power efficient wireless communication over very long distances. Among these solutions, LoRa (standing for Long Range Technology) is a new physical layer LPWAN solution, designed and patented by Semtech Corporation, which also manufactures the chipsets.

2.4.1 Overview on LoRa's Chirp Spread Spectrum Implementation

The LoRa Physical (PHY) is a derivative of Chirp Spread Spectrum (CSS) with integrated Forward Error Correction (FEC). CSS, initially developed for radar applications, uses signals with constant amplitude and passes the whole bandwidth in a linear (or non-linear) fashion from one end to the other end in a fixed time. This uniform distribution of a symbol over a wide bandwidth provides robustness to frequency-selective noise, caused by low cost crystals, and interference, despite the lower spectral efficiency. Using some additional precautions, CSS in LoRa PHY can also be more resilient to multi-path interference and to the Doppler effect than other more conventional modulations. The number of bits that LoRa encodes in a symbol is a tunable parameter, called SF, which stands for Spreading Factor. This means that a chirp using spreading factor SF maps 2^{SF} bits into a symbol. The number of times per second that the modem adjusts the phase is called the chip-rate and defines the modulation bandwidth BW. The chip-rate is a direct subdivision of the quartz frequency (32 MHz). LoRa uses three different bandwidth: 500 kHz, 250 kHz and 125 kHz. Assuming the latter bandwidth, LoRa symbols are modulated over an up-chirp of 125 kHz bandwidth and different pseudo-orthogonal spreading factors are used based on data-rate requirement and channel conditions. LoRa uses $SF7$ to $SF12$ spreading factors, depicted in the spectrogram

of Figure 2.3 [60] and determines the duration of a symbol T_s, according to the Equation (2.1).

$$T_s = \frac{2^{SF}}{BW}[sec] \tag{2.1}$$

Given Equation (2.1) the bit-rate for a fixed pair of SF and BW is computed as shown in Equation (2.2).

$$R_b = \frac{SF}{T_s}[bits/sec] \tag{2.2}$$

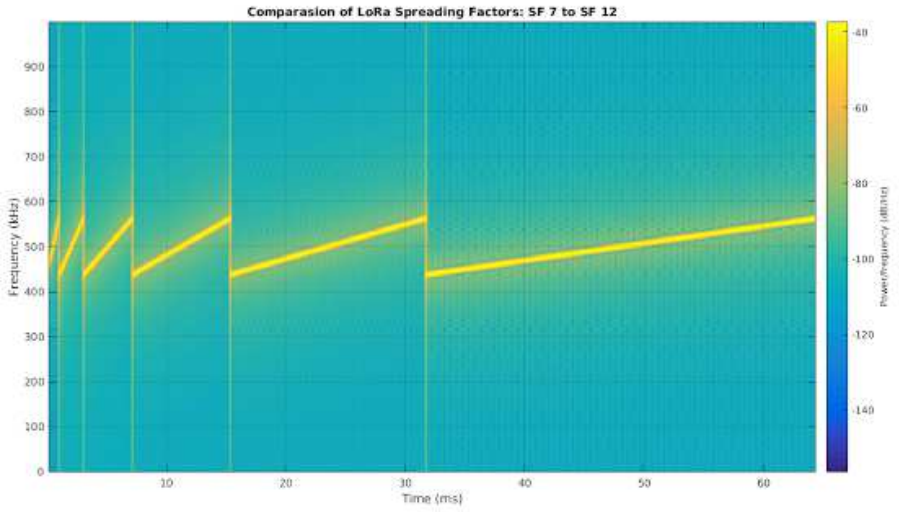


Figure 2.3: LoRa PHY. Spectrogram of different Spreading Factors: $SF7$ to $SF12$.

As shown by Equation (2.1) and summarized in Table 2.1 [61], assuming a fixed modulation bandwidth BW, an increase of SF of 1 will yield symbols that last twice the duration. Analogously, bigger BW increases the symbol-rate at which chirps are transmitted, and, consequently, the bit-rate R_b of the modulation. An increase in the transmit time for a chirp (i.e. symbol) gives the frame a higher robustness to interference or noise. On the other hand, this effect may be partially counterbalanced by two shortcomings: firstly, for higher spreading factors the number of possible symbols increases, making the occurrence of symbol errors more likely, and, secondly, transmitting longer frames increases probability of collisions.

Table 2.1: LoRa Phy. Bit-rate [bits/s] and SNR [dB]within the range of spreading factors and bandwidths.

SF	125 kHz	250 kHz	500 kHz	SNR
7	6835	13671	27343	−7.5 dB
8	3906	7812	15625	−10 dB
9	2197	4396	8793	−12.5 dB
10	1220	2441	4882	−15 dB
11	671	1342	2685	−17.5 dB

The choice of SF affects receiver sensitivity, which is defined by the Equation (2.3).

$$S = -174 + 10\log_{10} BW + NF + SNR[dB] \tag{2.3}$$

where the first term is due to thermal noise at the receiver in 1 Hz of bandwidth, NF is the noise figure at the receiver (which is fixed for a given hardware setup), and SNR is the signal to noise ratio required by the underlying modulation scheme. SNR values for different spreading factors are represented in Table 2.1, where it can be seen that increasing the spreading factor allows for a better sensitivity.

2.4.2 Physical Layer Packets

A LoRa physical frame includes a 4.25-up-chirps-minimum-long preamble, 2 synchronisation symbols, a physical payload and an optional Code Rate (CR). The latter is the FEC rate used by the LoRa modem that offers protection against bursts of interference, and can be set to either 4/5, 4/6, 4/7 or 4/8. An higher CR offers more protection, but increases time on air. Radios with different CR (and same SF and BW), can still communicate with each other if they use an explicit header, as the CR of the payload is stored in the header of the packet, which is always encoded at CR 4/8 [62]. The nominal bit-rate of the data signal is calculated as in Equation (2.4)

$$R_b = SF \cdot \frac{\left\lceil \frac{4}{4+CR} \right\rceil}{\left\lceil \frac{2^{SF}}{BW} \right\rceil} \tag{2.4}$$

Data are encoded in the signal as instantaneous changes in the frequency of the chirp, or lack thereof. Figure 2.4 shows the spectrogram of an example frame, formed by 8 up-chirp symbols as preamble, used to detect LoRa chirps, next 2 down-chirp symbols, acting as time synchronizer, followed by the 5 modulated symbols of the payload. The jump in the frequency represents the

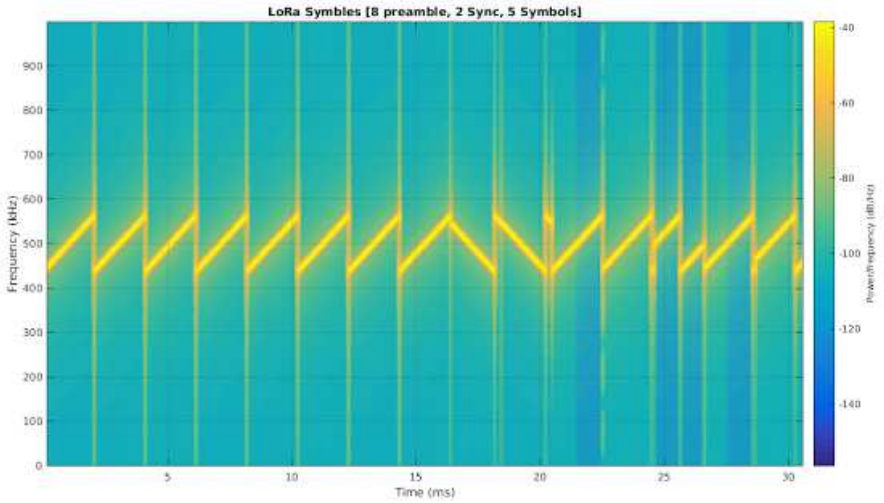


Figure 2.4: LoRa PHY Frame spectrogram: 8 up-chirps as preamble, 2 down-chirp as time synchronizer and 5-shifted-chirps-long payload.

modulated symbol. A decoding process is proposed in [63], and it consists in a de-chirping operation on the signal, followed by the calculation of the Fast Fourier Transform (FFT) of the signal, with a number of bins equal to the number of symbols that corresponds to the used SF.

Chapter 3

Connection Oriented Issues

In connection-oriented communication, paired devices maintain the link active, despite no data flow occurs. Even if BLE has been proved to be less energy-demanding than other radio standards, BLE could yet be not low power enough for devices running massive transmissions and powered by coin cells or energy harvesting applications, as typically envisioned in Internet of Things (IoT) scenarios. In this chapter, a BLE-based application is proposed in Section 3.1 to enable data communications between so-called smart shoes [15], described in subsection 3.1.1, and a smartphone that receives the data over its BLE interface, for possible relaying to a remote server, described in subsection 3.1.2. In order to implement an efficient data transfer from the smart shoes to the smartphone, a BLE configuration based on the indication transfer mechanism and in-node data-reduction is designed and presented in Section 3.2, particularly in subsections 3.2.1 and 3.2.2, and tested and evaluated in subsection 3.2.3, to ensure adequate throughput and reliability, while preserving the low-power behavior of the node. The work described in this chapter has been presented in [17].

3.1 Proposed System

In the proposed system architecture (Fig. 3.1), BLE is used to enable data communications between a pair of wearables, so-called smart shoes [6] and acting as peripheral devices, and a smartphone that receives the data over its BLE interface, acting as central node. The smart shoes used in this work have been prototyped in the lab; they are equipped with sensors and electronics to collect and transmit data about the user's gait and step profile, for a long-term monitoring of possibly decreasing performance. In this section further details about the hardware architecture of the proposed system are provided.

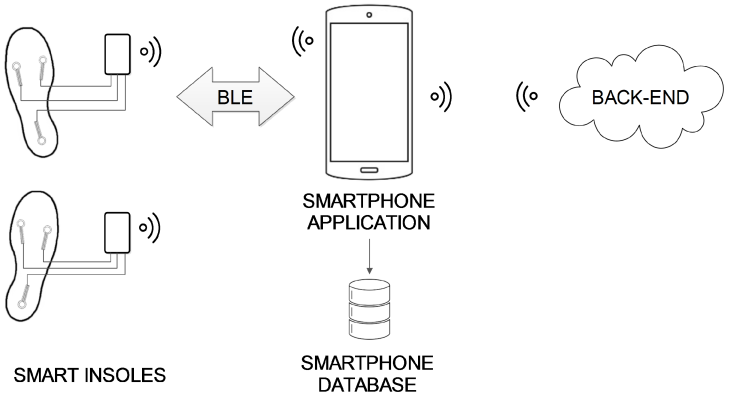
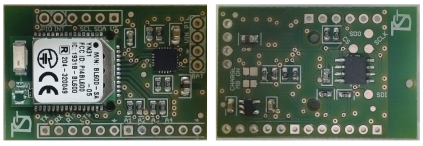


Figure 3.1: Proposed system architecture.

3.1.1 The smart shoes as slaves

Wearable smart shoes collect information on the user's physical activity, through insoles equipped with force sensors, and a processing board, able to classify different dynamic states and to transmit related data to a supervising system. Some essential design requirements have been accounted for, tailored on the user's needs, such as: reduced obtrusiveness, limited power consumption, adequate precision and reliability, and easy maintenance [15]. To fulfill them, hardware and software components have been carefully selected and designed. The hardware components of the smart shoe are a sensorized insole, equipped with Force Sensing Resistors (FSRs), a board, and a wireless inductive charger. The FSRs are widely used in wearable devices for well-being applications, like sensorized footwear (e.g. [64, 65, 66, 67]). In fact, they enable temporal analysis of the body weight distribution on the foot insole, and of the walking activity. In the proposed system, the transducer FSR 402 Short by Interlink Electronics [68] is used. It consists of two metal lines whose resistance depends on the pressure applied on their surface. The correct positioning of the transducers on the insole is crucial to ensure a reliable detection of the subject's physical activity and gait. As stated in [15], there are two possible configurations; in the one here chosen, FSRs are placed to match the heel, the 1st and the 5th metatarsal heads. This way, an analysis of the steps can be performed, identifying the different gait phases. The signal acquisition and data transmission procedures are implemented by a printed circuit board (PCB) developed ad hoc for our purposes, called TST board (see Fig. 3.2). Such an ad-hoc design aimed at getting low average power consumption and small form factor, as requested.



(a) Front view (b) Rear view

Figure 3.2: TST board host by the smart shoe, for data acquisition and transmission.

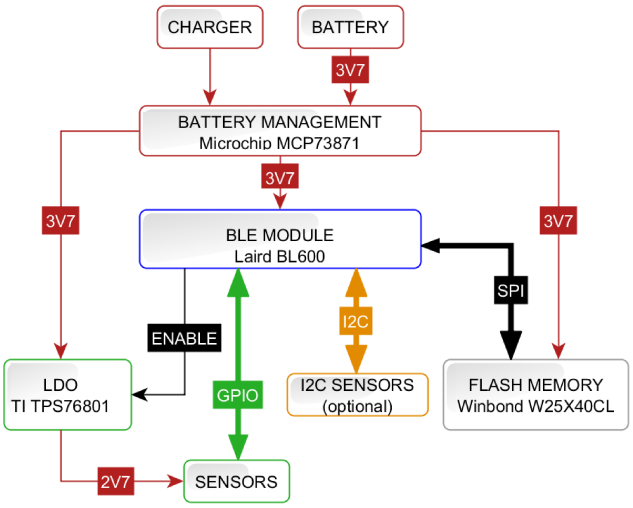


Figure 3.3: The hardware scheme of the electronic board.

In fact, the board can be host in a hole dug in the shoe sole, and covered by a lid. The user walks comfortably, without noticing the presence of the hardware components. Moreover, such a solution allows the easy maintenance, since the hardware components are easily accessible.

The Laird BL600 [69] module is the heart of the TST board, as shown in Fig. 3.3; it is based on the nRF51822 IC from Nordic Semiconductor, built around an ARM Cortex^{TM} M0 micro controller unit (MCU), and a 2.4 GHz Smart compatible Bluetooth transceiver. The module includes an LDO section and exposes a wide variety of external communication ports (both analog and digital, such as GPIO, SPI, I2C, UART, and JTAG). A ©Microchip MCP73871 [70] Power Management IC is in charge of the energy management, ensuring power supply during the battery charging phase. This subsystem is handled by the Laird BL600 firmware module, which provides information on the battery charge, and allows to implement energy management strategies. To facilitate the battery re-charging operation, a commercial inductive charging system has

been implemented. The analog section of the board (for FSRs reading) is based on an LDO TI TPS76801QD, selected for its low power consumption in the low-power state (500 nA), and manageable via firmware. Finally, the data storage is provided by a serial flash memory, the IC Winbond W25X40CL, interfaced to the BLE module by an SPI bus. The board also includes the I2C interface for the future integration of an accelerometer sensor. The latter will handle the module awakening, when in motion.

3.1.2 Smartphone as master

In a BLE star topology network, low consumption devices (slaves, i.e. smart shoes) are interfaced to the central node (Master, i.e. smartphone), that is in charge of searching within a short distance range the slaves available for communication, creating and managing the network. It may then forward the arriving data to a local or remote server for further processing. To allow using such a system even outdoor, a device provided with on-board memory capabilities for data storage and mostly portable, is required as a master, that is why a smartphone has been chosen. In fact, in recent years mobile devices have rapidly evolved, converting from devices designed exclusively to telephony, to real computers, enriched with reasonable memory footprint, computational capabilities, sensors (e.g. cameras, accelerometers and geo-positioning systems), and wireless interfaces (e.g. NFC, WiFi, Bluetooth, etc.). A smartphone application has been consequently implemented, using the Android APIs that support the BLE stack, available since the Jelly Bean 4.3 version. Such an application starts automatically after device booting, listening for connection availability from the slaves all the time.

3.2 Performance Evaluation

In the previously described system architecture, data are transmitted from the wearables to a smartphone, as in [71], but processing is performed locally by the shoe electronics. In [72], a continuous gait monitoring is also achieved using smart shoes, but details on the protocol managing communications from the two insoles to the smartphone are not given. A BLE configuration based on a syncronous transfer mechanism is properly designed. An operation model has been developed, able to provide design guidelines according to the performed test, aiming at ensure adequate throughput and reliability, while preserving the low-power behavior of the wearables. The methods and the following discussion of the obtained results are dealt with in the following subsections.

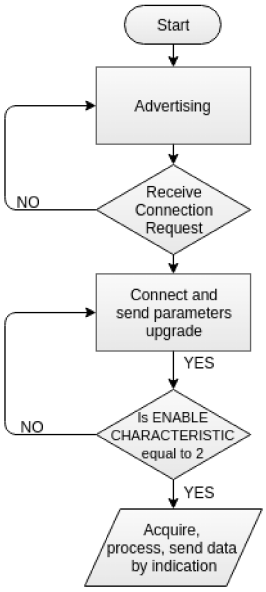


Figure 3.4: Flow chart of the application running on the TST board.

3.2.1 Smart shoes application Design

The application of the TST boards assigns the master/central/client role to the smartphone, and the slave/peripheral/server roles to each shoe (right and left), at the LL/GAP/ATT level. The use of BLE connection-oriented approach allows a synchronized data exchange. The software implementation has to trade off the limited power availability of slaves, minimizing the duty-cycle of data exchange, with the requested application throughput and data reliability. The TST board application flow chart, implemented by an event-driven programming language called Smart BASIC, is shown in Figure 3.4.

At the beginning, the advertiser, i.e. the smart insole, enters advertising mode in order to indicate its capability to establish a data channel connection with the scanning/initiator. The advertiser broadcasts the advertising message until the connection establishment. An event is generated when the advertiser receives a connection-request (CONN_REQ) by the Initiator (the smartphone), entering connection mode. The master sets the connection parameters, but the slave can request their upgrade. At the end of the upgrade procedure (at least 6 s, imposed by the protocol) the connection link is ready to exchange data, with a communication duty cycle scheduled on the basis of the updated T_{CI}. If $SL > 0$, the Slave can wake up every T_{ECI}, if the buffer of data to transmit is empty. The remaining T_{CI} can be used for retransmission of a possible lost packet, before the generation of successive packet.

Table 3.1: Step phases identification by binary coding of transducer outputs.

FSR1	FSR2	FSR3	Step Phase
1	0	0	Heel Contact (H)
1	0	1	Foot Contact (F)
1	1	0	
1	1	1	
0	0	1	Push Off (P)
0	1	0	
0	1	1	
0	0	0	Limb Swing (S)

To implement the data streaming between the slaves and the master, we add to the GATT server a custom service composed of two characteristics: *Data Characteristic* (20 bytes with *indicate* and *read* properties) and *Enable Characteristic* (1 byte with *write* and *read* properties). The former is for data exchange: to guarantee adequate efficiency and data reliability, the transfer is realized via the indication mechanism. This way the slave will automatically send the new data (indication) and wait for the confirming message from the master. The maximum allowed payload of 20 bytes is used for data exchange. The *Enable Characteristic* is used to enable or disable the data streaming, setting the byte to 2 or 0, respectively. The slave, every 10 ms, reads the FSRs ADC outputs from the binded GPIOs. An on-board pre-processing of FSRs values is implemented to decrease the over-the-air throughput and, consequently, the transceiver power consumption. The pre-processing algorithm, at each time step, receives as inputs the analog FSRs values and generates an integer value, representing a different foot-support condition. In order to reduce the data to transmit, an in-node data-reduction policy is performed: setting proper thresholds and using binary coding of the data, up to 8 different foot-support states are identified. Not all the possible binary combinations correspond to a different state; some of them are related to the same state, as detailed in Table 3.1.

The four identified states are used for a simple step detection algorithm. Each detected step phase, represented by 2 bits, is laid in RAM-buffer. When the buffer amounts to 20 bytes (every 10 ms$\cdot(20\cdot 8$ bits$)/(2$ bits$) = 800$ ms), its content is passed to the FIFO-buffer for transmission (Figure 3.5). Once the indication is sent, the slave must wait for the master confirming packet before flushing the transmission buffer. Once the confirming packet is received, the slave flushes the buffer and prepares it to host the next data packet, according to the application data-rate of 200 bps. In order to keep an acquisition timing

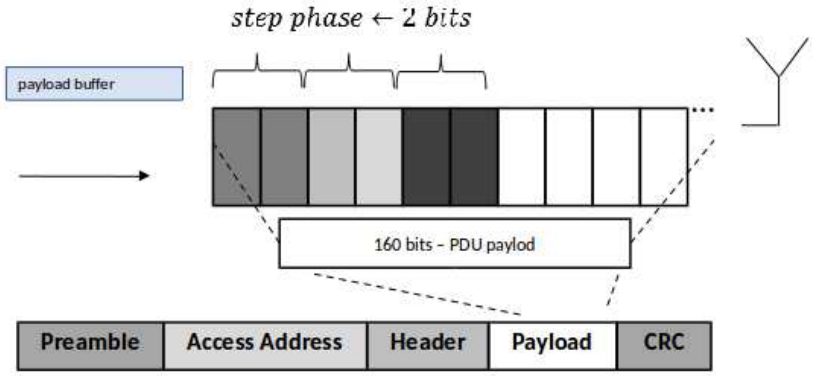


Figure 3.5: Payload encapsulation. Each detected step phase, represented by 2 bits, is laid in RAM-buffer. When the buffer amounts to 20 bytes, every 800 ms, its content is encapsulated in the BLE frame for transmission.

of 10 ms, T_{ECI} must be equal to 800 ms.

Let define the goodput as the application-level throughput, i.e. the amount of data, excluding protocol overhead bits as well as retransmitted data packets. The desired goodput, which guarantees the successful data exchange without loss of data, can be obtained as:

$$GP_{des} = N_{max} \cdot \frac{L_{data}}{T_{ECI}} \tag{3.1}$$

where N_{max} is the maximum number of data packets successfully indicated and confirmed within an T_{ECI}, and L_{data} is the amount of user data in a packet, equal to 20 bytes. Due to the limitations of the BL600 chipset, the Maximum Transmission Unit (MTU) size equals 20 bytes. This implies that only one data packet can be indicated in each CE. Moreover, when transferring data by the indication mechanism, only one packet between indication and confirmation may be exchanged within a CE. The procedure is shown in Figure 3.6 and explained below. At the first CE, the master polls the slave with an empty PDU; the slave sends the Indication PDU, closing the current CE. The latter first packet exchange is defined as *Indication Round Trip* (*IRT*). At the second CE, the master responds with the ATT confirming packet. The slave positively acknowledges the reception of confirming packet with an empty PDU. The latter second packet exchange is defined as *Confirming Round Trip* (*CRT*). *IRT* and *CRT* together represent the *Indication/Confirming Round Trip* (*ICRT*). Given a certain T_{CI}, one indication can be sent and confirmed, every $2 \cdot T_{CI}$. If no packets are lost, it is necessary to have at least $SL = 1$ (2 CI)

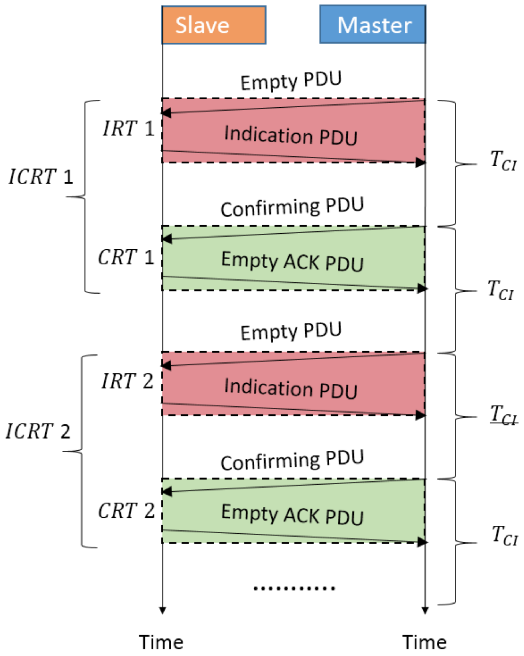


Figure 3.6: The indication/confirmation mechanism of data exchange.

in order to receive the confirming packets within the same T_{ECI}, also keeping the desired goodput equal to the minimum requested over-the-air throughput (overhead excluded). A performance evaluation will be provided in subsection 3.2.3 removing the ideal hypothesis of lack of retransmissions. The results obtained for different levels of signal strength are used to choose an appropriate value of SL in order to reach the desired goodput. The power consumption estimation will be also provided.

3.2.2 Smartphone application design

The flow diagram of the smartphone application is depicted in Figure 3.7. The application automatically starts after device booting, listening for connection availability from the slaves all the time. This process is periodically repeated to allow the smartphone connection to the TST boards as soon as they are within the BLE coverage range. When the slaves are detected, the master tries to connect to them and, if successful, it enables the GATT indication for each of them. By enabling the indication mechanism, the Master schedules synchronous waiting phases, listening for data from the smart shoes. When an indicated packet is delivered to the master, the latter parses the payload, splitting the single step phases and stores them in the internal DB together

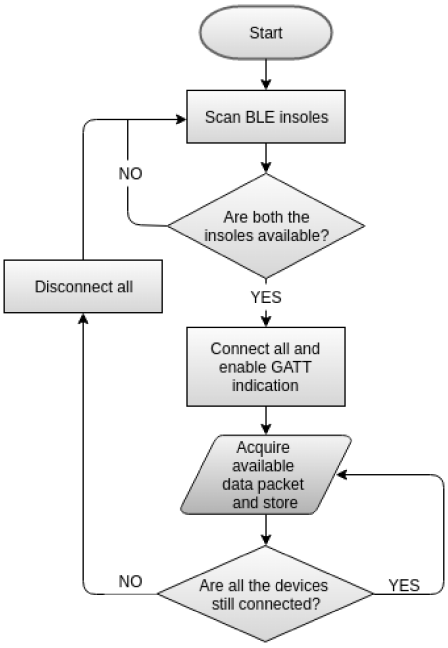


Figure 3.7: Application flow chart: it starts after the device booting and ends when the latter is shouted down.

two identifiers: the shoe type, i.e. right or left, and the arrival timestamp. The time information is not contained in the delivered payload, because the indication mechanism is featured by an application-layer ACK mechanism, that synchronizes the communication and foresees the retransmission of the lost packets. This way, the step phases in a sequence are labeled with an increasing timestamp, with time step equal to the data-acquisition time interval (i.e. 10 ms). When the user is indoor, all the information stored in the internal DB are periodically sent to an external server, i.e. local or remote server. The representation used for data is a simple Comma Separated Values (CSV) text. In order to use the system even outdoor, it was necessary to implement a method to recognize when the user is at home. The smartphone, indeed, continuously stores the data in the internal DB and sends them to the server once back at home. The system is capable to recognize the home connectivity, by checking the SSID of the WiFi network to which the phone is connected. By means of a broadcast receiver, i.e. a feature provided by Android APIs, the mobile phone keeps listening the connection changes, waking up when the home network is identified. Assuming that the user leaves the house and comes back after several hours, the memory footprint of all data acquired in such a period would be considerable. Therefore, the use of the CSV format is very important, since

it reduces the redundancy of the information and makes the message light. As reliability is a design requirement, the throughput chosen for the BLE communication is quite high, and the amount of stored data rises quickly. Choosing a data format characterized by a greater redundancy, such as JSON, causes a significant growth of DB's memory footprint, slowing down the communication and decreasing the application efficiency. If the BLE connectivity falls for at least one slave, in order to keep the simmetrical communication with the pair of shoes, the master drops both the links, waits for the connection timeout, and repeats the procedure starting from the scan phase.

3.2.3 Results and discussion

From an application-level perspective, the presented wearable system requires a low goodput, because a single PDU payload contains an amount of data, which covers a long acquisition time interval (800 ms). The latter may represent the duration of a whole step in a normal walking gait, so, a single lost packet brings to the missed detection of one step. This makes it necessary to have an application-level ACK mechanism. Generalizing for BLE-applications that have low goodput and throughput, but strict reliability requirements, we introduce a possible solution, based on the choice of the indication data transfer mechanism, and proper values of SL (T_{ECI}). The indication mechanism allows a control over the successful data exchange, while the use of non-zero SL allows to retransmit the lost packet, avoiding the loss of synchronization between the application throughput and transmission goodput. Let Γ_{IRT} and Γ_{CRT} be the probabilities of successful IRT and CRT within a T_{ECI}, respectively. Define Γ_{ICRT} to denote the probability of an $ICRT$ being successful within a T_{ECI}. It can be formulated as:

$$\Gamma_{ICRT} = \Gamma_{IRT} \cdot \Gamma_{CRT} \tag{3.2}$$

Define Ω_{ICRT} to represent the case of having an unsuccessful $ICRT$ within an T_{ECI}, which can be expressed as the union of combined mutually exclusive events:

$$\Omega_{ICRT} = \Gamma_{IRT} \cdot \Omega_{CRT} + \Omega_{IRT} \tag{3.3}$$

where Ω_{IRT} and Ω_{CRT} denote the probabilities of an unsuccessful IRT and CRT within a T_{ECI}, respectively.

$$\Omega_{IRT} = 1 - \Gamma_{IRT} \tag{3.4}$$

$$\Omega_{CRT} = 1 - \Gamma_{CRT} \tag{3.5}$$

In order to maintain the desired goodput, keeping Γ_{ICRT} equal to 1 is necessary. The solution is to take advantage of an SL greater than 1. Let k be within the number of available SL intervals for possible retransmission attempts, it follows:

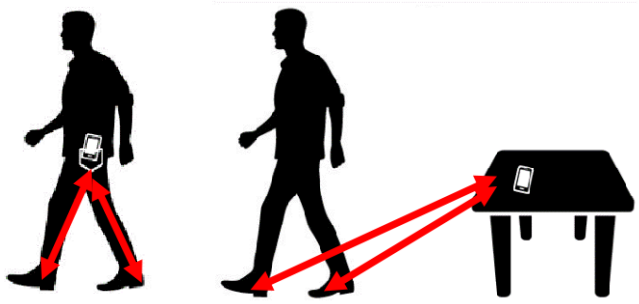
$$\Gamma_{ICRT} = \sum_{k=1}^{k_{lim}} \Gamma_{ICL}(k) = 1 \tag{3.6}$$

where $\Gamma_{ICL}(k)$ represents the probability an ICRT successfully ends in the $k-th$ CI out of the available SL intervals, while k_{lim} refers to the maximum value of k, whereby the ICRT will be completed with a full probability of success. The experimental work aimed at finding a value for k_{lim} in different conditions of signal strength. The results, shown in Table 3.2, have been obtained with the system presented in Section 3.1. We programmed the slaves in order to send an indication packet (with transmission power equal to 4 dBm), waiting for a confirming packet before sending the next indication. The T_{CI} has been set equal to 100 ms. Time T_{ICL}, i.e. the indication/confirming latency, has been measured during 2 minutes of recorded streaming data, for 5 times, for different values of RSSI (Received Signal Strength Indicator) of the Master (smartphone) receiving radio. Table 3.2 summarizes the percentage of Γ_{ICL}, derived from the recorded values of T_{ICL}, for each k and RSSI and highlights in gray the different values of k_{lim} related to a certain value of RSSI. The maximum found value of k_{lim} is equal to 6 and it is relative to an RSSI of -75 dBm, i.e. a condition of very weak signal strength. As a consequence, the maximum indication/confirming latency is equal to $k_{lim} \cdot T_{CI}$ and the SL must be equal or greater to $k_{lim} = 6$ to keep the desired goodput and to guarantee the consistence of the streaming with the application settings, as dictated by Eq. 3.6. Considering the outdoor scenario (Figure 3.8a), in which the user wears the smart shoes and typically carries the smartphone inside his pants pocket, the mean value of RSSI is approximately -60 dBm and the relative value of k_{lim} is equal to 3. In indoor scenario (Figure 3.8b), instead, the user might not have the smartphone with him; accordingly, the variable distance of connected devices, the complex multipath channels and nonline-of-sight (NLOS) conditions bring to lower values of RSSI and higher values of k_{lim}, compared to the typical devices configuration when using them in external environment.

With fixed T_{CI} (e.g. 100 ms), the use of a not null SL may also guarantee a lower power consumption by the slave, if, and only if, the latter is able to avoid as much as possible the awakening at each SL CIs. With reference to the values of RSSI and Γ_{ICL} the absorbed average currents, weighted by the Γ_{ICL} probabilities, have been estimated and summarized in Table 3.3. Assuming a linear model for the discharge characteristic of a Li-polymer rechargeable battery [73], we also estimate the battery life duration for each value of RSSI,

Table 3.2: Experimental results on the number of SL needed to end the $ICRT$, related to different values of RSSI.

Γ_{ICL} [%]		k						
		1	2	3	4	5	6	7
RSSI [dBm]	-50	97.7	2.2	0.1	0.0	0.0	0.0	0.0
	-55	92.8	6.8	0.4	0.0	0.0	0.0	0.0
	-60	87.8	11.7	0.5	0.0	0.0	0.0	0.0
	-65	79.2	17.8	2.6	0.4	0.0	0.0	0.0
	-70	65.6	27.6	5.3	1.2	0.3	0.0	0.0
	-75	56.2	30.1	10.5	2.2	0.7	0.3	0.0



(a) Outdoor (b) Indoor

Figure 3.8: a) Outdoor scenario: short distance between shoes and smartphone, because the latter is carried by the user; b) indoor scenario: variable distance between shoes and smartphone, unpredictable RSSI.

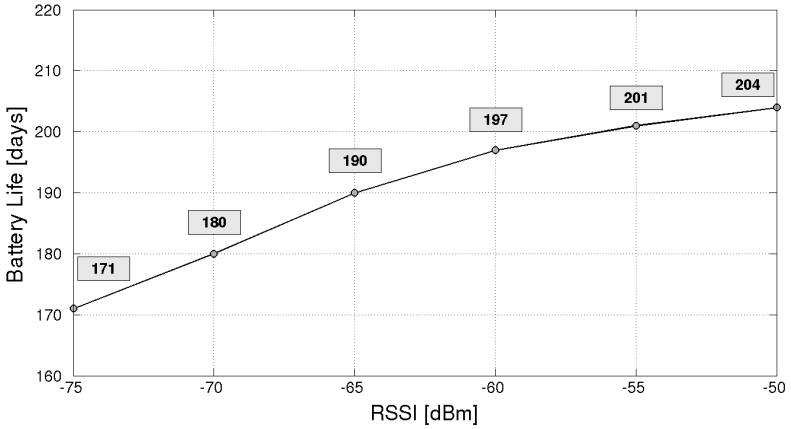


Figure 3.9: Insole battery life estimation, assuming a time-invariant RSSI.

Table 3.3: Insole average current consumption, assuming a time-invariant RSSI.

$RSSI[dBm]$	$I_{avg}[\mu A]$
-50	114
-55	116
-60	118
-65	122
-70	130
-75	136

dividing the nominal capacity (at 0.2C equal to 560 mAh) by the average current consumption of Table 3.3. The obtained values, expressed in days, are plotted in Figure 3.9.

Considering the typical RSSI of -60 dBm in outdoor scenario, the estimated battery life of 197 days demonstrates the suitability of BLE technology and application design in order to minimize the power consumption, keeping high data reliability.

Chapter 4

Periodic Random Access Protocol

MAC protocols for WSNs are usually designed to guarantee a trade-off between a suitable access to the communication medium and an efficient energy-saving policy. The neighbor discovery problem mainly relates to the nodes' ability to discover each other as quickly as possible without any prior knowledge or synchronization, bounding the energy waste at the same time. Usually, these protocols exploit duty-cycled radio activations. It is the case of the here denoted Periodic-Random-Access-to-Link-Layer (PRALL) protocol, implemented by BLE and introduced in Section 4.1. This protocol can be used also for transmitting small-sized data, without any prior connection establishment, expecially for event-driven applications in WSN. Therefore, the data-delivery-latency estimation issue is reflected in the discovery-latency estimation problem. In subsection 4.1.1 the PRALL discovery procedure is modeled and, finally, in Section 4.2 a discussion about the model results are provided (subsections 4.2.1 and 4.2.2), ending in subsection 4.2.3 with a modified PRALL proposal aimed at limiting the discovery-latency for some critical parameterizations.

4.1 Problem Description

In the neighbor-discovery and connectionless operation design, an important energy-efficiency metric to consider is the discovery-latency, which is the time the node spends in active state, from the beginning of transmission to the successful delivery of advertisement and/or data to the receiver node. High discovery-latency leads to increased energy consumption; therefore, limiting it is a good practice. In the first subsection we provide an overview on the PRALL protocol, used by BLE at the MAC-layer, while in the second subsection a deterministic approach is proposed to quantify the mean discovery-latency for the most of the possible parameters configurations.

4.1.1 Overview on PRALL protocol

In BLE PRALL protocol, the advertiser periodically sends a burst of three packets, each one on a different channel. This operation is called advertising-event. The interval between two advertising-events is labeled as advertising-interval and denoted by T_{adv}. Asynchronously, the scanner duty-cycles between scan- and idle- phases with period T_{si}, that is the scan-interval. The scan duration per period is called scan-window and denoted by d_{sw}. The scanner senses the 3 channels in a round-robin fashion. This implies that the duration of a complete scan is equal to $3T_{si}$. The advertising-interval is the sum of two terms, as shown in Equation (4.1).

$$T_{adv}(n) = T_{adv,0} + \rho(n) \tag{4.1}$$

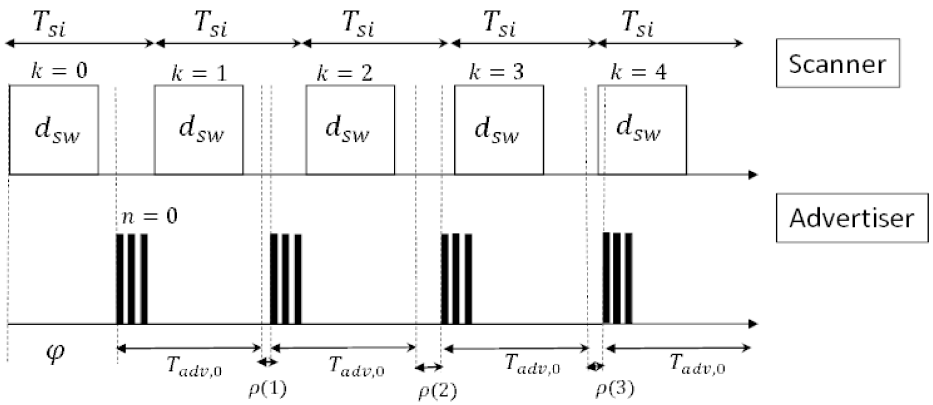


Figure 4.1: Timeline of the scanner-advertiser in the discovery procedure. The scanner duty-cycles transition from scan phase to idle phase, listening to advertiser messages on the channel 37, 38 or 39. The advertiser sends a burst of three packets on the three advertising channels in round-robin fashion. The advertising interval is composed of a static term $T_{adv,0}$ and a random additive term $\rho(n)$.

$T_{adv,0}$ is a settable parameter that characterizes the periodic advertiser behaviour, while ρ is a dynamic additive term, randomly chosen up to a small-size bound time (10 ms for BLE), for each n-th advertising-event. It is produced by an hardware Random Number Generator (RNG) and has a double aim:

(i) reducing the probability of advertising packets collision;

(ii) preventing an advertiser from missing consecutive scan-events.

Given the Equation (4.1) and the scanner-advertiser timeline in Figure 4.1, the starting-time of n-th advertising-event can be modeled by Equation (4.2).

$$t_a(n, \phi) = \phi + n \cdot T_{adv,0} + \sum_{i=1}^{n} \rho(i) \tag{4.2}$$

where ϕ represents the offset between the starting-time of the first advertising-event and the starting-time of the left-closest scan-event. Setting out T_{si} and d_{sw} for the scanner and $T_{adv,0}$ for the advertiser, the estimation of the discovery latency for the tuple $(T_{si}, d_{sw}, T_{adv,0})$ results in finding the number n of advertising-events which verifies the hit-condition expressed by Equation (4.3).

$$k \cdot T_{si} - d_{early}(ch) \leq t_a(n, \phi) \leq k \cdot T_{si} + d_{sw} - d_{late}(ch) \tag{4.3}$$

where ch represents the sensing channel within the k-th scan-window, and is computed step by step as in Equation (4.4).

$$ch(k) = mod(k, 3) + 37 \tag{4.4}$$

The durations $d_{early}(ch)$ and $d_{late}(ch)$ represent the time before the beginning (the first) and the end (the second) of the k-th scan-event that allows the n-th advertising-event to fall within the k-th scan-window; they are function of ch and, consequently, depend on the k-index as shown in Table 4.1. It follows that the effective scan-window d'_{sw} is shorter than d_{sw}. More in detail, its value is $(d_{sw} - d_a)$, where d_a denotes the time for sending an advertising packet on a single channel and listening to a possible response. The BLE protocol foresees that this time must be up to 10 ms, but usually, the value is much lower, almost 1 ms for passive scanning. In the case of active scanning this value can be larger, but always limited to 10 ms.

Table 4.1: Effective scan-window parameters. The duration $d_{early}(ch)$ and $d_{late}(ch)$ are functions of ch and represent the time before the beginning (the first) and the end (the second) of the scan-event, that allows the advertising-event to fall within the scan-window. d'_{sw} represents the effective scan-window, which is shorter than d_{sw}.

Channel	d_{early}	d_{late}	d'_{sw}
37	0	d_a	$d_{sw} + d_a$
38	$d_a + d_{ch}$	$2d_a + d_{ch}$	$d_{sw} + d_a$
39	$2d_a + 2d_{ch}$	$3d_a + 2d_{ch}$	$d_{sw} + d_a$

Finally, the discovery-latency is computed by using the Equation (4.5) for any random offset ϕ.

$$d(\phi) = t_{a,n}(\phi) + d_a \tag{4.5}$$

Unfortunately, the random value $\rho(n)$ makes the problem stochastic, hence, any implemented model can only estimate the couple of values (n, k) which verifies with high probability the inequality in Equation (4.3). Moreover the Equation (4.3) and (4.5) are referred to a single value of ϕ, that behaves like a random uniform variable, due to the asyncronous feature of PRALL operation. Despite these considerations, in subsection 4.1.2 we approach the discovery-latency estimation with a deterministic model.

4.1.2 PRALL deterministic-model

We formally define the problem, starting with these assumptions:

(i) the frequency-hopping is not considered, thus the scanner senses a single advertising channel as in STEM-B [45]; therefore, the duration of a complete scan collapses to one scan interval T_{si};

(ii) the random term $\rho(n)$ is forced to the first moment of the uniform distribution $\mu_\rho = E(\rho) = \frac{1}{2}(\rho_b - \rho_a)$ to make the problem deterministic.

The assumption (i) lays down the domain of the random variable ϕ between 0 and T_{si}. In order to face the problem in a full-deterministic way, we cannot consider ϕ as an uniform distributed variable, but as a discrete set, denoted by Φ, populated by all the initial offsets $\phi \in [0, T_{si}[$, spaced of δ_ϕ time-unit. Under the assumptions (i) and (ii) the Equation (4.2) shall be updated as in Equation (4.6):

$$t_{a,n}(\phi) = \phi + n \cdot (T_{adv,0} + \mu_\rho) + d_a = \phi + n \cdot T_{adv} + d_a \tag{4.6}$$

where $\phi \in \Phi$. As said in subsection 4.1.1, PRALL uses the random delay ρ in order to prevent an infinite loop of missing advertiser-scanner hits. In fact, for certain tuples $(T_{si}, d'_{sw}, T_{adv,0})$, there are some offsets ϕ that do not bring to a successful advertiser-scanner hit in lack of a random delay. This implies that the latency for those tuples will tend towards infinite for certain offsets ϕ, making also the mean discovery-latency undefined. We denote by Φ_{nh} the subset of Φ, populated by all the initial offsets ϕ that do not allow the convergence of the model. The complementary of Φ_{nh} is denoted by $\Phi_h = \Phi^*_{nh}$ and contains all the ϕ for which the discovery-latency is computable. Mathematically, finding the elements of Φ_{nh}, is possible exploiting the inequality expressed by Equation (4.7).

$$m \cdot gcd(T_{si}, T_{adv}) + d'_{sw} < \phi < (m+1) \cdot gcd(T_{si}, T_{adv}) \tag{4.7}$$

where m is an integer from 0. If ϕ falls within the intervals produced by Equation (4.7), overlaps between advertising-event and scan-event will never be found, otherwise the hits will occur at least every $T_{adv}/lcm(T_{si}, T_{adv})$ transmissions of the advertiser. As a consequence, a deterministic approach to the discovery issue will be able to quantify the mean discovery-latency for all the parameterizations that do not result in an undefined latency. In other terms, for a given parameterization, only if the subset Φ_{nh} is empty, the mean discovery-latency can be reckoned. It is important to note that, if the Equation (4.7) is not verified for $m = 0$ then no intervals for m greater of 0 will satisfy the disequality and, this implies the subset Φ_{nh} will be ever empty. Otherwise, if the inequality is proved for $m = 0$, it can be semplified by the Equation (4.8):

$$d'_{sw} < \phi < gcd(T_{si}, T_{adv}) \tag{4.8}$$

Thus, if the greatest-common-divisor (gdc) between T_{si} and T_{adv} is less than the effective scan-window d'_{sw}, then at least one item of ϕ is also element of Φ_{nh}. For parameterization in which Φ_{nh} is empty we have demonstrated that the discovery-latency has a finite value. In order to calculate the latter, firstly it is necessary to distinguish some cases.

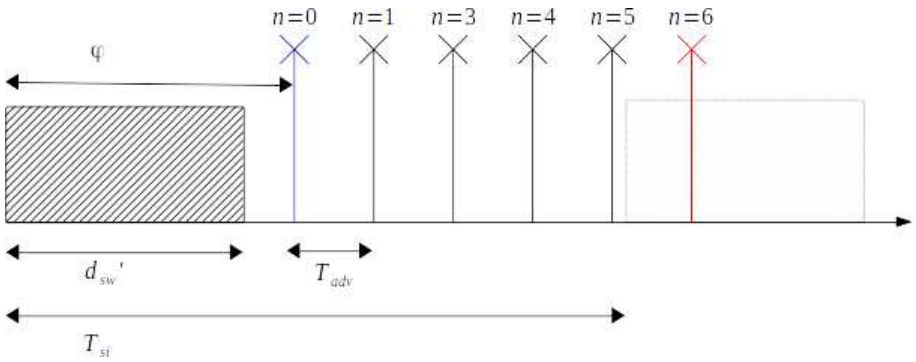


Figure 4.2a: Scheme of scanner-advertiser discovery procedure for the case $T_{adv} \leq d'_{sw}$. The advertiser hit the scan window in at least one scan interval.

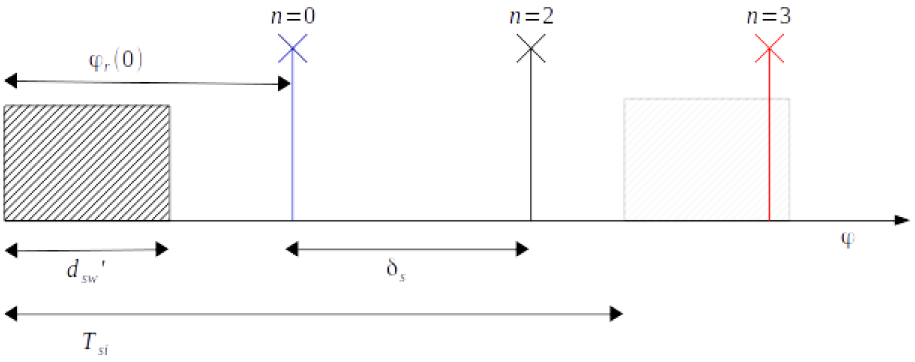


Figure 4.2b: Scheme of scanner-advertiser discovery procedure for the case $T_{adv} > d'_{sw}$. The relative anchor-point shifts with step of δ_s until the hit of the scan-window occurs.

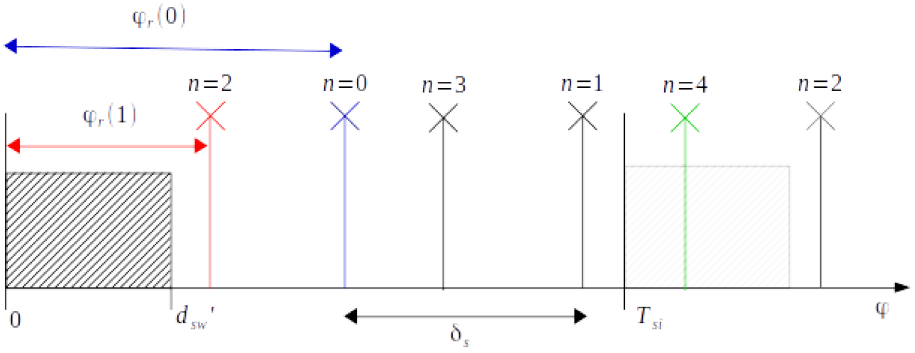


Figure 4.2c: Scheme of scanner-advertiser discovery procedure for the case $T_{adv} > d'_{sw}$. The relative anchor-point shifts with step of δ_s until the overtake of the scan-window occurs.

1. If $T_{adv} \leq d'_{sw}$, then the discovery-latency calculation results in the simplest case. Considering the timeline depicted in Figure 4.2a, it becomes clear that the advertiser will hit the scan window in at least one scan interval. In fact, the gap between the n-th advertising anchor-time and the right-closest scan-window shrinks until falling to zero or becoming negative. Mathematically, it is expressed by the Equation (4.9).

$$d(\phi) = \begin{cases} \left\lceil \frac{T_{si}-\phi}{T_{adv}} \right\rceil T_{adv} + d_a & if\ \phi \geq d_{sw} \\ 0 & otherwise \end{cases} \tag{4.9}$$

2. Whenever $T_{adv} > d'_{sw}$ a successful discovery within one scan interval is not guaranteed anymore. It is expedient to use a different representation, considering the advertising anchor-points in ϕ-time-domain. This approach semplifies the handling of the two cases $T_{adv} \geq T_{si}$ and $T_{adv} < T_{si}$, which will be treated together, considering a single time-domain, equal to a single scan-window T_{si}, wherein the advertising anchor-point relatively moves, shifting to the right with step of $\delta_s = (T_{adv} \bmod T_{si})$. For the sake of clarity, we use different notations to represent the relative anchor-points and the initial offset: $t_{a,r}$ and $\phi_r(0)$. As such, we have to further distinguish two sub-cases: $\delta_s \leq d'_{sw}$ and $\delta_s > d'_{sw}$.

   2.1. $\delta_s \leq d'_{sw}$. The relative anchor-point $t_{a,r}$ shifts with step of δ_s without ever overtaking the scan-window. Thus, the latency is easily expressed by the Equation (4.10), that is similar to Equation (4.9) except for δ_s taking the place of T_{adv}, and for the different notation for ϕ.

$$d(\phi) = \begin{cases} \left\lceil \frac{T_{si} - \phi_r(0)}{\delta_s} \right\rceil T_{adv} + d_a & if\ \phi_r(0) \geq d_{sw} \\ 0 & otherwise \end{cases} \tag{4.10}$$

   2.2. $\delta_s > d'_{sw}$. The relative anchor-point $t_{a,r}$ shifts with step of δ_s until the hit or the overtake of the scan-window occurs. In the former case, depicted in Figure 4.2b, the latency can be calculated already using the Equation (4.10), while in the latter case the computation becomes more involved. In fact, as it can be seen in Figure 4.2c, the anchor-point gap from the right-neighbor scan-window shrinks until the first $n = \left\lfloor \frac{T_{si} - \phi_r(0)}{\delta_s} \right\rfloor$, while drastically grows after the scan-window overtaking. The new position of the anchor-point is calculated as in Equation (4.11).

$$t_{a,r}(n, \phi_r(i)) = \delta_s - [(T_{si} - \phi_r(i)) \bmod \delta_s] \tag{4.11}$$

   where we define $\phi_r(i)$ as the first offset of $t_{a,r}$ from the left-closest scan-window after the i-th scan-window overtake. The procedure must be iterated exploiting in sequence the Equations (4.10) and (4.11) until the anchor-point will fall within the right-closest scan-window, that will occur when the Equation (4.12) will be verified.

$$0 \leq \delta_s - [(T_{si} - \phi_r(i)) \bmod \delta_s] \leq d'_{sw} \tag{4.12}$$

Algorithm 1 implements the above explained procedure. It takes the tuple $(T_{si}, d'_{sw}, T_{adv,0})$ as input and computes the mean of the discovery-latencies

swiping all the initial offsets ϕ, selected with pipe step factor $\delta_\phi = 1$ ms.

Algorithm 1 Mean discovery-latency calculation. The function takes as input arguments the scan-interval T_{si}, the effective scan-window d_{sw} and the advertising-interval $T_{adv,0}$, and gives back the mean discovery-latency d_{mean}.

function *deterministicDiscoveryLatency*$(T_{si}, d'_{sw}, T_{adv,0})$
  $T_{adv} \leftarrow T_{adv,0} + \mu_\rho$
  if $d'_{sw} \geq gcd(T_{si}, T_{adv})$ **then**
    $d_{cM} \leftarrow 0$
    for $\phi = 0$ to T_{si} with step of δ_ϕ **do**
      if $\phi \leq d'_{sw}$ **then**
        $n \leftarrow 0$
      else
        if $T_{adv} \leq d'_{sw}$ **then**
          $n \leftarrow \left\lceil \frac{T_{si}-\phi}{T_{adv}} \right\rceil$
        else
          $\delta_s \leftarrow (T_{adv} \bmod T_{si})$
          $n \leftarrow 0$
          $\phi_r \leftarrow \phi$
          do
            $n \leftarrow n + \left\lceil \frac{T_{si}-\phi_r}{\delta_s} \right\rceil$
            $\phi_r \leftarrow \delta_s - [(T_{si} - \phi_r) \bmod \delta_s]$
          while $\phi_r > d'_{sw}$
        end if
      end if
      $d_{cM} \leftarrow d_{cM} + nT_{adv} + d_a$
    end for
    $d_{mean} \leftarrow \frac{d_{cM}}{\left\lfloor \frac{T_{si}}{\delta\phi} \right\rfloor}$
  else
    $d_{mean} \leftarrow +\infty$
  end if
  return d_{mean}
end function

4.2 Discussion

In the previous section a deterministic model of PRALL protocol has been proposed and implemented by the Algorithm (1). In this section, the output of the model will be analyzed and compared with simulation results. Furthermore, the discussion results will show off some shortcomings of PRALL protocol for

critical parameterization. Finally, an improvement of PRALL, aimed at outperforming the capacity of upper-bounding the latency, has been proposed and discussed.

4.2.1 Deterministic Model Evaluation

Figure 4.3 depicts an example outcome of the Algorithm (1) for the parameterization reported in Table 4.2; on the x-axis the ascending values of $T_{adv,0}$ are given, while on the y-axis the related latencies are expressed as the mean numbers $n_{mean}(T_{adv})$ of the advertising-intervals necessary to successfully hit a scan-window. As can be seen, the curve results in some peaks, and, some of these lack of the latency value.

Table 4.2: Scanning and Advertising configuration used for model evaluation.

Parameter	**Value [ms]**
T_{si}	100
d'_{sw}	20
$T_{adv,0}$	$[20 \div 250]$

As said in Section 4.1, there are some parameters settings, which do not allow the model to converge, because the modeled advertiser cannot bridge the gap from the closest scan-window without randomizing the delay ρ. In other words, the infinite values are due to the assumption that the random delay can be deterministically expressed as the first moment of the uniform distribution μ_ρ. We can found all the $T_{adv,0}$ which act as vertical asymptotes for the latency, exploting the condition in Equation (4.8), that is below re-proposed in Equation (4.13), where $(T_{adv,0} + \mu_\rho)$ replaces T_{adv} in order to match the abscissa-unit.

$$d'_{sw} < \phi < gcd(T_{si}, T_{adv0} + \mu_\rho) \tag{4.13}$$

Given the parameterization in Table 4.2, the Equation (4.13) allows to identify the vertical asymptopes, which are centered in $T_{adv,0} = \{mT_{si} + 20, mT_{si} + 45, mT_{si} + 70, mT_{si} + 95\}$ ms, where m is an integer from zero and characterizes the periodicity T_{si} of peaks. In fact, the whole trend of n_{mean} at the varying of $T_{adv,0}$ shows a repetitive pattern over the $T_{adv,0}$-domain, $D(T_{adv,0})$. The repetitive sub-domain, denoted by $D_u(T_{adv,0})$, extends from $\overline{T}_{adv,0}$ and $(\overline{T}_{adv,0} + T_{si})$, where $\overline{T}_{adv,0}$ is an arbitrary value of T_{adv}. This evidence speeds up the computation of n_{mean}, which can be obtained varying $T_{adv,0}$ up to only a single T_{si}-wide sub-domain. In view of the above, a first conclusion can be formulated: the deterministic model is not suitable for the cases in which

the greatest-common-divisor between the scan- and the advertising- intervals is greater than the effective scan-window d'_{sw}.

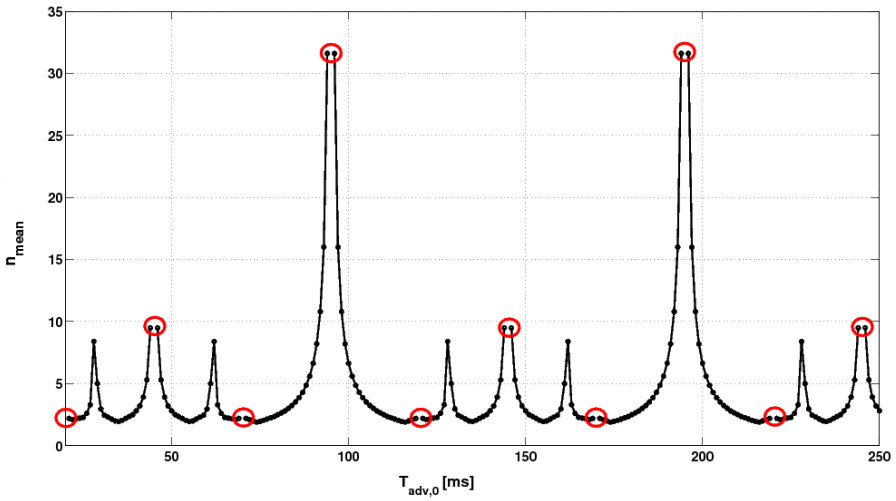


Figure 4.3: Results of the Algorithm (1) for the parameterization in Table 4.2. On the x-axis the ascending values of $T_{adv,0}$ are given, while on the y-axis the related latencies are expressed as the mean numbers $n_{mean}(T_{adv})$ of the advertising-intervals necessary to successfully hit a scan-window. The undefined values of n_{mean} are highlighted with a red-circle.

In order to estimate the discovery-latency in these cases, it is necessary to exploit a probabilistic approach, as suggested for example by Kindt et al. in [51]. This model exploits an iterative-probabilistic model, starting from the characterization of the probability density function (pdf) of the anchor-point at each n-th iteration, passing through the calculation of the cumulative probability that a total of n advertising-events lead to a close to full hit-probability in a range of k scan-events. With reference to the expression of the n-th anchor-point in Equation (4.2), the pdf of the latter results in a gaussian distribution centered in $\mu_g = n \cdot \mu_\rho$ with standard deviation $\sigma_g = \sqrt{\frac{n}{12}}\rho_b$, where ρ_b is the upper-bound delay (10 ms in BLE). Further details of this approach will be provided in the next chapter, where, in addition, a modification of the model in [51] is presented.

4.2.2 Model-Simulation Comparison

In order to evaluate the ability of the deterministic model and to give a reliable description of the discovery-latency estimation, we have implemented a custom discrete-event simulator in ©Matlab, aimed at the simulation of the discovery process. More in details, for each value of $T_{adv,0}$, varying in a range of T_{si}, the simulator performs $10^3 \cdot T_{si}$ iterations, picking up an initial random offset ϕ between 0 and T_{si} for each of them. Each iteration ends when the n-th advertising-event hits the k-th scan-event and the reached n is recorded. Finally, all stored n are averaged over the total number of the performed iterations, returning the $n_{mean,sim}(T_{adv,0})$. In Figure 4.4 the Algorithm 1 results are compared to the performed simulation outcomes; the parameterization in Table 4.2 is already used. At first glance, the similar trend between the curves appears clearly, except for the peaks values lacking in the former.

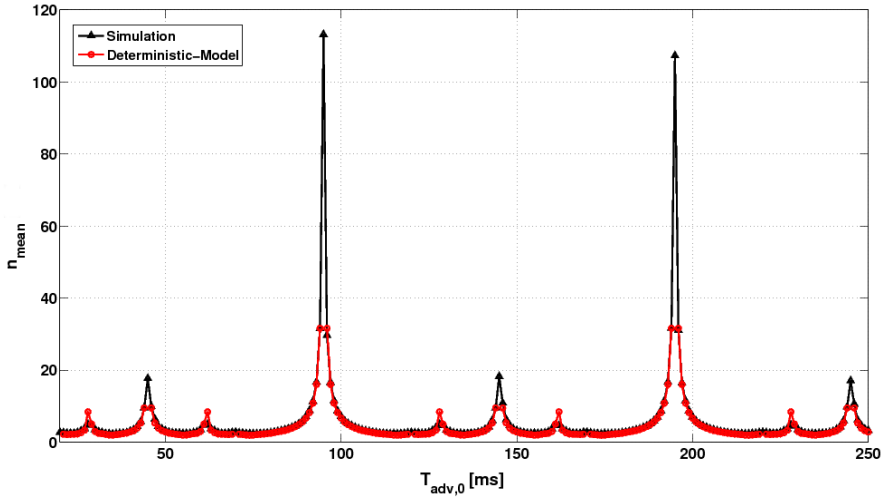


Figure 4.4: Comparison between simulator results, black triangle-pointed, and Algorithm (1) results, red circle-pointed, for the parameterization in Table 4.2. On the x-axis the ascending values of $T_{adv,0}$ are given, while on the y-axis the related latencies are expressed as the mean numbers $n_{mean}(T_{adv})$ of the advertising-intervals necessary to successfully hit a scan-window.

In order to allow for a quantitative comparison between simulation and model-computed values, we have calculated the Root Mean Square Error (RMSE), the Root Mean Square Error of prediction (RMSEP) and the Maximum Deviation (MD) exploiting the Equation (4.14) (4.15) (4.16), but excluding from

Table 4.3: Model-Simulation comparison.

	RMSE	**RMSEP**	**MD**
$dc = 80\%$	0.045 ± 0.017	0.073 ± 0.057	0.382 ± 0.200
$dc = 50\%$	0.131 ± 0.048	0.032 ± 0.020	1.297 ± 0.631
$dc = 20\%$	0.615 ± 0.189	0.084 ± 0.051	5.611 ± 2.743

the computation the undefined peaks.

$$RMSE = \sqrt{\frac{\sum_{D_u(T_{adv,0})}(n_{mean,sim}(T_{adv,0}) - n_{mean}(T_{adv,0}))^2}{\mathbf{card}(D_u(T_{adv,0}))}} \tag{4.14}$$

$$RMSEP = \sqrt{\frac{\sum_{D_u(T_{adv,0})}(n_{mean,sim}(T_{adv,0}) - n_{mean}(T_{adv,0}))^2}{\mathbf{card}(D_u(T_{adv,0}))}} \tag{4.15}$$

$$MD = max(|n_{mean,sim} - n_{mean}|) \tag{4.16}$$

The RMSE, RMSEP and MD have been reckoned for three different settings of the scan duty-cycle $dc = \frac{d'_{sw}}{T_{si}} \in \{20\%, 50\%, 80\%\}$, varying T_{si} from 50 ms to 500 with steps of 25 ms. Table 4.3 summarizes the average of the RMSE, RMSEP and MD values up to all the tested T_{si} settings. It further highlights the similarity between the two trends, particularly for scan duty-cycle $dc = 50\%$, which the minimum error has been obtained for. For dc greater or lower than 50%, the errors increase, leading to a slight over-estimation of the latency. We can distinguish two situations:

(a) if dc decreases, then the idle-time $d_{idle} = T_{si} - d'_{sw}$ increases as well as the number of the secondary peaks. As shown in Figure 4.4, the maximum deviation between the two curves relates to these peaks. This is further demonstrated by MD, which results in the largest value for $dc = 80\%$.

(b) if dc increases, then d_{idle} shrinks, and, consequently, the assumption formalised by Equation (4.6), i.e. μ_ρ in place of $\sum_{i=1}^{n} \rho(i)$, weakens, and, therefore, brings to an over-estimation of the latency.

4.2.3 Improved PRALL

Figures 4.3 and 4.4 highlighted that the latency drastically increases in some configurations, leading to noticeable peaks. The latters are centered in specific and repetitive positions. Among these, the peaks centered in $T_{adv,0} =$

$(T_{si} - \mu_\rho) + mT_{si}$ result in the highest latencies. These values cannot be often computed by the simulator without overtake the upper-limit computational time, suggesting the discovery-latency procedure can reach unacceptable values in real scenario. Thus, such a long latency must be avoided, using coordinated parameterizations for advertisers and scanners. Another solution could be performed, if the advertiser was able to trigger on the achievement of a certain threshold of n undelivered advertisements, modifying its own scheduling. This implies an acknowledgement mechanism which allows the advertiser to receive the advertisement-delivery confirmation (ACK) from the scanner. Such a mechanism is for example foreseen by BLE specifics and denoted as active-scan, despite the passive-scan. The difference is that an active scanner can send a scan request to confirm the advertisement delivery and/or request additional information from the advertiser, while a passive scanner can only receive data from advertising devices. The Figure 4.5 below shows the sequence of packets exchange in active scan. We propose an improved version of PRALL, here denoted as modified-PRALL (mPRALL), exploitable only if the advertiser has or can obtain the knowledge of the scan schedule. The solution is based on a simple observation: given that for $T_{adv,0} = (T_{si} - \mu_\rho) + mT_{si}$ the latency peak drastically arises, it is also true that not for all the offsets ϕ this configuration leads to a long discovery blindness. In particular the range of ϕ, that results in critical growing of n, is centered on the half of the scanner idle-time, that is $d_{idle} = T_{si} - d'_{sw}$. This is because, while n grows, the anchor-point oscillates about its central value, unable to close the gap from the neighbor scan-window within a reasonable time. In other terms, the random delay ρ is not sufficient to unlock the impasse. Thus, the idea is to implement a threshold of n, after that the advertiser will use a larger and fixed value of ρ in order to time-shift the anchor-point in a more favourable position. This operation may prove contra-indicated for ϕ out of the considered range, so the protocol improvement could not provide a pareto-optimal solution; for this reason fixed $\rho_f = \gamma T_{si}$ must be periodically alternate to random picked ρ as depicted by the flow-chart in Figure 4.6. The performed simulation has shown a good compromise for threshold n_t, set as in Equation (4.17), $\omega = 2$ and $\gamma = 3/4$.

$$n_t = \left\lfloor 12 \frac{(T_{si} - d'_{sw})^2}{(3\rho_b)^2} \right\rfloor \tag{4.17}$$

In order to evaluate the suitability of the improved version of PRALL we have properly modified the simulator.

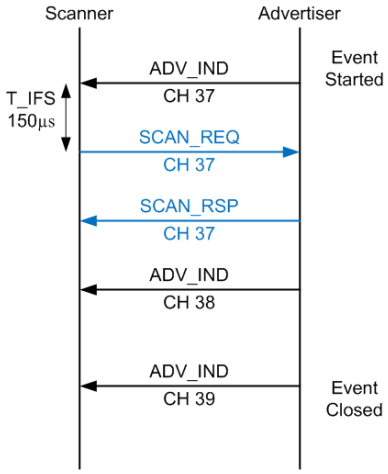


Figure 4.5: Sequence of packets exchanged in the active-scan procedure. The advertiser sends a burst of three packets on the three advertising channels in round-robin fashion. The scanner listens to incoming advertisements once for each channel 37, 38 or 39. When the scanner receives an advertising packet, it replies with an unicast scan-request (SCAN_REQ). After the reception of the latter, the advertiser closes the ACK procedure broadcasting a response (SCAN_RESP).

Table 4.4: Model-Simulation comparison.

	AR	**MDP**
$dc = 80\%$	1.327 ± 0.080	15.336 ± 11.955
$dc = 50\%$	1.395 ± 0.091	136.757 ± 80.771
$dc = 20\%$	1.231 ± 0.101	270.447 ± 109.997

Figure 4.7 shows the comparison between PRALL and mPRALL discovery returned by the simulator, using the configuration in Table 4.2, for the sake of consistency. At first glance the exploited modifications lead to an important reduction in latency for maximum peaks, less evident for the secondary ones. To give a sample quantification of the outperforming, we have computed two performance indexes, i.e. Average Ratio (AR) and Maximum Positive Deviation (MPD), defined by the Equations 4.18 and 4.19, where $n_{mean,sim}$ and

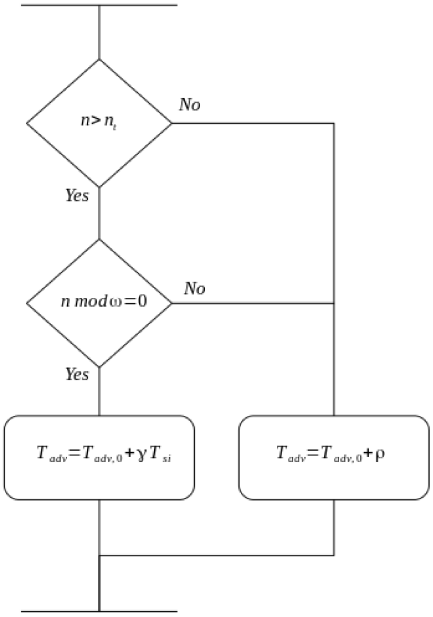


Figure 4.6: Flow-chart of ρ picking-up in mPRALL. When the advertiser reaches the upper threshold of unsuccessful attempts, it switches ρ from γT_{si} to a random value picked up in a range between 0 and ρ_b, i.e. 10 ms for the BLE.

$\hat{n}_{mean,sim}$ refer to standard PRALL and modified PRALL, respectively.

$$AR = \frac{\sum_{D_u(T_{adv,0})} n_{mean,sim}(T_{adv,0})}{\sum_{D_u(T_{adv,0})} \hat{n}_{mean,sim}(T_{adv,0})} \tag{4.18}$$

$$MPD = max(n_{mean,sim}) - max(\hat{n}_{mean,sim}) \tag{4.19}$$

AR and MDP have been computed for three different settings of the scan duty-cycle $dc \in \{20\%, 50\%, 80\%\}$, varying T_{si} from 50 ms to 500 with steps of $25ms$. Table 4.4 summarizes the average of AR and MDP values up to all the tested T_{si} settings. It clearly highlights the PRALL outperforming by the proposed improved version in terms of discovery efficiency. The value of the averaged MPD-index for $dc = 20\%$ emphasises the noticeable reduction of the maximum peak of latency. In addition, the averaged AR-index, always greater than 1, demonstrates that mPRALL represents a pareto-optimal solution, because of the robustness of the latency reduction for different configurations.

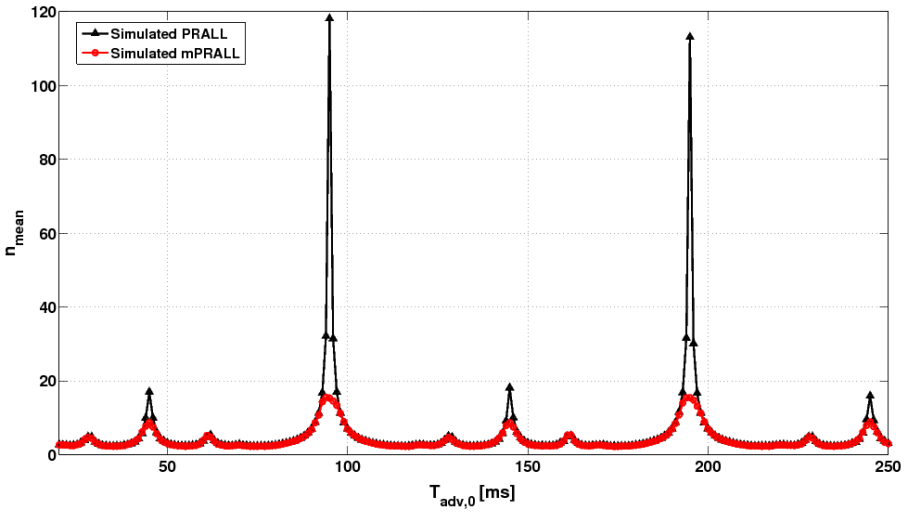


Figure 4.7: Comparison between standard PRALL simulator results, black triangle-pointed, and mPRALL simulator results, red circle-pointed, for the parameterization in Table 4.2. On the x-axis the ascending values of $T_{adv,0}$ are given, while on the y-axis the related latencies are expressed as the mean numbers $n_{mean}(T_{adv})$ of the advertising-intervals necessary to successfully hit a scan-window.

Chapter 5

Connectionless and connection-oriented devices coexistence

Connectionless communication can be used to discover devices available for connection or to asyncronously exchange data. The latter is the case of the sensors that need to turn on the radio to attempt data transmission only when a sensor-event occurs. To evaluate the energy efficiency of this communication paradigm, it is necessary to characterize the time the sensor spends from the detection and transmission of an event to the successful delivery of a packet to the central node. This time can be denoted as discovery-latency or data-delivery-latency. Anyway, the main advantage of connectionless communication is the greater energy saving with respect to the connection-oriented communication, specifically in the case of few and unpredictable data to be exchanged. If an application requires that both the unconnected and connected devices share a single central-node, proper parameters configuration is necessary to keep the data-delivery-latency low, while at the same time avoiding undesired overlaps between scanning and connection phases, due to a bad medium access management. Section 5.1 will provide the description of the model used for the connectionless, in subsection 5.1.1, the connection-oriented BLE communication, in subsection 5.1.2, and, the developed tool for parameters configuration in subsection 5.1.3. Section 5.2 will provide some results on the developed models in subsection 5.2.1 and shows how to use the provided tool in subsection 5.2.2 and 5.2.3, with the aim to balance and optimize the BLE-network setting. The methods and the results of the topic described in this chapter are included in [18].

5.1 Problem Description

The BLE protocol recommends several parameter settings for connectionless and connection-oriented communications, and their proper tuning to balance

and optimize the performance for a wide range of applications in terms of discovery-latency, energy efficiency and throughput. Some application scenarios require that unconnected sensors, that monitor relatively unpredictable and rare events, could operate in coexistence with sensors that are always connected, continuously communicating health or environmental data. The coexistence of connection-oriented and connectionless networks, sharing a single central/collector node can be required to reduce the number of the handling devices (acting sometimes as gateways), keeping the network complexity low and limiting the packet traffic congestion. The central device must be programmed in order to better schedule scanning and connection duty-cycles, thus optimizing the use of TDMA scheme.

5.1.1 Connectionless Model

As widely said the most important metric to measure the energy efficiency of MAC-protocol for neigbour discovery is the discovery-latency, which is the time the advertiser (or tag) spends in active state, from the start of transmission to the successful delivery of the advertising packet. High discovery-latency leads to increased energy consumption; therefore, limiting it is good practice.

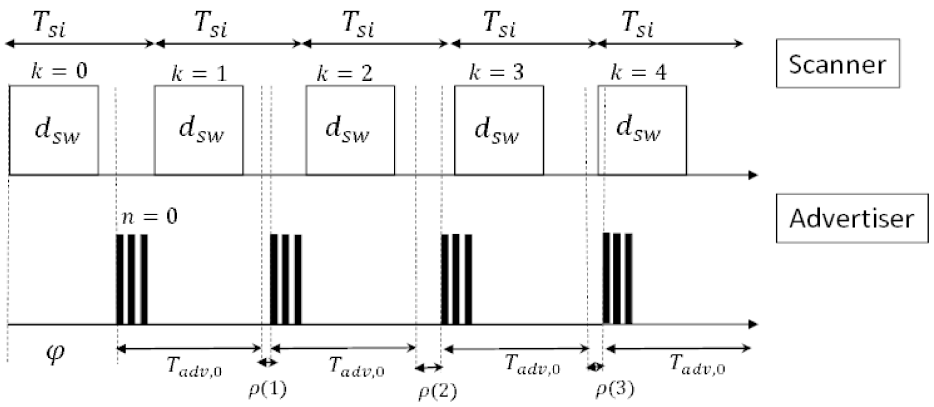


Figure 5.1: Timeline of the scanner-advertiser in the discovery procedure. The scanner duty-cycles transition from scanning phase to idle phase, listening to tag messages on the channel 37, 38 or 39. The tag (or advertiser) sends a burst of three packets on the three advertising channels in round-robin fashion. The advertising interval is composed of a static term $T_{adv,0}$ and a random additive term $\rho(n)$.

The proposed model for discovery-latency estimation is based on [51], but it differs for the ability to reduce the algorithm computational complexity, and for an extensive stochastic approach to the problem description. As explained in the Chapter 4, PRALL protocol foresees that the advertiser periodically performs a triple transmission of the advertising packet with a periodicity of T_{adv}, while, the scanner periodically listens for incoming adverts on the three channels in a round-robin fashion for a bounded time-window, denoted as scan-window d_{sw}. The scan period, named scan-interval and expressed as T_{si}, is irrespective of T_{adv}, due to the asymmetric paradigm adopted by PRALL protocol. Below, for the sake of clarity, we propose the Equations 5.1, 5.2, and 5.3, and the Figure 5.1, once again.

$$T_{adv}(n) = T_{adv,0} + \rho(n) \tag{5.1}$$

$$t_{a,n}(n, \phi) = \phi + n \cdot T_{adv,0} + \sum_{i=1}^{n} \rho(i) \tag{5.2}$$

$$k \cdot T_{si} - d_{early}(ch) \leq t_{a,n}(n, \phi) \leq k \cdot T_{si} + d_{sw} - d_{late}(ch) \tag{5.3}$$

where ch represents the listening channel of the k-th scan-window, and is computed step by step as $ch(k) = mod(k, 3) + 37$. Setting out T_{si} and d_{sw} for the scanner and $T_{adv,0}$ for the tag, Algorithm 2 estimates the expected discovery-latency, referred to as d_{exp}, by finding the couple of values (n, k) which verifies with high probability the condition expressed by Equation (4.3).

The durations $d_{early}(ch)$ and $d_{late}(ch)$ represent the time before the beginning (the first) and the end (the second) of the k-th scan-event that allows the n-th advertising-event to fall within the k-th scan-window; they are function of ch and, consequently, depend on k as shown in Table 5.1. It follows that the effective scan-window d'_{sw} is shorter than d_{sw}. More in detail, its value is $(d_{sw} - d_a)$, where d_a denotes the time for sending an advertising packet on a single channel and listening to a possible response. The protocol foresees that this time must be up to 10 ms, but usually, the value is much lower, almost 1 ms for passive scanning. In the case of active scanning this value can be larger, but always limited to 10 ms. At each n-th advertising-event the tag randomly chooses $\rho(n)$ through a pseudo random-number generator (RNG). In terms of probability the random variable ρ can be modeled through the uniform distribution with boundaries 0 and 10 ms. The probability density function (pdf) $f(t_{a,n})$ is computed as in Equation (5.2) taking into account that:

(i) ϕ, i.e., the offset between the first scan-event and the first advertising-event anchor points, is modeled through the uniform distribution with boundaries 0 and $\phi_{max} = 3T_{si}$,

(ii) $(nT_{adv,0})$ shifts $f(t_{a,n})$,

(iii) $\sum_{i=1}^{n} \rho(i)$ is the sum of n independent random variables ρ.

Table 5.1: Effective scan window parameters. The duration $d_{early}(ch)$ and $d_{late}(ch)$ are functions of ch and represent the time before the beginning (the first) and the end (the second) of the scan event, that allows the advertising event to fall within the scan window. d'_{sw} represents the effective scan window, which is shorter than d_{sw}.

Channel	d_{early}	d_{late}	d'_{sw}
37	0	d_a	$d_{sw} + d_a$
38	$d_a + d_{ch}$	$2d_a + d_{ch}$	$d_{sw} + d_a$
39	$2d_a + 2d_{ch}$	$3d_a + 2d_{ch}$	$d_{sw} + d_a$

The pdf of the term (iii) is the convolution of n uniform distributions. For $n = 1$ the pdf is a uniform distribution with boundaries 0 and 10 ms. For $n = 2$ the pdf is a triangular distribution. In contrast, for $n > 2$ the Central Limit Theorem (CLT) can be applied with sufficient approximation, describing the pdf of (iii) as a Gaussian distribution. The latter, shifted by (ii), takes $\mu_g = n{\times}5$ ms as mean and $\sigma_g = \sqrt{\frac{n}{12}}{\times}10$ ms as standard deviation. Finally, the pdf of $t_{a,n}$ is produced by the convolution between the pdf of (i)+(ii) and the pdf of the random variable ϕ by (iii). Algorithm 2 for each n-th advertising-event calculates the expected value μ_f and the half-width hw_f of the distribution function $f(t_{a,n})$, that is the half time interval width, which at least 99% of $f(t_{a,n})$ falls within. These are used to evaluate k_{min} and k_{max}, which are the lowest and highest indices of the scan-events, that the n-th advertising-event probabilistically overlaps with. The for-loop at each step, from k_{min} to k_{max}, updates the channel ch, and consequently the time $d_{early}(ch)$, $d_{late}(ch)$ of the k-th scan-event and the current duration of the advertising event, denoted by $d_{a,evt}(ch)$ and calculated as in Table 5.2. Then, the effective start and end time of the k-th scan-event, $t_{k,s}$ and $t_{k,e}$ respectively are evaluated.

Given those values, the algorithm calculates p_k, i.e., the probability for the n-th advertising-event being received successfully by the scanner, according to Equation (5.6), as the difference between the cumulative functions in Equation (5.4) and Equation (5.5).

$$F_{f(t_{a,n})}(t_{k,s}) = P(f(t_{a,n}) \leq t_{k,s}) \tag{5.4}$$

$$F_{f(t_{a,n})}(t_{k,e}) = P(f(t_{a,n}) \leq t_{k,e}) \tag{5.5}$$

$$p_k = F_{f(t_{a,n})}(t_{k,e}) - F_{f(t_{a,n})}(t_{k,s}) = \int_{-\infty}^{t_{k,e}} f(t_{a,n}) - \int_{-\infty}^{t_{k,s}} f(t_{a,n}) \tag{5.6}$$

Algorithm 2 Discovery latency estimation. The function takes as input arguments the scan interval T_{si}, the scan window d_{sw} and the advertising interval $T_{adv,0}$, and gives back the expected discovery latency d_{exp}.

function $discoveryLatency(T_{si}, d_{sw}, T_{adv,0})$
  $\rho_{max} \leftarrow 10$ ms
  $\phi_{max} \leftarrow 3T_{si}$
  $\epsilon \leftarrow 0.0001$
  $n \leftarrow 0$
  $p_{cM} \leftarrow 1$
  $d_{exp} \leftarrow 0$
  while $(p_{cM}) \geq \epsilon$ **do**
    $f(t_{a,n}) \leftarrow pdf(\phi + n \cdot T_{adv,0} + \sum_{1}^{n} \rho)$
    $\mu_f \leftarrow \frac{\phi_{max}}{2} + n \cdot T_{adv,0} + \frac{\rho_{max}}{2}$
    $hw_f \leftarrow getSigma(f(t_{a,n}))$
    $k_{min} \leftarrow floor(\frac{\mu_f - hw_f}{T_{si}})$
    $k_{max} \leftarrow ceil(\frac{\mu_f + hw_f}{T_{si}})$
    $p_{hit} \leftarrow 0$
    for $k = k_{min}$ to k_{max} **do**
      $ch \leftarrow mod(k, 3) + 37$
      $(d_{early}, d_{late}) \leftarrow getInterval(ch)$
      $d_{a,evt} \leftarrow getAdvEvntDuration(ch)$
      $t_{k,s} \leftarrow k \cdot T_{si} - d_{early}$
      $t_{k,e} \leftarrow k \cdot T_{si} + d_{sw} - d_{late}$
      $p_k \leftarrow \int_{-\infty}^{t_{k,e}} f(t_{a,n}) - \int_{-\infty}^{t_{k,s}} f(t_{a,n})$
      $p_{hit} \leftarrow p_{hit} + p_k$
      $d_{exp} \leftarrow d_{exp} + p_k \cdot p_{cM} \cdot (n \cdot (T_{adv,0} + \frac{\rho_{max}}{2}) + d_{a,evt})$
    end for
    $p_{cM} \leftarrow p_{cM} + (1 - p_{hit})$
    $n \leftarrow n + 1$
  end while
  return d_{exp}
end function

For the calculation of d_{exp}, two other variables, p_{hit} and p_{cM}, are required:

(i) p_{hit} is the probability for a successful reception of the n-th advertising-

event, given that all previous events have not been received at the scanner,

(ii) p_{cM} is the cumulative missed probability that n *advertising events* do not lead to a successful reception.

Table 5.2: Current advertising event duration. It depends on *ch* and represents the current part of the k-th advertising event which the advertiser has already run.

Channel	$d_{a,evt}$
37	d_a
38	$2d_a + d_{ch}$
39	$3d_a + 2d_{ch}$

The joint probability $p_{cM} \cdot p_k$ gives the probability that the n-th advertising-event is needed to close the cumulative missed probability p_{cM} to zero. The probability p_{cM} is the stop condition of the while-loop. In fact, when p_{cM} reaches a value under the given threshold ϵ, the algorithm outputs the final estimation of d_{exp}. The estimation accuracy depends on ϵ: the smaller ϵ, the better the accuracy, but the higher the computational cost. The value 0.0001 for ϵ allows a good accuracy.

5.1.2 Connection-oriented model

The following discussion is restricted to the case in which the central node handles up to only 2 connected devices. The connection-interval is the interval time between two consecutive *connection events* and is referred to as T_{ci}. It must be set as multiple of 1250 µs. Within the connection phase, master and slave negotiate the connection parameters:

(i) connection-interval (T_{ci}) from 7.5 ms to 4 s in steps of 1.25 ms,

(ii) *slave latency*, that is the number of connection events the slave could skip in lack of packets to send,

(iii) *supervision timeout*, maximum time the master must wait in the case of a lack of slave link before closing the connection.

Given one device, denoted by A, connected to the master, if a second slave, denoted by B, requests to establish a connection with the same master, the latter must properly space between A's and B's anchor points in order to avoid the two connections experiencing colliding transmissions. When choosing the offset δ_{AB} (shown in Figure 5.2) the central node must take into account:

(i) the slave request, i.e., a minimum and a maximum value of acceptable T_{ci},

(ii) the condition of *no-time-coincidence*, expressed by Equation (5.7).

$$gcd(T_{ci,A}, T_{ci,B}) + \tau < (2 \cdot gcd(T_{ci,A}, T_{ci,B}) - \tau) \tag{5.7}$$

The first term in Equation (5.7) is the greatest common divisor (gcd) between the connection-interval of A and B, τ is the sum of d_{ce} and d_{ch}: the former is the duration of a connection event (up to 10 ms) and the latter is the channel hop duration.

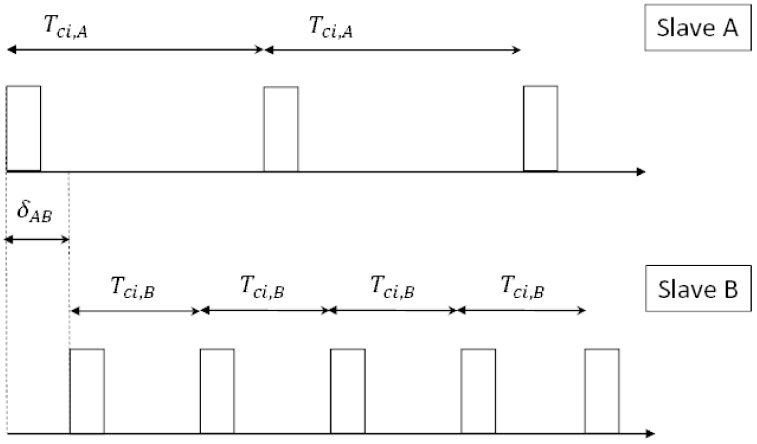


Figure 5.2: Master scheduling of slaves. The slaves A and B have connection intervals $T_{ci,A}$ and $T_{ci,B}$. In order to avoid connection events overlaps, the master must properly choose the initial time distance of B's anchor point from A's anchor point.

Whether the condition Equation (5.7) is verified, the master has just to choose δ_{AB} within the interval shown in Equation (5.8), in order to avoid overlaps.

$$m \cdot gcd(T_{ci,A}, T_{ci,B}) + \tau < \delta_{AB} < (m+1) \cdot gcd(T_{ci,A}, T_{ci,B}) \tag{5.8}$$

where m is an integer. If δ_{AB} does not fall within the interval given by Equation (5.8), overlaps will occur every $T_{ci,A(B)}/lcm(T_{ci,A}, T_{ci,B})$ transmissions of slave A(B). Else if the condition Equation (5.7) was not verified, any δ_{AB} the master would choose will result in collision. As a result, the master would not be able to handle two connections.

5.1.3 Coexistence Issues Modeling

The previous paragraph discussed the scheduling techniques the master must implement to avoid overlaps between two slaves transmissions.

Algorithm 3 Parameter designer tool. The function takes as input the connection intervals $T_{ci,A}$ and $T_{ci,B}$, the advertising interval $T_{adv,0}$, and the percentage of packets p_{succ} the master must guarantee to exchange with the slaves without overlaps, and gives back the scan-interval T_{si}, the scan-window d_{sw}, and the derived expected discovery latency d_{exp}.

function $parameterDesigner(T_{ci,A}, T_{ci,B}, T_{adv,0}, p_{succ},)$
  $d_{ce} \leftarrow 10$ ms
  $d_{ch} \leftarrow 150\mu s$
  $\tau \leftarrow d_{ce} + d_{ch}$
  if $(gcd(T_{ci,A}, T_{ci,B}) + \tau + 2 \cdot d_{ch}) < (2 \cdot gcd(T_{ci,A}, T_{ci,B}) - (tau + 2 \cdot d_{ch}))$ **then**
    $\delta_{AB} \leftarrow gcd(T_{ci,A}, T_{ci,B}) + \tau$
    $r \leftarrow lcm(T_{ci,A}, T_{ci,B})/min(T_{ci,A}, T_{ci,B})$
    $T_{si} \leftarrow min(T_{ci,A}, T_{ci,B})$
    $t_{start} \leftarrow \delta_{AB}$
    $t_{k,start} \leftarrow max(T_{ci,A}, T_{ci,B})$
    for $k = 1$ *to* r **do**
      if $t_{k,start} \leq t_{start}$ **then**
        $t_{k,start} \leftarrow \lceil t_{start}/max(T_{ci,A}, T_{ci,B}) \rceil \cdot max(T_{ci,A}, T_{ci,B})$
      end if
      if $(t_{k,start} - t_{start}) > min(T_{ci,A}, T_{ci,B})$ **then**
        $\vec{d_{sw}}(k) \leftarrow T_{SI} - (\tau + d_{ch})$;
      else
        $\vec{d_{sw}}(k) \leftarrow t_{k,start} - (t_{start} + \tau + d_{ch})$;
      end if
      $t_{start} \leftarrow t_{start} + T_{si}$
    end for
    $d_{sw} \leftarrow SelectScanWindow(\vec{d_{sw}}, p_{succ})$
    $d_{exp} \leftarrow DiscoveryLatency(T_{si}, d_{sw}, T_{adv,0})$
    return T_{si}, d_{sw}, d_{exp}
  else
    return *null*
  end if
end function

Here, we analyze the coexistence of connectionless and connection-oriented devices interfacing a single central device. In this case, the central node should avoid not only the overlap of connection-events, but also the coincidence between connection- and scan- events. In fact, if the central node scheduled a connection-event overlapping with a pre-scheduled scan-event, the connection-event would be skipped, with the resulting loss of the slave's packet. It follows that an improper choice of scan parameters could result in an undesired loss of data from connected devices. The developed Algorithm 3 takes as input the connection-intervals $T_{ci,A}$ and $T_{ci,B}$, the advertising-interval $T_{adv,0}$, and the percentage of packets p_{succ} the master must guarantee to exchange with the slaves without overlaps, and gives back the scan-interval T_{si}, the scan-window d_{sw}, and the derived expected discovery-latency d_{exp}. Firstly, the algorithm checks the condition given in Equation (5.9). This is obtained from Equation (5.7) by adding the time necessary to switch from connection-event to scan-event and viceversa ($2 \cdot d_{ch}$).

$$gcd(T_{ci,A}, T_{ci,B}) + \tau + 2 \cdot d_{ch} < 2 \cdot gcd(T_{ci,A}, T_{ci,B}) - (\tau + 2 \cdot d_{ch}) \tag{5.9}$$

If this is not verified, the algorithm exits without results, otherwise it goes to the following steps. The offset δ_{AB} must be chosen as:

$$\delta_{AB} = gcd(T_{ci,A}, T_{ci,B}) + \tau \tag{5.10}$$

Given that the A's and B's anchor points present a repetitive timeline pattern, for each $r = lcm(T_{ci,A}, T_{ci,B})/min(T_{ci,A}, T_{ci,B})$, the algorithm calculates T_{si} as the minimum value between A's and B's connection-intervals. Later, it evaluates the r scan-windows d_{sw}, including them in the vector $\vec{d_{sw}}$. The function $SelectScanWindow(\vec{d_{sw}}, p_{succ})$ finds the scan-window duration that makes the master able to successfully receive at least the percentage p_{succ} of packets exchanged with the slaves. Algorithm 3 can also be used fixing $T_{ci,A}$ and $T_{ci,B}$, varying the value of $T_{adv,0}$. The way the algorithm is used depends on which parameters can be adjusted and on network design requirements for the particular application scenario.

5.2 Discussion

This section is divided into two subsections: the former provides some results on discovery latency issue, approaching the Algorithm 2, while the latter shows how to use Algorithm 3 and understand its results, with the aim to balance and optimize the networks setting.

5.2.1 Connectionless Model Results

Algorithm 2 estimates the discovery latency of a tag in advertising phase, not taking into account the collision amongst homogeneous BLE devices, the interference to one or more channels and the reception of corrupted packets. As widely explained in Section 5.1, it uses an iterative-probabilistic model which simplifies the one presented in [51].

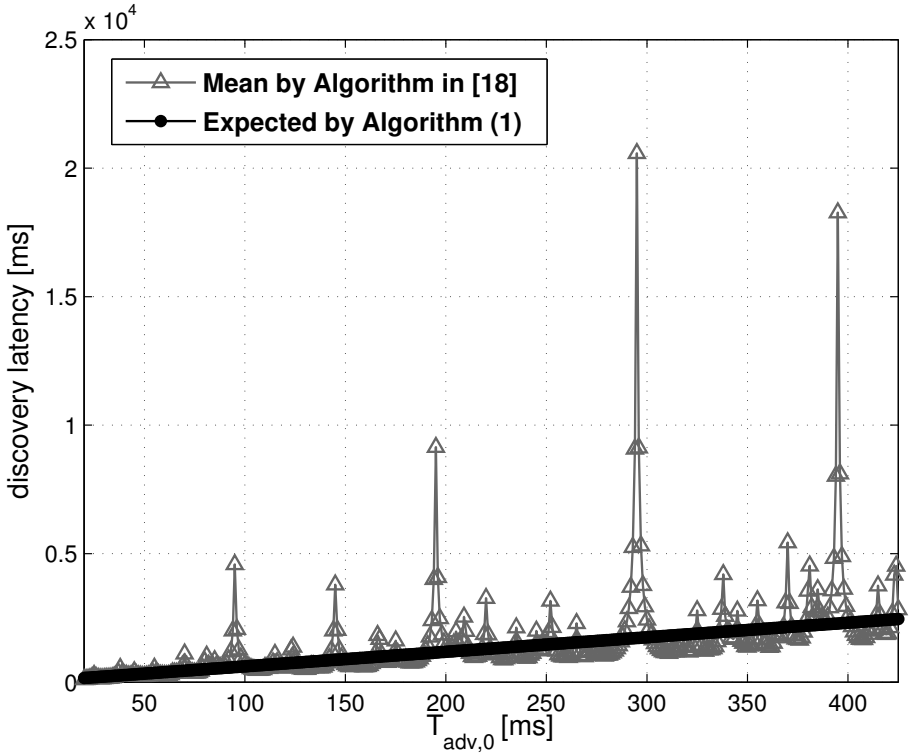


Figure 5.3: Comparison of models outcomes. The triangle-pointed line represents the mean discovery latency $\bar{d}_{mean}$ obtained by the model of Kindt et al. [51]; the solid line represents the expected discovery latency d_{exp} produced by Algorithm 2 for $T_{si} = 100$ ms, $d_{sw} = 25$ ms, $d_a = 10$ ms.

In the following, we provide an example of a possible scenario, in order to discuss Algorithm 2 results, also comparing them to those of Kindt et al. [51]. Table 5.3 summarizes the scanner and tag parameters designed if a tag self-advertises while one scanner is in listening mode. The Figure 5.3 compares the results obtained by the application of the model in [51] and Algorithm 2.

Table 5.3: Scanning and Advertising configuration used for the example scenario.

Parameter	**Value [ms]**
T_{si}	100
d_{sw}	25
$T_{adv,0}$	$[20 \div 425]$

The solid line represents the expected discovery latency d_{exp} produced by the former, while the triangle-pointed line represents the mean discovery latency, denoted by $\bar{d}_{mean}$, obtained by the model in [51]. The simulations are performed for the following configuration: $T_{si} = 100$ ms, $d_{sw} = 25$ ms, $d_a = 10$ ms.

As visible in Figure 5.3, the amplitude of $\bar{d}_{mean}$ periodically becomes very high, giving rise to noticeable peaks for $T_{adv,0}$ close to multiples of T_{si} and d_{sw}. It is important to clarify that the algorithm in [51] estimates the discovery latency averaging the delays obtained by varying the offset ϕ from 0 to $3T_{si}$. For $T_{adv,0}$ close to multiples of T_{si} or d_{sw}, the discovery latency drastically increases, but only for a few values of ϕ. The average $\bar{d}_{mean}$ is strongly affected by these few but high values. Algorithm 2 generalizes the description of the discovery latency, not using any kind of average, but directly inserting ϕ in the probabilistic model. As a result, the solid line shows a linear relationship between $T_{adv,0}$ and d_{exp} estimated by Algorithm 2. This result produces a manageable representation of d_{exp} trend that is easier to integrate in more complex models.

5.2.2 Coexistence Model Results

When two devices request to establish a connection link to a central node, the latter must assign proper connection intervals to them and space their anchor points in order to manage the data exchanges without overlaps. In fact, assuming the master was previously connected to the slave A with connection interval $T_{ci,A}$, it must properly design the configuration for B for the incoming connection. The choice is constrained by the presence of one already active connection, and must fit the condition Equation (5.7). If the central node has also to listen to advertising devices, it must properly set the connection interval of the second device B and fit the condition Equation (5.9) in order to better schedule both the slaves and tags. Fixing $T_{ci,A}$ and $T_{adv,0}$, Algorithm 3 gives back the values of $T_{ci,B}$ that reach the right balance between a limited d_{exp} and a high percentage p_{succ} of successful connection-events. The histogram in Equation (5.4) depicts the results produced from Algorithm 3, fixing $T_{ci,A}$ and

$T_{adv,0}$ equal to the typically used value 100 ms and varying $T_{ci,B}$ in the interval $[50 \div 600]$ ms with steps of 50 ms. It shows $T_{ci,B}$ and d_{exp} on the x- and y-axis, respectively. There are three bars per each couple of values $(T_{ci,B}, d_{exp})$: black for $p_{succ} = 100\%$, grey for $p_{succ} = 80\%$, white for $p_{succ} = 50\%$. Some observations can be derived:

(i) because of the condition of no-time-coincidence expressed by Equation (5.9) the master can accept only values of $T_{ci,B}$ multiple of $gcd(T_{ci,A}, T_{ci,B})$; indeed, the histogram depicts bars only for values of $T_{ci,B}$ multiple of 50 ms,

(ii) the black bar is usually higher (or equal) than the grey one, and the latter greater (or equal) than the white one; this depends on the percentage p_{succ} that must be guaranteed,

(iii) for some values of $T_{ci,B}$ the bars have similar height; in these cases the discovery latency d_{exp} is not sensitive to p_{succ}, at least above a certain percentage threshold, e.g., for $T_{ci,B} = 100$ ms the value of d_{exp} is equal to 38.47 ms independently of p_{succ} from 50% to 100%.

The Table 5.4 gives the values of T_{si} and d_{sw} that produce the results shown in Figure 5.4. Notice that when $T_{ci,B}$ and $T_{ci,A}$ are multiples, the scanning duty cycle systematically overtakes the 50%, keeping d_{exp} at low values. This fact points up that choosing the connection intervals equal or multiple to each other is a good practice to maximize the scanning duty cycle and, consequently, to limit the discovery-latency. Algorithm 3 can also be used by fixing $T_{ci,A}$ and $T_{ci,B}$ and varying $T_{adv,0}$. For example, assigning to the parameters $T_{ci,A}$ and $T_{ci,B}$ the values 100 ms and 50 ms respectively, the algorithm gives back the discovery latencies d_{exp} related to ascending values of $T_{adv,0}$ in the range $[20 \div 1000]$ ms, for given percentages of p_{succ}. For these settings the scan parameters are easily evaluated: $T_{si} = 50$ ms, $d_{sw} = 29.55$ ms for $p_{succ} \in \{100\%, 80\%\}$ and $d_{sw} = 39.70$ ms for $p_{succ} = 50\%$. The histogram in Figure 5.5 shows the advertising-interval $T_{adv,0}$ on the x-axis and the discovery-latency d_{exp} on the y-axis. There are three bars per each couple of values $(T_{adv,0}, d_{exp})$: black for $p_{succ} = 100\%$, grey for $p_{succ} = 80\%$, white for $p_{succ} = 50\%$. As it can be seen, for fixed value of scan-interval and scan-window the discovery-latency linearly rises by increasing the advertising-interval, according to the results discussed in the previous paragraph. This implies that tags which transmit very frequently are able to minimize the discovery-latency, particularly when the scan window is tiny compared to the scan-interval, i.e., for a low receiver duty-cycle.

Table 5.4: Configuration for example scenario. The Table shows the values of T_{si} and d_{sw} produced by Algorithm 3 for $T_{ci,A}$ and $T_{adv,0}$ equal to 100 ms and varying $T_{ci,B}$ in the interval $[50 \div 600]$ ms with steps of 50 ms.

Parameter	**Value [ms]**					
$T_{ci,B}$	50.00	100.00	150.00	200.00	250.00	300.00
T_{si}	50.00	100.00	100.00	100.00	100.00	100.00
$d_{sw}, p_{succ} = 100\%$	29.55	79.55	29.55	79.55	29.55	79.55
$d_{sw}, p_{succ} = 80\%$	29.55	79.55	29.55	79.55	79.55	79.55
$d_{sw}, p_{succ} = 50\%$	39.70	79.55	79.55	89.70	89.70	89.70
$T_{ci,B}$	350.00	400.00	450.00	500.00	550.00	600.00
T_{si}	100.00	100.00	100.00	100.00	100.00	100.00
$d_{sw}, p_{succ} = 100\%$	29.55	79.55	29.55	79.55	29.55	79.55
$d_{sw}, p_{succ} = 80\%$	79.55	79.55	79.55	89.70	89.70	89.70
$d_{sw}, p_{succ} = 50\%$	89.70	89.70	89.70	89.70	89.70	89.70

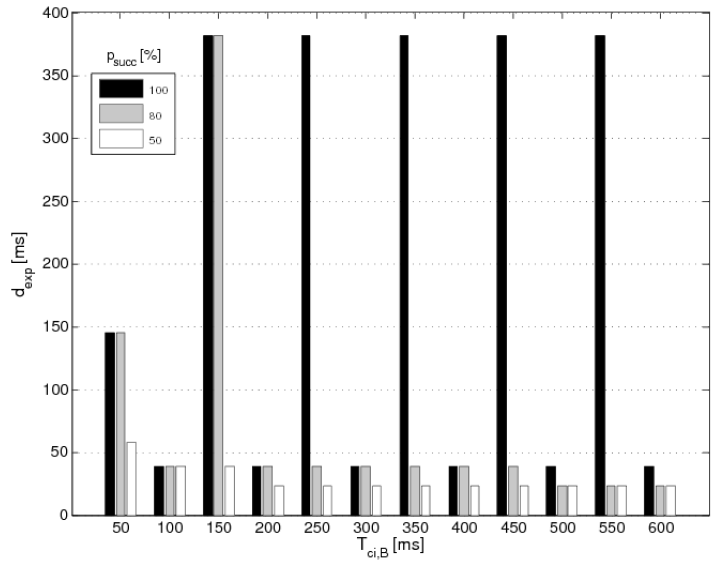


Figure 5.4: Results of the first parameters design. The histogram depicts $T_{ci,B}$ and d_{exp} on the x- and y-axis, respectively. There are three bars per each couple of values $(T_{ci,B}, d_{exp})$: black for $p_{succ} = 100\%$, grey for $p_{succ} = 80\%$, white for $p_{succ} = 50\%$.

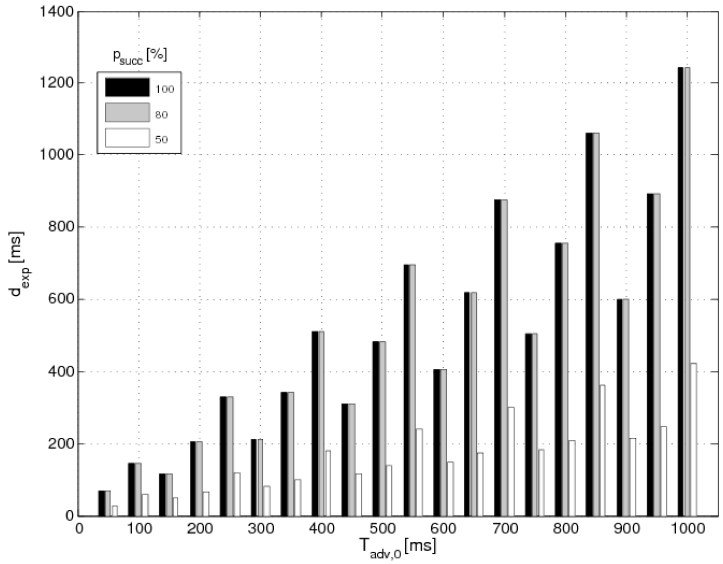


Figure 5.5: Results of the second parameters design. The histogram shows the advertising-interval $T_{adv,0}$ on the x-axis and the discovery-latency d_{exp} on the y-axis. There are three bars per each couple of values $(T_{adv,0}, d_{exp})$: black for $p_{succ} = 100\%$, grey for $p_{succ} = 80\%$, white for $p_{succ} = 50\%$.

5.2.3 Application in Home Automation and AAL Context

When using BLE in wireless home-automation and AAL applications, the number of devices becomes a limitation as BLE implements a star topology and the central node can manage only a few devices in persistent connection. Due to the TDMA technique, the number of time slots available for connected devices is further limited if the application requires short connection intervals. The connectionless topology allows us to increase the number of sensors while reducing the connected nodes. On the other hand, the central node should run heavier scheduling operations to handle both the types of communications. The presented tool is aimed at supporting the design and the configuration for these use cases. A possible scenario architecture is represented by the co-existence of the network systems presented in [17, 19] and discussed in Chapter 3 and 6 (Figure 5.6). In the former, two wearable devices (smart shoes) are connected to a central node, while the latter shows a platform for assistive home technologies based on the Message Queuing Telemetry Transport (MQTT) protocol, which also foresees a gateway, acting as central node for BLE sensors. Different types of event-driven sensors installed in the house are in charge of

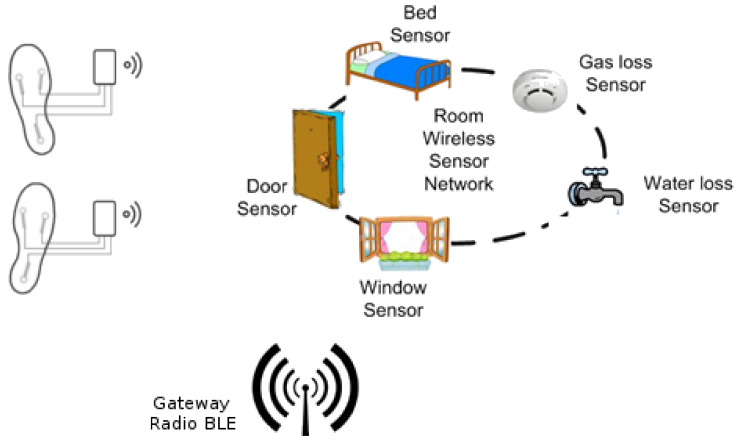


Figure 5.6: Home-automation example scenario. Two wearable devices, i.e. smart shoes, are connected to the master/gateway, while event driven sensors communicate with the master/gateway without connection.

detecting the events of interest, specifically:

- magnetic sensors placed on the entry door and on the windows, to detect opening and closing events;
- a Passive Infrared (PIR) sensor to be installed in the bathroom, to detect the user's presence;
- a bed sensor placed under the mattress;
- a water loss sensor.

These sensors do not need persistent connection. For this reason, a connectionless communication is recommended. If the two cited networks require to share the central node, properly tuning some critical BLE parameters becomes necessary. As highligthed in Chapter 3, the smart shoes need to have the connection-intervals equals to 100 ms and slave-latencies equal to 7 connection-intervals. As shown in Table 5.4, for $T_{ci,A}$ and $T_{ci,B}$ equal to 100 ms, the suggested values of scanning parameters, produced by Algorithm 3, are: $T_{si} = 100$ ms and $d_{sw} = 79.55$ ms. For these settings:

(i) the percentage p_{succ} of connection events successfully run up to all those scheduled reaches the 100%;

(ii) the *expected discovery latency* will be 38.47 ms for sensors with *advertising interval* $T_{adv,0}$ equal to 100 ms.

The tiny value of the obtained expected discovery-latency, quantitatively proves the suitability of this possible configuration and demonstrates that equal settings of slaves connection parameters maximize the time available for scanning operations. The increase in the number of the slaves makes d_{sw} collapse. In an extended scenario, considering the example of more than a pair of smart shoes, that require persistent connection with an interval of 100 ms, the master could manage at most 4 pairs of devices at the same time, with the minimum offset δ_{AB} being equal to 10.150 ms, according to the condition Equation (5.10). In this case, the central node can use a maximum scan window of 17.650 ms to perform the discovery process. For this d_{sw}, Algorithm 2 estimates a value of discovery latency equal to 1015.4 ms for $T_{adv,0} = 100$ ms, and 3860.4 ms for $T_{adv,0} = 400$ ms. The former value of expected discovery latency implies the sensor is expected to self-advertise for almost 4 s, from the event detection to the successful delivery of data. This condition would cause a noticeable delay in the sensor feedback and an increased power consumption for the transmitting node.

With respect to slaves' data reliability, the full compliance with the condition in Equation (5.10) and the adoption of the obtained scan-window are enough to guarantee that p_{succ} will reach 100%. Thus, in the case of slaves with equal connection-intervals, the connection-events will never overlap with scan-events.

Chapter 6

Telemetry Middleware in IoT Applications

The paradigmatic shift brought by the Internet of Things (IoT) has already revolutionized many key sectors, like environmental monitoring, grid and energy management, manufacturing, and it can be seen as a promising solution in the Ambient Assisted Living (AAL) domain. The increasing pervasiveness of IoT technologies relates to its ability of implementing networks of low power sensors and actuators, and, at the same time, of enabling advanced services and applications. As well known, the IoT paradigm implies access to the Internet, but not necessarily at the edges of the network. Anyway, the MQTT is widely considered an appropriate protocol, mainly beacause of its lighter weight compared to the Hyper-Text Transfer Protocol (HTTP), and its data-centric communication. Section 6.1 provides an overview of middleware solutions designed for Internet of Things (IoT) in health and wellness domains, and presents how the MQTT, can be effectively applied in assistive scenarios too, with different architectural options and communication technologies. In Section 6.2 we propose and evaluate the use of the MQTT payload structure over a LoRa physical communication link, to jointly exploit the lightweight pub-sub paradigm offered by MQTT and the long range, low power wireless capabilities offered by LoRa, to deliver building automation services.

6.1 MQTT Middleware in AAL Systems

IoT applications may be classified into two general categories: applications for ambient data collection and analytics, and real time reactive applications. The former require collecting data from a variety of sensors (e.g., sensors located in living environments, wearable devices), processing them in a subsequent time, and generating information that is used to take decisions and enable actions. The latter deal with systems operating in real time, and reacting according to the observed sensor values. The data collection and analytics applications are getting more and more popular, especially in the healthcare domain, where

personalized health tracking and monitoring has become vital to enable improved and affordable healthcare [74]. For this category, the availability of open, lightweight, and context-aware IoT middleware is essential to fulfill the expected requirements. Dementia monitoring pertains to both categories, as on one hand it requires the capability to react immediately to specific situations raising alarms (like anomalous exit from home during the night, or smoke detection), and, on the other hand, it has the major aim of collecting data about the people with dementia (PwD), to enable long term behavioral analysis for evaluating decreasing independency, and detecting worsening of cognitive impairments (loss of memory, loss of problem-solving capability, difficulties in language use).

Different approaches to the design of IoT middleware may be identified. Some proposals fall into the category of *service-based solutions*, built on the Service-Oriented Architecture (SOA) [75], that enables easy addition or deployment of different IoT devices as services. Web applications are provided to access the raw data collected by IoT devices. *Cloud-based solutions* limit the type and number of IoT devices the user can deploy, but data collection and interpretation are made easier, and use cases can be pre-programmed. Finally, other *frameworks* emphasize open, plug-and-play IoT architectures, with a variety of IoT devices that can be exposed as reusable agents, and IoT applications may be composed based on them. An example applied to health is given in [76].

Looking at the application domain each available middleware is conceived for, the majority of the solutions are general purpose, and can be applied in different systems. This is the case for OpenIoT [77], Terraswarm [78], Global Sensor Network (GSN) [79], Calvin [80], and NodeRED [81], just to mention some solutions.

The VIRTUS IoT middleware [82] provides a valid alternative to solutions based on SOA. It leverages an Instant Messaging protocol (XMPP) to guarantee a (near) real time, secure, and reliable communication channel among heterogeneous devices. The middleware has been exploited in a healthcare case study, through the implementation of a cost-savvy remote body movement monitoring system, aimed at classifying patients' daily activities, designed as a modular architecture and deployed in a large scale scenario.

EcoHealth (Ecosystem of Health Care Devices) [83], is a Web middleware platform for connecting doctors and patients using attached body sensors, thus aiming to provide improved health monitoring and diagnosis for patients. The platform is able to integrate information obtained from heterogeneous sensors in order to provide mechanisms to monitor, process, visualize, store, and send notifications regarding patients' conditions, and vital signs at real time, by using Internet standards.

A *cloud-based* middleware specifically designed for fitness, health and well-

ness applications is GoogleFit [84], an open IoT ecosystem that lets users to control their fitness data and build fitness applications, from one central location. A cloud service stores data from a variety of devices and apps: third-party sensors can be connected to this service through APIs for subscribing to a particular fitness data type or a particular fitness source, for querying of historical data or persistent recording of the sensor data from a particular source. The IoT devices available in the cloud are accessible by applications via either vendor's specific APIs or RESTful APIs. A permission and user control module handles the privacy and security of data, by prompting for user consent before GoogleFit's apps can read or store fitness data. GoogleFit provides native support for IoT devices equipped with BLE (Bluetooth Low Energy) communication interface; if a new sensor that does not communicate in BLE has to be added, it is necessary to implement the so called *FitnessSensorService* class, as well as the supported data type, if not available. GoogleFit has a narrow application scope and does not provide a framework for the composition of general IoT applications, and there are significant issues with privacy, security, and unpredictable latency.

Almost all the IoT applications developed for the wellness domain rely on the use of wearable sensors and devices, as well as health monitoring is typically performed by resorting to biomedical sensors for which an active role of the monitored user is required (e.g. in performing a blood pressure measurement). Neither the use of wearables, nor the active involvement of the patient, are acceptable and realistic assumptions when dealing with dementia monitoring, due to the cognitive impairments of the monitored user [85]. Based on these requirements, this chapter investigates the possible adoption of the light-weight MQTT in assistive technology to unobtrusively and passively monitor PwDs at home, by means of several types of sensors equipped with wired or wireless communication interfaces.

6.1.1 MQTT Middleware

The Organization for the Advancement of Structured information (OASIS) approved MQTT Version 3.1.1 as an OASIS Standard in 2014, and it has been defined as the reference standard for IoT [86]. MQTT is open, simple, and designed to be easy to implement: these features make it more suitable than other protocols, like the HTTP, for adoption on devices with limited resources, in environments such as M2M communication and IoT.

The pub/sub mechanism involves three types of components: a client that sends messages, called *publisher*, a second client which receives messages, called *subscriber*, and a *broker*, in charge of managing the communication. One of the broker functions is to filter the messages according to their *topic*. Each client

subscribes to one or more topics of interest, and receives the messages published on these topics by other clients.

MQTT is based on the Transmission Control Protocol (TCP), so clients that want to initiate a communication need to send a *CONNECT* message to the broker, which responds with a *CONNACK* message. Then, the connection is kept open and the client can publish and receive messages until it disconnects. If required, the broker can request client authentication and perform authorization.

MQTT offers three levels of Quality of Service (QoS):

- 0 - At most once. Each message is transmitted at most once, so this is the lowest QoS level and provides a best effort delivery.
- 1 - At least once. The message can be sent more than once, until the sender gets an acknowledge from the receiver.
- 2 - Exactly once. This is the safest but slowest QoS level. The message is delivered exactly once, avoiding duplications by following a four steps handshake between sender and receiver.

The client can choose both the QoS level for each message published, and the QoS level when it subscribes to a topic, depending on the requirements of its application.

MQTT is called lightweight, because the message header requires 2 bytes only, while other 2 bytes (variable header) are optionally used in some messages. The protocol foresees 14 types of messages to perform the different actions, and for the corresponding acknowledgments: connect, publish, subscribe, disconnect, ping. Some types of messages require a payload, such as the *CONNECT* and the *PUBLISH*, while messages for ping request and response just contain the header.

Over years MQTT has become quite widespread, and several open source libraries are available [87], for different devices and programming languages. A variety of brokers have been implemented too, thus making easy and immediate for developers to design their own application, and test the protocol functioning.

6.1.2 Monitoring Sensors and Home Gateway

The main components that take part in the monitoring system [88] are: a network of sensors set up in the house, a home gateway, a remote server and a user interface to show the notifications related to the events detected by the sensors. They are schematically shown in Fig. 6.1. Different types of sensors installed in the house are in charge of detecting the events of interest, specifically:

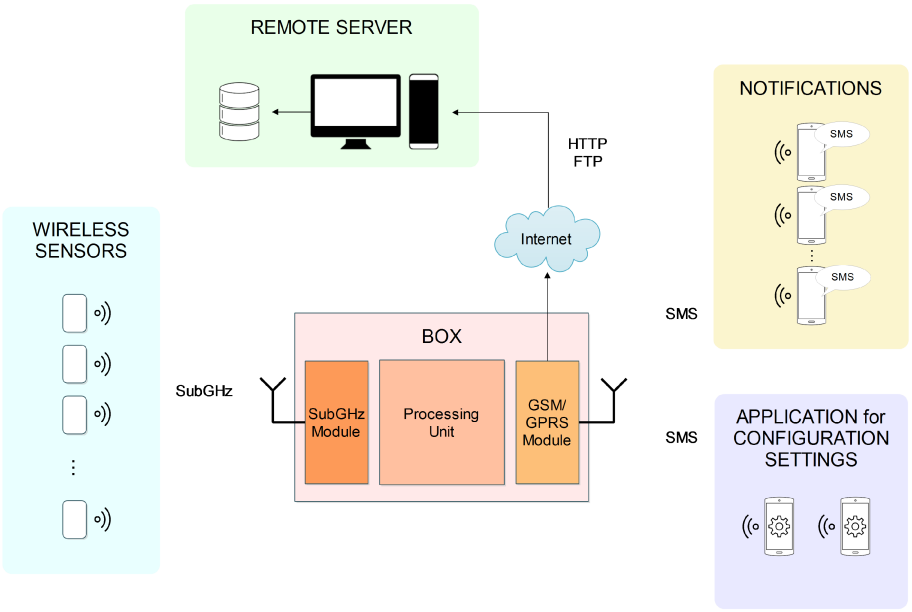


Figure 6.1: The home monitoring system components

- magnetic sensors placed on the entry door and on the windows, to detect opening and closing events;
- a Passive Infrared (PIR) sensor to be installed in the bathroom, to detect the user's presence;
- a bed sensor placed under the mattress, to detect the user going to bed or exiting from bed.

Only environmental sensors are installed, for a non-intrusive monitoring, and selected upon an assessment of caregivers' needs in assisting the PwD. Based on research outcomes about the requirements expressed by informal caregivers of people affected by dementia [89], a technological support in caregiving should help to: i) check if the monitored person leaves his/her bed during the night, and how long he/she stays out of the bed, to possibly raise an alarm; ii) check how much time the monitored person spends in the lavatory; iii) monitor the entrance door and the main windows, to check if the patient leaves the room or the house, especially during the night. Moreover, the type of sensors employed do not collect personal health data of the users and information remains anonymous. This feature represents a key point for the system acceptability. Each sensor is connected to proper electronic boards that provide the communication capability either with a gateway or with the remote server, depending

Table 6.1: Detected actions following the "exit from bed" event in the night (09:00 PM to 06:00 AM).

Detected action	Day 1	Day 2
Presence in the bathroom	76%	15%
Door opening	14%	2%
Window opening	0%	0.2%
None	10%	82.8%

on the configuration adopted, among those proposed in this section. Table 6.1 gives an example of the percent occurrence rates of different activities inferred by processing the events detected by the selected sensors, and limited to the night hours, over two different days.

The role of the gateway is to enable the communication between systems or devices using different technologies. It is introduced in the architecture to connect the sensors with the remote server, in case they cannot communicate directly. Regardless the presence of the gateway, the remote server (physically represented by a desktop computer) is always needed, however, its functionalities change depending on the architecture. In fact, in case of a direct connection of the sensors, it only represents a central node that enables the messages coming from the sensors to reach the user interface. Otherwise, it processes the incoming information, creates a text message in a human-readable format, and forwards it to the user interface.

In a monitoring scenario, it is very important to provide feedbacks to the caregiver, by notifying the occurred events. Thanks to its portability and the many onboard communication interfaces, the smartphone is the ideal candidate to serve as the caregiver's interface. More precisely, an Android application implementing the MQTT protocol has been developed. It exploits both the Wi-Fi and the WAN data connection to communicate with the server, which provides the information. If the user (i.e. the caregiver) wants to receive notifications, he has to register for the service simply by sending a request. At this point, whenever a sensor detects an event, the server will send a message to the smartphone with details about the event occurred. The latter will display them as a notification, together with a sound signal. When the smartphone is offline, all the messages can be stored by the server and then sent again when the smartphone becomes connected again. This way, no notifications will be lost. The MQTT protocol intrinsic characteristics make it particularly suitable to differentiate the messages according to the topic they belong to. This allows, for example, to variate the notification sound or the smartphone vibration according to the severity of the event that has occurred, and, thus, to manage different alarm levels.

6.1.3 Mapping MQTT to different architectures

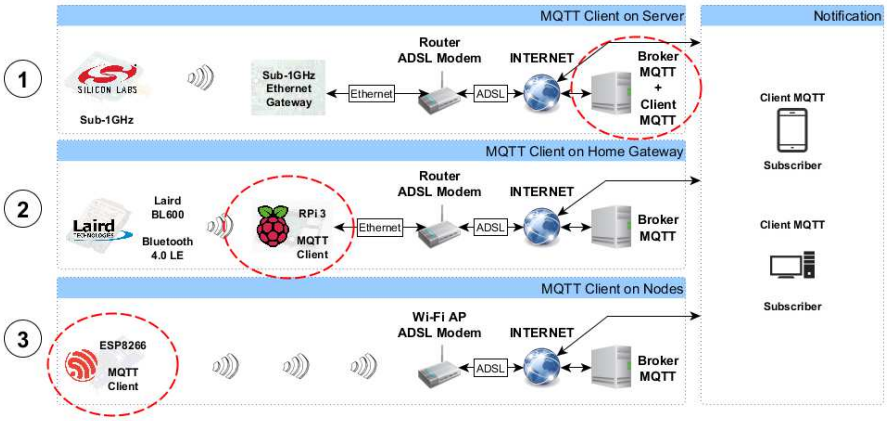


Figure 6.2: Representation of the different architectures and MQTT integration levels

In this section three alternatives are proposed, in which MQTT is introduced at different levels of the communication architecture, with increasing penetration into the home user, from top to bottom:

1. at the server;
2. at the home gateway;
3. at the sensor nodes.

To ease understanding, the different alternatives are represented in Fig. 6.2. For each configuration, the location of the MQTT publisher clients is highlighted in circles. The subscriber clients can be implemented on several devices, even simultaneously, such as smartphones and other mobile devices or computers.

In case 1, MQTT is adopted only for events notification. This is the condition in which MQTT is added to an already existing platform, to facilitate notification delivery, but it is not possible to integrate MQTT natively in the sensor nodes firmware. Each sensor is connected to an electronic board based on a Silicon Lab Si1000 family chip [90], that provides a subGHz communication link to the gateway, at a frequency of 868 MHz. When an event occurs, the electronic board receives an interrupt that wakes up the device from deep-sleep state, and implements the transmission. This wireless communication is connectionless, which exposes to collision problems; however, it is not a relevant factor in our case study. On the other side, the gateway is connected to

the home ADSL modem, so it can forward the acquired data to the remote server through the Internet. The server acts as an MQTT client that publishes the messages on a defined topic. The server also hosts the MQTT broker that manages all the messages. In order to deliver the notifications to the caregiver's smartphone, the caregiver has to specify the address of the broker, and subscribe to the correct topic, using the proper functionalities supported by the mobile application.

Case 2 is an intermediate integration level: the MQTT client resides in the home gateway, implemented on a single-board computer, the Raspberry Pi (RPi) equipped with both BLE and Ethernet interfaces. The BLE technology is used to receive messages from the sensors, whereas the Ethernet interface is connected to the home ADSL modem to reach the remote server.

The BLE node sensor, based on LAIRD BL600 [69] board, implements an event-driven data exchange. The sensor node, which is normally in a state of minimum power consumption, wakes up when a sensor interrupt happens, and begins the BLE advertising phase to indicate its capability to establish a connection to the master. The latter (i.e., the RPi) is continuously in devices scanning mode, and, once received the advertising packet from the sensor, sends back a connection request. If the sensor node receives the request, the connection is established and the node assumes the role of slave. The slave uses the indication operation as the mechanism to transmit data to the master. The indication is an unsolicited message, which requires confirming response. The indication/confirmation operation follows a stop-and-wait scheme at application level (ATTribute level). When the sensor node receives the confirming packet, it sends the disconnection message and goes back in a low power consumption state, until the next sensor activation. The master translates the indicated data into an MQTT Message and publishes it on a proper topic. The power consumption of the sensor node depends on the used BLE node technology, the proper choice of protocol parameters, as the Advertising Interval and Connection Interval durations [49], and the time spent by the slave in advertising phase until the connection establishment.

Case 3 represents the deepest level of integration of MQTT into the monitoring platform, directly on the sensor nodes. The ESP8266-12 ([91]) boards provide communication capabilities to the sensors, through a built-in Wi-Fi interface, and they implement an MQTT client. Thus, by means of a WiFi home Access Point (AP), sensors are able to connect directly to the broker and publish messages whenever an event is detected. Once detected a sensor activation, the node, that normally stays in a low power consumption mode, begins the Wi-Fi connection phase. When the connection is established, the node is capable to reach the remote MQTT broker: it starts the MQTT connection, assuming the role of MQTT Client, and once connected, it publishes a

Table 6.2: Power consumption of sensor nodes in the different use cases.

	$I_{avg}[mA]$	$\Delta t_{avg}[s]$
Case 1	17.295	0.080
Case 2	0.549	0.353
Case 3	81.940	7.526

message with QoS 1 on the proper topic. Level 1 of QoS foresees the exchange of a pair of messages at application level: the published data packet and the confirming packet. The correct reception of the confirmation packet guarantees the data has been published on the topic; the node may go back in a state of minimum power consumption, after the sequential closing of the MQTT and Wi-Fi connections.

Power consumption is mostly affected by the technology used, and the operations performed in Wi-Fi and MQTT connection phases. The chosen QoS for the publishing operation, or the payload size of published message, basically do not affect the average absorbed current. Furthermore, the re-establishing of Wi-Fi and MQTT connection for each sensor event, increases the latency between the event occurrence and its availability on the broker. This penalizes both the communication delay and the power consumption, in the case of frequent sensor events.

Table 6.2 summarizes the power consumption measurements on the sensor nodes in all the three use cases. The absorbed average current I_{avg} has been measured on the time interval, Δt_{avg}, needed to take the sensor event and transmit the data, and, then, going back to a state of minimum power consumption. The performed analysis has the only objective to explore different applicative solutions. In fact, in the current configurations, the different sensor nodes have different transmissive technologies and rules: this prevents a significant comparison in terms of power efficiency.

The Organization for the Advancement of Structured information (OASIS) approved MQTT Version 3.1.1 as an OASIS Standard in 2014, and it has been defined as the reference standard for IoT [86]. MQTT is open, simple, and designed to be easy to implement: these features make it more suitable than other protocols, like HTTP, for adoption on devices with limited resources, in environments such as M2M communication and IoT.

The pub/sub mechanism involves three types of components: a client that sends messages, called *publisher*, a second client which receives messages, called *subscriber*, and a *broker*, in charge of managing the communication. One of the broker functions is to filter the messages according to their *topic*. Each client subscribes to one or more topics of interest, and receives the messages published on these topics by other clients.

MQTT is based on the Transmission Control Protocol (TCP), so clients that want to initiate a communication need to send a *CONNECT* message to the broker, which responds with a *CONNACK* message. Then, the connection is kept open and the client can publish and receive messages until it disconnects. If required, the broker can request client authentication and perform authorization.

MQTT offers three levels of Quality of Service (QoS):

- 0 - At most once. Each message is transmitted at most once, so this is the lowest QoS level and provides a best effort delivery.
- 1 - At least once. The message can be sent more than once, until the sender gets an acknowledge from the receiver.
- 2 - Exactly once. This is the safest but slowest QoS level. The message is delivered exactly once, avoiding duplications by following a four steps handshake between sender and receiver.

The client can choose both the QoS level for each message published, and the QoS level when it subscribes to a topic, depending on the requirements of its application.

MQTT is called lightweight, because the message header requires 2 bytes only, while other 2 bytes (variable header) are optionally used in some messages. The protocol foresees 14 types of messages to perform the different actions, and for the corresponding acknowledgments: connect, publish, subscribe, disconnect, ping. Some types of messages require a payload, such as the *CONNECT* and the *PUBLISH*, while messages for ping request and response just contain the header.

Over years MQTT has become quite widespread, and several open source libraries are available [87], for different devices and programming languages. A variety of brokers have been implemented too, thus making easy and immediate for developers to design their own application, and test the protocol functioning.

6.2 MQTT and LoRa in building automation architecture

Fig. 6.3 shows the system architecture implemented within our project. From left to right: the remote node (RN), i.e. the sensor or actuator, equipped with LoRa transceiver (a Semtech SX1276 IC [92]); the concentrator/gateway (GW), with two main elements: the LoRa radio frequency (RF) module [92] (named local node - LN), and a Raspberry Pi3 acting as the Micro Controller

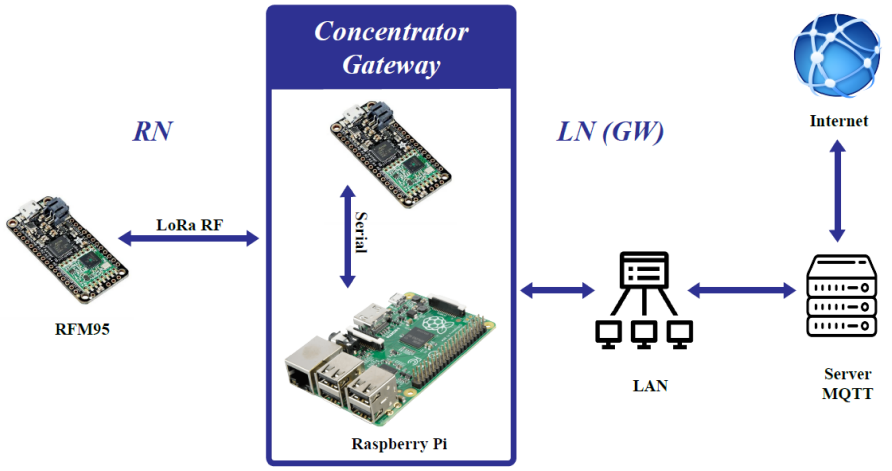


Figure 6.3: Main components of the proposed system architecture.

Unit (MCU). The concentrator connects to RN's over LoRa, and to an MQTT broker via a generic data connection (e.g. LAN-Ethernet, 4G). Finally, the backend server hosts a database and an MQTT broker. Any other MQTT client (e.g. a smartphone) may reach the broker via an internet connection. The concentrator receives frames formatted according to LoRa specifications from the RN, provides their conversion into MQTT messages, and publishes them on specific topics managed by the MQTT broker, over the available data connection.

6.2.1 The remote node

The RN is implemented on a M0 Adafruit Feather equipped with SX1276 IC [92]. The RFM95 transceiver of the radio interface uses a 868/915 MHz LoRa modem: the working frequency selected for use is 868 MHz. The maximum duty-cycle of the EU 868 ISM band is 1% and it results in a maximum transmission time of 36 sec/hour in each sub-band, for each end-device. Any sensor used for building automation purposes (e.g. light or temperature sensor), and connected to the RN, generates data carried into the payload of a LoRa frame, with a size limited to 255 bytes, more than enough for the purposes of our project. The general structure of a LoRa frame is shown in Fig. 6.4. The preamble field allows the receiver to synchronize to the incoming data. The header, in the pre-defined explicit mode, contains the packet length information (in bytes), and indicates whether a 16-bit Cyclic Redundancy Check (CRC) for the payload is present at the end of the frame. Additionally, it includes a CRC

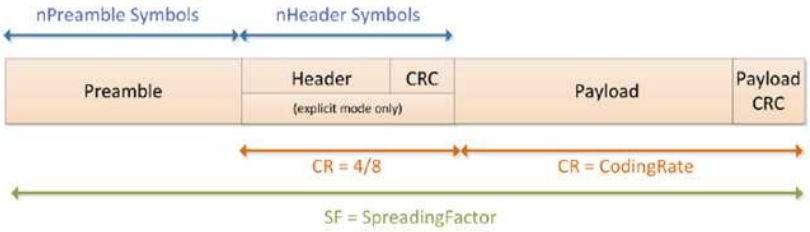


Figure 6.4: A LoRa frame typical structure.

to allow the receiver to discard packets with invalid headers.

In the specific architecture developed within the project, the LoRa frame payload formatted by the nodes has this syntax:
msgID=* * * *&value=*&nodeID=*&time=* * * * * * * * * *, where each * denotes a decimal. So, four decimals are used to encode the message identifier, one for the value, one for the node (sensor or actuator) identifier, and ten decimals for the message timestamp.

6.2.2 The concentrator/gateway

Two main components are included in the GW, i.e. the LN that provides the physical LoRa interface, and a RaspberryPi acting as the MCU. The MCU is equipped with a Linux-based Raspbian O.S.; message formatting (from LoRa on the serial port to MQTT, and viceversa) is implemented by a Python script exploiting the Paho MQTT library [93]. A background thread processes the received MQTT messages, whereas other functions in the script monitor the serial port to check for possible new messages provided by the LN. The concentrator collects LoRa frames through the LN's radio interface, and provides protocol conversion to MQTT Control Packets (CPs) [94]. The message to publish by the MQTT broker has a variable length, and it may be any type of data (binary, textual, XML or JSON). MQTT supports three levels of Quality of Service (QoS), depending on the way it manages message retransmission. In QoS level 0, MQTT operates in a "fire and forget" fashion, meaning that a best effort delivery is provided, with no guarantee about the real message reception. In QoS level 1, MQTT guarantees that a message is received at least once (but it may even be received more than once); finally, in QoS level 2 a unique reception of the message is guaranteed.

By defining the *publishers*, the *subscribers* and the *topics* managed by the MQTT broker, and setting the proper QoS level, it is possible to implement the rules governing the building automation functionalities. As an example: to switch on/off lamp A according to the light intensity value measured by sensor B, the system will request actuator A to *subscribe* to the B *light intensity*

value topic, onto which sensor B is going to *publish* its data. This way, A will receive timely notification about any value transmitted by B, and will act upon accordingly. A QoS level 2 is adequate for this type of communication, to avoid unnecessary switch on/off events due to multiple receptions of the same message, or missing actuations due to lost messages.

6.2.3 The communication protocol

The data transmission protocol implemented for the project provides a LoRa-to-MQTT and MQTT-to-LoRa format conversion, operated by the RaspberryPi used as the MCU at the GW.

Assuming for the sake of simplicity that the RN acts as a sensor, the communication from it to the GW, and back, takes place according to the following steps:

1) The sensor information is encoded according to the format described in 6.2.1 The timestamp is set to a proper value, and the message is sent to the GW as a LoRa frame;

2) Once received by the LN at the GW, the whole payload of the LoRa frame is copied to the serial port connected to the MCU (the RaspberryPi);

3) The MCU copies the LoRa frame payload into an MQTT message, changes the original timestamp value to the local value computed by the MCU, and publishes the MQTT message on the predefined topic;

4) The MQTT broker delivers the message to all the nodes subscribed to that topic, among which the GW itself is listed;

5) Being a subscriber, the RaspberryPi (i.e. the GW) receives the message, changes the timestamp value to the current local one, and sends back the message to the LN on the serial connection;

6) The LN copies the MQTT message with the new timestamp into a LoRa frame payload and sends it back to the RN;

7) Finally, the RN extracts the received LoRa message ID and timestamp. They are compared to the original ones (step 1): if the message ID is the same but the timestamp values are different, this is taken as a proof that the original message generated at step 1 has been delivered by the MQTT broker.

According to the flow of messages described by the aforementioned steps, the communication protocol implemented is a bidirectional one. It enables some kind of acknowledgment about the message delivery, but it is not able to prove

the success of communication. In fact, what the LoRa RN gets back from the GW is a proof that the MQTT broker has delivered its original message to the clients subscribed to that specific topic. However, this does not ensure that the target client has received the message: it could be temporarily switched off, and no feedback about this condition is available for the RN originating the message, nor for the MQTT broker, or for the GW. It is left to upper layer-2 protocols the execution of procedures able to guarantee the message delivery to the destination node.

A similar flow of steps, as the one described above, happens when an MQTT client (like a software app running on a smartphone to let the user control the home or building automation facilities) needs to send a message to a LoRa RN, for example an actuator:

1) The MQTT client publishes a message on a specific topic;

2) The MQTT broker delivers the message to all the clients subscribed to that specific topic, among which the GW is included;

3) The GW MCU extracts the message ID and stores it locally; the message timestamp is updated to the actual value set by the GW, and the new message is copied onto the serial port connecting the MCU to the LoRa LN;

4) The LoRa LN copies the message from the serial port into a LoRa message and transmits it to the RN over a LoRa link;

5) The RN compares the node ID value of the message to its own ID. If the message is accepted, its ID value is compared to the last stored one, for freshness verification. The payload is copied to the local memory, then the message is re-transmitted, as is, back to the LoRa GW;

6) At the GW, the message goes through all the steps described previously;

7) The message, published onto a topic the GW is subscribed to, gets back to the GW. Here, after checking the message ID is the same already stored within the GW at step 3), the message is deleted to avoid further loops.

Steps 6) and 7) in the aforementioned flow ensure the possibility to check that the message originated from an MQTT client has reached the destination node (actuator) through LoRa.

When addressing the selection of a wireless communication technology for applications related to building and home automation, it is important to take into account aspects related to security, in order to avoid the risk the system gets controlled by unauthorized entities. According to the specifications, LoRa uses two AES (Advanced Encryption Standard) security keys, as shown in Fig.

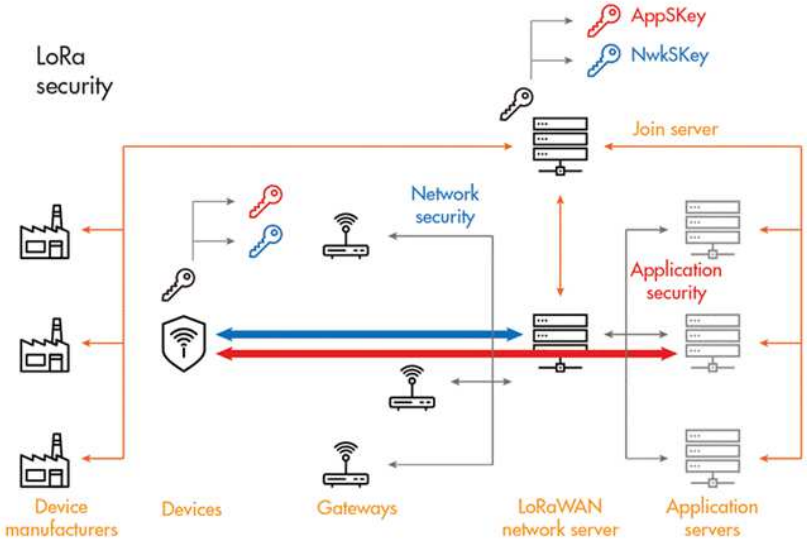


Figure 6.5: Key management in LoRa [95].

6.5, that are unique to each LoRa device: the *NwkSKey*, and the *AppSkey*. Both keys have a length of 128 bits. The network session key (*NwkSKey*) is used for interaction between the node and the network to check the validity of messages (MIC - Message Integrity Check). The application session key (*AppSKey*) is used for encryption and decryption of the payload. There are two ways for LoRa devices to join the network: Activation by Personalization (ABP) and Over-the-Air Activation (OTAA). In the ABP case, the session keys are stored in the LoRa nodes. Dynamically activated devices (OTAA) use a third 128 bit AES application key (named *AppKey*) to derive the two session keys during the activation procedure. The choice of one of these two options is tipically based on implementation constraints.

6.2.4 Radio coverage evaluation

As a first experimental evaluation, we conducted several measurement campaigns to collect data about the indoor radio coverage provided by LoRa inside a huge facility (in our University campus), to possibly replicate a realistic building automation scenario. Fig. 6.6 shows the location of the LoRa GW (inside the Telecommunications Lab) and the selected Points of Measurement (PoM). Received Signal Strength Indicator (RSSI) values were collected in 7 different PoMs on the same floor, by transmitting LoRa packets of 20 bytes payload, with a Spreading Factor (SF) equal to 7. At each position, the transmitter was switched on for a total time of 30 minutes. The LoRa enabled RN

was programmed to transmit around 4 packets per minute, in order to collect 120 packets per PoM. In reality, due to environmental conditions affecting the transmissions, the effective number of packets received at the GW from each PoM was smaller, as detailed in Table 6.3.

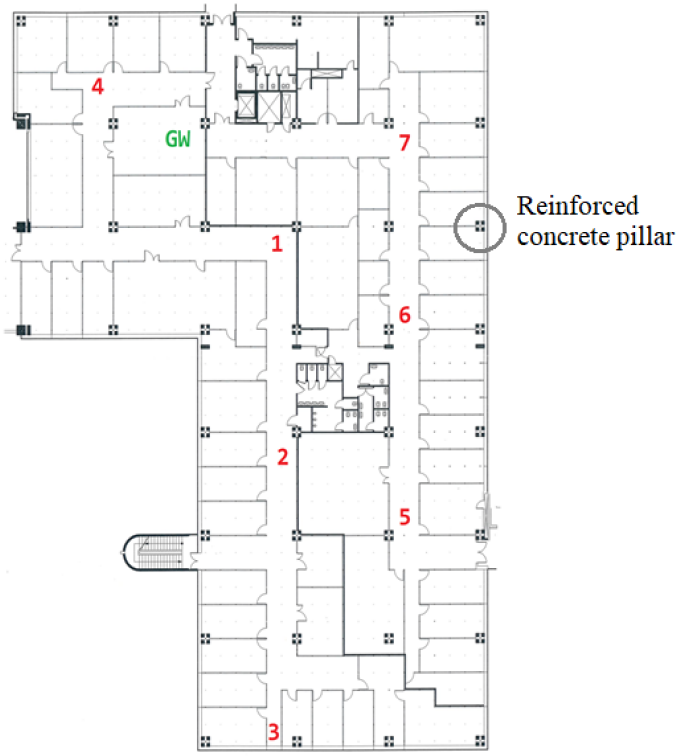


Figure 6.6: PoMs and GW location for LoRa RSSI measurements on the same floor.

Fig. 6.7 shows the distribution of the number of packets received for each RSSI value at each PoM, during a 30-minute-long measurement campaign. The privileged position of PoM 4 is quite evident from the graph, as it provides the packets with the highest RSSI values over all the positions tested. PoM 7 and PoM 1 are almost equivalent from the point of view of the distribution of received packets with similar RSSI values, whereas PoM 3 results more favorable than PoM 5, despite the greater distance from the GW. We can explain this behavior by looking at Fig. 6.6: even if PoM 3 is farther than PoM 5 from the GW, and the number of walls crossed is higher, however in order to reach PoM 5 the LoRa signal interacts with reinforced concrete pillars and gets more attenuated.

Distances between each PoM and the GW are reported in Table 6.4, that also

Table 6.3: Amount of packets received at each PoM over 30 minutes transmission

PoM	no. of packets received
1	109
2	111
3	111
4	110
5	112
6	101
7	110

Table 6.4: Average RSSI values in indoor LoRa propagation (SF = 7)

PoM	Avg RSSI (dB_m)	distance (m)	no. of walls
1	-61	13	3
2	-79	29	7
3	-83	53	13
4	-42	9	1
5	-95	40	11
6	-75	27	6
7	-60	22	1

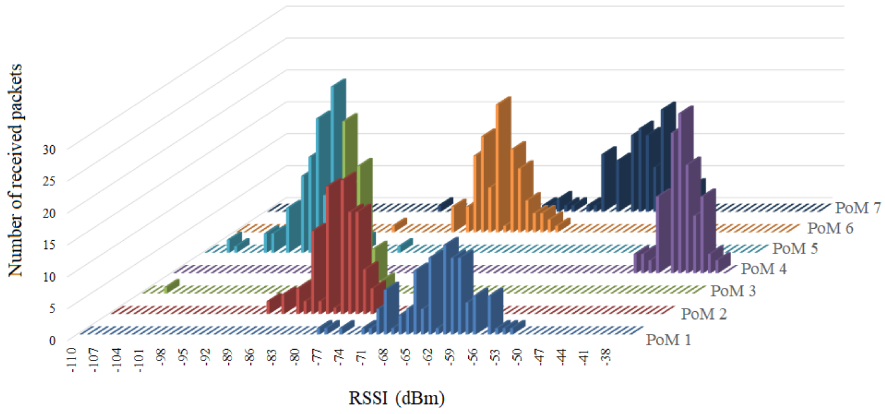


Figure 6.7: Distribution of the number of packets received for each RSSI value at each PoM.

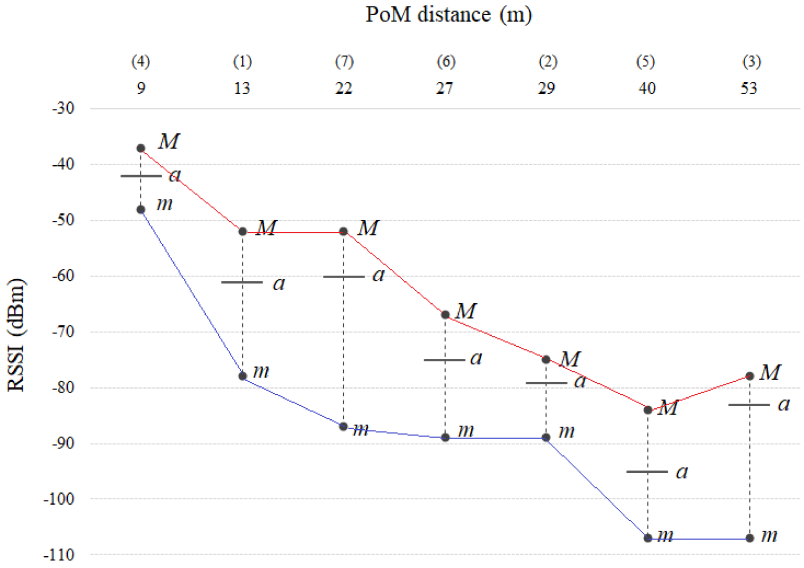


Figure 6.8: Minimum (m), maximum (M), and average (a) RSSI value at each PoM, as a function of its distance from the GW. The PoM identifiers are given in brackets.

summarizes the average RSSI values obtained at each position, depending on the different distances and the number of walls the LoRa signal went through. It is possible to verify that the shortest distance (9 m) joint with the smallest number of walls (1) provides the highest RSSI (-42 dB_m). The variation of the minimum, maximum, and average RSSI at each PoM (according to the RSSI distributions presented in Fig. 6.7), with respect to the distance from the GW, is reported in Fig. 6.8.

6.2.5 System performance

First of all, we can state that the physical layer technology chosen for the proposed architecture, namely LoRa, is adequate to provide the necessary signal strength for the indoor enviroment considered. In fact, as shown in Fig. 6.8, in any case, the RSSI is far above the receiver sensitivity of -142 dB_m, so the LoRa radio coverage on the floor is adequate to support the transmission of data and commands related to the building automation system. In [96] a similar scenario is proposed with the use of Zigbee technology. Due to the short coverage range, the system architecture is more complex because in order to achieve the same coverage more nodes are needed. In fact for covering two floors, a network consisting of 16 routers is needed. By using Wi-Fi technology, as described in [97],

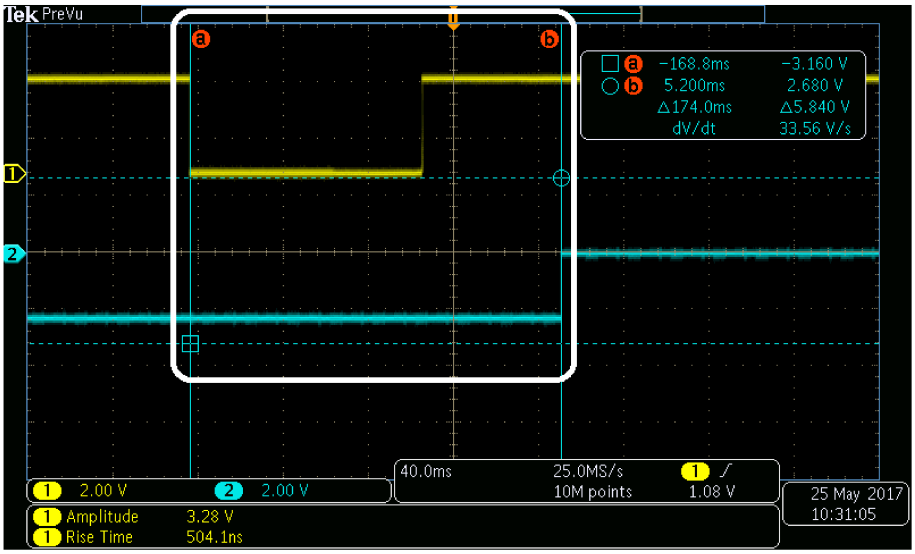


Figure 6.9: Oscilloscope screenshot showing a delay of 174 ms between command transmission and execution.

the ESP8266 Wi-Fi module can draw spikes of 250 mA in transmission, while the RFM95 LoRa module used for this project reaches 120 mA at the maximum, during a +20 dB_m transmission. This means a shorter battery lifetime for Wi-Fi based networks, so higher maintenance costs. We can show another important result of the proposed solution, i.e. the time requested by a message to cause a state change in the actuator node is compatible with the delay requirements posed by real-time home and building automation applications, i.e. 200 ms for delay-sensitive use cases [98]. Fig. 6.9 shows a screenshot of the oscilloscope displaying a delay of 174 ms between a button-pressure event and light activation. In [99] it is shown that by using the ZigBee technology, the latency for one hop is about 7.35 ms, so for the building automation scenario this leads to a total latency of 170 ms. Such a technology provides low latency but requires networks with more elements and hence higher costs and even more complex procedures to add or remove nodes to/from the network itself.

Chapter 7

Conclusions

In the introductory Chapters 1 and 2 we have given a general overview of the several kinds of Wireless Sensor Networks (WSN), limiting the dissertation on the "last 100-meters networks", i.e. BAN, PAN and LAN, with a short introduction on Low Power Wide Area Networks (LPWANs) and LoRa (Long Range) technology. In particular the energy efficiency issue has been addressed, relating it to the management of the sensors access medium (MAC), in terms of duty-cycled radio activation and enabled communication paradigm, i.e. connectionless or connection-oriented. Among the several technology solutions adopted for short-range applications, the Bluetooth Low Energy (BLE) deserved closer attention, due to its low-cost and low-energy features, as well as the widespread integration in smartphones and many other mobile devices, that provide an easy tunnel to Internet. Such strength points, coupled with the integration of 6LowPAN supported from the version 4.2 of BLE, make this protocol the ideal solution for Internet of Things (IoT), defined in [100] as "a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols".

In Chapter 3, we analyzed the BLE standard and its use in a developed wearable application for Ambient Assisted Living (AAL). We introduced the whole system, consisting of a pair of smart shoes interfaced to a smartphone, acting as central node and gateway. The peculiar streaming scenario, characterized by low throughput, but strict reliability requirements, led to design a solution that provides the system with an ad-hoc in-node data-reduction and an ACK mechanism at application layer, as well as a duty-cycling policy at Link Layer, able to synchronize the application-level buffer with the transmission-level buffer, keeping the wearable nodes in shorter periodical waking-time. The use of the indication transfer mechanism and the optimization of the connection parameters (Connection Interval and Slave Latency) represented a good solution for energy consumption efficiency and drew some general guidelines for this kind of application, very common in IoT and, specifically in AAL. This work has been published in [17].

The other main topic addressed in this thesis is represented by the mod-

eling of the discovery-procedure implemented by BLE at MAC layer, here denoted as Periodic-Random-Access-to-Link-Layer (PRALL), which is mainly described in Chapter 4 and partially in Chapter 5. PRALL are generally used for neighbor-discovery, but also for connectionless data communication. The energy efficiency issue has been addressed by modeling the discovery-latency estimation, firstly with a deterministic model in Chapter 4, and secondly with a probabilistic approach in Chapter 5. The former has proved to match the simulation results for most of the parameterizations and, according to the simulator output, it has highlighted the drawbacks of PRALL protocol for some scanner-advertiser configurations. Starting from this evidence, a modified version of PRALL (mPRALL) has been proposed, and its performace in terms of latency has been evaluated, performing simulations for several parameterizations. The obtained performance-indexes have shown the ability of MPRALL to outperform PRALL, in scenarios where mPRALL can be used (full knowledge of network configuration for each node).

In Chapter 5, we proposed a tool aimed to support the design and the configuration of BLE devices, allowing a single central node to manage both the connected and the asynchronous sensors and/or actuators. The medium access management of the central node has been modeled, focusing on the MAC scheduling procedure in both connectionless and connection-oriented communication. The developed models were then merged into a single tool. The results highlighted the suitability of the proposed tool for a proper design of the device parameters, maintaining, for example, useful timing for discovery process of event-driven sensors and avoiding undesired overlaps between advertising events and connection events, due to inaccurate scanning and connection parameters configuration.

In Chapter 6, we have addressed the data-centric communication paradigm exploited by one of the most promising light-weight alternative to HTTP for IoT, i.e. Message Queue Telemetry Transport (MQTT), to investigate the suitability of this paradigm, firstly, in an AAL scenario and, secondly, in building automation context. In the first work the specific use case of assistive technology for the home monitoring of patients with dementia has been investigated, proposing three possible architectures featuring different degrees of integration of the MQTT protocol into the technology platform, based on different communication technologies. The results obtained, and the power consumption estimation provided for each case, show that MQTT may be effectively adopted for a quick and reliable distribution of notification messages among the different actors involved by the platform. In the second work, the MQTT data-structure has been adopted over LoRa physical communication link to support real-time building automation services. Extensive experimental measurement campaigns proved LoRa is well suited to ensure adequate radio coverage in indoor sce-

narios, even in big and bulky buildings, resulting in a received signal strength well over the receiver sensitivity. An ad-hoc communication protocol to ensure a basic acknowledgment mechanism to the messages exchanged among nodes of the building automation system, and between them and any third-party MQTT client, has been designed and its performance tested, thus verifying the real-time requirements of the building automation service are fulfilled.

In conclusion, when looking at the state of the art technologies, they should give a clear indication of the proliferation of architectures, schemes, protocols and services in WNS, and, generally in IoT. This will inevitably lead to a fragmentation of the IoT. Hence, the convergence on standardized protocols as BLE and LoRa for short- and long-range communication, respectively, or MQTT for the access to Internet could be the key to roll out the IoT tangle. But, on the other side, the research in telecommunications cannot stop on the strong points of this protocols, but has to improve them starting from their drawbacks and shortcomings, expecially in energy-efficiency, self-organization capability and data-reliability for power-constrained sensors.

Acronyms

AAL Ambient Assisted Living

ABP Activation by Personalization

ADC Analog to Digital Converter

ADSL Asymmetric Digital Subscriber Line

AES Advanced Encryption Standard

ANT/ANT+ ANT ultra-low-power technology

API Application Programming Interface

AR Average Ratio

ARM Advanced RISC Machine

ATT Attribute Protocol

BAN Body Area Network

BLE Bluetooth Low Energy

B-MAC Beacon-MAC

BT Classic Bluetooth

BW BandWidth

CE Connection Event

CI Connection Interval

CLT Central Limit Theorem

CR Code Rate

CRC Cyclic Redundancy Check

CSS Chirp Spread Spectrum

CSV Comma Separated Values

CRT [1] Chinese Remainder Theorem [2] Confirming Round Trip

CTS Clear To Send

DB DataBase

FEC Forward Error Correction

FFT Fast Fourier Transform

FHSS Frequency Hopping Spread Spectrum

FIFO First-Input First-Output

FSR Force Sensing Resistor

GAP Generic Access Profile

GATT Generic Attributes

GFSK Gaussian Frequency-shift keying

GPIO General Purpose Input/Output

GSN Global Sensor Network

GW GateWay

HTTP Hyper-Text Transfer Protocol

I2C Inter-integrated Circuit

ICRT Indication/Confirming Round Trip

IEEE Institute of Electrical and Electronics Engineers

IoT Internet of Things

IRT Indication Round Trip

ISM Industrial, Scientific and Medical

JSON JavaScript Object Notation

JTAG Joint Test Action Group

L2CAP Logical Link Control and Adaption Protocol

LDO Low-DropOut regulator

LL Link Layer

LN Local Node

LoRa Long-Range technology

LPWAN Low Power Wide Area Network

6LowPAN Low-Power Wireless Personal Area Networks

MAC Medium Access Control

MAN Metropolitan Area Network

MCU Micro-Controller Unit

MD Maximum Deviation

MOSS Many-to-One-Sensor-to-Sink

mPRALL modified Periodic Random Access to Link Layer

MPD Maximum Positive Deviation

MQTT Message Queue Telemetry Transport

MTU Maximum Transmission Unit

NF Noise Figure

NFC Near Field Communication

NLOS Nonline-Of-Sight

OTAA Over-the-Air Activation

PAN Personal Area Network

PDF Probability Density Function

PHY Phisical Layer

PIR Passive Infra-Red

PMAC Pattern-MAC

PoM Points of Measurement

PRALL Periodic Random Access to Link Layer

PwD People with Dementia

QoS Quality of Service

REST REpresentational State Transfer

RF Radio Frequency

RI-MAC Receiver-Initiated asynchronous MAC protocol

RMAC Routing enhanced MAC protocol

RMSE Root Mean Square Error

RMSEP Root Mean Square Error of Prediction

RNG Random Number Generator

RSSI Received Signal Strength Indication

RTS Request To Send

SF Spreading Factor

SIG Special Interest Group

SL Slave Latency

S-MAC Sensor-MAC protocol

SMP Security Manager Protocol

SNR Signal Noise Ratio

SOA Service-Oriented Architecture

SPI Serial Peripheral Interface BUS

SSID Service Set IDentifier

STEM-B Sparse Topology and Energy Management-Beacon

STEM-T Sparse Topology and Energy Management-Tone

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

T-MAC Timeout-MAC protocol

UART Universal Asynchronous Receiver-Transmitter

WAN Wide Area Network

WiFi Wireless Fidelity

WSN Wireless Sensor Network

X-MAC low power MAC protocol

XML eXtensible Markup Language

XMPP Instant Messaging protocol

Bibliography

[1] D. J. Cook and S. K. Das, Eds., *Smart Environments*. John Wiley & Sons, Inc., sep 2004. [Online]. Available: https://doi.org/10.1002/047168659x

[2] *Fundamentals of Wearable Computers and Augmented Reality, Second Edition*. CRC Press, 2017. [Online]. Available: https://www.amazon.com/Fundamentals-Wearable-Computers-Augmented-Reality/dp/1138749311?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1138749311

[3] B. Adryan, P. Fremantle, and D. Obermaier, *The Technical Foundations of Iot*, ser. Artech House mobile communications library. Artech House, 2017. [Online]. Available: https://books.google.it/books?id=YkfxAQAACAAJ

[4] E. Tsimbalo, X. Fafoutis, E. Mellios, M. Haghighi, B. Tan, G. Hilton, R. Piechocki, and I. Craddock, "Mitigating packet loss in connectionless bluetooth low energy," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, dec 2015. [Online]. Available: https://doi.org/10.1109/wf-iot.2015.7389068

[5] C. Caione. (2013) Ultra-low power wsns: distributed signal processing and dynamic resource management. [Online]. Available: http://amsdottorato.unibo.it/5397/1/Caione_Carlo_tesi_V2.pdf

[6] R. Li, C. Zheng, and Y. Zhang, "Study of power-aware routing protocal in wireless sensor networks," in *2011 International Conference on Electrical and Control Engineering*. IEEE, sep 2011. [Online]. Available: https://doi.org/10.1109/iceceng.2011.6057824

[7] J. Wang, Q. Gao, H. Wang, and W. Sun, "A method to prolong the lifetime of wireless sensor network," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, sep 2009. [Online]. Available: https://doi.org/10.1109/wicom.2009.5300990

[8] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, apr 2007. [Online]. Available: https://doi.org/10.1109/mwc.2007.358967

[9] H. Patel and V. Shah, "A review on energy consumption and conservation techniques for sensor node in WSN," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*. IEEE, oct 2016. [Online]. Available: https://doi.org/10.1109/scopes.2016.7955508

[10] K. S. A. Rasbi, H. Shaker, and Z. T. Sharef, "Survey on data-centric based routing protocols for wireless sensor networks," *International Journal of Electrical, Electronics and Computers*, vol. 2, no. 2, pp. 9–16, 2017. [Online]. Available: https://doi.org/10.24001/eec.2.2.3

[11] U. D. Prasan, "Energy efficient and qos aware ant colony optimization (eq-aco) routing protocol for wireless sensor networks," *International Journal of Distributed and Parallel systems*, vol. 3, no. 1, pp. 257–268, jan 2012. [Online]. Available: https://doi.org/10.5121/ijdps.2012.3122

[12] M. A. Hail and S. Fischer, "IoT for AAL: An architecture via information-centric networking," in *2015 IEEE Globecom Workshops (GC Wkshps)*. IEEE, dec 2015. [Online]. Available: https://doi.org/10.1109/glocomw.2015.7414020

[13] S. Spinsante, V. Stara, E. Felici, L. Montanini, L. Raffaeli, L. Rossi, and E. Gambi, "Chapter 4 - the human factor in the design of successful ambient assisted living technologies," in *Ambient Assisted Living and Enhanced Living Environments*, C. Dobre, C. Mavromoustakis, N. Garcia, R. Goleva, and G. Mastorakis, Eds. Butterworth-Heinemann, 2017, pp. 61 – 89. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128051955000041

[14] E. Gambi, L. Montanini, L. Raffaeli, S. Spinsante, and L. Lambrinos, "Interoperability in iot infrastructures for enhanced living environments," in *Proceedings of the 4th International BlackSea Conference on Communications and Networking*. IEEE, 2016.

[15] A. D. Santis, E. Gambi, L. Montanini, L. Raffaeli, S. Spinsante, and G. Rascioni, "A simple object for elderly vitality monitoring: The smart insole," in *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, Sept 2014, pp. 1–6.

[16] A. D. Santis, A. Del Campo, E. Gambi, L. Montanini, G. Pelliccioni, D. Perla, and S. Spinsante, "Unobtrusive monitoring of physical activity in AAL - a simple wearable device designed for older adults," in *Proceedings of the 1st International Conference on Information and Communication Technologies for Ageing Well and e-Health*. SCITEPRESS - Science and and Technology Publications, 2015. [Online]. Available: https://doi.org/10.5220/0005497102000205

[17] A. Del Campo, L. Montanini, D. Perla, E. Gambi, and S. Spinsante, "Ble analysis and experimental evaluation in a walking monitoring device for elderly," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2016, pp. 1–6.

[18] A. Del Campo, L. Cintioni, S. Spinsante, and E. Gambi, "Analysis and tools for improved management of connectionless and connection-oriented BLE devices coexistence," *Sensors*, vol. 17, no. 4, p. 792, apr 2017. [Online]. Available: https://doi.org/10.3390/s17040792

[19] A. Del Campo, E. Gambi, L. Montanini, D. Perla, L. Raffaeli, and S. Spinsante, "MQTT in AAL systems for home monitoring of people with dementia," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, sep 2016. [Online]. Available: https://doi.org/10.1109/pimrc.2016.7794566

[20] S. Spinsante, G. Ciattaglia, A. Del Campo, D. P. Davide Perla, G. Cancellieri, and E. Gambi, "A lora enabled building automation architecture based on mqtt," in *2017 AEIT International Annual Conference (AEIT)*. IEEE, 2017 in press.

[21] P. Dutta, D. Culler, and S. Shenker, "Procrastination Might Lead to a Longer and More Useful Life." [Online]. Available: http://www-static.cc.gatech.edu/fac/Constantinos.Dovrolis/hotnets07/papers/hotnets6-final129.pdf

[22] B. Jang, J. B. Lim, and M. L. Sichitiu, "An asynchronous scheduled mac protocol for wireless sensor networks," *Computer Networks*, vol. 57, no. 1, pp. 85 – 98, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128612003246

[23] F. Alfayez, M. Hammoudeh, and A. Abuarqoub, "A survey on mac protocols for duty-cycled wireless sensor networks," *Procedia Computer Science*, vol. 73, no. Supplement C, pp. 482 – 489,

2015, international Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187705091503495X

[24] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2002, pp. 1567–1576 vol.3.

[25] S. Sung, H. Kang, E. Kim, and K. Kim, "Energy consumptlon analysls of s-mac protocol in single-hop wireless sensor networks," in *2006 Asia-Pacific Conference on Communications*, Aug 2006, pp. 1–5.

[26] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, June 2004.

[27] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 171–180. [Online]. Available: http://doi.acm.org/10.1145/958491.958512

[28] A. Akl, T. Gayraud, and P. Berthou, "A metric for evaluating density level of wireless sensor networks," in *2011 IFIP Wireless Days (WD)*, Oct 2011, pp. 1–3.

[29] F. Tong, W. Tang, R. Xie, L. Shu, and Y. C. Kim, "P-mac: A cross-layer duty cycle mac protocol towards pipelining for wireless sensor networks," in *2011 IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–5.

[30] T. Kaur and D. Kumar, "Tdma-based mac protocols for wireless sensor networks: A survey and comparative analysis," in *2016 5th International Conference on Wireless Networks and Embedded Systems (WECON)*, Oct 2016, pp. 1–6.

[31] C. Yu, R. Fiske, S. Park, and W.-T. Kim, "Many-to-one communication protocol for wireless sensor networks," *Int. J. Sen. Netw.*, vol. 12, no. 3, pp. 160–170, Nov. 2012. [Online]. Available: http://dx.doi.org/10.1504/IJSNET.2012.050454

[32] C. M. G. Algora, V. A. Reguera, N. Deligiannis, and K. Steenhaut, "Review and classification of multichannel mac protocols for low-power and lossy networks," *IEEE Access*, vol. 5, pp. 19 536–19 561, 2017.

[33] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 95–107. [Online]. Available: http://doi.acm.org/10.1145/1031495.1031508

[34] A. El-Hoiydi and J. D. Decotignie, "Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," in *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No.04TH8769)*, vol. 1, June 2004, pp. 244–251 Vol.1.

[35] J. Beaudaux, A. Gallais, J. Montavont, T. Noel, D. Roth, and E. Valentin, "Thorough empirical analysis of x-mac over a large scale internet of things testbed," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 383–392, Feb 2014.

[36] G. Vigneshwar and T. Senthil, "Life time maximization analysis with application to ll mac amp; ri-mac protocol in wireless sensor networks," in *2013 IEEE Conference on Information Communication Technologies*, April 2013, pp. 156–160.

[37] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 181–192. [Online]. Available: http://doi.acm.org/10.1145/1614320.1614341

[38] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 71–84. [Online]. Available: http://doi.acm.org/10.1145/1460412.1460420

[39] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 350–361. [Online]. Available: http://doi.acm.org/10.1145/1791212.1791253

[40] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, 2002, pp. 200–209 vol.1.

[41] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *IEEE Transactions on Computers*, vol. 59, no. 11, pp. 1562–1575, Nov 2010.

[42] M. Bakht, M. Trower, and R. Kravets, "Searchlight: Won't you be my neighbor?" 01 2012.

[43] E. B. Hamida, G. Chelius, and E. Fleury, "Revisiting neighbor discovery with interferences consideration," in *Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks*, ser. PE-WASUN '06. New York, NY, USA: ACM, 2006, pp. 74–81. [Online]. Available: http://doi.acm.org/10.1145/1163610.1163623

[44] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 137–145. [Online]. Available: http://doi.acm.org/10.1145/501431.501435

[45] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, Jan 2002.

[46] D. I. Inc., "Ant message protocol and usage," 2014, revision 5.1. [Online]. Available: www.thisisant.com

[47] ——, "Why ant+: The facts," 2015. [Online]. Available: www.thisisant.com/business/why-ant/facts

[48] ——, "Ant an11: Ant channel search and background scanning channel," 2009, revision 2.1. [Online]. Available: www.thisisant.com

[49] Bluetooth-SIG, "Specification of the bluetooth system, 4.0," vol. 6, Jun. 2010.

[50] J. Liu, C. Chen, and Y. Ma, "Modeling neighbor discovery in bluetooth low energy networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1439–1441, September 2012.

[51] P. Kindt, D. Yunge, R. Diemer, and S. Chakraborty, "Precise energy modeling for the bluetooth low energy protocol," *CoRR*, vol. abs/1403.2919, 2014. [Online]. Available: http://arxiv.org/abs/1403.2919

[52] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/9/11734

[53] K. Cho, G. Park, W. Cho, J. Seo, and K. Han, "Performance analysis of device discovery of bluetooth low energy (ble) networks," *Computer Communications*, vol. 81, no. Supplement C, pp. 72 – 85, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366415003886

[54] W. S. Jeon, M. H. Dwijaksara, and D. G. Jeong, "Performance analysis of neighbor discovery process in bluetooth low-energy networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1865–1871, Feb 2017.

[55] P. Østhus, "Concurrent operation of bluetooth low energy and ant wireless protocols with an embedded controller," 2011.

[56] 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*, vol. 23, no. 5, pp. 60–67, October 2016.

[57] Lora. [Online]. Available: https://www.lora-alliance.org/what-is-lora

[58] Sigfox. [Online]. Available: http://www.sigfox.com

[59] Weightless. [Online]. Available: http://www.weightless.org

[60] Lora: Symbol generation. [Online]. Available: http://www.sghoslya.com/p/lora-is-chirp-spread-spectrum.html

[61] D. Magrin. Network level performances of a lora system. [Online]. Available: http://tesi.cab.unipd.it/53740/1/dissertation.pdf

[62] M. Bor and U. Roedig, "Lora transmission parameter selection," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2017, pp. 27–34.

[63] M. Knight and B. Seeber, "Decoding lora: Realizing a modern lpwan with sdr," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[64] E.-H. Jang, Y.-J. Cho, S.-Y. Chi, J.-Y. Lee, S. S. Kang, and B.-T. Chun, "Recognition of walking intention using multiple bio/kinesthetic sensors for lower limb exoskeletons," in *Control Automation and Systems (ICCAS), 2010 International Conference on*. IEEE, 2010, pp. 1802–1805.

[65] J. A. Majumder, I. Zerin, C. P. Tamma, S. I. Ahamed, and R. O. Smith, "A wireless smart-shoe system for gait assistance," in *Biomedical Conference (GLBC), 2015 IEEE Great Lakes*. IEEE, 2015, pp. 1–4.

[66] N. Rana, "Application of force sensing resistor (fsr) in design of pressure scanning system for plantar pressure measurement," in *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on*, vol. 2. IEEE, 2009, pp. 678–685.

[67] N. Pinkam and I. Nilkhamhang, "Wireless smart shoe for gait analysis with automated thresholding using pso," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*. IEEE, 2013, pp. 1–6.

[68] I. Electronics, "Fsr® 400 series data sheet," 2014, <http://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf>.

[69] L. Technologies, "Bl600 series, single-mode ble module featuring smartbasic," <http://cdn.lairdtech.com/home/brandworld/files/Product%20Brief%20-%20BL600.pdf>.

[70] M. T. Inc., "Mcp73871, stand-alone system load sharing and li-ion/li-polymer battery charge management controller," <http://ww1.microchip.com/downloads/en/DeviceDoc/20002090C.pdf>.

[71] J. Shubham, B. Carlo, R. Yanzhi, G. Marco, C. Yingying, and C. C. Fabiana, "Lookup: Enabling pedestrian safety services via shoe sensing," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '15. ACM, 2015, pp. 257–271. [Online]. Available: http://doi.acm.org/10.1145/2742647.2742669

[72] I. González, J. Fontecha, R. Hervás, and J. Bravo, "An ambulatory system for gait monitoring based on wireless sensorized insoles," in *Sensors 2015*, 2015, pp. 16 589–16 613. [Online]. Available: http://doi.org/10.3390/s150716589

[73] B. Industries, "Specification of li-polymer rechargeable battery. model no.: Lp-443440-1s-3," 2012, <http://datasheet.octopart.com/LP-443440-IS-3-BAK-Industries-datasheet-12501472.pdf>.

[74] *User Modeling, Adaptation and Personalization, Proceedings of the 23rd International Conference UMAP 2015*. Springer, Lecture Notes in Computer Science LNCS9146, 2015, dublin, June 29 - July 3.

[75] M. P. Papazoglou and D. Georgakopoulos, "Introduction: Service-oriented computing," *Commun. ACM*, vol. 46, no. 10, pp. 24–28, Oct. 2003. [Online]. Available: http://doi.acm.org/10.1145/944217.944233

[76] J. Singh and J. M. Bacon, "On middleware for emerging health services," *Journal of Internet Services and Applications*, vol. 5, no. 1, pp. 1–19, 2014. [Online]. Available: http://dx.doi.org/10.1186/1869-0238-5-6

[77] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, L. Skorin-Kapov, and R. Herzog, *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*. Cham: Springer International Publishing, 2015, ch. OpenIoT: Open Source Internet-of-Things in the Cloud, pp. 13–25. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16546-2_3

[78] "The swarm at the edge of the cloud," Terraswarm Research Center, 2014, http://ecedha.org/docs/default-source/source/the-swarm-at-the-edge-of-the-cloud.pdf.

[79] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, ser. VLDB '06. VLDB Endowment, 2006, pp. 1199–1202. [Online]. Available: http://dl.acm.org/citation.cfm?id=1182635.1164243

[80] P. Persson and O. Angelsmark, "Calvin - merging cloud and iot," *Procedia Computer Science*, vol. 52, pp. 210 – 217, 2015, the 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915008595

[81] "Node-red, a visual tool for wiring the internet of things," Node RED, 2015, http://nodered.org.

[82] M. Bazzani, D. Conzon, A. Scalera, M. A. Spirito, and C. I. Trainito, "Enabling the iot paradigm in e-health solutions through the virtus middleware," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, June 2012, pp. 1954–1959.

[83] P. Maia, T. Batista, E. Cavalcante, A. Baffa, F. C. Delicato, P. F. Pires, and A. Zomaya, "A web platform for interconnecting body sensors and improving health care," *Procedia Computer Science*, vol. 40, pp. 135 – 142, 2014, fourth International Conference on Selected Topics in Mobile Wireless Networking (MoWNet'2014). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914014082

[84] "Google fit," Google, 2015, https://developers.google.com/fit/.

[85] N. Pombo, S. Spinsante, C. Chiatti, P. Olivetti, E. Gambi, and N. Garcia, "Assistive technologies for homecare: Outcomes from trial experiences," in *ICT Innovations 2015, Proceedings*, 2015, pp. 240–249.

[86] "Mqtt 3.1.1 specification," OASIS MQTT TC, 2016, http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[87] A. Pultier, "Mqtt libraries," 2016, https://github.com/mqtt/mqtt.github.io/wiki/libraries.

[88] L. Montanini, L. Raffaeli, A. D. Santis, A. Del Campo, C. Chiatti, G. Rascioni, E. Gambi, and S. Spinsante, "Overnight supervision of alzheimer's disease patients in nursing homes - system development and field trial," in *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health*. SCITEPRESS - Science and and Technology Publications, 2016. [Online]. Available: https://doi.org/10.5220/0005790000150025

[89] C. Chiatti, J. M. Rimland, F. Bonfranceschi, F. Masera, S. Bustacchini, L. Cassetta, F. Lattanzio, and on behalf of the UP-TECH research group, "The up-tech project, an intervention to support caregivers of alzheimer's disease patients in italy: preliminary findings on recruitment and caregiving burden in the baseline population," *Aging & Mental Health*, vol. 19, no. 6, pp. 517–525, 2015, pMID: 25188811. [Online]. Available: http://dx.doi.org/10.1080/13607863.2014.954526

[90] S. Labs, "Si1000/1/2/3/4/5 ultra low power, 64/32 kb, 10-bit adc mcu with integrated 240–960 mhz ezradiopro® transceiver," <https://www.silabs.com/Support%20Documents/TechnicalDocs/Si1000.pdf>.

[91] Espressif Systems IOT Team, "ESP8266EX Datasheet, v.4.3," online, 2015, <http://bbs.espressif.com/>.

[92] "SX1276 data sheet," Semtech Corporation. [Online]. Available: http://www.semtech.com/apps/filedown/down.php?file=sx1276.pdf

[93] "Python Client - documentation," Eclipse Foundation. [Online]. Available: https://eclipse.org/paho/clients/python/docs/

[94] "OASIS Standards Track Work Product, MQTT Version 3.1.1," OASIS, 2014. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[95] "LoRa Technical Specifications," LoRa Alliance. [Online]. Available: https://www.lora-alliance.org/

[96] R. Hsiao, D. Lin, H. Lin, C. Chung, and S. Cheng, "Integrating zigbee lighting control into existing building automation systems," 2012.

[97] R. K. Kodali and S. Soratkal, "Mqtt based home automation system using esp8266," in *Humanitarian Technology Conference (R10-HTC), 2016 IEEE Region 10*. IEEE, 2016, pp. 1–5.

[98] H. Zhu, Z. Pang, B. Xie, and G. Bag, "Ietf iot based wireless communication for latency-sensitive use cases in building automation," in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, June 2016, pp. 1168–1173.

[99] J. Zhang, A. Huynh, Q. Ye, and S. Gong, "Reliability and latency enhancements in a zigbee remote sensing system," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*. IEEE, 2010, pp. 196–202.

[100] E. T. P. on Smart Systems Integration (EPoSS). (2009) Internet of things in 2020: Roadmap for the future. [Online]. Available: https://www.smart-systems-integration.org/public/documents/publications/Internet-of-Things_in_2020_EC-EPoSS_Workshop_Report_2008_v3.pdf/download