Study of adaptation mechanisms of the wireless sensor nodes to the context for ultra-low power consumption

Andreina Liendo Sanchez

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : OPTIQUE ET RADIOFREQUENCES

Arrêté ministériel : 25 mai 2016

Présentée par

Andreina LIENDO SANCHEZ

Thèse dirigée par **Dominique MORCHE**, Directeur de recherche, CEA

préparée au sein du **Laboratoire CEA/LETI**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Etude des mécanismes d'adaptation des noeuds de capteurs sans fil dans le contexte de très faible consommation d'énergie

Study of adaptation mechanisms of the wireless sensor nodes to the context for ultra-low power consumption

Thèse soutenue publiquement le **25 octobre 2018**,
devant le jury composé de :

Monsieur DOMINIQUE MORCHE
INGENIEUR DE RECHERCHE, CEA GRENOBLE, Directeur de thèse
Monsieur MARCELO DIAS DE AMORIM
DIRECTEUR DE RECHERCHE, CNRS DELEGATION PARIS, Rapporteur
Monsieur GUILLAUME VILLEMAUD
PROFESSEUR ASSOCIE, INSA LYON, Rapporteur
Monsieur ANDRZEJ DUDA
PROFESSEUR, GRENOBLE INP, Président
Monsieur FRANCK ROUSSEAU
MAITRE DE CONFERENCES, GRENOBLE INP, Examinateur
Monsieur FRANÇOIS PÊCHEUX
PROFESSEUR, SORBONNE UNIVERSITES - PARIS, Examinateur

Université de Grenoble
Ecole doctorale Electronique, Electrotechnique, Automatique, Télécommunication et Signal (EEATS)
Spécialité: Optique et Radiofrequences

THESE

Study of adaptation mechanisms of the wireless sensor nodes to the context for ultra-low power consumption

– en Français –

Etude des mécanismes d'adaptation des noeuds de capteurs sans fil dans le contexte de très faible consommation d'énergie

préparée à

STMicroelectronics

– et –

CEA-LETI

par

Andreina LIENDO

Directeur de thèse	:	**Dominique MORCHE**
Co-encadrant	:	**Roberto GUIZZETTI**
Co-encadrant	:	**Franck ROUSSEAU**

Pour obtenir le grade de :
Docteure de l'Université de Grenoble

Thèse soutenue devant le jury le **25 Octobre 2018**

M. Andrzej DUDA	**Président du jury**
M. Marcelo DIAS DE AMORIM	**Rapporteur**
M. Guillaume VILLEMAUD	**Rapporteur**
M. François PECHEUX	**Membre du jury**
M. Franck ROUSSEAU	**Membre du jury**

Abstract

The Internet of Things (IoT) is announced as the next big technological revolution where billions of devices will interconnect using Internet technologies and let users interact with the physical world, allowing *Smart Home*, *Smart Cities*, smart everything. Wireless Sensor Network are crucial for turning the vision of IoT into a reality, but for this to come true, many of these devices need to be autonomous in energy. Hence, one major challenge is to provide multi-year lifetime while powered on batteries or using harvested energy. Bluetooth Low Energy has shown higher energy efficiency and robustness than other well known WSN protocols, making it a strong candidate for implementation in IoT scenarios. Additionally, BLE is present in almost every smartphone, turning it into perfect ubiquitous remote control for smart homes, buildings or cities. Nevertheless, BLE performance improvement for typical IoT use cases, where battery lifetime should reach many years, is still necessary.

In this work we evaluated Bluetooth Low Energy performance in terms of latency and energy consumption based on analytical models in order to optimize its performance and obtain its maximum level of energy efficiency without modification of the specification in a first place. For this purpose, we proposed a scenarios classification as well as modes of operation for each scenario. Energy efficiency is achieved for each mode of operation by optimizing the parameters that are assigned to the Bluetooth Low Energy nodes during the neighbor discovery phase. This optimization of the parameters was made based on an energy model extracted from the state of the art. The model, in turn, has been optimized to obtain latency and energy consumption regardless of the behavior of the nodes at different levels: application and communication. Since a node can be the central device at one level, while it can be the peripheral device at the other level at the same time, which affects the final performance of the nodes.

In addition, a novel battery lifetime estimation model was presented to show the actual impact that energy consumption optimization have on nodes lifetime in a fast (in terms of simulation time) and realistic way (by taking

into account empirical data). Performance results were obtained in our Matlab based simulator based on OOP paradigm, through the use of several IoT test cases. In addition, the latency model used for our investigation was experimentally validated as well as the proposed parameter optimization, showing a high accuracy.

After obtaining the best performance possible of Bluetooth Low Energy without modification of the specification, we evaluated the protocol performance when implementing the concept of Wake-Up radio, which is an ultra low power receiver in charge on sensing the communication channel, waiting for a signal addressed to the node and then wake the main radio up. Thus, the main radio which consumes higher energy, can remain in sleep mode for long periods of time and switch to an active mode only for packet reception, therefore saving considerable amount of energy. We demonstrated that Bluetooth Low Energy lifetime can be significantly increased by implementing a Wake-Up radio and we propose a modification of the protocol in order to render this protocol compatible with an operating mode which includes a Wake-Up radio. For this, we studied the Wake-Up radio state of the art and evaluated Bluetooth Low Energy devices lifetime when a selected Wake-Up radio is implemented at the master side.

Contents

List of Figures

List of Tables

Glossary

$\boldsymbol{T_{AI}}$ Advertising Interval. 19, 20, 24, 26, 33, 36, 64, 91–93

$\boldsymbol{T_{CI}}$ Connection Interval. 16, 17, 23, 99

$\boldsymbol{T_{SI}}$ Scanning Interval. 23, 24, 26, 27, 32, 33, 91–93

$\boldsymbol{T_{ST}}$ Supervision Timeout. 23, 99

$\boldsymbol{T_{SW}}$ Scanning Window. 23, 24, 26, 33, 91–93

$\boldsymbol{T_{apk}}$ Time to transmit an advertising packet. 25

$\boldsymbol{t_{ch}}$ Time between two consecutive advertising packets. 25

ATT Attribute Protocol. 14

BLE Bluetooth Low Energy. iii, iv, vi, viii, 4, 5, 7–9, 11–15, 18, 19, 22–27, 30, 31, 33, 35, 36, 38, 44, 46–49, 55, 56, 58, 59, 61–63, 66, 67, 69, 71–80, 83, 84, 87–91, 99, 103–106, 109, 112–116, 119–122

BR Basic Rate. 12

CDF Cumulative Distribution Function. 28

CL Critical Latency. 59, 82–84, 116

CM Connected Mode. 19, 25, 30, 31, 34, 35, 46, 47, 49, 58, 59, 71, 73–76, 106, 112

CRC Cyclic Redundancy Check. 15, 21

DL Discovery Latency. 9, 19, 25, 26, 30–33, 35–38, 47, 49, 59, 60, 63, 64, 67, 68, 81–83, 89, 91–93, 111, 112, 120

EDR Enhanced Data Rate. 12

EIRP Effective Isotropic Radiated Power. 103

FSM Finite State Machine. 15

GATT Generic Attribute Profile. 14

HCI Host Controller Interface. 87–90

IF intermediate frequency. 104

IoT Internet of Things. i, 1, 2, 4, 7, 9, 58–60, 71, 83

LL Link Layer. 12, 14, 15, 18–20, 30, 31, 49, 88, 89, 99

masterSCA master Sleep Clock Accuracy. 77, 106

ND Neighbor Discovery. 19, 23, 25, 30, 31, 34–36, 38, 46, 47, 49, 58, 59, 62–64, 67, 69, 71–74, 76, 78, 99, 109–111, 116, 119, 120

OCF OpCode Command Field. 90, 91

OGF OpCode Group Field. 90

OOK On-Off Keying. 104, 106, 117

OOP Object Oriented Programming. 9, 46, 47, 49

PDF Probability Density Function. 33

PDU Protocol Data Unit. 21–23

RFID Radio Frequency Identification. 103

slaveSCA slave Sleep Clock Accuracy. 77

SMP Security Manager Protocol. 14

UML Unified Modeling Language. 47, 48

WuR Wake-Up radio. 9, 10, 99–101, 103–117, 119, 121, 122

Chapter 1

Introduction

1.1 Context and Problematic

1.1.1 The IoT

The IoT is the paradigm announced as the next big technological revolution where billions of devices will interconnect using Internet technologies, allowing users to interact with the physical world. It is expected that by 2020, around 50 billion devices will be internetworked according to Ericsson [10]. According to F. J. Riggins and S. F. Wamba [11], forecasts for the number of connected devices on the Internet of Things for the year 2020 vary considerably, from 26 billion (Gartner [12]), to 30 billion (ABI research [13]), to 50 billion (Cisco [14]), to 75 billion (Morgan Stanley [15]). One of the key enablers for the IoT are wireless communications, that allow seamless interaction with numerous devices. Many of these devices are expected to be tiny autonomous integrated systems with sensors and actuators disseminated everywhere, allowing *Smart Home*, *Smart Cities*, *Smart Campus*, smart everything.

The development of IoT includes three components: a) embedded intelligence: smart devices designed to complete a task without human help, b) connectivity: smart devices are generally connected to other devices or a device network via different protocols and c) interaction: smart devices that exchange information by themselves and communication is not only human-human or human-thing but thing-thing [16]. A simple example that illustrates the concept can be a smart thermostat that learns about the user preferences and automatically adjusts the temperature in a room, so in addition to a comfortable environment at home, it will help save on heating and using energy more efficiently.

To enable connectivity different types of networks are used such as: Wire-

less Personal Area Network (WPAN), Local Area Network (LAN), Wide area network (WAN), Wireless Sensor Networks (WSN), where devices can be connected via different protocols as: RFID, ZigBee, 3/4G, Wi-Fi, LoRaWAN, Bluetooth, etc.

Much has been investigated to define all that is needed for IoT's implementation, going from architecture, technologies and security issues to possible fields of applications [16, 17, 18, 19, 20, 21], these investigations show that Wireless Sensor Networks (WSN) are crucial for turning IoT's vision into a reality.

A WSN is a self-configuring local area network whose nodes, equipped with some kind of sensor, wirelessly send their measurements to a set of receiving nodes called sinks. WSN nodes must meet requirements of autonomy, low power consumption, low cost and robustness. One of the most critical aspects of these networks is the effective management of energy, which makes the communication protocols very important to avoid waste of energy in collisions or useless channel listening. Recent research in physical (PHY)-layer has demonstrated promising progresses on link reliability and energy efficiency. However, in modern medium access control (MAC) design, energy efficiency has become one of the key requirements and is still a hot research topic [22].

1.1.2 WSN Protocols and Energy Consumption

A WSN node is typically battery operated and thus energy constrained. The operational life of a miniaturized system, capable of sensing, storing and receiving/transmitting data, is relatively short, requiring periodic maintenance. For some applications where location of nodes hinders access to them, a simple chore as replacing the batteries becomes a very arduous an intensive labor and even dangerous in some cases, thereby increasing the costs of deploying such networks. It is therefore desired to have WSN nodes running on very little power, even for sensor running on wired protocols such as PLC [22] as the energy consumption has an associated cost.

There are several major sources of energy waste in WSN [23]:

- **Idle monitoring:** listening to receive possible traffic that is not sent. The power consumption in sensing the communication channel is as high as that of actually receiving a packet.
- **Collision:** when a transmitted packet is corrupted and then discarded there is energy wasted on a packet which is never recovered. Additionally, nodes usually retransmit the packet leading to a waste of power.

- **Control packet overhead:** such as packets use for synchronization and security purposes. Sending and receiving control packets consumes energy too.
- **Overhearing:** sometimes nodes pick up packets coming from other nodes in the vicinity which not addressed for this node (or at least the packet header), thus causing energy waste.

Many solutions to this idle listening problem are based on the technique of duty cycling. The basic idea of low duty cycle is to reduce the time a node is idle or spends overhearing an unnecessary activity by putting the node in the sleep state. The most ideal condition of low duty cycling is when a node sleeps most of the time and wakes up only to transmit or receive packets. Usually after a node wakes up, it listens to the channel for any activity before transmitting or receiving packets. If no packet is to be transmitted or received, the node returns to the sleep state. A whole cycle consists of a listening period and a sleep period, the duty cycle is measured as the ratio of the listening period length to the whole period length. A small duty cycle means that a node is asleep most of the time in order to avoid idle listening and overhearing. However, a trade-off between duty cycle and latency must be achieved as latency increases with the decrease of duty cycle.

WSN use MAC protocols to coordinate the time when nodes will access a shared communication channel, MAC stands for Medium Access Control, the MAC layer is the lower sublayer of the data link layer (layer 2) of the seven-layer OSI model, it provides addressing and channel access control mechanisms that make it possible for several nodes to communicate within a multiple access network that uses the same shared physical medium.

The channel access control mechanisms provided by the MAC layer are also known as a multiple access protocol. Duty-cycling MAC protocols can be categorized based on their channel access strategy into two groups, contention-based and contention-free protocols. Contention-based protocols are intended to resolve a collision after it occurs or try to avoid it. These protocols execute a collision resolution protocol after each collision, they are often called random access protocols or "listen before talk", as nodes use the medium without pre-coordination. Example of this type of protocols are the Aloha protocol, Carrier Sense Multiple Access (CSMA) and Multiple Access with Collision Avoidance (MACA). Contention-free protocols, are designed to ensure that a collision can never occur. In these protocols, the nodes are following some particular schedule which guarantees collision-free transmission times. Typical examples of such protocols are: Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code

Division Multiple Access (CDMA). These protocols are either centralized or rely on a global time reference.

In terms of energy consumption, contention-based protocols exhibit substantial waste of energy due to high idle listening and overhearing, as nodes may wake up and listen to messages intended for another node within the network and false wake-up may also occur when a radio using a different protocol is transmitting in the same frequency band. Another important source of energy waste are packet collisions. Meanwhile, contention-free protocols could incur in significant overheads for resource allocation (e.g. time slots), especially for sensors with dynamic data traffics or topologies. Additionally, clock drift and timing errors are also the cause for a significant waste of energy for these protocols. One way to mitigate the energy lost is to implement hybrid protocols containing features of both contention-based and contention-free protocols [22]. However, overall energy consumption of WSN nodes is not only subject to the chose among contention-based, contention-free or hybrid protocol, but to fulfilling specific application objectives such as traffic, topology, range, etc.

Among the preferred technology for WSN applications we can find ZigBee which is based on CSMA/CA protocol, ANT and BLE which are based on TDMA, SIGFOX and LoRa which are based on ALOHA protocol. Among these, BLE has been proven to be more efficient in terms of energy consumption [24, 25, 26] for the world of small amount of data and short range applications which is within our target. BLE is a *de facto* standard[27] for low-energy short-range personal communications, and as such a key radio technology for the IoT [28]. Bluetooth, and BLE in particular, is one of the most widespread standard wireless communication technology today, available in millions of devices for leisure, medical, or home automation applications, where smartphones play the crucial role of control center for the user and gateway to the Internet. BLE has been designed to provide lower energy consumption, security, ease of use and to avoid interference with higher power IEEE 802.11 networks. This research focus on performance improvement in terms of energy consumption of BLE for a large variety of applications for the IoT. As said earlier, BLE is based on TDMA protocol and we mostly focus on one of the main causes of energy loss of this protocol which is the overhead.

1.1.3 Bluetooth Low Energy

Bluetooth is a standardized wireless technology for data exchange over short distances that has been primarily designed for low-power consumption, with a short range and based on low-cost transceiver microchips. Four major

versions have been released since its creation in 1994 by Ericsson, where one of the main improvement across versions is the data rate, the newer the version, the higher the data rate. High speed data link was one of the original objectives since it was targeted at data exchange between portable devices like smartphones.

Bluetooth v4.2, includes classic Bluetooth, Bluetooth High Speed, and Bluetooth Low Energy —aka. BLE, Bluetooth LE or Bluetooth Smart— providing Basic Rate, Enhanced Data Rate and Low Energy respectively. This work deals with Bluetooth Low Energy (BLE) which is only a subset of Bluetooth and is aimed at very low power applications, targeting small and low cost sensors running on a coin cell battery unlike the other versions.

BLE operates channel hopping communications in the 2.4 GHz ISM band. It uses 40 radio frequency channels between 2402 MHz and 2480 MHz with 2 MHz spacing. Three channels are used for advertising and were specifically chosen to avoid IEEE 802.11 frequencies as much as possible, thus reducing interference. The 37 remaining channels are used to exchange application data using adaptive frequency hopping for co-existence and robustness. This is the physical layer of BLE.

1.1.4 BLE Typical Applications

BLE technology is broadly used for automotive, home automation, consumer electronics, medical health, smartphones, pc peripherals, wearables, sports fitness, retail location based services [29]. Bluetooth SIG has provided with a list of profiles [29] intended to enable Bluetooth devices to be efficiently implemented for different applications such us collecting sensor information, health, sports and fitness, environmental sensing and proximity applications, among others. These profiles define the behavior for both central and peripheral role devices and moreover, they include recommended scanner/advertiser configuration to ensure optimal performance of the protocol in terms of latency and energy consumption.

Proximity is one of Bluetooth profile which enables presence and distance monitoring between two devices [29], it enables actions to be taken when two devices are close to each other. Typically, this is discussed as a mobile device moving through space, although it might also be two mobile devices moving relative to each other. One of the best known applications of this type is iBeacon which is a Bluetooth technology that helps a device understand its location and surroundings with a high degree of accuracy. iBeacon enables a device to display web pages, negotiate transactions and control nearby machines. Although the technology is mostly used in retail, it is potentially useful in a wide variety of contexts such as museum guides, retail store en-

hancement [30], finding the nearest parking lot, calculate position based on reference points, push information (advertising) and tracking luggage at the airport [31] among others.

The profile [29] establishes two roles: proximity monitor which is the collector and the peripheral device (advertiser), and proximity reporter which is looking for an iBeacon and has the central device role (scanner), normally at the user side. The proximity from the iBeacon to the targeted device can be estimated by either dividing the surrounding area of the iBeacon into regions as shown in figure Figure 1.1 or either by using the strength of the signal from the iBeacon (RSSI) to calculate the distance in meters.

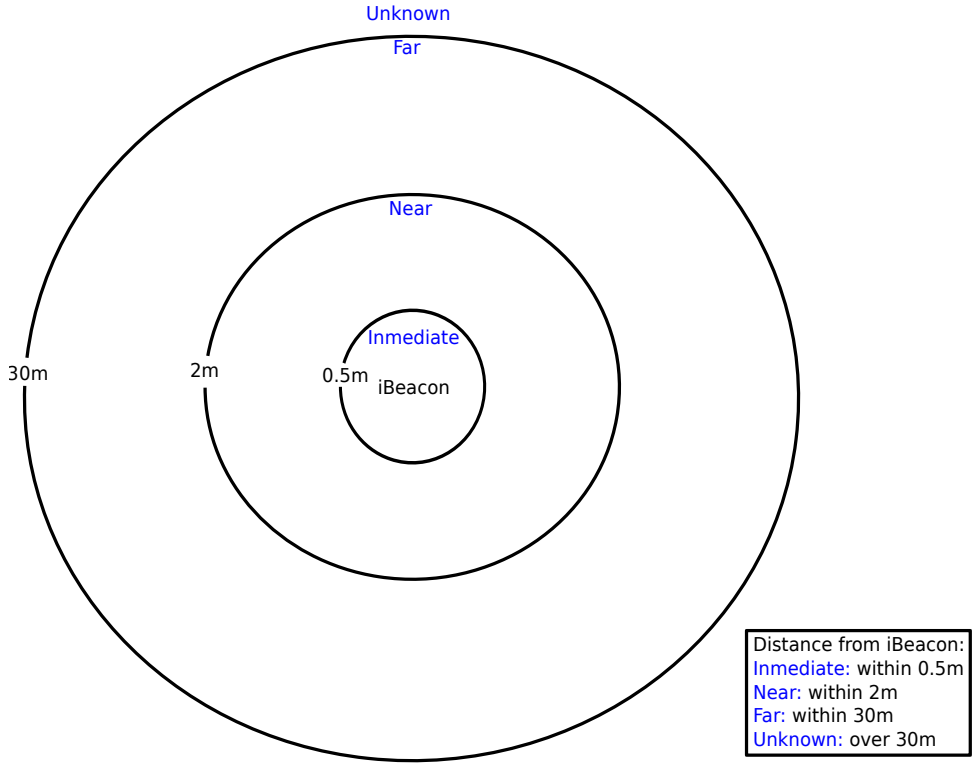


Figure 1.1: iBeacon regions

1.1.5 BLE limitations for typical scenarios

BLE is designed for periodic transfers of very small amounts of data, such as beacons providing proximity detection and localized information, and leisure or medical devices monitoring vital parameters. One important aspect of energy consumption evaluation for BLE is the scenario, and the suitability

Table 1.1: Recommended Advertising Interval, Scan Interval and Scan Window Values (Table 5.1 and 5.2 in[9])

Advertising/Scan Duration	**Parameter**	**Value**
First 30 seconds (fast connection)	Adv. Interval	20 ms to 30 ms
After 30 seconds (reduced power)	Adv. Interval	1 s to 2.5 s
First 30 seconds (fast connection)	Scan Interval	30 ms to 60 ms
	Scan Window	30 ms
After 30 seconds (reduced power)	Scan Interval	1.28 s
– Option 1	Scan Window	11.25 ms
After 30 seconds (reduced power)	Scan Interval	2.56 s
– Option 2	Scan Window	11.25 ms

of the system behavior to the scenario. A simple example will illustrate this fact: suppose that a user triggers the advertising process on his/her device, a heart rate monitor, but no scanner is available. Then the device could spend all of its energy advertising without any scanner listening, hence the necessity for higher level control of the process. This behavior is specified in a collection of profiles associated to the core Bluetooth specification, and typical values are shown in Table 1.1, where two advertising and scan modes are defined: if a connection is not established in the first 30 s the devices switch to a lower power mode.

BLE does not seem suitable for long running applications with few events, that nevertheless constitute a large portion of envisioned IoT applications. In this work we consider applications such as metering and home automation that are typical of the Smart Building IoT. Such network comprise tens of nodes that communicate relatively infrequently, every 5 to 15 minutes or only a few times a day, with only a few bytes of payload and relaxed latency requirements of 200 ms or above[32]. Among some of the specific uses cases within this category we can list: HVAC, lighting control, structural integrity monitoring and access control. Given the need to explore different scenarios, we propose to classify scenarios in section 3.3 in order to adapt the protocol to the application requirements and we present two test cases where we consider only application parameters regardless of the protocol and topology used, with the objective of evaluating BLE performance for such use case requirement in section 2.5.

In some cases, communication between two devices can be initiated when the device on sensor side detects activity, in some others it is triggered by the user by placing the battery or by pressing a button, or just by presence;

and depending on each device role, the scanner gets to be available waiting for an advertising packet or conversely, it may be the advertiser who is available waiting to be discovered by the scanner. In other words, one of the two devices must remain active for longer periods of time thus consuming a considerable amount of energy.

For a given use case, there could be a permanent source of energy or a sufficiently large battery capacity operated device at the scanner side (such as smartphone or tablet), and a limited battery capacity operated device on the advertiser side (such as coin cell battery); while having the opposite case for a different use case. So, the energy consumption and discovery latency constraint must be evaluated differently depending on the scenario.

1.2 Objective of this Thesis

This work aims to contribute to the optimization of BLE performance in order to implement it in a wide range of IoT related applications. The general objective is to achieve energy consumption reduction since it is required to extend battery lifetime, while providing a good trade-off between energy consumption and latency.

In order to achieve the general objective of this thesis, specific objectives are:

1. **Energy model:** it is necessary to predict energy consumption of the protocol to be able to evaluate and contrast different energy optimization methodologies.

2. **Energy optimization:** the total energy consumption of a BLE device depends on a set of parameters used to configure the devices allowing communication between two or more devices and which impacts on latency and energy consumption, but also on the overall communication pattern, which is imposed in great measure by specific *application requirements*. An optimization methodology is required to reduce energy consumption based both protocol configuration and application requirements.

3. **Wake-up radio implementation evaluation:** after optimizing BLE performance to reduce energy consumption to a minimum, Wake-Up radio (WuR) implementation evaluation is desired since it promises longer autonomy and system reactivity.

1.3 Thesis Structure

In Chapter 2 we introduce the BLE performance evaluation methodology for this work, we do a review of the existing energy models and present our proposed energy model optimization for which BLE performance is evaluated. In addition we present a novel model for battery lifetime estimation which is designed for lithium battery like coin cell type. Along this work most of the results are given in terms of battery lifetime. They are all based on a a Panasonic CR2032 coin cell battery with a nominal capacity of 225 mAh.

Based on the mentioned optimized energy model, a BLE parameter optimization for neighbor discovery process is presented in Chapter 3. In order to evaluate the impact of the proposed parameter optimization, the first battery lifetime results are given for a retail store use case and a wireless medical telemetry use case. The results using the parameter optimization are then compared with the Bluetooth SIG recommended profiles which are introduced in Section 1.1.5. Then we present our methodology to mitigate the effects of overhead due to network synchronization. It consists in categorizing all possible use cases into three types of scenarios and implementing new operating modes of BLE for each type of scenario. The methodology is evaluated based on two test cases: a temperature and humidity monitoring use case and light switches control use case, for which we present the architecture of our Matlab based simulator where we have implemented the energy and battery models and which is based on Object Oriented Programming (OOP) paradigm.

In Chapter 4, we present a set of experiments which are intended to validate Discovery Latency (DL) model. The results proof that our methodology can be used for a wide range of IoT scenarios and for any BLE device regardless of the manufacturer. Experiments are performed in a platform called WalT whos architecture and features are presented in Chapter 4.

In Chapter 5, we evaluate the possibility to implement WuR (an ultra low power radio receiver within BLE and its benefits which are given in terms of lifetime. We do an overview of WuR state of the art and present results under different scenarios where the WuR is implemented at the central node side. In general we evaluate to types of scenarios and propose a new mode of operation for BLE in order to efficiently adapt the protocol with the use of the WuR. Finally conclusions of this work and open challenges are presented.

Chapter 2

Modeling and Evaluation Methodology

In this chapter we discuss performance evaluation metrics of BLE and existing models for discovery latency and energy consumption estimation. In addition, a proposed model optimization, along with a novel battery lifetime estimation model are introduced.

2.1 Performance evaluation

With respect to wireless communication performance evaluation, our research is focused on two important metrics: energy consumption and latency. Energy consumption is one of the most critical and restraining factors in WSN. We concentrate in the energy consumed during a total amount of time instead of peak power. Improvements in energy efficiency are generally achieved by adopting a more efficient technology or by application of commonly accepted methods to reduce energy losses, our work targets the optimization of the protocol in order to minimize the usage of the radio by spending most of its lifetime in sleep mode, thus extending its lifetime.

Determining how energy consumption will affect device lifetime can be a quite complex task. Furthermore, most of these devices are intended to run on batteries, as they may be placed where no other infrastructure is available, hence calculating the battery lifetime of such a product is always necessary. Batteries are components based on chemical reactions to generate electricity. The actual behavior of such components is very non-linear and strongly dependent on the use conditions such as temperature and discharge rate. Furthermore battery lifetime doesn't only depend on the battery capacity which is usually given in mAh, but also on how the battery capacity itself

is affected by the usage pattern, meaning that its lifetime depends on both the battery capacity and the node behavior. We will use analytical models to estimate both the energy consumption of the node and the lifetime of the battery.

The other important metric is latency. Usually, using a very low duty cycle protocol will extend battery lifetime but it gives rise to another problem: the increase in latency. If a receiving node spends most of the time in sleep mode, there are high chances that it will not be available at the moment of packet arrival, forcing the transmitting node to repeatedly send the packet, resulting in high latency and increase of energy consumption for the transmitting node. So, estimating the battery life prior to deployment of the actual node requires an analytical method which coarsely captures the node behavior. Therefore, understanding the communication protocol and assessing the energy consumption of each of the device's states, is indispensable. A description of the BLE protocol is then given in the next section.

2.2 BLE Protocol Description

In this section we describe in detail the Link Layer (LL) of the protocol. BLE point-to-point packet transfer resides in the LL in which we concentrate our research. The latency estimation for BLE depends on the communication mode which are Neighbor Discovery and Connected Mode and are both explained in the Section 2.2.2. Our work focus mostly on the duration of Neighbor Discovery process often called Discovery Latency. Also, we will extract the most significant parameters that influence the energy consumption of the BLE nodes.

2.2.1 BLE Protocol Stack

BLE was designed for low bandwidth and low latency applications. BLE is not a protocol for file transfer or streaming but for sending small bundles of data unlike the other Bluetooth versions Basic Rate (BR)/Enhanced Data Rate (EDR), as BLE is targeted for a world of coin cell batteries. The goal of BLE is to enable a wide variety of small devices to act as peripherals for smartphones and other mobile devices. Ii has been said that BLE will be to smartphones *universal peripheral connectivity* as USB is for PC's [33].

The Bluetooth low energy technology is designed for addressing two alternative implementations:

- Smart device

- Smart Ready device

Smart devices support only the BLE standard. It is used for applications in which low power consumption and coin cell battery is the key point (as sensors). Smart Ready devices support both BR/EDR and BLE standards (typically a mobile or a laptop device). We focus pour work on Smart devices.

BLE has several protocol layers which form the BLE protocol stack. The BLE protocol stack allows devices to locate, connect and exchange data with each other. It is composed of three main parts which are Controller and Host stacks, plus applications which is on top of the two main stacks. The controller stack contains the radio interface, and the host stack deals with the high level data. The entire protocol stack is shown in Figure 2.1.

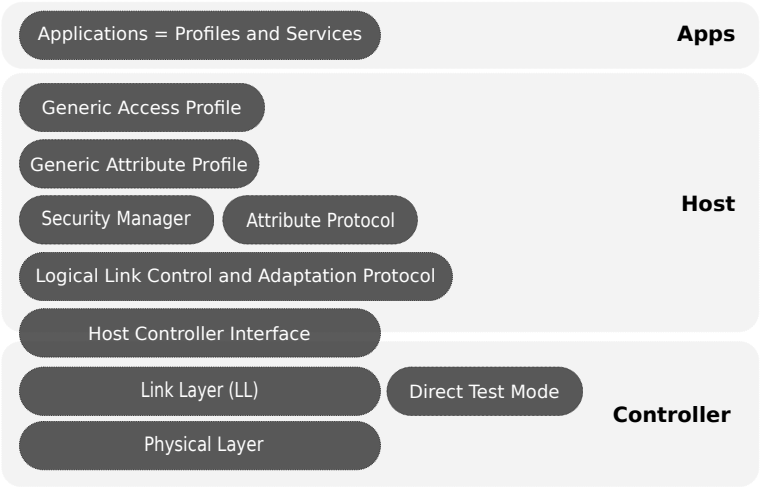


Figure 2.1: BLE Protocol Stack

The **Physical Layer (PHY)** is a $1Mbps$ adaptive frequency-hopping Gaussian Frequency Shift Keying (GFSK) radio. It operates in the license free 2.4 GHz ISM band. BLE uses 40 RF channels (0-39), with 2 MHz spacing. There are two channels types:

1. Advertising channels that use three fixed RF channels (37, 38 and 39) for sending/receiving advertising channel packets used for discoverability/connectability and broadcasting.

2. Data physical channel uses the other 37 RF channels for bidirectional communication between connected devices or in other words for application data exchange.

BLE is an Adaptive Frequency Hopping (AFH) technology that can use only a subset of all the available frequencies in order to avoid the frequencies used by other no-adaptive technologies. This allows to move from a bad channel to a known good channel by using a specific frequency hopping algorithm which determines the next good channel to be used.

The **Link Layer (LL)** defines how two devices can use a radio for transmitting information between each other. It will be explained in higher detail in the next section.

The **Host Controller Interface Layer** provides a mean of communication between the host and controller either through software or by a hardware interface such as SPI, UART or USB.

The **Logical Link Control and Adaptation Protocol (L2CAP)** takes multiple protocols from the upper layers and encapsulates them into the standard BLE packet format (and vice versa). It also takes large packets from the upper layers and breaks them up into chunks that fit into the 27 bytes maximum effective payload size (v4.0 4.1) of the BLE packets on the transmit side, and vice versa.

The **Attribute Protocol (ATT)** forms the basis of data exchange in BLE applications, is a simple client/server protocol based on attributes presented by a device. Each server contains data organized in the form of attributes, each of which is assigned a 16-bit attribute handle, a universally unique identifier (UUID), a set of permissions, and a value. The ATT allows a device to expose certain pieces of data, known as attributes, to another device. The device exposing attributes is referred to as the Server and the peer device using them is called the Client.

The **Security Manager Protocol (SMP)** provides a framework to generate and distribute security keys between peers to encrypt a LL communication.

The **Generic Attribute Profile (GATT)** defines a framework for using the ATT protocol, and it is used for services, characteristics, descriptors discovery, characteristics reading, writing, indication and notification. It comes on top of the ATT. It adds a data model and hierarchy, it defines how data is organized and exchanged in between different applications.

2.2.2 BLE Communication Modes

The way in which two BLE devices are able to exchange data, is determined by the LL of the protocol stack. It defines packet formats, protocol data, control procedures, mapping from and allocation of RF Channel, encryption, packet error checking (via Cyclic Redundancy Check (CRC)), etc.

The LL of the BLE protocol specification establishes five states machines through which at least two devices must transit in order to establish a communication: `stand by`, `advertising`, `scanning`, `initiating`, and `connection` (see Figure 2.2). A device LL allows only one state to be active at a time. A device can go from any state to `stand by` state.

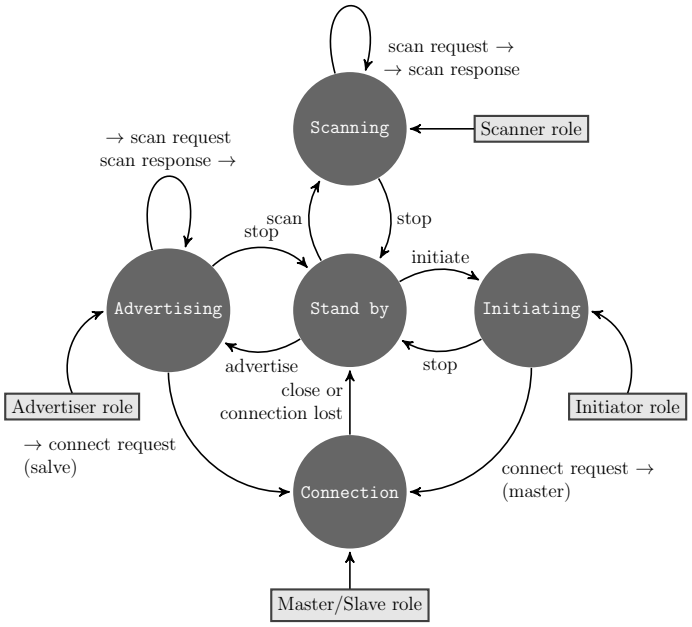


Figure 2.2: BLE Finite State Machine

In what follows we will explain in detail how BLE devices must transit through all the Finite State Machine (FSM) states. This has the objective to deeply understand the LL of the protocol and consequently to understand the existing energy models of the protocol. The models are based on the LL and presented in Section 2.3. A model optimization is then presented in Section 2.3.4.

A device starts in `stand by` state and remains there until the host layer tells to do otherwise. A device in the `advertising` state is called *advertiser*. This state is required if a device wants to be discoverable or connectable —to let other devices know that it wants to join the network— or wants to broadcast data. In other words, devices can advertise:

- To broadcast promiscuously,
- To transmit signed data to a previously bonded device,
- To advertise their presence to a device wanting to connect,

- To reconnect asynchronously due to a local event,
- To act as a location beacon.

In the `scanning` state a device will receive advertising packets, it is used simply to listen if there are any devices requiring to join the network, it will periodically listen to the three advertising channels. A device in `scanning` state is called *scanner*.

After successfully receiving an advertising packet, a *scanner* goes into the `initiating` state and listens for advertising channel packets from a specific device(s) (represented by the first packet in Figure 2.3) and it may respond to these packets to initiate a connection by sending a *connection request* (`CONNECT_REQ` packet in Figure 2.3) to the advertiser and moves to `Connection` state. A device in `initiating` state is called *initiator*. A *scanner* device who enters into a `Connection` state is called `master` whereas an *advertiser* device who enters into a `connection` state is called `slave`. A device enters the `Connection` state when the *initiator* sends a *connection request* or when an *advertiser* receives a *connection request*. However, at this point a connection is considered to be created but not yet established. A connection is only considered to be established once a data channel packet has been received from the peer device.

The master has the flexibility to schedule the starting moment of the first connection event at a time of its choosing called *anchor point*. The slave will listen waiting for the first packet from the master during a time duration called *transmit window* (see Figure 2.3). The *connection request* packet contains three parameters used to determine the *transmit window*: the *connection interval* T_{CI} which is the interval between connection events, the *transmit window offset* which is a multiple of 1.25 ms in the range of $[0,T_{CI}]$, and the *transmit window size* which is a multiple of 1.25 ms in the range of $[1.25, \min(10,T_{CI}-1.25)]$ ms. This three parameter are illustrated in Figure 2.3.

The *transmit window* starts at 1.25 ms $+transmitWindowOffset$ after the end of the *connection request*. After successfully receiving the first packet from the master, the slave will send application data and a connection is considered to be established. The master ensures that a connection event closes at least T_IFS (150μ,according to the specification) before the anchor point of the next connection event. The next connection event anchor point happens T_{CI} after the first anchor point.

In addition, because of sleep clock accuracies, there is uncertainty for the slave about the exact timing of the master's anchor point. Therefore, the slave is required to re-synchronize to the master's anchor point at each

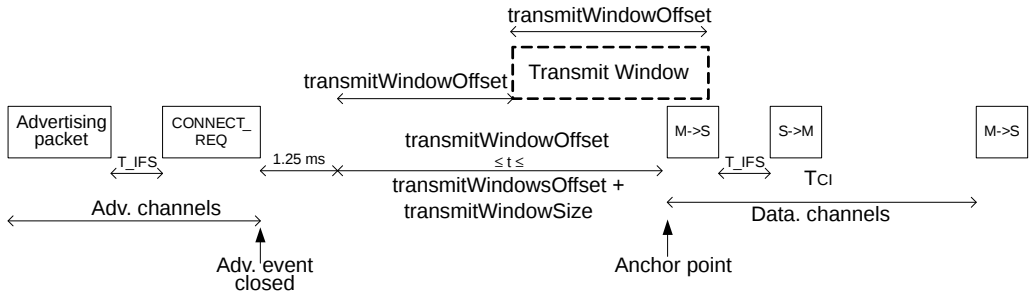


Figure 2.3: Master's view on connection setup with a non-zero transmit window offset [1]

connection event. The slave calculates the time when the master will send the first packet of a connection event (an empty-payload packet) taking into account clock jittering (and the `transmit window` in the case of a first connection event).

In order to calculate the extra time it will need to listen, the slave also uses the master's sleep clock accuracy ($masterSCA$) from the *connection request*, together with its own sleep clock accuracy ($slaveSCA$) and the *anchor point* of the last connection event when it received a packet from the master (`timeSinceLastAnchor`). For more detail in how sleep clock accuracies are considered by BLE Core, we refer the reader to [1]. This extra listening time is called the Window Widening (ww) and is shown in Figure 2.4.

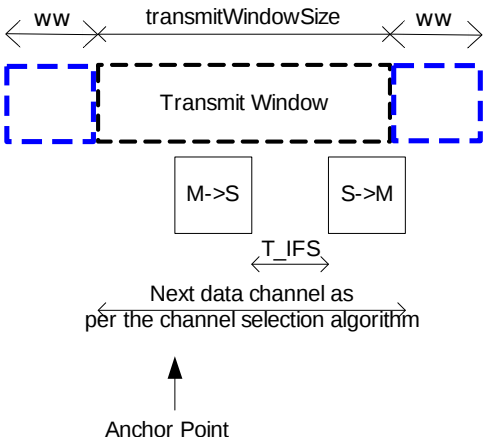


Figure 2.4: Window Widening

During connection setup or during a connection parameter update, the slave listens waiting for the master's anchor point, during a time equal to ww before the start of the transmit window and until ww after the end of the transmit window. The slave applies the ww during every connection event. Assuming the clock inaccuracies are purely given in parts per million (ppm), it is calculated according to Eq. 2.1.

$$ww = \left(\frac{masterSCA + slaveSCA}{1000000} \right) timeSinceLastAnchor \tag{2.1}$$

Moreover, a simpler way to describe the LL is by grouping the FSM states within two categories: *connection-oriented* and *connectionless*, where `connection` state is *connection oriented*, `stand by` state doesn't belong to any category and the rest of states are all *connectionless*. Based on that, BLE communication can be categorized into two modes of communication: Neighbor Discovery (ND) when a device transits through the non-connected states and Connected Mode (CM) when transiting the connected states.

To summarize, during ND, communication is asynchronous and devices transit through 4 states: `stand by`, `scanning` and `initiating` states for scanners, or `stand by` and `advertising` states for advertisers, using only the 3 advertising channels. In CM a device transits through `connection` and `stand by` state, and uses the 37 data channels.

In addition, the scanning state has two substates: the `passive scanning` and the `active scanning`. In the `passive scanning` the device only listen and never transmit anything, while in the active scanning, when a new device is discovered by the LL, a *scan request* is sent to the advertiser and a *scan response* is expected in reply. It can be used to either request extra information to the *advertiser* before entering a *connection* or to communicate asynchronously as shown in Figure 2.5.

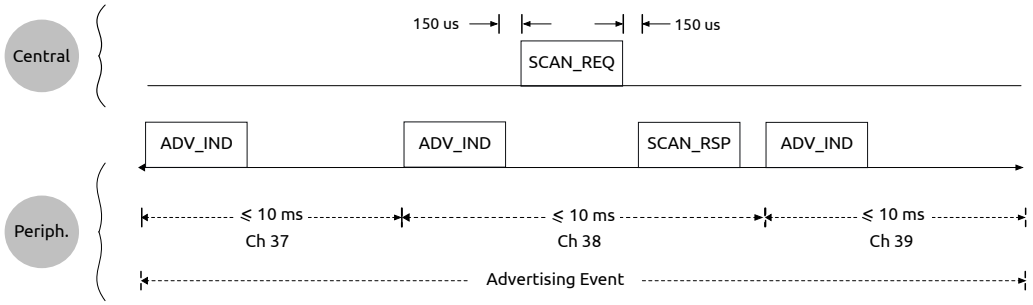


Figure 2.5: Active Scanning

In general, one of the most important features of BLE is the possibility for the nodes to spend a certain amount of time in sleep mode in order to save some energy. The time devices spend in active or sleep mode, or in other words in each of the states depicted in this section, should be determined based on the targeted application and within a range which is defined by the BLE specifications. Depending on the chosen configuration, devices will spend more or less energy and the latency will be affected as well. In the next section we present all the parameters that affect BLE performance.

2.2.3 Energy-Affecting Parameters

The BLE standard specifies a large range of feasible parameter values for both ND and CM that allows flexibility for selecting appropriate parameter values for BLE devices. The choice of these parameters will greatly affect the performance of the devices especially during ND, as very long DL will incur in high energy consumption, where DL is the time interval between the moment the advertiser enters into advertising sate up to the moment when the advertising packet is received by an initiator. Here we break down all these parameters.

2.2.3.1 In neighbor discovery phase

- **Advertising Interval** T_{AI} such that $T_{AE} = T_{AI} + \rho$ as shown in Figure 2.6. We have called T_{AE} the time between two consecutive advertising events, where T_{AI} is the advertising interval, and ρ is the random advertising delay in the range of $[0, 10]$ ms which is computed for each advertising event to alleviate synchronization effects. For each advertising interval, the advertiser sends packets in the 3 advertising channels during a time which is hardware and packet size dependent. The device remains in sleep mode from the end of the last transmitted packet to the starting time of the next advertising event. The advertising interval T_{AI} is an integer multiple of 0.625 ms in the range of [20 ms,10.24 s] or [100 ms,10.24 s] depending on the advertising type.

- **Advertising packet size** advertising packets contain the advertiser address and advertising data, or initiator address depending on the type of advertising. LL defines the packet format which is a unique format for both advertising channel packets and data channel packets. Each packet consists of four fields: the preamble, the Access Address, the Protocol Data Unit (PDU) and the CRC as shown in Figure 2.7

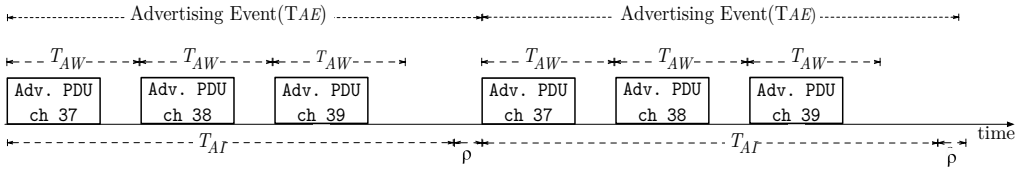


Figure 2.6: Advertiser in Neighbor Discovery

(where the shortest packet is 80 bits in length and corresponds to an empty payload packet).

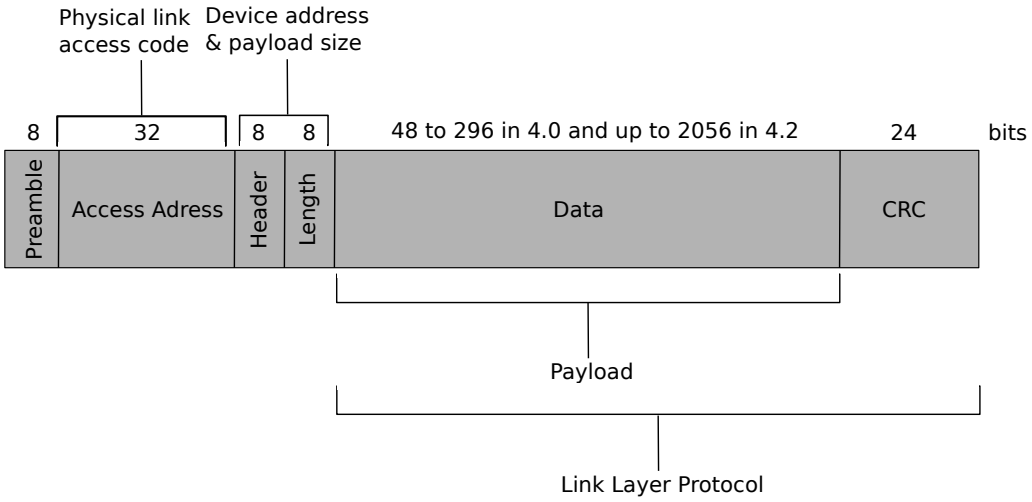


Figure 2.7: BLE Generic Packet Structure

Advertising PDU's has a 16-bit header and a variable payload size. The payload size depends on the type of advertising events as follows:

- ADV_IND: connectable undirected advertising (visible in Figure 2.5). Device can establish a connection after successful advertising packet reception, so a *connection request* from the *initiator* is possible. Also a *scan request* is possible.
- ADV_DIRECT_IND: connectable directed advertising event. A *connection request* from the *initiator* is possible. The PDU is sent to a specific device whose address is known. A *connection request* from the *initiator* is not possible.
- ADV_NONCONN_IND: non-connectable undirected advertising event. Neither *connection request* or *scan request* from the *initiator* are possible. Can be used to broadcast data asynchronously.

  - ADV_SCAN_IND: scannable undirected advertising event. A *scan request* from the *scanner* is possible whereas a *connection request* is not.

  Where an ADV_DIRECT_IND PDU has a payload size of 12 Bytes and the rest of the events PDU may have a payload size that ranges from 6 to a maximum of 37 Bytes.

- **Scan Interval** T_{SI} is the time between two consecutive scanning events. During a scanning event, the scanner will listen to the 3 advertising channels, one by one, waiting for an advertising packet.

- **Scan Window** T_{SW} defines how long a scanner will listen waiting for an advertising packet. When $T_{SW}<T_{SI}$, the scanner will spend some time in low-power mode between the end of the last scanning window and the starting time of the next scanning event, allowing the scanner to save some energy. It shall be $T_{SW}\leq T_{SI}\leq$ 10.24 s. It is allowed to have a $T_{SW}=T_{SI}$ in which case the device will be scanning continuously.

2.2.3.2 In connected mode

- **Connection Interval** T_{CI} is the interval between two connection events on master side, and shall not overlap. T_{CI} is a multiple of 1.25 ms in the range of [7.5 ms, 4 s].

- **Supervision Timeout** T_{ST} defines the maximum time between two received data PDUs before the connection is considered lost. It shall be a multiple of 10 ms in the range of [100 ms, 32 s].

- **Slave latency** defines the number of consecutive connection events that the slave device is not required to listen for the master. It should not cause a supervision timeout and shall be an integer in the range of $[0, \min(T_{ST}/T_{CI}, 500) - 1]$. This allows the slave to sleep much more than the master. (See Figure 2.8)

- **Payload Size** BLE has a maximum payload size of 37 B in v4.0 and 257 B in 4.2.

All energy-affecting parameters according to the BLE specifications are summarized in Table 2.1. Moreover, the total energy consumption of a BLE device depends on the parameters listed above but also on the overall communication pattern. ND is a very energy hungry phase, where the scanner must listen for long periods of time, while the advertiser might have to send

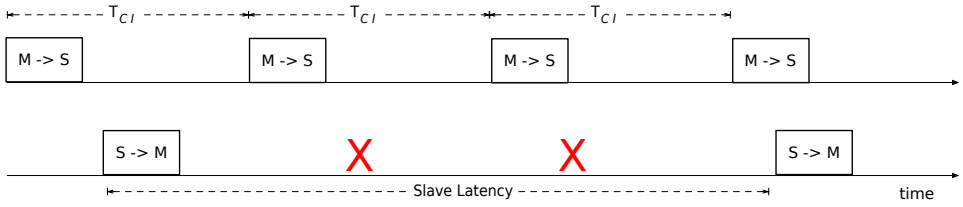


Figure 2.8: BLE Connected Mode with Slave Latency non-zero

a lot of packets until contact is made. The duration of this phase is non-deterministic but we can evaluate average and worst case values based on a suitable model.

Table 2.1: BLE energy-affecting parameters

Name	**Notation**	**Value according to standard**
Scan Interval	T_{SI}	$\leq 10.24\,\text{s}$
Scan Window	T_{SW}	$\leq T_{SI}$
Advertising Interval	T_{AI}	20 (or 100) ms $\leq T_{AI} \leq 10.24\,\text{s}$ integer multiple of 0.625 ms
Advertising Delay	ρ	pseudo-random value in $[0, 10]$ ms
Advertising Event	T_{AE}	$= T_{AI} + \rho$
Adv. Indication interval	T_{AW}	$\leq 10\,\text{ms}$ (low duty cycle)

After ND, both devices establish a connection and remain in CM as long as needed depending on application requirements or user actions. For well chosen ND parameters and a long connection period, the energy spent in ND may become negligible. Existing energy models are discussed in the next section.

2.3 BLE Energy Consumption Models

2.3.1 State of the art analysis

Recently, there have been a few studies on BLE energy modeling: [2] proposes the estimation of CR2032 coin cell battery lifetime for a custom scenario usage based on current consumption measurements on a CC2541 BLE Texas Instrument device in its Salve role during CM. The current consumed during each of the internal radio states of Connection Events are summed up including sleep mode and the current consumption due to MCU processing

of the different layers of BLE protocol stack. This is a quantitative model of the PHY layer and measurements for each different use case are needed. [2] has set a precedent for subsequent energy analysis, since they are based on quantitative models of the PHY Layer of the same type as we explain later. Figures 2.9 and 2.10 show the measurements from [2] on which the lifetime estimation is based. The weakness of this method is that only the slave in CM is taken into account and non-linear behavior of the battery is not considered.

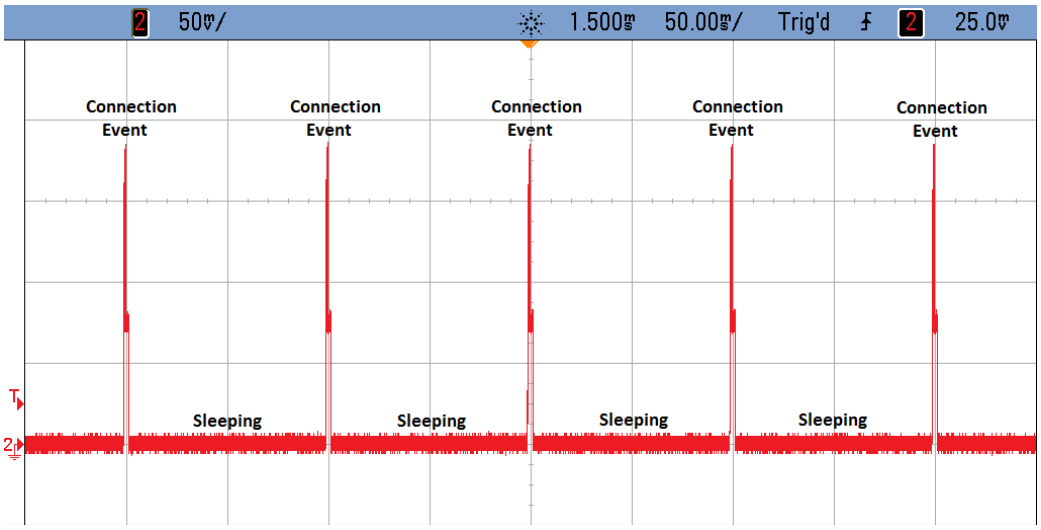


Figure 2.9: CC2541 BLE Texas Instrument device current consumption measurements during Connected Mode (extracted from [2])

Authors in [3] and [4] present a probabilistic analysis to estimate the duration of the ND phase and energy consumption. These models assume that there's an initiator which is scanning periodically on the three channels and an advertising packet is sent at a random starting point with respect to the three consecutive scanning intervals. Physical Layer in both [3] [4] are based on quantitative models that provide with an average current consumption as in [2].

In [3], the time for scanning the three channels is divided into three regions and within each region, the possible advertising starting points are further categorized into different bins which have three possible different sizes as seen in Figure 2.11 (where T_{apk}+t_{ch}= T_{AW},T_{apk} is the time to send an advertising packet which is packet size dependent and T_{apk} is the time between consecutive advertising packets). Then the probability of an advertiser to start in the i_{th} bin equals to the length of all three scanning intervals divided by the length of that bin. If the advertiser begins advertising in the

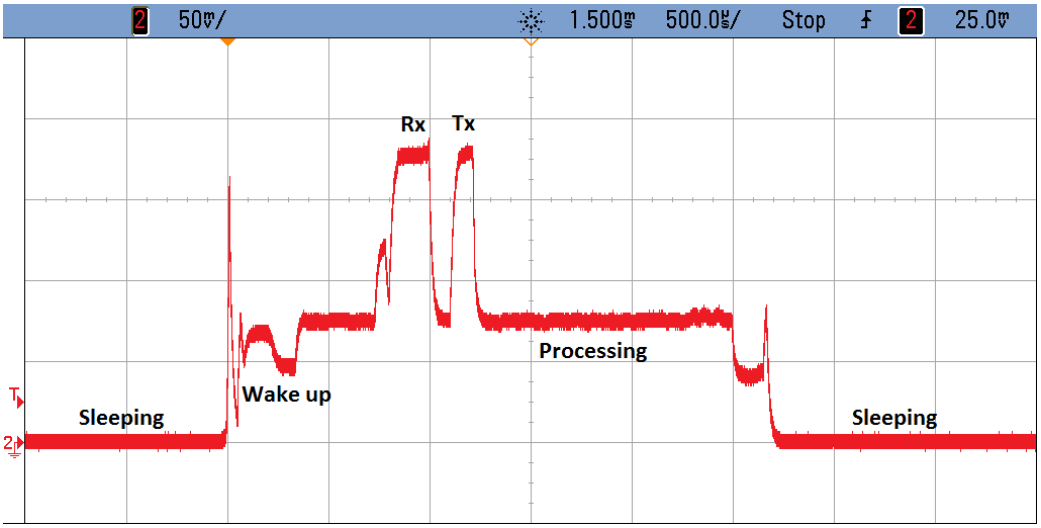


Figure 2.10: CC2541 BLE Texas Instrument device single event current consumption measurements during Connected Mode (extracted from [2])

i_{th} bin within region n ($i = 0, 1, ..., kn + 1, n = 1, 2, 3$), the DL will be equal to $iT_{AI}+A_n$ ($A_n = nA + (n-1)b$), which is the time since the start of the received advertising event until the time when the advertising packet of the corresponding advertising channel is received by the scanner. The energy consumption is given by the energy consumed during T_{AI} multiplied by the number of advertising events that were not received by the scanner plus the energy consumed during the advertising event with a successful reception on the scanner side. Finally an average energy consumption is obtained for a given $T_{SI}-T_{SW}-T_{AI}$ configuration. This model is only valid when $T_{SW}<T_{SI}$.

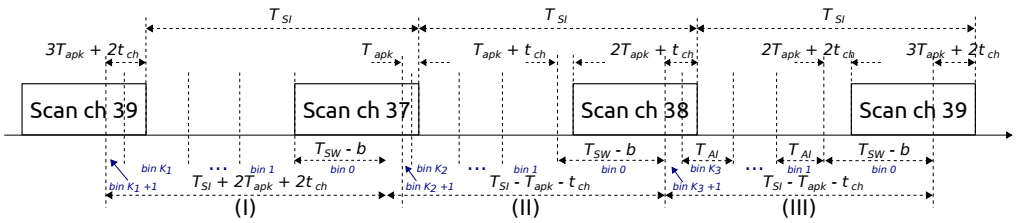


Figure 2.11: Discovery latency and energy consumption model from [3]

Similar as in [3], authors of [4] have estimated advertiser DL based on the starting point of advertising events with respect to scanning events. The authors consider that the scanner starts listening at time $t = 0$ and the advertiser starts advertising after a random delay with a uniform distribution which has been defined as ϕ and is a random delay relative to the beginning

of the first scan event k with a maximum value of $3T_{SI}$.

Moreover, an advertising event may start earlier than the subsequent scan event and still be received successfully by the scanner, but also there are some points within a scan event that lead to a lost advertising packet, this is due to the fact that each advertising event consist of three packets on three different channels, so for estimating the probability of a packet reception, an *effective scanning window* is defined. This window goes from *tearly* and *tlate* and is estimated for each advertising channel as shown in Figure 2.12.

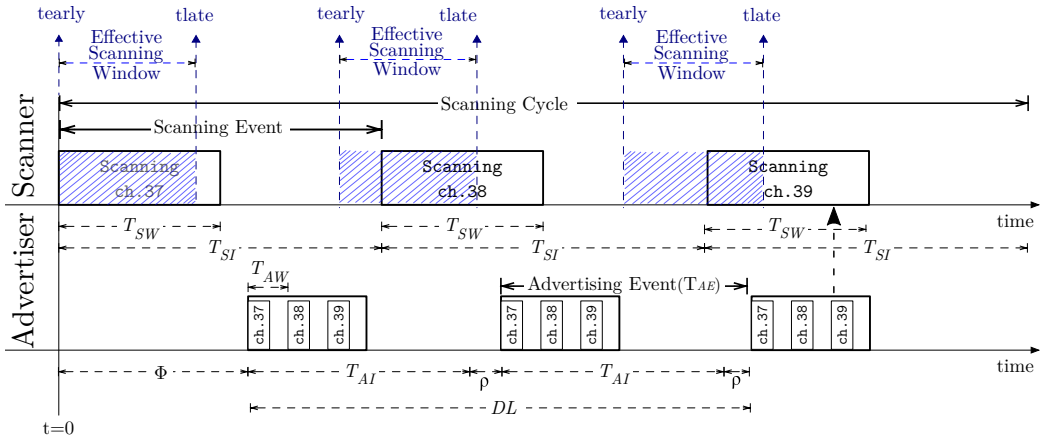


Figure 2.12: Discovery latency and energy consumption model from [4]

The probabilistic model predicts the expected number of advertising events before one of them is received successfully. For a given ϕ, the algorithm calculates the probability P_{hit} for a successful reception of each advertising event n. ρ is modeled as a random variable with uniform distribution. For two consecutive advertising events, an uniform distribution is considered, for three consecutive events a symmetric triangular distribution is assumed, and according to the central limit theorem, for more than three events, a Gaussian distribution is assumed.

The central limit theorem establishes that the Cumulative Distribution Function (CDF) of the sum of independent random variables, converges to a Gaussian CDF as the number of terms grows without limit, even if the original variables themselves are not normally distributed [34]. ρ is modeled as a random variable, having the distribution:

$$f(\rho) = \begin{cases} \frac{1}{10ms} & if \quad 0ms \leq \rho \leq 10ms \\ 0ms & else \end{cases} \tag{2.2}$$

In general, the mean and the variance of a random variable with uniform distribution is defined as $\mu = \frac{a}{2}$ and $\sigma^2 = \frac{a^2}{12}$ respectively [34]. Then as we

said earlier, for more than three events a Gaussian distribution is assumed, accordingly the mean is assumed as $\mu = n5ms$ and standard deviation $\sigma = \sqrt{\frac{n}{12}}10ms$. Moreover, the model in [4], includes all operating modes of the LL for both ND and CM and has been validated in [4] via measurements on a Bluegiga BLE112 device which is based on Texas Instruments CC2540.

In [35], authors estimate the probability of successful reception for each of the advertising channels. Since the model considers the ND used to immediately establish a connection after successful reception of the advertising packet, then the probability that the advertiser has enough time to exchange control packets (scan request and scan response) with the scanner after sending an advertiser packet is estimated. The probability is estimated for each advertising channel. Then probabilities are estimated when the network is composed by m advertisers and n scanners and the expected advertiser DL is calculated.

The model in [36] estimates DL and energy consumption during ND based on the Chinese Remainder Theorem. Their activities are expressed on the discrete-time axis. The time is quantized into a discrete component, called a time slot with the fixed length of Δ and similar to the model in [4], DL is estimated from advertiser's perspective who may start advertising a certain amount of time (in this case time slots) after a time of reference $t = 0$, when the scanner starts scanning. The scanner discovers the advertiser when the first slot of the advertising event falls inside the discovery region. Then the probability of a successful reception is estimated considering possible interference coming from other communication systems as well as possible collisions.

To summarize, models in [4, 36] estimate both advertiser and scanner energy consumption, whereas the models in [2, 3, 35] estimate only advertiser energy consumption, so with these three models, determining a possible trade off between advertiser and scanner energy consumption is not possible. The model from [2] only considers energy consumption during CM, models in [3, 35, 36] consider only ND, whereas the model from [4] takes into account both ND and CM. The model from [36] is the only one that considers interference from radios using other communication protocols and possible collisions. The model in [35] considers collisions due to several BLE devices in a network. All of these models estimate DL and energy consumption from advertiser's perceptive.

Although the energy model proposed in [4] doesn't consider interference or packet collision and is based in single advertiser single scanner communication, it is the most complete in the SotA as it comprises all operating modes of the LL for both ND and CM and have been proven to be very precise via experimental measurements performed in a testbed. Our research is

build upon this model and firstly we will describe the model in further detail in the next section.

2.3.2 Adopted probabilistic model description

In this section we present the adopted model in further detail. Due to its proved accuracy and because it includes all operating modes of the LL for both ND and CM as previously discussed, we have chosen the model from [4] to work on top of it for the BLE performance evaluation. The model estimates the advertiser DL which is the elapsed time between the starting point of the first advertising event, until the starting point of the advertising event with highest probability of successful reception by the scanner (t_{ideal} without considering ρ and t_{real} considering ρ). To calculate the total advertiser DL, the time $t_{advEvent}(ch)$ must be added, which is the time spent until the reception of the packet as shown in Figure 2.13. It depends on the channel over which reception takes place.

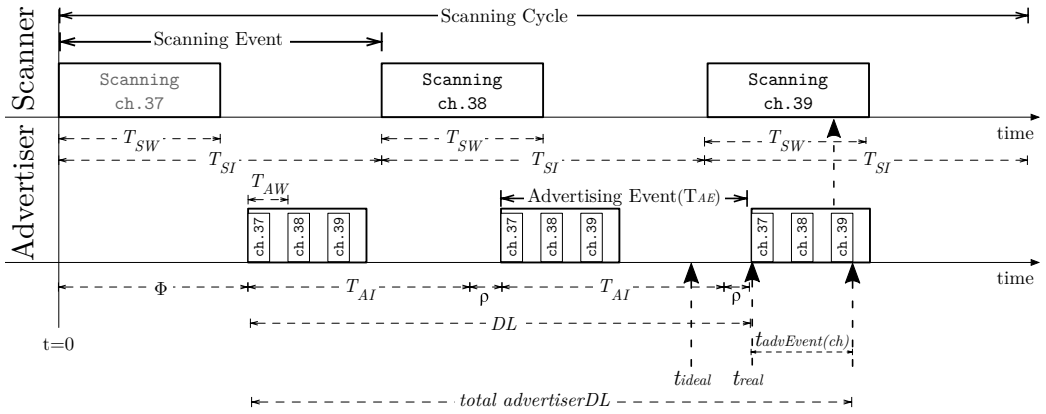


Figure 2.13: Total advertiser discovery latency depending on the advertiser channel over which a successful reception takes place

The model is used to estimate the probability for a successful reception of an advertising packet during ND, considering that an advertiser shall send n packets before a scanner is able to receive it. Given a certain T_{AI}-T_{SI}-T_{SW} configuration, the total amount of time a device spends transmitting/receiving during ND is estimated, until the communication takes place between a single advertiser and a single scanner. Consequently the energy consumed during that time is estimated.

The condition to consider a successful reception is that the starting point of an advertiser event falls into a region within a scanner interval called

effective scanning window as explained in the previous section. The effective scanning window is different for each of the advertising channels and depends on four parameters: scanning interval T_{SI}, scanning window T_{SW}, duration of an advertising packet transmission T_{apk} and the duration of hopping to the next channel t_{ch} (see Figure 2.13, where $T_{AW} = T_{apk} + t_{ch}$). The effective scanning window is estimated as seen in Table 2.2, where $tearly$ is the time with respect to each advertising channel from which, if an advertising event n begins, the packet can be successfully received even if $tearly$ is not within the scanning window. $tlate$ is the time with respect to each advertising channel from which, if an advertising event n begins, the packet can't be successfully received, even if $tlate$ is within the scanning window.

Table 2.2: Effective Scanning Window

Channel	**tearly**	**tlate**	$tsemin(k)$	$tsemax(k)$
37	0	T_{apk}	0	$(k-1)T_{SI} + T_{SW} - tlate$
38	$T_{apk} + t_{ch}$	$2T_{apk} + t_{ch}$	$(k-1)T_{SI} - tearly$	$(k-1)T_{SI} + T_{SW} - tlate$
39	$2T_{apk} + 2t_{ch}$	$3T_{apk} + 2t_{ch}$	$(k-1)T_{SI} - tearly$	$(k-1)T_{SI} + T_{SW} - tlate$

For each possible ϕ, the probability of successful reception P_{hit} is estimated for every advertising event n, considering that all previous events were not received by the scanner. ϕ increases from 0 to $3T_{SI}$ in steps Δ whose length affects the accuracy of the estimated DL as it'll be further discussed in Chapter 3. The random delay ρ is modeled as a random variable with uniform distribution.

The probability of successful reception P_{hit} is calculated according to line 10 of Algorithm 1, where n and k are the current evaluated advertiser and scanner events respectively.

The n_{th} and k_{th} events are considered according to lines 13 and 6 of Algorithm 1. $t_{advEvnt}(ch)$ is the duration of an advertising event successfully received by the scanner on the current channel. The current channel is obtained according to line 9 of Algorithm 1. P_{hit} gives the approximate probability of an advertising event having started between $tsemin(k)$ and $tsemax(k)$, which are the minimum and maximum values of the effective scanning window for the scan event k and they are defined in Table 2.2.

For calculating the probability P_{hit} of an advertising event being successfully received, the Probability Density Function (PDF) of the start of an advertising event over time t (which depends on both n and ϕ) is required. The shape of the distribution depends on n and is assumed as mentioned in Section 2.3.1: for two consecutive advertising events, an uniform distribution

Algorithm 1 BLE Average Advertiser Discovery Latency

1: $DL_{adv} \leftarrow 0$
2: **for** $\phi = 0$ to $3T_{SI}$ step Δ **do**
3: $n \leftarrow 0$, $DL_{exp} \leftarrow 0$, $P_{hit} \leftarrow 0$, $PcM \leftarrow 1$, $ch \leftarrow 37$
4: **while** $1 - PcM \leq \varepsilon$ **do**
5: $t_{ideal} \leftarrow \phi + nT_{AI}$, $P_{hit} \leftarrow 0$
6: $K_{min} = \lfloor \frac{t_{ideal}}{T_{SI}} \rfloor$, $K_{max} = \lfloor \frac{t_{ideal}+n5ms}{T_{SI}} \rfloor$
7: **for** $K = K_{min}$ to $K = K_{max}$ **do**
8: $ch \leftarrow mod(j, 3)$
9: $(tsemin(k), tsemax(k)) \leftarrow getInterval(ch))$
10: $P_{hit} \leftarrow P_{hit} + pk(t_{real}, n, \rho\sqrt{\frac{n}{12}}, t_{ideal}, tsemin(k), tsemax(k), \rho)$
11: $DL_{exp} \leftarrow DL_{exp} + pk.PcM(n(T_{AI} + 5ms) + t_{advEvent}(ch))$
12: **end for**
13: $PcM \leftarrow PcM(1 - P_{hit})$, $n \leftarrow n + 1$
14: **end while**
15: $DL_{adv} \leftarrow DL_{adv} + DL_{exp}$
16: **end for**
17: $\overline{DL_{adv}} \leftarrow \frac{DL_{adv}}{\frac{3T_{SI}}{\Delta}}$

is considered, for three consecutive events a symmetric triangular distribution is assumed, and according to the central limit theorem, for more than three events, a Gaussian distribution is assumed. Then the approximate distribution for the time $t(n, \phi)$ is as shown in Algorithm 2.

Continuing with the details of Algorithm 1, PcM is the probability that the n_{th} advertising event does not lead to a successful reception (cumulative miss probability), it is calculates in line 13 of Algorithm 1. With increasing values of n, the probability that one of the advertising events is received successfully grows. Thus, PcM shrinks with growing n and the algorithm finishes if $(1-PcM)$ is smaller than a lower bound $\epsilon = 0.9999$. The expected DL for a given ϕ offset is calculated according to line 11 of Algorithm 1. It was considered for the expected value of ρ to be $5ms$ and the error is neglected. Then the average DL for a given $T_{SI}-T_{SW}-T_{AI}$ configuration is obtained by integrating the expected DL results over all possible values of ϕ. A numerical integration is then performed by multiplying the results with Δ and computing the sum of these values.

To estimate the energy consumed during ND, all parameter values are presented in terms of electric current in amperes A and electric charge in coulombs C. From these values, the power and energy consumption is easily obtained for a given supply voltage. During a ND or CM event, the current consumed by the device is given by a sequence of different n phases from which consumed charge is calculated by Eq. 2.3.

$$Q_n = t_n I_n \tag{2.3}$$

Then having the charge consumed during each phase, the charge con-

Algorithm 2 CDF of the starting time t for the n_{th} advertising event

if $n = 1$ **then**
  if $t < t_{ideal}$ **then**
    $F(t) = 0$
  else
    $F(t) = 1$
  end if
else if $n = 2$ **then**
  if $t_{ideal} < t < t_{ideal} + \rho$ **then**
    $F(t) = \frac{t - t_{ideal}}{\rho}$
  else if $t < t_{ideal}$ **then**
    $F(t) = 0$
  else
    $F(t) = 1$
  end if
else if $n = 3$ **then**
  if $t \geq t_{ideal}$ **then**
    if $t < t_{ideal} + \rho$ **then**
      $F(t) = \frac{(t - t_{ideal})^2}{2\rho^2}$
    else if $t < t_{ideal} + 2\rho$ **then**
      $F(t) = \frac{1 - (t_{ideal} + 2\rho - t)^2}{2\rho^2}$
    else
      $F(t) = 1$
    end if
  else
    $F(t) = 0$
  end if
else
  $F(t) = \Phi(\frac{t - t_{real}}{\sigma})$
end if

sumed for an event with n phases can be computed as in Eq. 2.4. The amount of phases as it will be later described, varies from one manufacturer to another. Then the total charge consumed during ND or CM will be Q_E multiplied by the amount of events, including the time spent in sleep mode.

$$Q_E = \sum_{i=1}^{n} Q_i \tag{2.4}$$

2.3.3 Probabilistic Model Limitations

According to the BLE specifications, before establishing a connection, the scanner must first listen to the advertising channel during the ND in order to synchronize with the advertiser. The scanner/master is the initiator and is responsible for establishing a connection (CM) with one or multiple advertiser/slave. Once in CM, the slave can have only one connection and the master can have one or multiple connections. So, from BLE communication point of view, the scanner/master has the central device role whereas the advertiser has the peripheral role which is typically at the sensor side.

In some cases, ND can be activated when the device on sensor side detects activity, in some others it is triggered by the user by placing the battery or by pressing a button, or just by presence meaning that one of the two devices must remain active for longer periods of time, thus consuming a considerable amount of energy; but depending on each device role, the constrained DL or energy may be at the scanner side, contrary to what has been evaluated up to these days. Although ND communication is mainly used for device synchronization, it is feasible to use ND for data transmission purposes. If the advertiser is intended to send data within the advertising packets, it is possible for it to send data to multiple scanner. Proximity type of applications are an example where the advertiser who is configured to broadcast data, will advertise periodically and multiple scanner in range can receive the data. As a connection state is not required; from the application point of view, the advertiser and the scanner play the central and the peripheral role respectively, and furthermore, the scanner gets to be at the user side, meaning that for such use case, DL criticality is at the scanner side. This shows that efforts to improve performance should be focused on one side or another, depending on its inherent critical conditions which is in great measure imposed by the application.

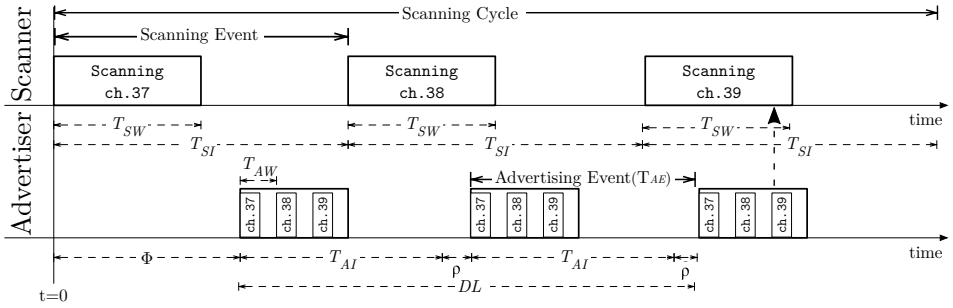
DL has been traditionally modeled assuming that the advertiser starts at a random time within one scanning cycle, where the scanner is listening to the three different advertising channels. The model from [4] considers that the advertiser starts at a given phase offset called ϕ after $t = 0$, moment at which the scanner starts scanning on channel 37 as shown in Figure 2.14(a). In order to determine scanner performance when constrained DL and energy consumption is at the scanner side, there is a need for new or improved models. Therefore, an extension and improvement of the model from [4] is presented in Section 2.3.4.

In addition, the ability to optimize the parameters that guarantee optimal performance with these models is possible but limited. It is still necessary to develop mechanisms that allows obtaining an approximate of the appropriate parameters that adapt to a specific application requirement in a fast and efficient way. A method for the optimization of BLE configuration parameters during ND is presented later in Chapter 3.

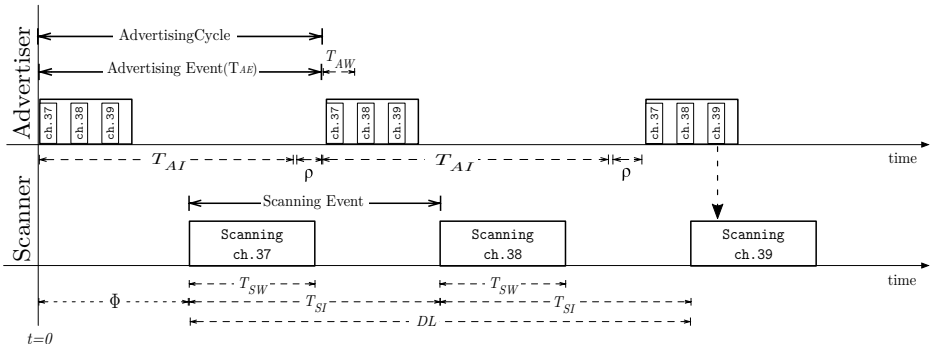
2.3.4 Improvement of the Probabilistic Energy Model

In this Section we present an improvement of the adopted probabilistic energy model which translates into an extension of Algorithm 1, which was explained in the previous Section. DL is estimated from the scanner perspective when playing the peripheral role at application level. The scanner starts scanning

after a given phase offset ϕ relative to time $t = 0$ when the advertiser starts advertising. Analogously to the model in [4], ϕ is delimited by one entire advertiser cycle. The behavior of the system when the scanner plays the peripheral role at the application level is depicted in Figure 2.14(b).



(a) The Scanner is available waiting for an advertising packet



(b) The Advertiser is available waiting to be discovered by the scanner

Figure 2.14: Discovery latency and energy consumption model from advertiser perspective 2.14(a) and from scanner perspective 2.14(b)

Similar as in Algorithm 1, the probability of successful reception of the advertising event n by the scanner during the scan event k, is estimated according to Algorithm 2. As ϕ is delimited by one entire advertiser cycle, we have considered that the advertiser starts advertising at a maximum of $T_{AI}+\rho$ time before $t = 0$, which is a full advertising cycle as shown in Figure 2.14(b) and the maximum value of ρ is considered. Before $t = 0$ any advertising event n has 0 probabilities of being received by the scanner, so PcM is increased to its maximum value and the next event n is evaluated.

All possible scanning events k for which a successful reception of a given advertising event n could happen, are considered according to line 9 of Algorithm 3 which is visible in the next page. The probability of successful reception is calculated in line 12 of Algorithm 3, where $tsemin(k)$ and $tsemax(k)$

Algorithm 3 BLE Average Scanner Discovery Latency

1: **for** $\phi = -T_{AI} + 10ms$ to 0 step Δ **do**
2: $n = 1$, $PcM = 1$, $\varepsilon = 0.9999$
3: **while** $1 - PcM \leq \varepsilon$ **do**
4: $t_{ideal} = \phi + (n-1)T_{AI}$, $t_{real} = t_{ideal} + (n-1)\frac{\rho}{2}$
5: **if** $t_{ideal} < 0$ **then**
6: $PcM = 1$, $n = n + 1$
7: *continue*
8: **end if**
9: $K_{min} = \lfloor \frac{t_{ideal}}{T_{SI}} \rfloor$, $K_{max} = \lceil \frac{t_{ideal}+n5ms}{T_{SI}} \rceil$, $P_{hit} = 0$
10: **for** $K = K_{min}$ to $K = K_{max}$ **do**
11: $tsemin(k, T_{SI}, T_{SW}, Tapk, tch)$, $tsemax(k, T_{SI}, T_{SW}, Tapk, tch)$
12: $p_k(t_{real}, n, \rho\sqrt{\frac{n}{12}}, t_{ideal}, tsemin(k), tsemax(k), \rho)$
13: $P_{hit} = P_{hit} + p_k$
14: **end for**
15: $PcM = PcM(1 - P_{hit})$, $n = n + 1$
16: **end while**
17: **end for**
18: $\overline{adv_{t_{real}}} = t_{real}(max(p_k)) + abs(\phi)$, $scan_{t_{real}} = t_{real}(max(p_k))$
19: $\overline{advDL} = mean(adv_{t_{real}})$, $\overline{scanDL} = mean(scan_{t_{real}})$

depend on the current advertising channel and is obtained as shown in line 11. As in the original Algorithm, PcM is the probability that the n_{th} advertising events don't lead to a successful reception (cumulative miss probability) and the algorithm finishes if $(1 - PcM)$ is smaller than a lower bound $\epsilon = 0.9999$. Unlike Algorithm 1, the phase offset ϕ is within a negative range meaning that the probability is evaluated for all possible cases when the advertiser starts advertising before the scanner starts.

For both Algorithm 1 and 3, the average DL has been computed such that all data related a specific event can be accessed once the algorithm has been executed as seen in line 18 of Algorithm 3, where $adv_{t_{real}}$ is $t_{real}(max(p_k)) - \phi$. The foregoing aims to evaluate not only average but also worst case, as discussed in Chapter 3, so we don't make use of d_{adv} and d_{exp} variables shown in algorithm 1. Finally, average DL is calculated as in line 19.

Algorithm 3 can be used for both cases: when analyzing BLE performance from advertiser or scanner perspective. The difference for this two cases lies in the definition of ϕ. When analyzing performance from advertiser perspective we use $\phi = [0, 3T_{SI}]$ and when analyzing performance from scanner perspective $\phi = [-(T_{AI} + 10ms), 0]$ accordingly with Figure 2.14. For the sake of simplicity, henceforth when we refer to the performance from advertiser point of view we will refer to Algorithm 1, and to Algorithm 3 when analyzing from scanner perspective. On the other hand, as it has been stated in [4], the precision of the model lies in the choice of Δ (ϕ step), we have adjusted Δ for Algorithm 1 according to Eq. 2.5, and for Algorithm 3 according to Eq. 2.6.

$$\Delta = \begin{cases} 1ms & if \quad 30ms \geq T_{SI} \leq 100ms \\ 5ms & if \quad 100ms < T_{SI} < 640ms \\ 93,6ms & if \quad T_{SI} \geq 640ms \end{cases} \tag{2.5}$$

$$\Delta = \begin{cases} 10\mu s & if \quad 20ms \geq T_{AI} \leq T_{SW} + \rho \\ 19.9\mu s & if \quad T_{SW} + \rho < T_{SI} < 6s \\ 29.9\mu s & if \quad T_{SI} \geq 6s \end{cases} \tag{2.6}$$

2.3.5 DL and Energy Model Comparison

We have implemented the improved model from previous Section in a Matlab OOP based simulator which is presented in Section 2.5. We set up a test scenario where two devices exchange packets during ND. The advertising packets contain a payload size of 37 Bytes in total: 6 Bytes of device address and 31 Bytes of effective Payload. Additionally, packets contain a 10 Bytes header, for a total advertising packet size of 47 Bytes which is the longest packet on BLE v4.0 & v4.1. Here we consider ND until the reception of the advertising packet on the scanner side, so no scan request or connection request has been considered.

Figure 2.15 shows the average DL for $T_{SI} = 3.2s$ and $T_{SW} = 2.56s$ for one connection during ND, where the difference of the results from both Algorithms can be appreciated.

Figure 2.16 shows the corresponding average energy consumption for $T_{SI} = 3.2s$ and $T_{SW} = 2.56s$. We can see the difference when evaluating from different perspectives. Advertising packet reception occurs faster when the scanner is available at the moment the advertiser starts advertising, however a much better DL/energy consumption trade-off between advertiser and scanner is obtained, when it is the advertiser who is available waiting to be discovered by the scanner. On the other hand, for small values of T_{AI}, the user experience is similar for both cases as DL of the advertiser with Algorithm 1, and DL of the scanner with Algorithm 3, remain very close to each other. Our results shows the impact of evaluating BLE from the different perspectives, thus our modeling methodology provides with more realistic DL and energy consumption estimation, as real conditions from the application are taking into account.

2.4 Battery Lifetime Estimation

In this Section we present an overview of the state of the art of battery lifetime modeling and a new battery lifetime model.

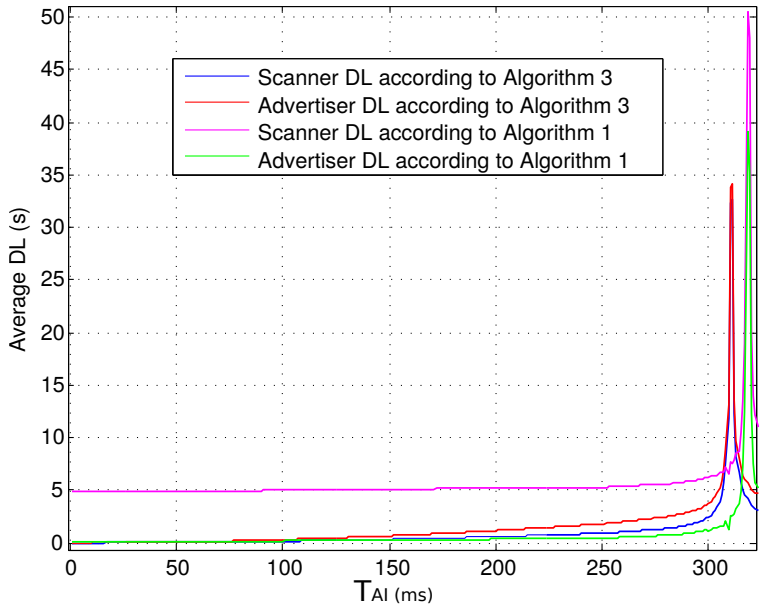


Figure 2.15: Average Discovery latency comparison between Algorithm 1 and Algorithm 3 for $T_{SI} = 3.2s$ and $T_{SW} = 2.56s$

2.4.1 Understanding the Battery Lifetime Estimation Challenge

A battery is a device that converts chemical energy into electrical energy. To familiarize the reader with the battery lifetime estimation problem, first we present some of the variables used to characterize battery operating conditions which are given by the manufacturer and commonly used to estimate battery lifetime:

- **Cut-off Voltage:** the minimum allowable output voltage. It is determined by complex electrochemical processes that depend on the timing and intensity of the load during discharge. This voltage generally defines the "empty" state of the battery. When the battery is no longer able to maintain a sufficiently high output voltage in response to the load presented by the device, it will fail to operate correctly.

- **Capacity or Nominal Capacity:** the total Ah available when the battery is discharged at a certain discharge current (in A). It is also referred to as rated capacity.

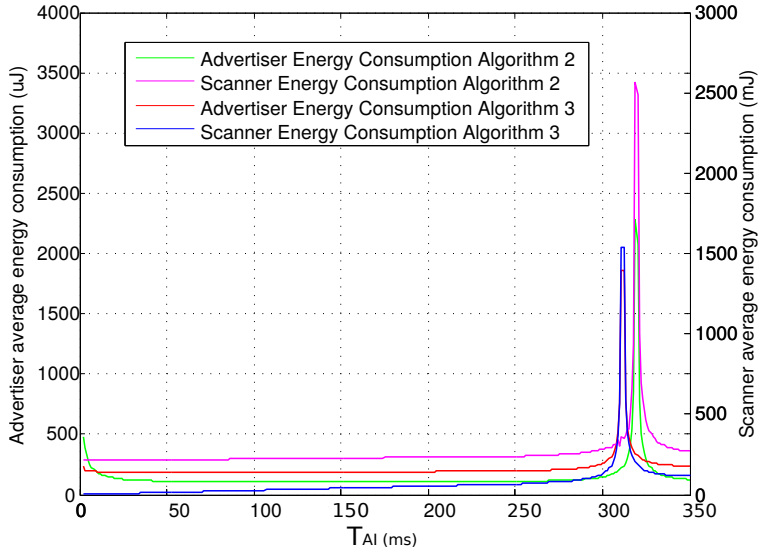


Figure 2.16: Average energy consumption comparison between Algorithm 1 and Algorithm 3 for $T_{SI} = 3.2s$ and $T_{SW} = 2.56s$

- **Nominal Voltage:** is the reference voltage in Volts (V) of the battery, sometimes referred to as average voltage of the battery. As the battery is used, the voltage will drop lower and lower until the minimum is reached (the cut-off voltage). The nominal voltage is the output voltage that the battery will maintain during the majority of the battery life. Once the battery reaches the cut-off voltage, cut-off circuitry disconnects the battery.

- **Continuous Standard Load:** is the current in A (continuously supplied) that will provide the expected capacity given by the manufacturer. It is also referred to as rated current.

- **State of Charge (SOC):** an expression of the present battery capacity as a percentage of its maximum capacity. SOC is generally calculated using current integration to determine the change in battery capacity over time.

- **Percentage of Self Discharge per Year:** is an internal chemical reaction in which the capacity of the battery is reduced when stored at room temperature ($20°$–$30°C$), without any connection between the electrodes. These chemical reactions occur within the cell even when no load is applied. Typically, manufacturers provide the self discharge of

the battery in terms of a percentage (%) of the original battery capacity per year.

- **Operating Temperature:** ($^{\circ}C$) manufacturers specify the ambient temperature range over which the battery will operate efficiently and safely

Knowing the current discharge of the battery and the total capacity in Ah, one can compute the theoretical lifetime of the battery using Eq. 2.7,

$$L = \frac{C}{I} \tag{2.7}$$

Where $L(h)$ is the battery lifetime, $C(Ah)$ is the rated maximum battery capacity and $I(A)$ is the discharge current. The complexity of the battery lifetime estimation lies in the fact that the total capacity of the battery is not a fixed parameter. It changes depending on the discharge rate given by a specific load. The nominal capacity is measured by the manufacturer by discharging the battery at the indicated continuous standard load. Manufacturers provide several curves describing the battery behavior under different conditions: capacity vs. load, voltage vs. load, voltage vs. lifetime, among others; all of them for different operating temperatures. The behavior of the battery is non-linear, so Eq. 2.7 can not be used under any given condition. According to this equation, the lower the discharge current the higher the lifetime, but looking at the curves given by the manufacturer it can be shown that for very small loads, the battery capacity does not exceed from a maximum value and therefore nor does the lifetime. In other words, below a certain discharge rate the battery capacity saturates.

2.4.2 Battery Lifetime Models: State of the Art

In practice, a sensor node may be equipped to read the battery voltage, allowing the prediction of the battery remaining energy. This is valid only for a node powered by an alkaline battery, because the battery voltage varies proportionally to its remaining energy. However, this technique is not useful in a node powered by a lithium battery like coin cell type, as the lithium battery maintains an almost constant voltage during its lifetime [37]. In consequence, several studies have been carried out in order to estimate lithium battery lifetime. In [38], the authors characterize a lithium battery by carefully measuring the battery's current and voltage output for the duration of the battery lifetime while an embedded board consumes power from the battery. They obtain the battery capacity under different load profiles until the battery's cut-off voltage is reached, then the effective capacity of the

battery is computed by monitoring how much current is consumed from the battery. In other words, the authors have characterized the capacity of the battery taking into account the effect of battery discharge rate on the maximum battery capacity, in order to accurately estimate the battery lifetime. The authors point out that the effective battery capacity may only be found through careful measurements. They do not provide lifetime estimations in this work.

The authors in [39] start from a known fact: if the battery is drained with a current greater than the rated current, the effective battery capacity is diminished and therefore its lifetime too. The authors use the battery efficiency to estimate its lifetime, being the battery efficiency the ratio of the effective battery capacity to its rated capacity on a per-cycle basis. The authors calculate the amount of energy consumed from the battery during n cycles, where each cycle is defined based on a time constant τ (set by the authors at $1ms$), and the duration of a processor cycle [39]. For estimating the energy consumed, they compute the average current drawn from the battery during n number of cycles. At each cycle n, they estimate the discharge current ratio which is the ratio of the average current drawn from the battery (calculated from cycle $n-1$) to the rated current. Having the discharge current ratio or also called discharge rate, and using the "discharge rate vs battery efficiency" curve provided by the manufacturer; the corresponding battery efficiency at each n cycle is extracted. At lower efficiency, lower energy remaining in the battery and therefore less remaining lifetime after n cycles. The authors assume that the remaining battery lifetime is proportional to its remaining efficiency. Therefore, if a battery with a rated capacity of $100mAh$, at 100% of efficiency has a lifetime of $1h$ drained at a rate of $100mA$; then at 60% of efficiency, it has a remaining lifetime of $36min$. Then if the battery is drained at a ratio 3 times higher ($300mA$), then the remaining battery lifetime at 60% of its efficiency is 3 times lower, that is $12min$.

Authors in [37], mathematically modeled the battery lifetime and validated it by comparing model results with lifetime measurement in a prototype using different duty cycles. The model is related to the physical characteristics of the battery and requires two parameters, α and β which are the total capacity in the battery when it is fully charged and the rate at which the active charges are filed at the electrode surface respectively. These parameters are obtained by discharging the battery to its cut-off voltage. Having obtained this two parameters, the energy consumed during a certain period of time can be estimated based on two effects: **rate capacity**, the lifetime decreases if the battery is discharged with a current greater than the given continuous standard load of the battery (similar as in [39]), and **recovery**

effect, it is possible to recover some of the battery energy if it is discharged with small currents separated by idle periods. The model recursively approximates the node's remaining energy after n periods of size Δ, based on the computed energy value at $n-1$. Finally, the battery is considered depleted when α less the consumed energy equals 0, and the remaining lifetime is estimated in terms of n periods. The model is implemented in a sensor node to predict the remaining energy in real time.

In [40], the study is focused on measuring the capacity that has been consumed from the battery when the cut-off voltage is reached, similar as in [38]. The consumed capacity is called the *rate dependent capacity*. A testbed hardware measures the battery output voltage which is the determinant of device lifetime. Authors measure battery lifetime under various continuous loads (100% of duty cycle) to determine the rate dependent capacities of the battery (integral of current over time), then these capacities are used to predict the battery lifetime using the same loads but at various duty cycles (less than 100% of duty cycle). The authors compare predicted and measured lifetime and they observed very low accuracy on the predicted lifetime for low duty cycles.

Authors in [41, 42], perform an indirect measurement of the battery SOC in order to estimate the remaining energy. In both studies, the output voltage and output current of the battery is carefully measured in order to estimate its effective capacity (integral of current over time). In [41], the objective is to accurately measure the SOC of the battery in order to avoid an overcharge or a deeply discharge that can damage the battery. In [42], the objective is to accurately measure the SOC of the battery in order to estimate the battery remaining energy. Based on the battery remaining energy, there will be a minimum amount of sensor nodes within a network that should be in the active mode when providing a requested service. Neither the authors in [41] or [42] provide remaining battery lifetime estimation.

Due to the non-linear behavior of the batteries, it is very difficult to obtain a good approximation of the lifetime without experimental or empirical data. The electrochemical models as presented in [43, 44, 45, 46], are the most accurate for a prediction of battery lifetime and very close to the experimental data, but this kind of models involve a high computational complexity and require an in-depth knowledge of the battery chemistry.

In summary, there are several methods in the literature to estimate battery lifetime. Model-based methods are very common, they consist of algorithms used to estimate the states of a battery from its measured parameters such as voltage and current or from empirical data given by the manufacturer. As stated previously, the most accurate methods are based on electrochemical models, however due to the need for an in-depth knowledge of the battery

chemistry they are out of the scope of our work. We focus on model-based methods such as in [37]-[42]. We have reviewed some of the researches done since early 2001 to the recent 2015. We can see that in general, the methodology hasn't changed a lot: researches indirectly measure battery lifetime or remaining energy by directly measuring the current drained from the battery. Authors in [37, 38, 41, 42, 40] use the battery capacity parameter to predict either its lifetime or its remaining energy by computing the consumed battery capacity via numerical integration of the consumed current; whereas authors in [39], use the battery efficiency parameter to estimate its lifetime. The efficiency is obtained via linear interpolation between the points of a curve given by the manufacturer.

Among the main drawbacks of these methods we can name: In [38] authors conclude that the effective battery capacity can only be determined via careful measurements, making necessary to perform measurements for every different use case, thus increasing the design time considerably. In [39], the authors assume that battery lifetime is linearly proportional to the battery efficiency which was not proven, additionally we have seen that the "discharge rate vs battery efficiency" curve is not frequently provided by the manufacturer nowadays for the most common commercially available coin cell batteries. In [37] authors use a mix of mathematical and physical models which is intended to be implemented on the node and has high computational complexity, which makes it not very good fit for our purpose. The implemented method by authors in [40] is not accurate for nodes running with low duty cycles and the range of duty cycles for which the method is accurate is not specified. In [41, 42], the initial SOC of the battery must be known to obtain accurate results which depends on the aging, self-discharge and previous use of the battery. These parameters may be unknown, thus introducing high inaccuracy in the results. The models are also intended to be executed on the nodes in real time.

2.4.3 Battery Lifetime Estimation: Our Proposal

In order to overcome the difficulties discussed in the previous section and to provide with fast but realistic lifetime estimation we propose a methodology that can be used to characterize the battery capacity analytically with empirical data provided by the manufacturer, so that it can be applied to any lithium battery and for any use case. Rather than implementing the model in a sensor node, we want to provide with a model that can predict the overall battery lifetime in order to determine the impact in battery lifetime of our proposed energy optimizations which are presented later.

We estimate the battery lifetime based on the data provided in the

datasheet of a Panasonic CR2032 coin cell battery with a nominal capacity of 225 mAh for a continuous load of 0.2 mA, the continuous standard load or rated current given by the manufacturer is the operating point at which maximum battery capacity is reached. The datasheet includes the "Capacity vs. Load Resistance" curve shown in Figure 2.17, we use it for our lifetime estimation method and we explain it throughout this section. From Figure 2.17 we can see that the operating point can be lower for very low temperatures. Battery lifetime results presented all along our work are based on the 20°C curve shown in Figure 2.17.

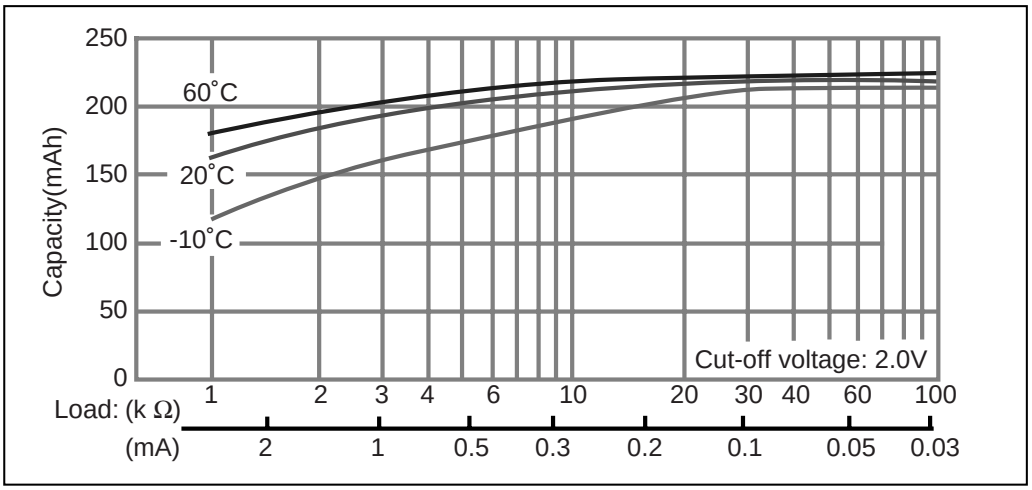


Figure 2.17: Battery Capacity vs Load Resistance Curve from a Panasonic CR2032 Datasheet

A wireless application subjects the battery to different conditions, the radio circuitry of a typical BLE device can draw anywhere from 1 μA in sleep mode to 17.5 mA during Rx/Tx, which far exceeds the rated drain current condition for which the battery capacity is given in the battery datasheet. The impact of this variations for a typical coin cell battery supporting typical WSN applications has been previously evaluated in [47], from which we can assume that, as long as the highest current drain is less than 30 mA and the average current is less than the rated drain current specified on the data-sheet, the battery capacity will remain close to the nominal value C_r, otherwise the battery capacity must be recalculated.

We recalculate the battery capacity for average current higher than the nominal value based on Peukert's law. It establishes that if the battery is discharged at a faster rate than the nominal value given by the manufacturer, the delivered capacity is lower than the rated one. The actual time

to discharge the battery in hours according to Peukert's law is given by Eq. 2.8.

$$t = H \left(\frac{C_r}{IH} \right)^k \tag{2.8}$$

Where H is the rated discharge time (in hours), C_r is the nominal battery capacity, I is the actual discharge rate drained from the battery, k is the Peukert's constant and t is the total lifetime of the battery that we refer as to L in the rest of our work. For example, a battery with a rated capacity of $100mAh$ and a rated current of $100mA$ given by the manufacturer, the battery will be fully discharged in $1h$. But if the battery is drained at the rate of $200mA$ the total battery capacity will less, one may think that the lifetime would be reduced by half as the discharge rate is twice higher. However, due to the non-linear behavior of the battery, the total lifetime will be actually less than $30min$. Therefore, according to Eq. 2.8 and if for example the battery have a Peukert's constant $k = 1.1$, it has a lifetime of $27.9min$. Knowing the lifetime of the battery given the actual discharge rate, we calculate the effective battery capacity by substitution of Eq. 2.7 in Eq. 2.8 as shown in Eq. 2.9.

$$C = \left(H. \left(\frac{C_r}{I.H} \right)^k \right) I \tag{2.9}$$

Peukert's law describes how the available battery capacity changes according to a specific discharge rate [48]. The Peukert's constant k is an indicator of the battery performance. The higher k, the less is the remaining battery capacity. The Peukert's constant of a battery is determined empirically. However, if the capacity is listed for two discharge rates, the Peukert exponent can be determined algebraically. We use the "Capacity *vs.* Load" given by the manufacturer to calculate the Peukert's constant k. Let L_1 and L_2 be the corresponding discharge times for two discharge rates given from the curve, C_1 and C_2 are its respective capacities, L_1 and L_2 are calculated using Eq. 2.7. The Peukert's constant can be then calculated according to Eq. 2.10 as follows:

$$k = \frac{\log \left(\frac{L_2}{L_1} \right)}{\log \left(\frac{C_1}{L_1} \right) - \log \left(\frac{C_2}{L_2} \right)} \tag{2.10}$$

We estimate a Peukert's constant of 1.1 for the Panasonic CR2032. As we said previously, as long as the drained current is not higher than the rated

current I_r, then the battery will be able to deliver its maximum capacity in which case we assume $C = C_r$ for our calculations, where C is the effective battery capacity and C_r is the rated battery capacity given by the manufacturer. Furthermore, if the current drained from the battery is higher than the rated one, we estimate the effective battery capacity according to Eq. 2.9. On the other hand, Peukert's law does not take into account the battery self-discharge current. Lifetime is first estimated according to Eq. 2.11, then we consider 1% of self discharge SD per year as shown in Eq. 2.12. In addition, a real maximum lifetime for typical WSN applications of ≈15 years has been estimated in previous studies [49], based on this we assume no more than 20% of battery capacity C_{SL} is left after 10 years of battery usage due to the consumption pattern (I) seen in our simulations (we refer the reader to Chapter 3), so the total battery lifetime L_t is given by Eq. 2.13.

$$L = \frac{C}{I} \tag{2.11}$$

$$L_{SD} = \frac{C(1-SD)^L}{I} \tag{2.12}$$

$$L_t = 10 + L(C_{SL}) \tag{2.13}$$

The flowchart shown in Figure 2.18 summarizes our model as follows:

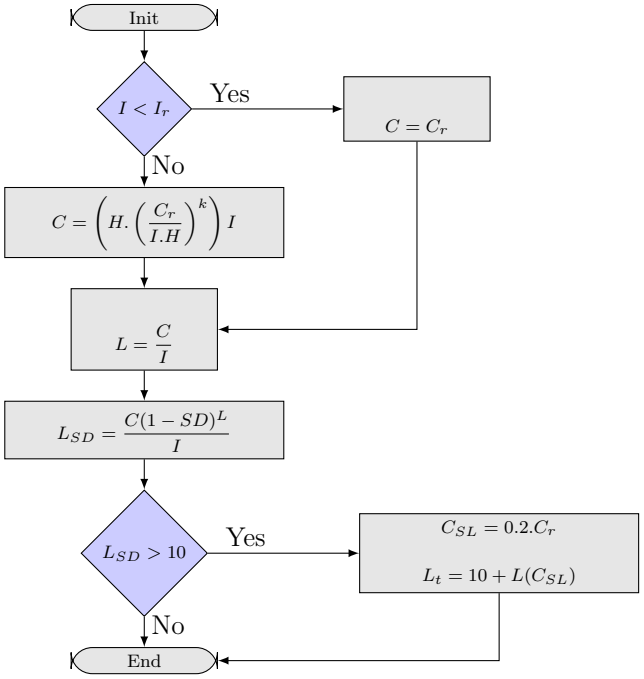


Figure 2.18: Battery Lifetime Estimation Flowchart

As a weakness, our model does not take into account temperature effects on battery capacity and recovery effects after periods in sleep mode. However, other temperatures different from 20°C can be consider from the curve in Figure 2.17, from which constant k must be calculated and battery lifetime can be estimated following the previous exposed methodology.

2.5 Matlab Based Simulator Architecture

To provide the required easy to extend and adaptive environment, Matlab OOP paradigm has been chosen. The architecture is divided in Classes which mainly result from logical separation due to their tasks. The implemented Classes with their main tasks and parameters are discussed below:

- **BLE** contains all parameters relative to the BLE protocol such as maximum payload size, throughput, header size, connection request and connection terminate packet size and time to be transmitted, among others. These parameters are used from different parts of the code and BLE is the only protocol supported by the simulator. The parameter `max_packet_size` depends on BLE Core version, the throughput is $1Mbps$ which is defined by the standard, so it takes 8μs to transmit one bit. Header size is 10 Bytes for all BLE versions and all modes of communication. Connection request (`con_req`) and connection terminate (`con_ter`) parameters have a length of 44 and 11 Bytes respectively. The variable `t_empty` is the time to transmit or receive an empty packet which is 80μs. and holds true for all BLE Core version. The BLE Class is presented in Matlab as follows:

MAX_PACKET_SIZE	127
raw_throughput	1000000
rho_max	0.0100
rho_average	0.0050
con_req	44
con_ter	11
header	10
t_empty	$8.0000e^{-05}$
t_con_req	$3.5200e^{-04}$
t_con_ter	$8.8000e^{-05}$

- **States** this is the main Class and contains all the different radio states through which the BLE transceiver transits during both ND and CM.

`tx`, `rx` and `sleep` represent transmission, reception and sleep states respectively. `adv37`, `adv38` and `adv39` include all the states of the BLE transceiver from wake-up state up to transmission of advertising packet in the corresponding advertising channel. This is used to estimate the final DL and energy consumption during ND knowing the channel at which reception takes place. `adv` includes all operating states of an advertiser, `scapre` includes all operating states of a scanner from wake-up state up to the inter-channel transition state before transmission of a connection request or other request type. `sca` includes `scapre` and all operating states during and after a request to the advertiser. `scaInt` is the transition state from `tx` to `rx` or viceversa.

- **STStatesND & TIStatesND** contains all the attributes listed in *States* and all current consumption and voltage values corresponding to each radio state of ND phase taken from the quantitative model of the PHY layer for BlueNRG and TI modules as shown in Figures 2.19 and 2.20 respectively. As it can be seen in the Figures, the total amount of states may vary from one SoC manufacturer to another.

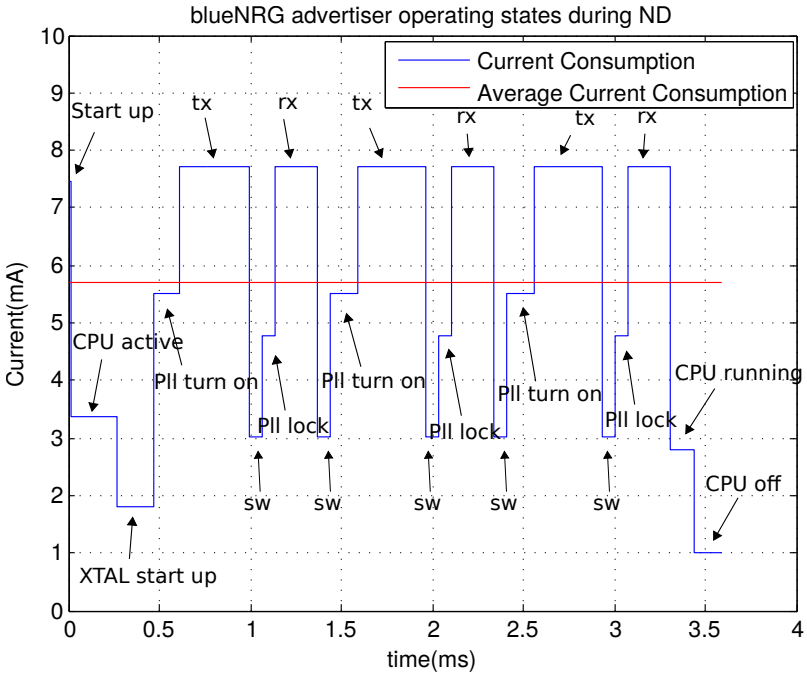


Figure 2.19: Advertiser operating states according to BlueNRG BLE device from STMicroelectronics (extracted from [5])

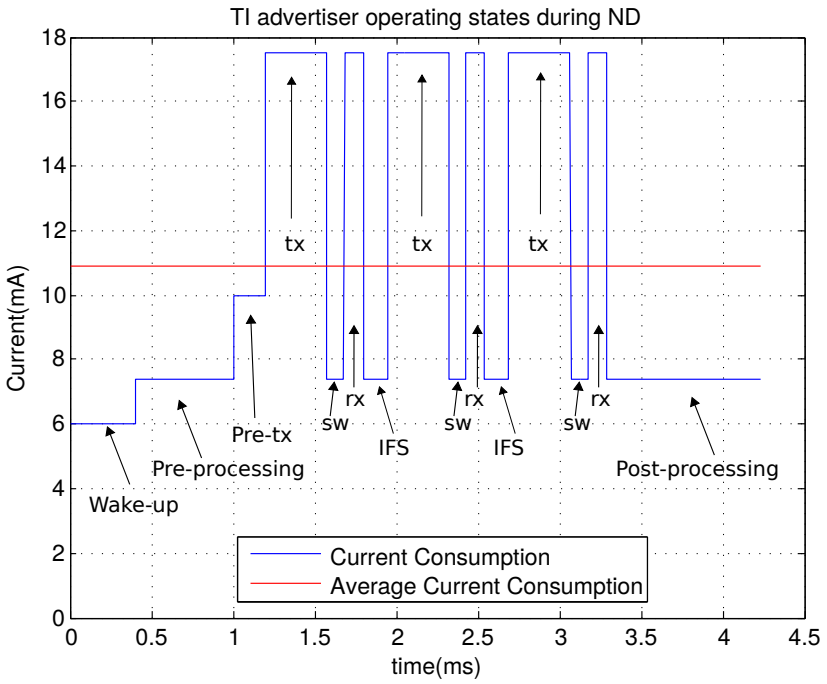


Figure 2.20: Advertiser operating states according to CC2540 BLE device from Texas Instruments (extracted from [6])

- **STStatesCM & TIStatesCM** the same as previous Class but contains the model for connection mode of BlueNRG and TI modules respectively.
- **Bat** the existence of Bat is independent from the existence of SoC and is used to calculate the lifetime of a coin cell battery.
- **SoC** all four previous states Classes and Bat are invoked from a method in SoC, as a result we obtain an object that contains all information about BLE module: manufacturer, ND and CM states with its corresponding current consumption and voltage and the energy consumed by its power supply which is a coin cell battery. It is presented in Matlab as follows:

For a better understanding of our simulator design, here we present our simulator architecture using Unified Modeling Language (UML), which is a modeling language used to visualize the design of a system, as for example a software architecture. As we said earlier, our simulator is build using OOP and its architecture is divided in Classes. These Classes are interrelated to

BLE Soc	TI CC2540
	$Vdd = 3V$
Battery	Panasonic CR2032
	$Cm = 0.225Ah$
	$I_{rated} = 0.0002A$
BLE Soc	BlueNRG
	$Vdd = 3V$
Battery	Panasonic CR2032
	$Cm = 0.225Ah$
	$I_{rated} = 0.0002A$

each other in specific ways. The following are such types of logical connections that are possible in UML and which are present in our simulator: association, aggregation, dependency and generalization. They are explained in detail as follows:

- **Association:** where one Class instance is using the other Class instance or vice-versa, or both may be using each other. But the main point is, the lifetime of the instances of two Classes are independent of each other and there is no ownership between two Classes.
- **Aggregation:** is same as association but there is an ownership of the instances. In our case, it means that a SoC can have several batteries from different manufacturers, but a battery from one manufacturer can belong to only one SoC.
- **Dependency:** where one Class depends on another Class but another Class may or not may depend on the first Class.
- **Generalization:** there is a main Class having some properties, functions etc. Then new Class is derived from this main Class and child Class(es) can have access to all the functionality of the main or parent Class. In our case the main Class is the *States* Class and the *TIStatesND, STStatesND, TIStatesCM* and *STStatesCM* Classes have all access ti *States* Class functionalities

Figure 2.21 visible in the next page shows the simulator architecture. It represents the 8 Classes mentioned above and the UML relationship between them. Furthermore, in order to validate the proposed operating modes for BLE, we use two test cases: a Temperature and Humidity Monitoring use case and a Light Switches use case. They are explained in detail in Chapter 3 where we also give lifetime results obtained from our Matlab simulator.

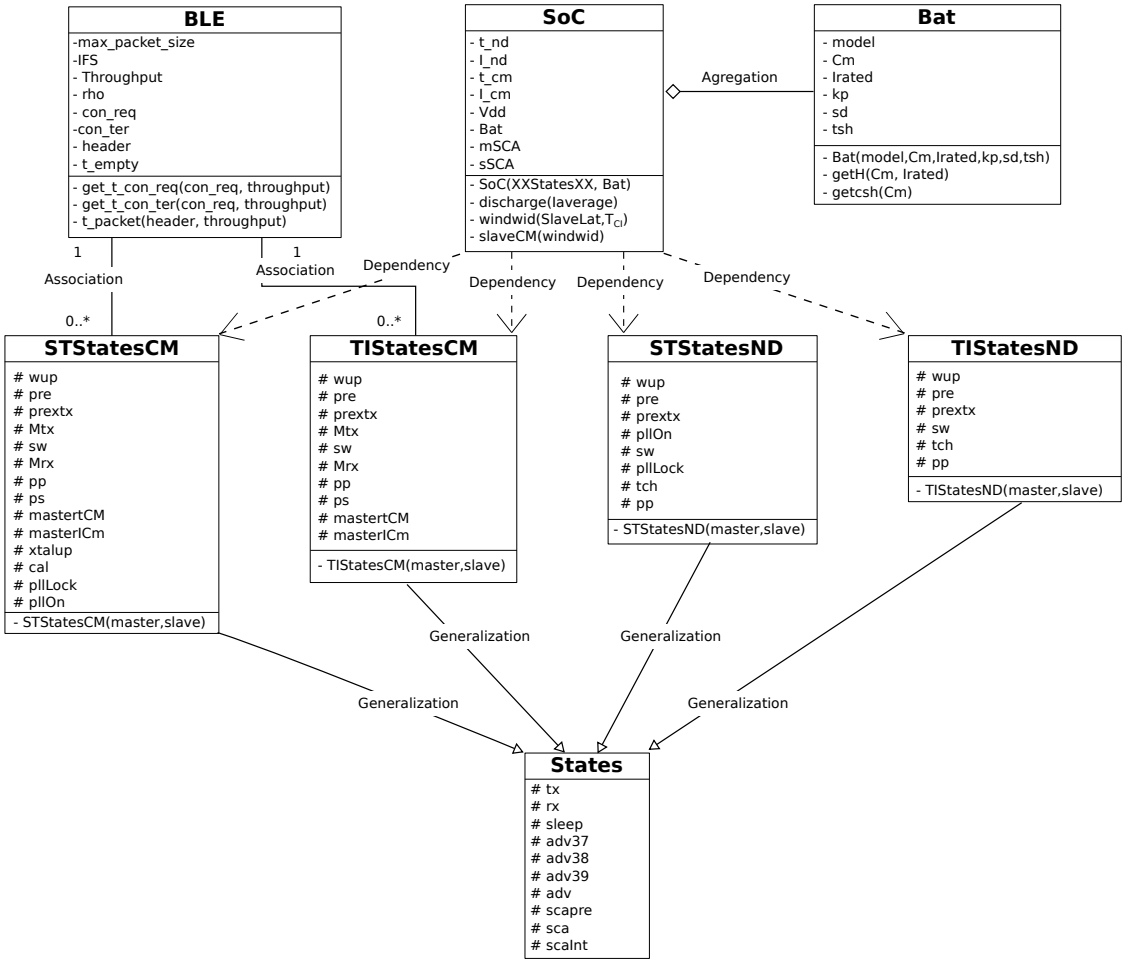


Figure 2.21: Simulator Architecture

2.6 Chapter Conclusion

In this chapter, we introduced two important metrics for the BLE performance evaluation: energy consumption and DL. In order to extract these metrics a comprehensive understanding of the protocol is needed. Consequently, we explained the LL communication modes (ND and CM) which are based on a five states FSM given by BLE specifications. With the purpose of maintaining synchronization and data exchange, each of this communication modes comprise a set of parameters that establishes the time that devices spend transmitting/receiving. The time it takes for a packet to be received by the receiving node, either during ND or CM directly depends on the choice of this parameters and subsequently, the energy consumption and battery lifetime.

We have studied existing DL and energy consumption models for BLE and we have compared five of them [4, 2, 3, 35, 36]. The model in [4], is the most complete in the SotA as it comprises all operating modes of the LL for both ND and CM and have been proven to be very precise via measurements. Our research is built upon this model and similarly we don't consider interference or packet collision and we evaluate single advertiser/slave single scanner/master communication. On the other hand, the model is not adapted to evaluate scanner performance when DL and energy consumption are critical at its side. We have optimized the algorithm proposed in [4] in order to render ND parameter optimization and to evaluate scanner performance when needed. We have validated the model and we show that, although the user experience remains similar for small values of T_{AI}, a much better DL/energy trade-off between advertiser and scanner is obtained when advertiser (typically the peripheral device) is available waiting to be discovered by the scanner.

In order to evaluate BLE performance, the impact of energy consumption over the battery lifetime must be evaluated as nodes are typically battery operated. We have proposed a model to estimate battery lifetime based on data provided by the manufacturer, Peukert's Law and discharge rate given by node behavior. The model doesn't take into account temperature variations or recovery effects after the node spends time in sleep mode. However, the model provides with fast and realistic approximation of lifetime. The model is designed to be used with any lithium battery.

Finally we introduced our Matlab based simulator which implements OOP paradigm, thus providing flexibility and allowing to evaluate different manufacturer BLE devices. It is used for calculating BLE devices lifetime and performance evaluation of parameter optimization and the different energy optimization methods which are presented in the next Chapter.

Chapter 3

Energy Consumption Optimization

In this Chapter we present our parameter optimization method based on algorithms discussed in Chapter 2. Network synchronization produces great waste of energy due to overhead. In order to mitigate this effect, we propose a method to optimize energy consumption. First we categorize all possible use cases we can find in practice into three types of scenarios and then we propose an energy consumption optimization by introducing new operating modes of BLE for each type of scenario. We give results of energy consumption and lifetime for typical IoT use cases.

3.1 Parameter optimization

BLE standard supports a wide range of parameter values for ND and CM. The choice of these parameters directly affect BLE devices performance. Therefore, it is of capital importance to use an optimal parameter configuration in order to ensure a fair trade-off between DL and energy consumption. An analytical model can offer a good guideline for choosing the right configuration, such as the one presented in [4]. However, results from this models are very general and need for large computation time. Getting results from Algorithm 1 can take from minutes to several hours or a few days when evaluating configurations that uses large values parameters. Hence, making the design process very long and not optimal, as getting the right parameters according to specific application requirements with these algorithms, is not straightforward.

On the other hand, Bluetooth SIG has provided with a list of profiles [29] intended to enable Bluetooth devices to be efficiently implemented for

different applications such as collecting sensor information, health, sports and fitness, environmental sensing and proximity applications, among others. These profiles define the behavior for both central and peripheral role devices and moreover, they include recommended scanner/advertiser configuration to ensure optimal DL and energy consumption during ND. However, to the best of our knowledge, there is no performance evaluation using these profiles and there is no evaluation considering the scanner as a peripheral device with constrained energy from application point of view. We evaluate these profiles for some IoT typical use cases and compare the results when using our parameter optimization method. In general, we give parameters that are estimated based on specific use case requirements and provide with a method to obtain an optimal configuration for any use case. These kind of evaluations would help designing new mechanisms to optimize energy consumption based on real use cases constraints and furthermore, on how to optimize the typical parameters implemented during ND and CM, based on application requirements. In this section we focus on the ND phase whereas CM is analyzed in Section 3.2.

Optimization of the ND phase is achieved using a look-up table based simulation in Matlab using Algorithm 3. With this algorithm, BLE performance is evaluated during ND given a certain (T_{SI}, T_{SW}, T_{AI}) configuration. In order to evaluate BLE performance based on requirements for a given use case, we reproduced the inverse process: we sweep into the look-up table to determine the (T_{SI}, T_{SW}, T_{AI}) configuration whose worst DL is no longer than a CL given by the use case, with a maximum difference of 5 ms below the given DL. Based on simulation results, we see that several configurations can satisfy a given latency requirement, for example, for a required advertiser DL of 200 ms at least a dozen of configurations can be used, with a scanner duty cycle ranging from 65% to 100%, using $T_{SI} \leq 400$ ms and an advertiser duty cycle of up to $\approx 5\%$, in which case, the configuration that provides the lowest energy consumption on the advertiser side, is automatically selected.

Analogously, several configuration can satisfy a given latency requirement when constrained DL is at the scanner side and the configuration that provides the lowest energy consumption for the scanner, is then selected. In Table 3.1 visible in the next page, we give a set of parameters which are suitable for a required DL between 200 ms and 1 s while providing the minimum average energy consumption possible on the advertiser side. For the same range of required DL, Table 3.2 shows a set of suitable parameters when the constrained Critical Latency (CL) is at the scanner side.

Based on our simulations, we have made some important observations: for any (T_{SI}, T_{SW}, T_{AI}) configuration where $T_{AI} < T_{SW}$ and given a certain CL, there is a maximum value of T_{AI} that will satisfy the CL requirement

Table 3.1: Parameter Optimization Respect to Advertiser CL

CL	$\mathbf{T}_{SI}$	$\mathbf{T}_{SW}$	$\mathbf{T}_{AI}$
200 ms	400 ms	300 ms	190 ms
300 ms	500 ms	400 ms	290 ms
400 ms	60 ms	50 ms	390 ms
500 ms	500 ms	500 ms	490 ms
600 ms	300 ms	300 ms	590 ms
700 ms	200 ms	200 ms	690 ms
800 ms	70 ms	60 ms	790 ms
900 ms	2.56 s	1.92 ms	890 ms
1 s	1.92 ms	1.28 ms	990 ms

Table 3.2: Parameter Optimization Respect to Scanner CL

CL	$\mathbf{T}_{SI}$	$\mathbf{T}_{SW}$	$\mathbf{T}_{AI}$
200 ms	400 ms	300 ms	190 ms
300 ms	600 ms	400 ms	290 ms
400 ms	800 ms	500 ms	390 ms
500 ms	900 ms	600 ms	490 ms
600 ms	1.28 s	640 ms	590 ms
700 ms	2.56 s	1.28 s	690 ms
800 ms	3.2 s	1.28 s	790 ms
900 ms	3.84 s	1.28 s	890 ms
1 s	4.48 s	1.28 s	990 ms

according to the worst DL. Most T_{AI} values below that maximum will satisfy the CL (except a few ones) and even though these smaller values can provide shorter average DL; the average energy consumption increases due to the worst case of DL despite this cases are rare. This is illustrated in Figure 3.1.

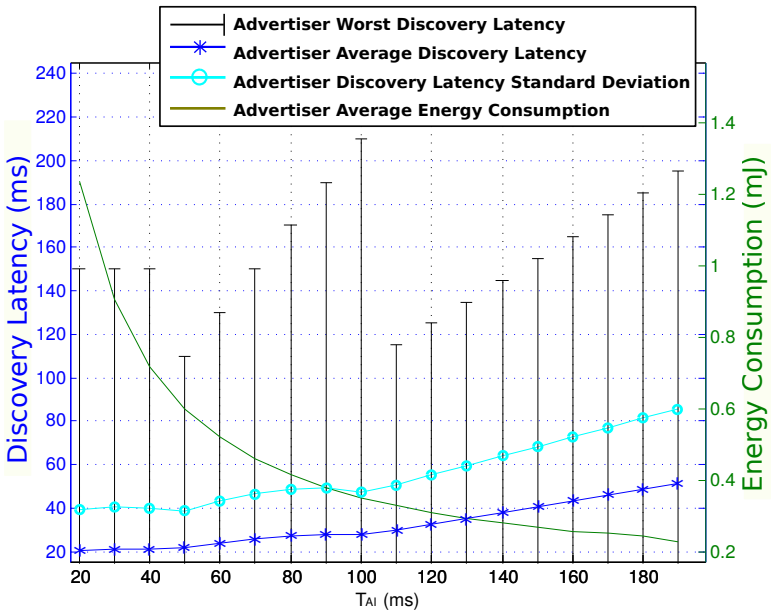


Figure 3.1: Discovery latency and energy consumption for $CL = 200\,\text{ms}$, $T_{SI} = 400\,\text{ms}$, $T_{SW} = 300\,\text{ms}$, $20\,\text{ms} \leq T_{AI} \leq 190\,\text{ms}$

In Figure 3.1 we can see, for example, that average and worst case of DL is lower for $T_{AI} = 50\,\text{ms}$ compared to $T_{AI} = 190\,\text{ms}$, but energy consumption

for $T_{AI} = 50\,\text{ms}$ is higher. Our simulation tool automatically provides with the set of parameters that will ensure the required maximum DL even for the worst case (which happens when the peripheral device starts at a certain ϕ for which DL is the longest for that set of parameters), but providing at the same time with the lowest energy consumption possible for that device.

It is also important to highlight the fact that the standard deviation is greater than the average value as seen in Figure 3.1. This indicates a bias, that is, the presence of extreme values. Indeed, there are a few occurrences where the DL is very large or in other words, there are a few occurrences where the DL is close or equal to the worst case, but they have a very big impact on the mean value. The mean is susceptible to distortion by the presence of extreme values, and because it is based on deviations from the mean, the standard deviation is susceptible to the same distortion which is determined by the fact that the deviation scores are squared.

Furthermore, in order to provide results for typical IoT scenarios based on our parameter optimization, we propose two use cases and compare them with results when using recommended SIG Profiles configurations. These two use cases are a retail store and a medical telemetry system. They are described in detail in the next sections.

3.1.1 Retail store use case

This test case targets iBeacon technology. It was announced in 2013 by Apple [50] as a new technology which provides a higher level of location awareness. iBeacon is an efficient, built-in, cross-platform technology for Android and iOS devices, which utilizes BLE for indoor positioning. Since the technology uses BLE, it offers the user less battery drain. A device which generates iBeacon advertisements is called beacon. Beacons establish a region around them by iBeacon signals. A device supporting an iBeacon application can determine if it has entered or exited from the region, and can approximate its distance to the beacon via signal strengths. An iBeacons transmit advertisement data frames containing different identifying fields as follows:

- **iBeacon prefix:** these are 9 fixed Bytes which indicate that the BLE device is actually an iBeacon device.
- **UUID:** universally unique identifier is a 16 Bytes length field, which can be used as an ID for all beacons used in an application. The purpose of the different ID is to distinguish beacons in a network, from all other beacons in outside networks. Technically an organization can

use multiple UUIDs if needed, for example to identify venues in different locations or to identify different business units.

- **Major:** is a 2 Bytes length field, which can be used to differentiate between beacons with same UUIDs. Major values are intended to identify and distinguish a group – for example all beacons in a certain floor or room could be assigned a unique major value.

- **Minor:** this 2 Bytes field can be used to differentiate between beacons with the same UUIDs and Major values. Minor values are intended to identify and distinguish an individual beacon within a group of beacons with the same assigned major value.

- **Tx Power:** is the strength of the signal measured at 1 meter from the beacon. This number is then used to determine how close the device is from the beacon.

All the previously presented fields constitute an iBeacon packet, it makes a total of 31 Bytes which is the payload for the BLE advertising packet. We evaluated the case where an iBeacon (advertiser) is placed around an item in order to push information about what's on sale so the user can find the item, receive extra information about it or even pay for it at the point of sale (POS). For this, the user must be in range with the beacon according to Figure 1.1. Supposing the user is walking around the beacon and does not stop within the region of range, then the time to pass near it must be considered to ensure that, even in the worst case, the time to establish a connection between the beacon (advertiser) and the smartphone (scanner); is long enough, so the user can successfully receive the notification.

In the worst case, where the user walks by 29.5m far from the beacon, for a total walking distance of 10.9m, the user has a maximum time to establish a connection of 11.22s if the user walks at speed of 3.5 Km/h [51] or 4.36s at a speed of 9 Km/h [52]. Best and worst case for walking distance are depicted in Figure 3.2 visible in the next page. For evaluation of this scenario in terms of energy consumption, the following conditions are taken into account:

- Ideal channel conditions: no interference and no collisions.

- The user passes through the region at a constant speed in a straight line without stopping. Is the worst case as there is not much time for ND.

- BLE on the user's smartphone is active at all moments.

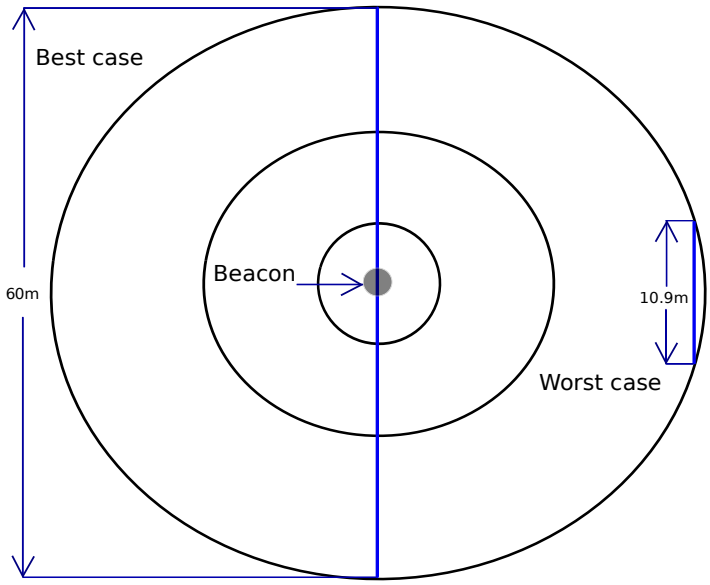


Figure 3.2: Worst case and best case walking distance around iBeacon

In order to ensure that the advertising is successfully received, even in the worst case, we have considered a CL of $4.36s$ in accordance with the above mentioned. In addition, if we consider the possibility of a packet loss due to either interference or collision, then an entire period would have to pass before the packet can be received. Hence, with the aim of providing reliability, we reduce this CL to 2.18s, thus even if there is a packet loss during the worst case of a given $T_{SI} - T_{SW} - T_{AI}$ configuration, the packet can still be received while the user is in range with the beacon.

We first estimate DL and energy consumption when implementing configurations proposed by the Bluetooth SIG and we compare them with the recommended configuration according to our parameter optimization method. Taking into account that the advertiser is considered to be active permanently but the ND is restricted to a CL of $2.18s$ on the user side, we do not consider dynamic reconfiguration as in the SIG Profiles, so we do not implement recommended configuration after 30s for this use case. We evaluate performance of this use case based on Algorithm 3, where the advertiser is considered to be available waiting to be discovered by the scanner.

Table 3.3, visible in the next page, shows the simulation results for average energy consumption, average DL and worst case of DL, for advertiser and scanner of a CC2540 from Texas Instruments and a BlueNRG from STMicroelectronics. These values represent the typical consumption during one connection between advertiser and scanner during ND. Our proposed config-

uration meets the requirements of the use case, while implementing a duty cycle that provides a longer lifetime for both scanner and advertiser. If we use the Bluetooth SIG configurations, using a T_{AI} of either $20ms$ or $30ms$, scanner duty cycle is 50% and advertiser duty cycle is minimum $\approx$ 33%, whereas with our proposition scanner and advertiser duty cycle is 25% and $\approx 0.18\%$ respectively, thus extending battery lifetime.

Table 3.3: Retail store use case discovery latency and energy consumption results

			TI		**BlueNRG**	
Proposed configurations by SIG: fast connection			**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 60ms, T_{SW} = 30ms$	$T_{AI} = 20ms$	E_{avg}	$231.16\mu J$	$1.1mJ$	$66.711\mu J$	$452.11\mu J$
		DL_{avg}	$34.9ms$	$22.5ms$	$34ms$	$20ms$
		DL_{wc}	$51.6ms$	$34.2ms$	$50.7ms$	$31.7ms$
	$T_{AI} = 30ms$	E_{avg}	$202.38\mu J$	$1.24mJ$	$57.78\mu J$	$476.78\mu J$
		DL_{avg}	$41ms$	$23.6ms$	$40.1ms$	$21.1ms$
		DL_{wc}	$71.6ms$	$39.2ms$	$70.7ms$	$36.6ms$
Proposed configuration in this work			**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 10.24s, T_{SW} = 2.56s$	$T_{AI} = 2.2s$	E_{avg}	$192.11\mu J$	$58.4mJ$	$60.07\mu J$	$25.7mJ$
		DL_{avg}	$2.17s$	$1.11s$	$2.22s$	$1.11s$
		DL_{wc}	$4.41s$	$2.21s$	$4.41s$	$2.21s$

Table 3.4 shows lifetime results for retail store use case. If we compare our results with the Bluetooth SIG configuration, we can see advertiser lifetime for the STMicroelectronics device, is improved by a factor of $\approx$ 89 with our proposition compared to the SIG Profile configuration of $T_{AI} = 20ms$ and $\approx$ 62 times higher with SIG configuration of $T_{AI} = 30ms$. In the other hand with the Texas Instruments device, lifetime is $\approx$ 105 times higher compared to the SIG Profile configuration of $T_{AI} = 20ms$ and $\approx$ 72 time higher compared to $T_{AI} = 30ms$. Implementing the recommended SIG Profile it wouldn't be possible to power the beacon (advertiser) with a coin cell battery, whereas with our proposition the beacon could run permanently with a coin cell battery of $225mAh$ battery capacity with a maximum lifetime of 2.32 years with the STMicroelectronics device in contrast with the 3.82 days of lifetime if using the SIG Profile.

Table 3.4: Retail store use case lifetime results

		TI		**BlueNRG**	
Proposed configurations by SIG: fast connection		**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
				Lifetime	
$T_{SI} = 60ms, T_{SW} = 30ms$	$T_{AI} = 20ms$	3.82 days	1.76 days	9.55 days	2.56 days
	$T_{AI} = 30ms$	5.54 days	1.76 days	13.82 days	2.56 days
Proposed configuration in this work		**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 10.24s, T_{SW} = 2.56s$	$T_{AI} = 2.2s$	1.1 years	4.05 days	2.32 years	5.62 days

In addition the lifetime of the scanner ≈ 2 times higher with our proposition. Although our calculation is based on the coin cell battery of $225mAh$ battery capacity, the scanner is expected to be running on a smartphone and thus to account with a much higher capacity. Nevertheless our results show that the battery at the scanner side is depleted at less twice slower, allowing the user to make efficient use of the battery at the same time that the smartphone can be used for the rest of its functionalities, without causing a total depletion of the battery.

Figure 3.3 shows a comparison for the lifetime results. It illustrates the ratio between lifetime with SIG Profile configurations and our proposition. Furthermore, it shows the ratio between the SIG Profile with $T_{AI} = 20ms$ and our proposition, followed by the ratio between the SIG Profile with $T_{AI} = 30ms$ and our proposition and finally the ratio between scanner lifetime using SIG profile configuration and the scanner lifetime using our proposition. The graphic includes the results for both the TI and the STMicroelectronics devices. The high ratio obtained for the advertiser lifetime it is due to the fact that duty cycle can be up to ≈ 183 times lower using our proposition compared to the SIG configuration, whereas scanner duty cycle can be up to ≈ 2 times lower using our proposition, thus assuring a trade-off between energy consumption and latency while complying with the use case requirements.

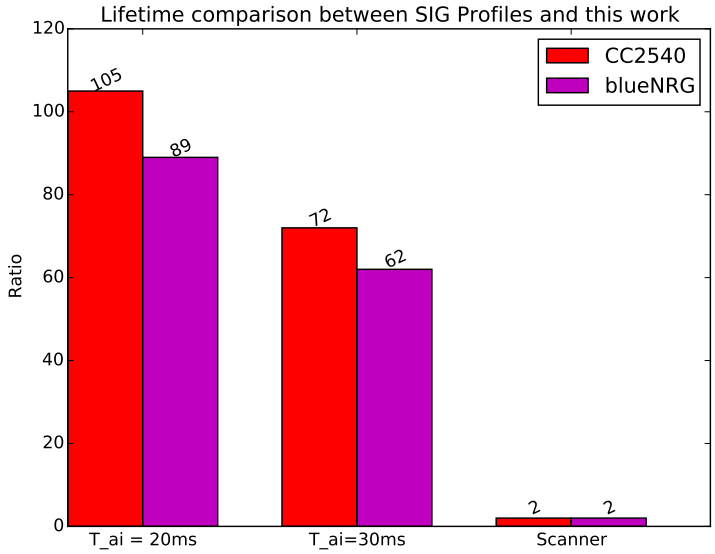


Figure 3.3: Lifetime comparison between SIG Profiles vs this work for the retail store use case

3.1.2 Wireless medical telemetry use case.

These systems include measurement and recording of physiological parameters and other patient-related information which are used in a broad range of environments such as health care facilities and patients' home. Wireless communication systems are envisioned to be used for implantable pacemakers, glucose monitors, insulin pumps and remote patient monitoring, among others [53]. We evaluate a remote patient monitoring system using Proximity profile for sending patient-related information to the hospital information technology system.

The system is based on a typical four layers architecture as in [7]. A first layer incorporates sensor nodes operating within a wireless network. The nodes can be placed on the human body as very small patches (on-body sensors), sewed into fabric (wearable sensors), or implanted under the skin (in-body sensors). Layer 2 contains user interaction devices, BLE based monitoring devices such as smartphones, tablets or PDAs. They act as an Access Point (AP) that are usually located within a room environment. The third layer consists of a Decision Measuring Unit (DMU). An automatic computing system which performs all major computing operations and is connected to the Internet. The role of the DMU is to collect, filter and analyze the information. The DMU is able to recognize patient's conditions. Subsequently, appropriate decisions are made automatically regarding the health status of the patient. The DMU is connected to a back-end medical institution such as a hospital in which physicians are able to consider people's health status. The last layer provides health care services to patients. The data stored in the DMU, is delivered to a remote server in a hospital, where medical professionals have access to it. The system architecture is shown in Figure 3.4 which is visible in the next page.

In order to evaluate BLE performance, we focus on Layer 1 and Layer 2, in which the smartphone or any other mobile device plays the central role (scanner) and the in-body implanted pacemaker plays the beacon or peripheral role (advertiser). A connection is required to collect data from the sensor to the smartphone or other mobile device via BLE. The key parameters to be evaluated are the connection time on the user side and sensor energy consumption during connection, this means advertiser DL and energy consumption during ND. The following conditions are taken into account:

- Ideal channel conditions: no interference.
- The scanner is active when sending data to hospital is required, whose frequency can range from every 3 months to every 12 months [54].

- Advertiser is active at all moments as it is implanted on the patient and is battery operated.
- Scanner is in range with the iBeacon at the moment of connection.

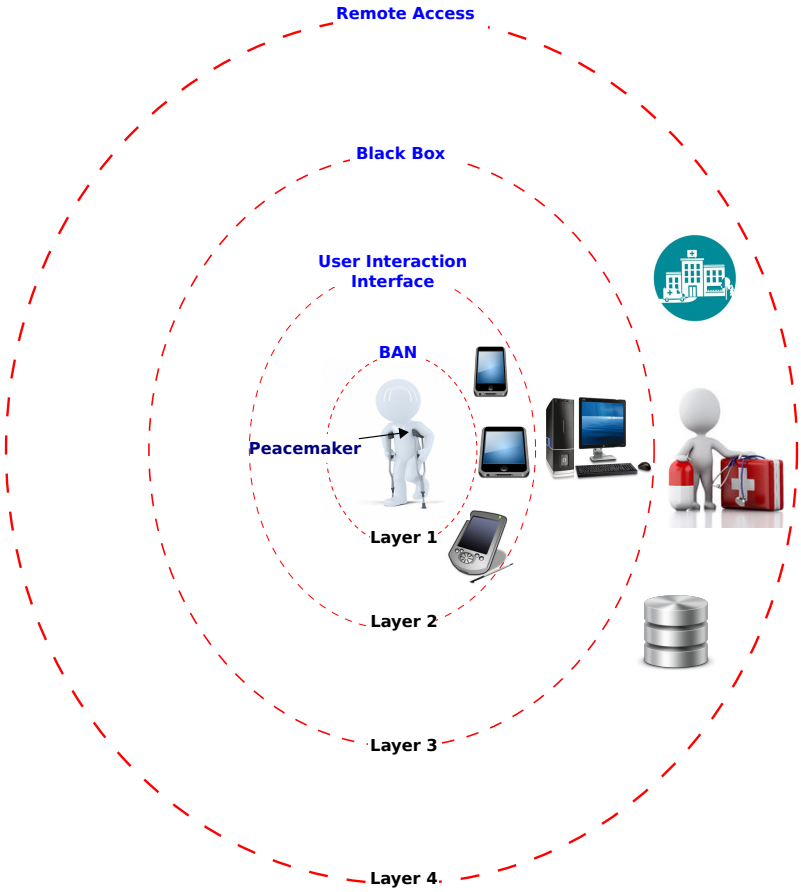


Figure 3.4: Typical Architecture of a Medial Telemetry System [7]

Performance is evaluated based on Algorithm 1, since the advertiser plays the peripheral role at application level and has constrained DL and energy consumption. Similar to the previous use case, we estimate average energy consumption, average DL and worst case DL for both scanner and advertiser during ND. For this use case, changing settings dynamically after 30 seconds is considered but contrary to the Bluetooth SIG we implemented only at the scanner side. In this case the dynamic reconfiguration as the data collecting device may be activated only when transmitting the data to the hospital is required, nevertheless this is not limitative. A fixed access point can be

considered too, in which case a fixed configuration is recommended where the scanner would be doing continuous scanning using a permanent source of energy.

Table 3.5 shows the simulation results for the medical telemetry system. For advertiser and scanner of a CC2540 from Texas Instruments as well as a BlueNRG from STMicroelectronics. These values represent the typical expense during one connection between advertiser and scanner during ND. Using the recommended configurations from the Bluetooth SIG scanner duty cycle is 50% and advertiser duty cycle goes from $\approx 0.2\%$ to $\approx 20\%$ whereas with our proposition scanner duty cycle is 60% and advertiser duty cycle is $\approx 0.04\%$, thus ensuring the longest lifetime possible at the advertiser side which is critical for this use case, as the sensor is implanted in the body. At the same time, our proposition provides with a good user experience and the best trade off possible between advertiser and scanner energy consumption.

Table 3.5: Medical telemetry system discovery latency and energy consumption results

			TI		**BlueNRG**	
Proposed configurations by SIG: fast connection			**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 60ms, T_{SW} = 30ms$	$T_{AI} = 20ms$	E_{avg}	$210.27\mu J$	$3.7mJ$	$63.32\mu J$	$1.6mJ$
		DL_{avg}	$26.3ms$	$131.9ms$	$25.4ms$	$129.3ms$
		DL_{wc}	$103.1s$	$285ms$	$102.1ms$	$282.4ms$
	$T_{AI} = 30ms$	E_{avg}	$208.33\mu J$	$3.9mJ$	$62.78\mu J$	$1.7mJ$
		DL_{avg}	$35.4ms$	$140.3ms$	$34.5ms$	$137.7ms$
		DL_{wc}	$143.1ms$	$337.7ms$	$142.1ms$	335.1
Proposed configurations by SIG: after 30s of ND			**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 60ms, T_{SW} = 30ms$	$T_{AI} = 1s$	E_{avg}	$215.89\mu J$	$30.7mJ$	$67.46\mu J$	$13.2mJ$
		DL_{avg}	$990.5ms$	$1.12s$	$989.6ms$	$1.16s$
		DL_{wc}	$3.18s$	$4.64s$	$3.01s$	$4.64s$
	$T_{AI} = 2s$	E_{avg}	$215.8\mu J$	$56.1mJ$	$69.81\mu J$	$24.1mJ$
		DL_{avg}	$1.93s$	$2.04s$	$1.93s$	$2.04s$
		DL_{wc}	$8.23s$	$9.88s$	$8.02s$	$9.88s$
Proposed configuration in this work			**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 500ms, T_{SW} = 300ms$	$T_{AI} = 10.24ms$	E_{avg}	$145.43\mu J$	$204mJ$	$53.5\mu J$	$88.2mJ$
		DL_{avg}	$4.1s$	$7.03s$	$4.09s$	7.06
		DL_{wc}	$10.25s$	$21.83ms$	$10.25s$	$21.82s$
Proposed configuration in this work after 30s of ND			**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 400ms, T_{SW} = 200ms$	$T_{AI} = 10.24ms$	E_{avg}	$183.52\mu J$	$253.1mJ$	$71.14\mu J$	$111.3mJ$
		DL_{avg}	$6.4s$	$9.59s$	$6.4s$	9.59
		DL_{wc}	$20.49s$	$21.78ms$	$20.49s$	$21.77s$

Table 3.6, visible in the next page, shows medical telemetry system lifetime results. Using $T_{AI} = 2s$, advertiser lifetime is improved ≈ 96 and ≈ 82 times for Texas Instruments and STMicroelectronics device respectively, compared to the SIG profile.

Additionally, we can see from Table 3.6, visible in the next page, that comparing the $T_{AI} = 20ms$ configuration with our proposition of $T_{AI} = 10.24ms$, advertiser lifetime is ≈ 408 and ≈ 281 times higher for Texas Instruments

Table 3.6: Medical telemetry system lifetime results

		TI		**BlueNRG**	
Proposed configurations by SIG: fast connection		**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
			Lifetime		
$T_{SI} = 60ms, T_{SW} = 30ms$	$T_{AI} = 20ms$	3.82 days	1.76 days	9.55 days	2.56 days
	$T_{AI} = 30ms$	5.54 days	1.76 days	13.82 days	2.56 days
Proposed configurations by SIG: after 30s of ND		**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 60ms, T_{SW} = 30ms$	$T_{AI} = 1s$	6.28 months	1.76 days	1.14 years	2.56 days
	$T_{AI} = 2s$	1 year	1.76 days	2.14 years	2.56 days
Proposed configuration in this work		**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 500ms, T_{SW} = 300ms$	$T_{AI} = 10.24ms$	4.27 years	1.46 days	7.36 years	2.11 days
Proposed configuration in this work after 30s of ND		**Advertiser**	**Scanner**	**Advertiser**	**Scanner**
$T_{SI} = 400ms, T_{SW} = 200ms$	$T_{AI} = 10.24ms$	4.27 years	1.71 days	7.36 years	2.54 days

and for STMicroelectronics device respectively. Even if we compare the best configuration of SIG Profiles which is $T_{AI} = 2s$ with our proposition, advertiser lifetime is still ≈ 4 and ≈ 3 times higher when using Texas Instruments and STMicroelectronics device respectively. With respect to the scanner, our proposition does not impact its lifetime as it does with the advertiser since we only focus on the advertiser which is implanted on the patient's body, so a battery replacement would imply an invasive procedure. We ensure to provide the longest advertiser lifetime possible, while providing a good user experience. We have considered an average of $4s$ of DL as a tolerable amount of time to wait before establishing a connection with the mobile device or other access point with BLE. Our results show that BLE is a suitable technology for this kind of applications since with the right configuration, a device can have an autonomy of more than 7 years with a coin cell battery of $225mAh$. Figure 3.5 shows a comparison for the lifetime results between the SIG Profile and our proposition, the Figure can be seen in the next page.

We have achieved to improve advertiser lifetime considerably during the ND phase. With our methodology it is possible to optimize devices energy consumption while guaranteeing a low latency at the same time. We assure a latency which is always below the maximum accepted value according to the use case. Furthermore, the solution previously presented corresponds only to applications where the communication is only done asynchronously and the data exchange between both devices is sporadic. However in the next section we also analyze the cases in which a permanent and synchronous connection can be necessary.

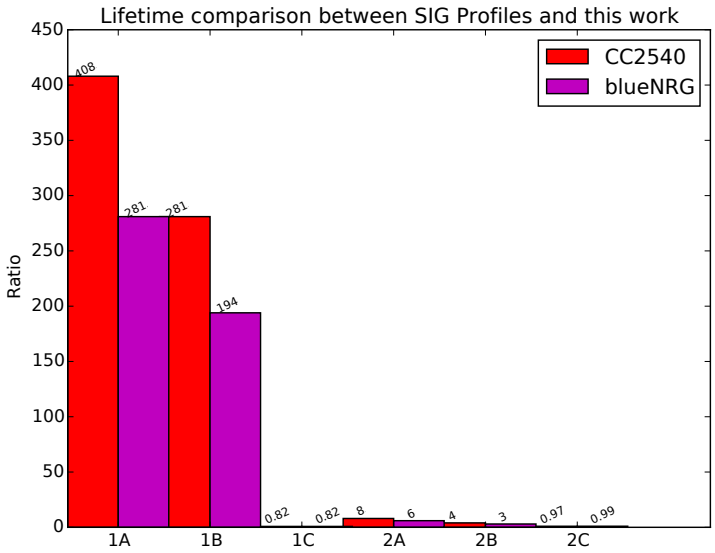


Figure 3.5: Lifetime comparison between SIG Profiles vs this work for the medical telemetry system use case. Group 1 represents the lifetime ratio between the SIG Profile and our proposition for ND during the first 30 seconds and Group 2 after 30 seconds. With respect to SIG Profiles, Group 1A represents $T_{AI} = 20ms$, 1B represents $T_{AI} = 30ms$ and 1C represents the scanner. Group 2A represents $T_{AI} = 1s$, group 2B represents $T_{AI} = 2s$ and group 2C represents the scanner

3.2 Low Duty Cycle Applications with BLE

Even though BLE has been proved to be more energy efficient and robust than other WSN protocol[24, 25], there is still a need for adapting the protocol to more generic IoT scenarios, where traffic patterns are different, with scarcer communication and where applications need to run autonomously during several years. In this section we discuss how the problem can be aboard with the use of BLE.

In CM, BLE devices have to interact regularly for two main reasons. First, to guarantee a *critical latency* (CL), which is the maximum communication latency that guarantees acceptable operation or user experience, and is application dependent. The worst case happens when an event occurs right after a network exchange, then a full period is needed before communication happens again. The same would happen if the event happens before a network exchange but there is packet loss due to high interference or packet collision.

Second, to guarantee correct synchronization, which depends on system constraints, mostly on the oscillator accuracy that has a direct impact on clock drift.

This second constraint together with the targeted operating conditions, for example a temperature range, define the maximum communication period for the system. A BLE module transmits application data during a connection event. The timing of connection events is determined by two parameters: T_{CI} and *Slave latency* as shown in Figure 3.6. According to the specification T_{CI} has a minimum value of 7.5 ms and maximum value of 4 s.

As mentioned in Section 2.2.3.2, a slave could save much more energy than the master by skipping communication events. The amount of events the slave is allowed to skip is determined by the *Slave Latency* parameter, it defines the number of consecutive connection events that the slave device is not required to listen for the master. It should not cause a supervision timeout and shall be an integer in the range of $[0, \min(T_{ST}/T_{CI}, 500)-1]$. The *Supervision Timeout (T_{ST})* parameter defines the maximum time between two consecutive received data PDUs before the connection is considered lost. It shall be a multiple of 10 ms in the range of [100 ms, 32 s]. Taking a deeper look into these specifications we can see that, if for example we use the minimum T_{CI} of 7.5 ms, the maximum T_{ST} will be 3.75 s, having a maximum *slave latency* of 499 events. Meaning that, if the Master doesn't receive a data PDU form the slave 3.75 s after the last communication event, the connection will be considered as lost and devices will go into ND process. As another example, if we use the maximum T_{CI} which is 4 s, then T_{ST} will be 32s, having a maximum slave latency allowed of 7 events.

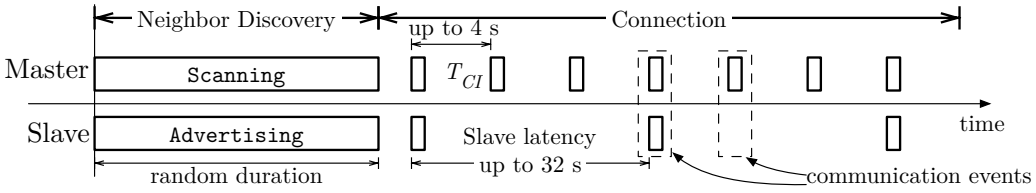


Figure 3.6: BLE communication pattern

In the case of rare applications events, this causes high communication overhead and represents a huge waste of energy: supposing that even though two devices need to communicate every X min in average, the master would have to wake up at least every 4 s and analogously, the slave would have to wake up at least every 32 s in order to remain synchronized.

For applications where events occur with low frequency, it seems reasonable to use a communication pattern where devices establish a connec-

tion only when needed instead of maintaining a connection, thus consuming considerably more energy due to overhead. That implies the need of using asynchronous communication patterns which leads our attention to ND process of BLE. In order to let devices spend most of the time in sleep mode between events, seems natural to think of configuring the devices such that CM is not use at least between two consecutive events, meaning that devices would have to synchronize with each other prior to an application event, in order to allow the data exchange but forcing the devices to perform ND regularly. During ND, the advertiser is not synchronized with the scanner and the reception of a packet by the scanner is not guaranteed, so it is likely that the advertiser will have to send a packet multiple times and the scanner has to keep its radio on, until a successful reception is achieved. This non-deterministic and energy hungry nature of ND behavior makes the possibility of switching from asynchronous to synchronous communication on a regular and indefinite basis, a very complex task.

Additionally, as it is already known, energy consumption and DL can vary greatly depending on the ND parameters chosen for both scanner and advertiser. Moreover, changing parameters to improve energy consumption on one side will degrade performance on the other side, however a trade off between scanner and advertiser energy consumption can be achieved by finding the right parameters, as we show in Section 3.4.3.3. In the next section, we first propose a scenarios classification in order to implement the right optimization technique for each case.

3.3 Proposed scenarios classification

There are two types of events occurrence with respect to their relation to time, *synchronous*, where events timings are known in advance, and *asynchronous*, where events happen randomly. If we now consider communications, we must add *critical latency* (CL) as a new parameter. We have defined CL as the minimum required time to ensure proper operation and/or user experience. We present a classification of the scenarios according to their CL and frequency of events occurrence.

We propose to distinguish three types of scenarios for which we propose different modes of operation depending on their CL and events type in the next section

1. *Continuous high frequency* are scenarios for which the CL is low, on the order of seconds or less, no matter if events are synchronous or asynchronous, to provide the illusion of real-time, like heart rate monitors

that communicate measurements several times per minute to trackers. This is the typical scenario for BLE.

2. *Random low frequency* are scenarios for which the period of interaction is large but events are asynchronous and need a low CL. The light switch scenario falls into this category, switches are used a few times a day, but CL must remain below 200 ms to provide acceptable user experience when turning on or off the light.

3. *Periodic low frequency* are scenarios for which the period of interaction is large and there is a very relaxed CL. We can accommodate asynchronous events, as long as we define an application-level duty cycle, where the application guarantees periodic communication episodes enforcing the CL. This is typically the case for the temperature and humidity monitoring scenario.

3.4 Proposed modes of operation

We define the operation modes as different ways of implementing the communication modes of BLE (explained in Chapter 2), allowing devices to be compatible with different type of applications while extending battery lifetime. We propose three possible modes of operation from which to choose depending on the type of scenario:

1. *Classic BLE:* devices perform ND, normally triggered by an action from the user and once they discover each other they establish a connection and stay in CM continuously and for long periods of time. This is the typical mode of operation for BLE.

2. *Fully asynchronous BLE:* the scanner listens asynchronously waiting for a packet from an advertiser. Once the data packet is received the advertiser goes to sleep mode to save energy and the scanner goes back to scanning state, in this case the scanner should be powered by a permanent source of energy. This is the mode where data could be sent within advertising packets for example and has not been previously evaluated for autonomous and long-running applications.

3. *Duty cycled BLE (DC-BLE):* application data exchange is quite rare, so both devices go to sleep mode for long periods of time, but they must perform ND prior to a data exchange, so optimization of this phase is necessary in order to achieve better energy consumption compared to classic BLE. We propose to enforce a duty cycle on top of BLE.

Figure 3.7 shows the typical behavior of single master to single slave BLE communication using the different modes of operation. Detailed aspects of the DC-BLE mode are shown in Figure 3.8, where we show the least favorable case when application data must be sent after setting up a connection. If the size of the data is small enough it can be optimized, using the non-connectable undirected advertising up to 31 B of data, or using the scannable undirected advertising for more than 31 B by sending application data within the scan response packets.

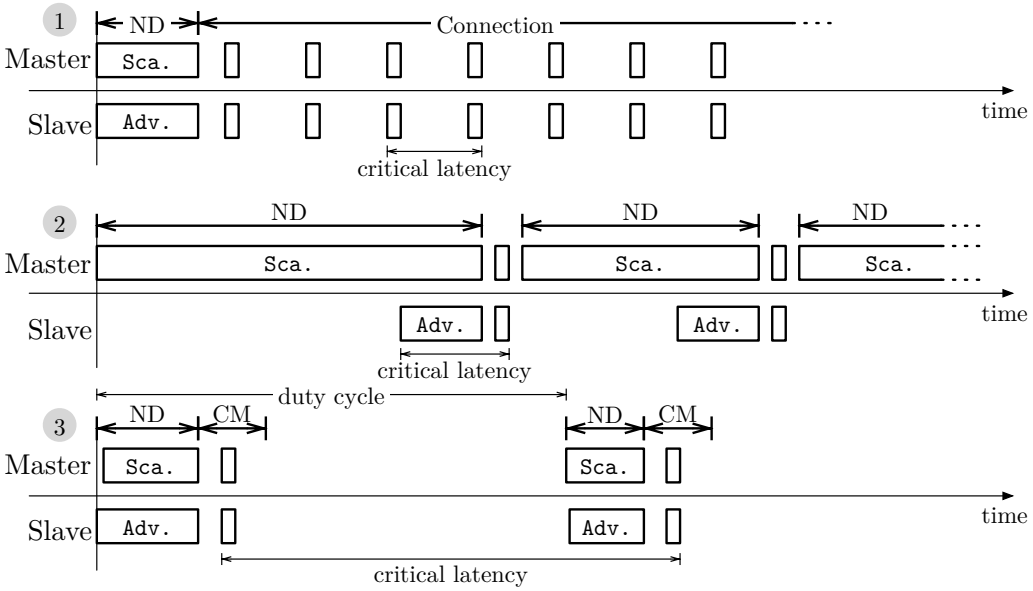


Figure 3.7: Proposed operation modes: 1) Classic BLE, 2) Fully Asynchronous BLE and 3) Duty-Cycled BLE

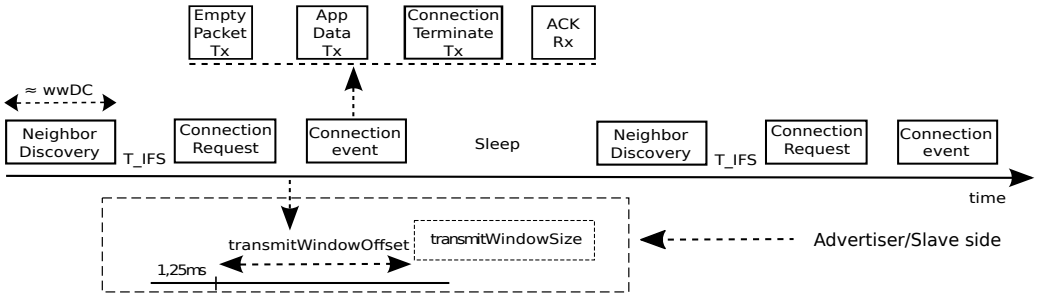


Figure 3.8: Proposed DC-BLE from Master's perspective

Node oscillators do not operate at the exact same frequency. This brings uncertainty in the slave of the exact timing of the master's anchor point due

to clock drift. During CM, slaves are required to re-synchronize to the master's anchor point at each connection event where it listens for the master. BLE spec establishes a listening time called `windowWidening` which is the time before and after the expected anchor point that the slave must listen to ensure a synchronization with the master. Analogously, for DC-BLE mode, where we set up a duty cycle on top of classic BLE, clock drift must be taken into account from the scanner side. During each communication event, we consider a time that the scanner should listen before and after the beginning of the scanning event in order to perform the ND at approximately the same time for both advertiser and scanner, thus increasing the probability of obtaining a low DL. We use the equation given in the specification for CM as shown below:

$$\texttt{windowWidening} = \frac{\texttt{masterSCA} + \texttt{slaveSCA}}{10^6} \times \texttt{tSLA} \tag{3.1}$$

where master Sleep Clock Accuracy (masterSCA) and slave Sleep Clock Accuracy (slaveSCA) are the master and slave sleep clock accuracy and `tSLA` is the time since the beginning of the last scanning event. We have used a 40 ppm clock accuracy as it is a typical value for WSN nodes.

3.4.1 Temperature and Humidity Monitoring use case:

Let's consider one node communicating to a central device to send temperature and humidity readings. The sensing node would periodically wake up, collect the environmental data and transmit this data. Temperature and humidity are monitored at several moments in the day, we consider that the average timing between transmissions is 43 minutes with no constraint on the latency[55].

3.4.2 Light Switches use case:

We also consider a use case in which one or several light switches are used to control one or several lamps. In this case, the main constraint is a CL of 200 ms, which is the maximum latency for a good user experience[32]. Typically, switches are operated only a few times a day.

3.4.3 Lifetime results on Low Duty Cycle test cases

In this section we present the lifetime results when implementing the different operating modes of BLE for the two chosen test-cases explained in the previous section. First we consider one single node communicating to a

central device to send temperature and humidity readings every 43 minutes in average[55]. We compare this scenario using DC-BLE against classic BLE and found that for TI device the advertiser/slave lifetime can reach more than 14 years, approximately 1.13× compared to classic BLE, whereas the scanner/master lifetime can reach more than 7 years when using the DC-BLE scheme which is ≈ 1.8× the lifetime of the master implementing classic BLE, see Table 3.7. When using classic BLE, ST master device is 2.7× better than TI master because the Rx/Tx current consumption of the TI device is 2.3× greater than that of ST. On the other hand, the performance of ST master device in DC-BLE mode is not better than in classic mode, due to the clock drift, which causes a large waste of energy that can be improved by reducing the current consumption during the sleep mode (which is 1.15μA), for which an optimization of the BLE radio is required.

Table 3.7: Lifetime in years for two test-cases, with TI CC2540 and ST BlueNRG devices, in Scanner/Master and Advertiser/Slave roles

		Temperature and Humidity Monitoring		Light Switches	
		TI	ST	TI	ST
Classic BLE	S/M	4.1	10.88	0.25	1.1
	A/S	13.05	14.08	0.25	0.76
Fully Async	S/M	N/A	N/A	0.25	0.25
	A/S	N/A	N/A	14.82	14.26
				58.3×	18.72×
DC-BLE	S/M	7.3	10.46	N/A	N/A
		1.8×	.96×		
	A/S	14.78	14.25	N/A	N/A
		1.13×	1.01×		

For the light switch control system, we first consider one single switch with a BLE device powered with a coin cell battery that communicates to a bulb with a BLE device that can be connected to the mains. The scanner mode is configured at the bulb end and as it is connected to a permanent source of energy, it is feasible to implement continuous scanning during ND, thus ensuring the minimum discovery latency and energy consumption on the advertiser side. As the CL is required to be 200 ms[32], when implemented classic BLE, the minimum T_{SI} is set to 200 ms which would lead to a lifetime of approximately 3.05 months (0.25 year), whereas when communicating asynchronously only 20 times per day sending a few bytes of

application data, the advertiser lifetime could be extended up to 14.82 years (58.3×) in the worst case where the data is not transmitted in the advertising packet but a connection is established and terminated right after the application data exchange as shown in Figure 3.8 in Section 3.4. For further details see Table 3.7.

3.4.3.1 Advertiser energy consumption

When using DC-BLE, regardless of the chosen parameters and even though ND is performed each time prior to a data exchange, our results show that energy consumption at the advertiser/slave side will always be better than classic BLE for periodic low frequency scenarios as long as the communication period (time between each application data exchange) is greater than 76 seconds for TI devices and 260 seconds for ST devices. In other words, DC-BLE can be implemented for applications where CL can be relaxed up to 76 s/260 s or above while still providing a good user experience.

For random low frequency applications, the fully asynchronous BLE is always better than classic BLE at the advertiser/slave side as communication is carried out only when needed and using the minimum amount of energy possible.

3.4.3.2 Scanner energy consumption

We estimate the lifetime of the scanner when using DC-BLE and compare this results against classic BLE. We found that the former provides longer lifetime for periodic and low frequency scenarios, as long as the CL is greater than 86 s for TI device and is never better for ST device because its performance is already better using classic BLE respect to TI device, the only way ST master using DC-BLE can overcome classic BLE is reducing the current consumption during sleep mode. It is important to highlight that at this point (when master lifetime is greater using DC-BLE), the lifetime of the DC-BLE slave is not better than the slave using classic BLE and communication is asymmetric, where the advertiser lifetime reaches up to twice the scanner lifetime. Further discussion about energy symmetry can be found in Section 3.4.3.3. On the other hand, for random low frequency applications, fully asynchronous mode is always worst than the classic mode at the scanner side. A permanent source of energy is the only choice since when powered on a coin cell battery, the lifetime would be reduced to only about 15 h. The main focus when using this mode is to increase the advertiser lifetime. Additionally this configuration can support many advertisers/slaves at the same time while having the minimum latency since these devices do not need to

be associated to the master. An example could be the light switch scenario where several switches can control one or many bulbs asynchronously, where the role of the switch can be a smartphone with no need to trigger an application on the scanner side. Results of comparison between Classic BLE and pour proposed DC-BLE using the TI and the ST device are illustrated in Figure 3.9 and 3.10 respectively.

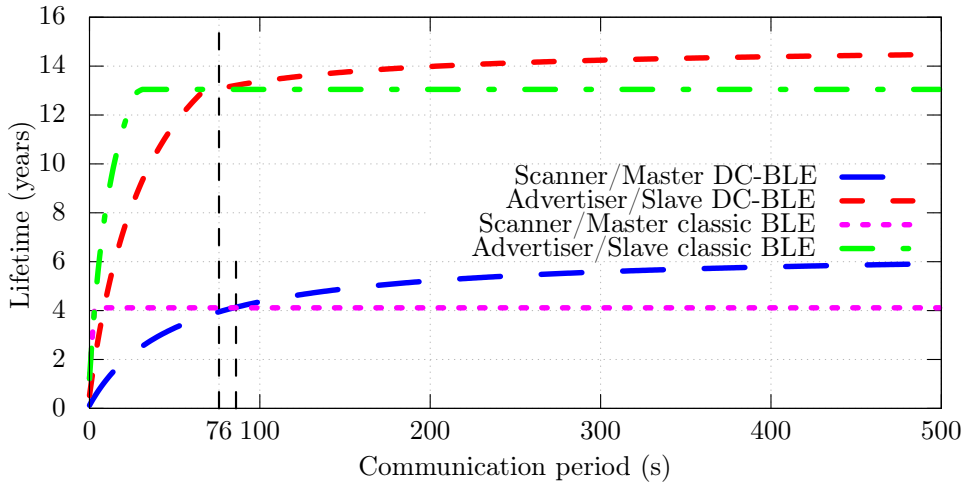


Figure 3.9: Advertiser/Slave and Scanner/Master Lifetime when using TI C2540 devices

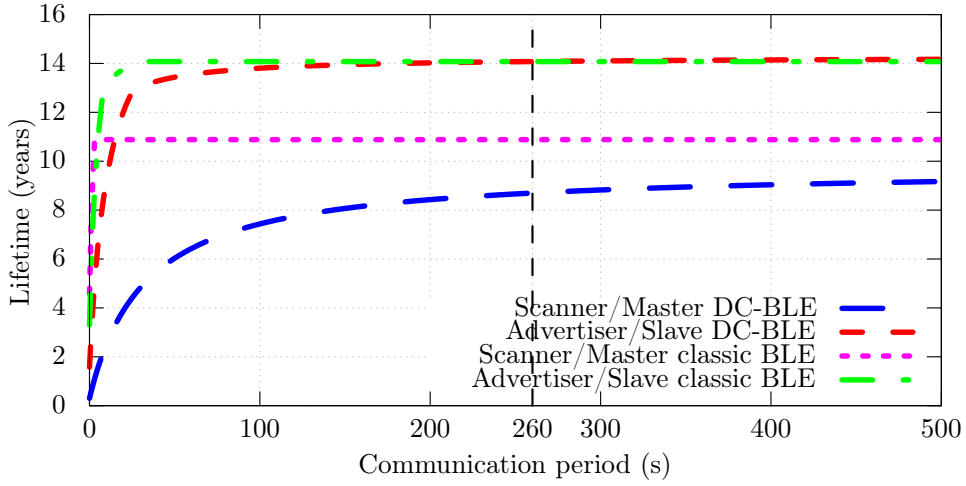


Figure 3.10: Advertiser/Slave and Scanner/Master Lifetime when using BlueNRG devices from ST Microelectronics

3.4.3.3 Scanner/Advertiser Consumption Trade-off

Mesh networks tend to break the master-slave assumption on which Bluetooth is built. Network nodes should be able to relay bidirectional traffic, then BLE devices must be able to switch from master to slave role and provide some kind of symmetry in energy consumption. Recent version 5.0, allows many-to-many connection by adding secondary advertising packets which are transmitted *connectionless* using the data channels, allowing to broadcast synchronous data[29]. Up to version 4.2, the master can connect to several slaves in the same network while a slave can only connect to one single master, and the master would have to perform scanning between connection events in order to listen for new nodes attempting to join the network, in which case the master would never sleep (similar to the fully asynchronous mode). However, multi-hop BLE communication is feasible with only advertising packets, and in this case it is desirable that scanners and advertisers have equal lifetimes. We consider random low frequency scenarios, where nodes should provide some kind of symmetry in energy consumption and be able to relay bidirectional traffic.

Let's consider the fully asynchronous mode, where the master does continuous scanning allowing to scan other devices and achieving the best performance on the slave side, but depleting the scanner's battery very rapidly. A lifetime trade-off between master and slave can be achieved by decreasing the duty cycle on the scanner side, thus extending its lifetime at the expense of decreasing the slave's lifetime and increasing the DL. As shown in Figure 3.11 in the next page, symmetric lifetime is possible when scanner duty cycle is 0.24% for the TI device and 0.58% for the ST device while the communication period on the slave side is of 1.06 s and 1.1 s respectively. the maximum lifetime they could both achieve would be of 0.6 years. As a drawback, DL will not be less than 1.5 s in average, making this solution not compatible for applications with lower required CL.

3.5 Assumptions and Limitations

Some applications appear to be random low frequency, like intrusion alarm systems or medical monitoring devices, since detection events are quite rare, but they actually fall into the continuous high frequency category. These kinds of sensitive applications need continuous monitoring of sensor presence which is implemented using link loss detection in BLE, based on frequent communication events.

In the following, we take the classic assumptions found in the literature

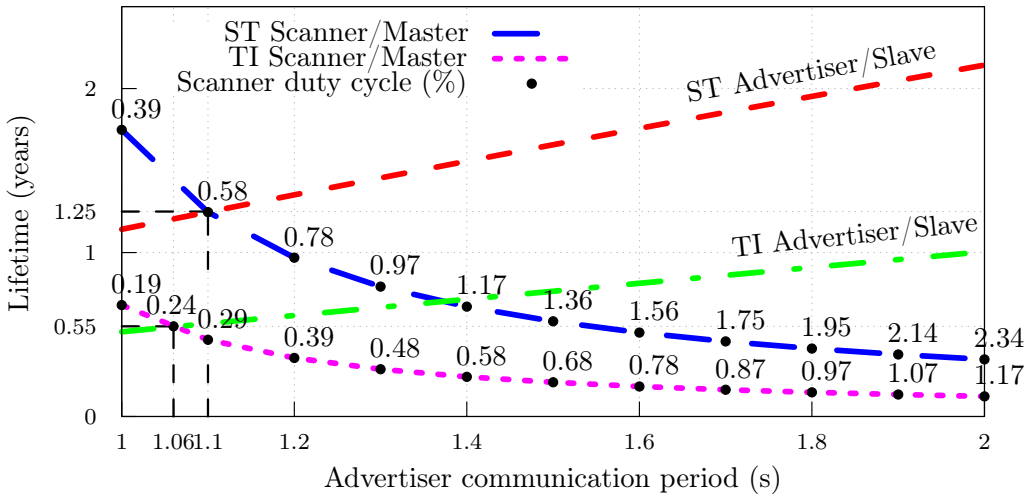


Figure 3.11: Master (central) and Slave (peripheral) trade-off for fully asynchronous mode of operation. Master's lifetime depending on the duty cycle when implementing $T_{SI} = 10.24\,\text{s}$ for the TI device and $T_{SI} = 5.12\,\text{s}$ for the ST device. Slave's lifetime depending on the communication period (application dependent). ST device performance when symmetry is achieved, is $\approx 2.3\times$ better than TI device performance.

we compare to. We focus primarily on point-to-point master-slave communication that are typical of BLE, which is asymmetric in nature. However, we give some insight on how BLE behaves in the more general scenario of multi-hop WSN, which is symmetric and where nodes must act as masters as well as slaves. For the estimation of the energy consumption we do not take into account security, so no Message Integrity Check is considered, which saves 4 B. Also, we consider that no user interaction is required, that `transmitWindowOffset` $= 0$, and that channel conditions are ideal with no interference or collisions.

Multiple slaves cause problems when using connected mode: the master has to schedule connection events for all of them while avoiding collisions and keeping some time for scanning for additional devices. In fully asynchronous and DC-BLE, we do not keep master and slave connected, hence we do not face this problem. Since we consider only one master with one slave in our evaluation, this is the best situation for classic BLE with respect to our proposition.

3.6 Chapter Conclusion

In this Chapter we introduced a new parameter optimization method for BLE based on the DL worst case for each $T_{SI} - T_{SW} - T_{AI}$ configuration. For this, we presented our extension to the model from [4], in order to evaluate devices performance when the scanner is at the user side, thus ensuring to provide with accurate estimation of scanner/advertiser DL and energy consumption and therefore right parameter optimization depending on the application. We provided DL, energy consumption and lifetime results for two typical IoT applications based on our parameter optimization and we compared them with performance results when using the configurations recommended by the Bluetooth SIG. The two applications used to evaluate parameter optimization are: retail store and a medical telemetry use case.

Additionally, we defined a new parameter called CL, as the minimum required time to ensure proper operation and/or user experience for a specific use case. Based on this, we introduced a classification of the scenarios according to their CL and frequency of events occurrence. They are classified into three categories: continuous high frequency, random low frequency and periodic low frequency, for which we proposed three BLE modes of operation: classic BLE, fully asynchronous BLE and duty cycled BLE (DC-BLE) respectively. These BLE modes of operation are designed without modification of the BLE specification and based on our parameter optimization. The idea is to ensure the most efficient implementation possible of the protocol by taking into account the use case requirements. It is sufficient to know within which scenario category the use case falls, and then to apply the right operating mode in order to obtain the best performance possible, thus extending devices lifetime. In order to evaluate the proposed operating modes, we used two use cases: a temperature and humidity monitoring and a light switches use case. We compared the temperature and humidity monitoring use case when using our proposed DC-BLE mode against when using classic BLE and we compared the light switches use case when using fully asynchronous BLE against when using classic BLE.

Using our propositions we have achieved the following lifetime improvements:

- Retail store use case: using the BlueNRG device, advertiser lifetime is 89 times higher. Using the CC2540 device, advertiser lifetime is 105 higher and scanner lifetime is twice higher using both the BlueNRG and the CC2540 device.
- Medical telemetry use case: using the BlueNRG device, advertiser lifetime is 281 times higher. Using the BlueNRG device, advertiser lifetime

is 408 times higher. No improvement is achieved on scanner side for this use case as the optimization is focused only on the advertiser side.

- Temperature and humidity monitoring use case: using the CC2540 device, advertiser lifetime is 13% higher and scanner lifetime is almost twice higher. Using the BlueNRG device, advertiser lifetime is only 1% higher and no improvement is achieved on the scanner side for this device.

- Light switch use case: using the CC2540 device, advertiser lifetime is almost 59 times higher and using the BlueNRG device advertiser lifetime is almost 19 times higher. No improvement is achieved on scanner side for this use case as the optimization is focused only on the advertiser side.

In general we have shown for a total of 4 test cases, how to take the performance of the protocol to its maximum level of energy efficiency without modifying the specification. STMicroelectronics devices show better performance except for the DC-BLE mode due to high current consumption during sleep mode. We also achieved symmetrical lifetime for advertiser and scanner when using fully asynchronous mode which is desirable for implementation of mesh networks.

Chapter 4

Experimental DL Model and Parameter Optimization Validation

In this chapter we present a set of experiments that were driven over a platform called Walt. These experiments aim to check the validity and accuracy of DL and thus energy consumption model presented in Chapter 2. In order to validate our parameter optimization method which is the basis for the proposed modes of operation, we run a series of tests that verify worst and average values of DL and energy consumption for Algorithms 1 and 3. First we introduce the Walt platform to familiarize the reader with its architecture. Then we explain how it was set up to run all tests to finally present the results which are compared against Matlab simulator results.

4.1 Testbed Platform: WalT

We have validated the proposed model on a real testbed developed on top of WalT. WalT is a cheap, reproducible and highly configurable platform for network experiments. WalT started as an attempt to fill the gap between small ad hoc experiments and large scale remote platforms like FIT IoT-lab [56]. A first prototype was developed as a FabLab project at Grenoble-INP Ensimag [57]. Development was then funded by Université Grenoble Alpes, Grenoble INP / UJF, through AGIR 2013-2014 WalT project. The Walt platform is basically composed by 5 different hardware elements as follows:

1. **WalT clients:** from which to access the server and the nodes. Its software only interacts with the WalT server who acts as a gateway

allowing to manage the other elements of the platform. Clients must reach the server through the external LAN.

2. **WalT server:** is the brain of the platform, it interacts with clients, nodes on which we run or drive experiments, switches and the docker hub (repository of WalT images). Any amd64 machine may be installed as a WalT server (in the current version of the platform it is an Intel NUC mini-PC).

3. **PoE switches:** are used in a WalT platform to provide:

   - Remote management through SNMP
   - Power-Over-Ethernet (IEEE 802.3af) capability on all ports (at least those where a WalT node is connected)
   - VLANs (IEEE 802.1q) management (allowing to isolate the WalT platform from the external LAN)
   - Link Layer Discovery Protocol (LLDP)

4. **PoE splitters:** are IEEE 802.3af compliant PoE splitters. They allow to negotiate the power requirements with the PoE switch and provide data and power through 2 dedicated connectors (RJ45 and DC jack connectors respectively)

5. **Raspberry Pi nodes:** these electronic boards are the nodes of the WalT platform. The boot procedure of WalT nodes is made of 2 steps: they first boot a minimal operating system stored on their SD card. Then, the minimal OS allows to connect to the server and boot the WalT image requested.

6. **BLE devices:** are USB-BT400 Mini Bluetooth dongle from Asus which are compatible with Bluetooth 4.0 Core

The architecture of WalT platform is presented in Figure 4.1 visible in the next page.

4.2 Host Controller Interface

Before entering into explanation of testbed set up, it is necessary to understand what is the Host Controller Interface (HCI) of BLE and how it works. As shown in Figure 2.1 in Chapter 2, BLE protocol stack is organized in 3 major building blocks: *Application*, *Host* and *Controller*. The Application

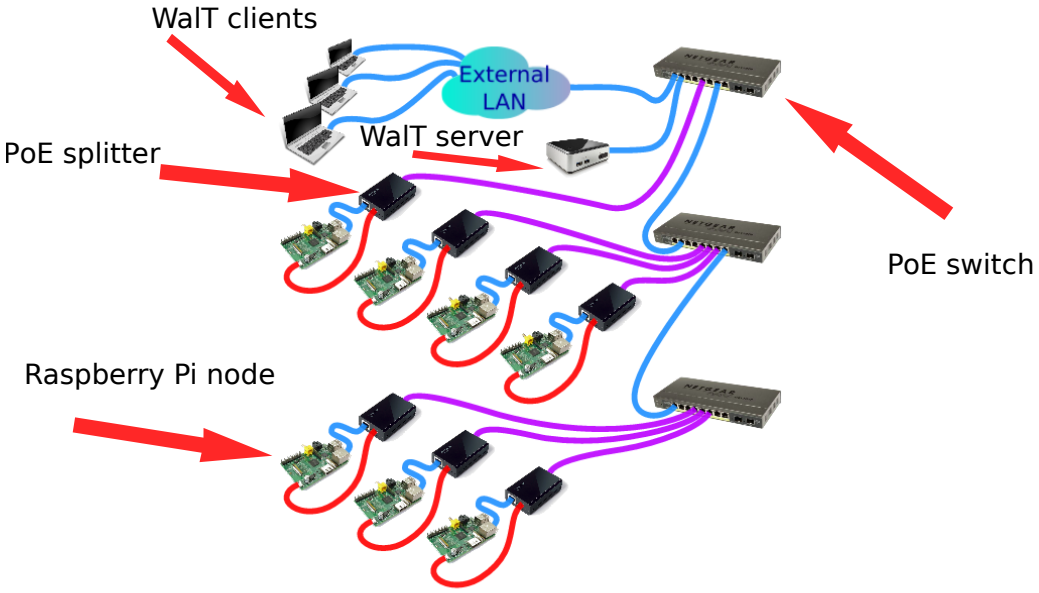


Figure 4.1: WalT Platform. Image extracted from [8]

block is the user application which interfaces with the BLE protocol stack. The Host covers the upper layers of BLE protocol stack, and the Controller covers the lower layers. The Host can communicate with the BLE module with an addition of something called HCI.

The purpose of HCI is to interface the Controller with the Host, it provides a command interface to the baseband controller and access to hardware status. This interface provides a uniform command method of accessing Controller capabilities. The HCI commands provide the Host with the ability to control connections to other BLE Controllers, for which the LL exchanges HCI control packets with remote BLE devices. HCI allows more powerful CPUs to control a BLE device over a serial interface, usually UART or USB. A typical example of this configuration includes most smartphones, tablets, and personal computers, where the host (and the application) runs in the main CPU, while the controller is located in a separate hardware chip connected via a UART or USB. In our case the Host runs in a Raspberry Pi node supporting BlueZ which is the official Linux Bluetooth protocol stack, and the Controller in the USB-BT400.

Figure 4.2 in the next page, shows typical interactions between HCI Commands and Events and the LL. The Figure is extracted from the BLE specifications [1] and it is used here to help the reader understand what happens at the level of abstraction used during our experiments and the interactions occurred between the two BLE devices. Figure 4.2 illustrates the interac-

tions of two devices named A and B. Each device includes a Host and a LL entity in this example where they exchange generic commands, events and group of transactions. In practice, they exchange commands to set advertising data and advertising parameters from Host (BlueZ stack) to Controller (USB-BT400). As response to these commands, notification of successful command is sent from Controller to Host of a device. Finally, as a result of a successful command, advertising packets are transmitted over the air from one device's Controller to another device's Controller. A detailed description of our experiments is given in the following section.

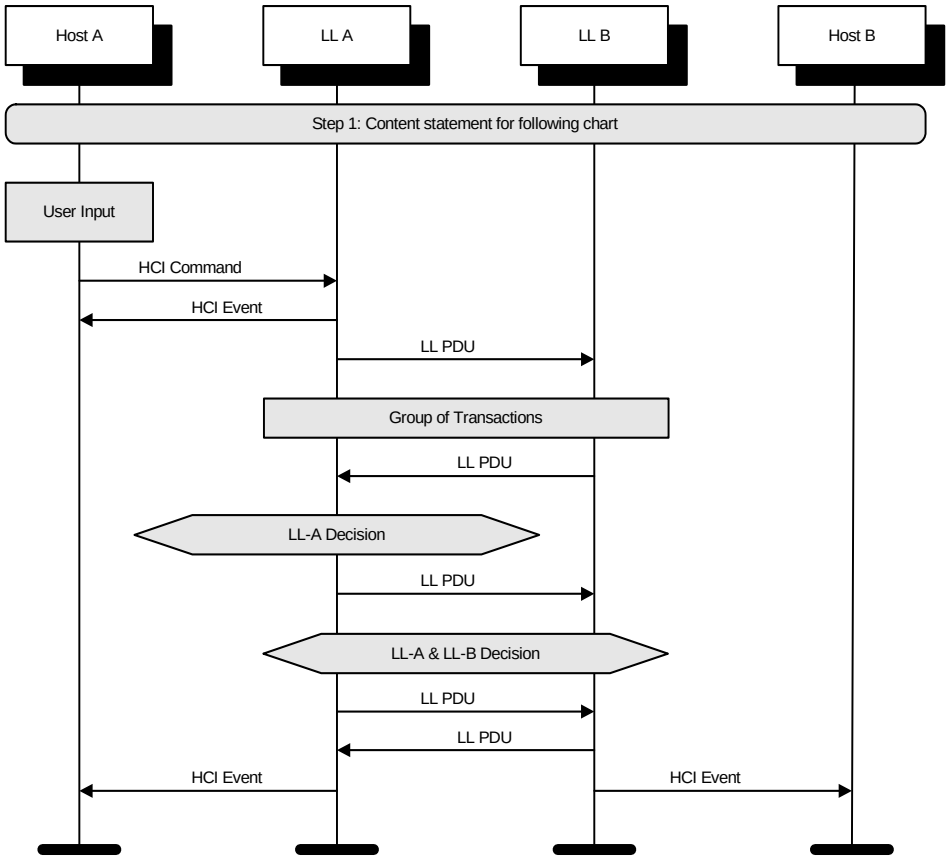


Figure 4.2: Example of message sequence chart (extracted from [1])

4.3 Testbed Setup

We have configured two nodes to play advertiser and scanner role. They have both a set of bash scripts that drive the communication between the

two BLE devices that are connected to each node. The communication is controlled via Bluez. The advertiser contains two scripts, one main script which contains all data related to $T_{SI} - T_{SW} - T_{AI}$ configuration and gets the second script running. This second script is in charge of getting one single scanner node script running as well as controlling the timing of the communication between the two devices.

The advertiser node stores a log file containing all network traffic captured with tshark, which is a terminal oriented version of Wireshark designed for capturing and displaying packets from a given interface. After finalizing with the execution of a test, the log file is then transferred to Walt server in order to process the data with a python based script. This script is in charge of calculating average advertiser and scanner DL, and extracting maximum DL values for a given $T_{SI} - T_{SW} - T_{AI}$ configuration.

In order to validate our parameter optimization method for both Algorithm 1 and 3, the advertiser and the scanner nodes were configured to exchange advertising packets with a given $T_{SI} - T_{SW} - T_{AI}$ configuration which is passed from the advertiser to the scanner as described earlier. Then tshark is started in each node to capture nodes traffic.

Figure 4.3 in the next page, shows the sequence followed to test Algorithm 1 where the advertiser starts advertising after a random phase offset ϕ. Once the advertiser is ready, after node configuration and start of tshark, the advertiser sends the "ready" message to a named pipe (also known as a FIFO) on the scanner, so it can set the time $t = 0$ for both nodes. Then the advertiser wait for a time duration of ϕ before it starts advertising. When the advertising packet is received by the scanner it sends a "Received OK" message to a named pipe on the advertiser and the process stops. This is repeated 100 times for each $T_{SI} - T_{SW} - T_{AI}$ tested configuration and the log is finally sent to the Walt server. Figure 4.4 shows the same test method but this time the advertiser starts advertising at time ϕ before the scanner starts scanning.

One key issue in this test is to set BLE devices configuration correctly and this is done with HCI. To interface Host and Controller, the HCI provides a Command Packet. It is used to send commands to the Controller from the Host. To set the different $T_{SI} - T_{SW} - T_{AI}$ parameters and initiate and stop advertising and scanning when needed we use HCI commands. An HCI command can be broken down as follows:

- **hcitool -i hci0:** a HCI instruction to the Bluetooth Adapter identified as hci0.
- **The Opcode:** Each command is assigned a 2 byte Opcode used to uniquely identify different types of commands. The Opcode parameter

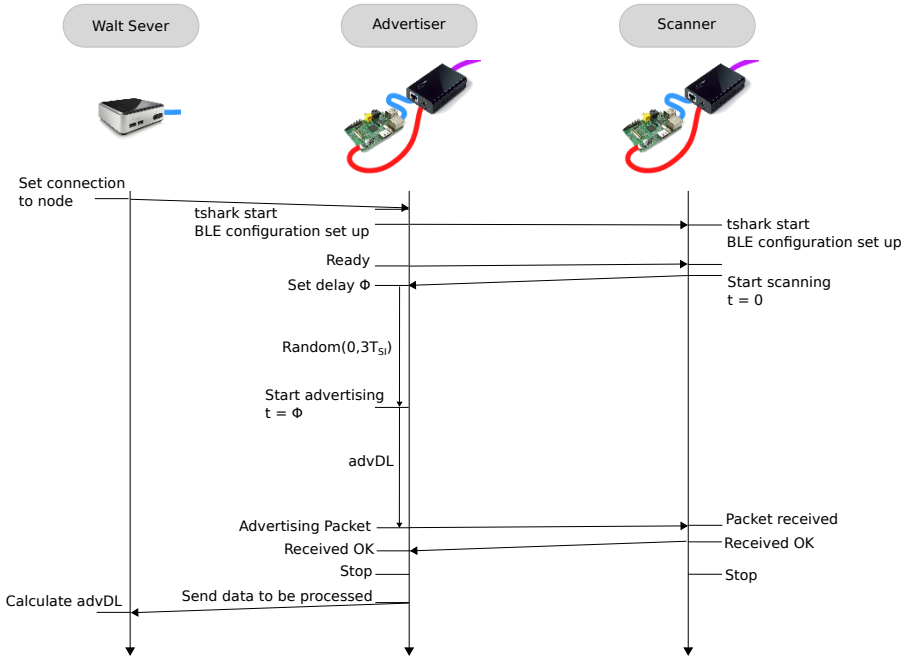


Figure 4.3: WalT Platform Test 1

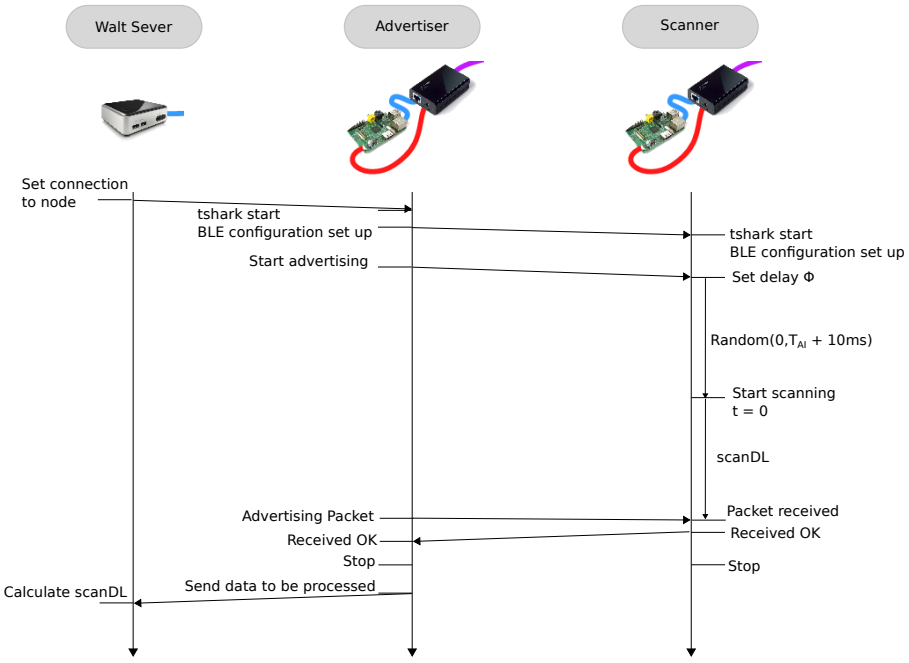


Figure 4.4: WalT Platform Test 2

is divided into two fields, called the OpCode Group Field (OGF) and OpCode Command Field (OCF) as follows:

- **cmd 0x08:** this is the OGF. A HCI Command Packet to the BLE Controller.
- **cmd 0x0008:** this is the OCF. A specific request to the Controller. In this case the request is to set the BLE advertising data.

Along the test, a set of HCI commands are required. They are listed in Table 4.1.

Table 4.1: HCI commands used to configure and control BLE advertising and scanning

OCF	Arguments/Description					
0x000a	01					
Set adv. enable	Enable					
0x0008	9 Bytes preamble	16 Bytes UUID	16 00	02 00	C5	00
Set adv. data	iBeacon prefix	Adv. data	Major	Minor	Power	Unused
0x0006	A0 00 A0 00	00	00 00	5C:F3:70:66:16:4B	07	00
Set adv. param.	min and max T_{AI}	Adv. type	Add. and direct add. type	MAC add.	Adv. ch	allow scan/conn req.
0x000c	01	01				
Set sca. enable	Enable	Filter dup. true				
0x000b	01	12 00	10 00	00	00	
Set sca. param.	Sca. type	T_{SI}	T_{SW}	Add. type	All adv.	

4.4 Experiment Results

In order to validate the model and the parameter optimization method, we performed a series of tests in Walt platform. Figure 4.5 shows the average advertiser DL estimated for $T_{SI} = 400ms, T_{SW} = 300ms$.

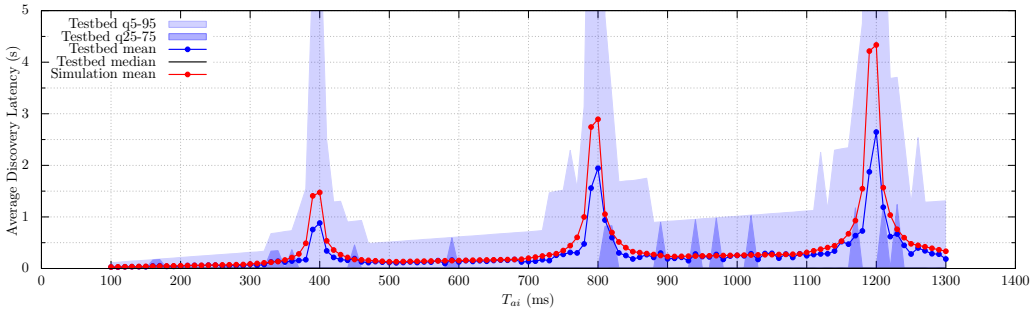


Figure 4.5: Advertiser discovery latency validation for Algorithm 1

Average DL has been estimated with Matlab simulator and compared with experimental results obtained from the testbed according to the Figure 4.3 from Section 4.3. Similarly, Algorithm 3 was validated according to

the Figure 4.4 from Section 4.3. We obtained average scanner DL results and is shown in Figure 4.6. The results correspond to a configuration of $T_{SI} = 400ms, T_{SW} = 300ms$ and $T_{AI} = [20ms, 10.24s]$. Results are illustrated in increments of $20ms$.

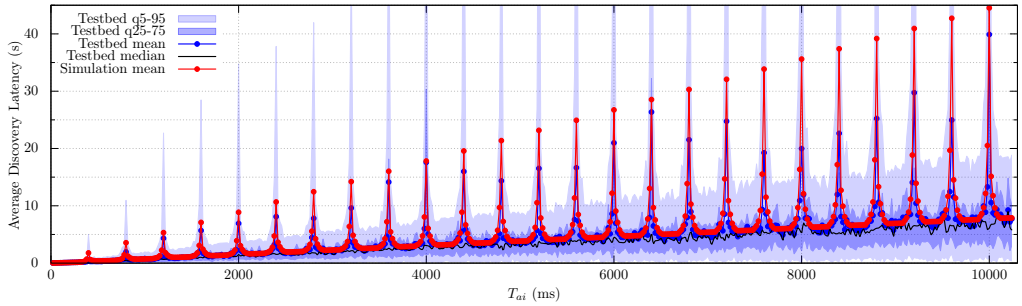


Figure 4.6: Scanner discovery latency validation for Algorithm 3

For the sake of legibility, we show detailed statistics only for testbed results and only the mean for simulation. In Figure 4.5 the advertiser is peripheral, and in Figure 4.6 the scanner is peripheral from application point of view. High average DL is obtained near the values of T_{AI} that are multiples of T_{SI}, corresponding to the peaks observed in both figures, as it was observed in previous work too[4]. Around these values, if an advertising event starts between the end of the scanning window and the beginning of the next scanning event, then the next advertising event is very likely to be missed, as the random delay ρ is not enough to compensate the difference between T_{SI} and T_{SW}. The peaks should be avoided in configurations, as very large average and worst case DL are obtained for these configurations, resulting in higher energy consumption and bad user experience.

Experimental results show the same pattern as the results obtained from simulations of the model. As said earlier, statistical details for simulation results were omitted from the Figures, since they fit perfectly to experimental results, except for extreme mean and maximum values at the peaks, they are just more regular as expected since they were obtained from an ideal system, with a very large number of trials. In both cases, the peaks are exactly at the same location, which means that the model and experiments behave similarly. However, we can see that there is a significant difference in the extreme values. This can be explained easily by the fact that we drive experiments from user space, where high precision timing is impossible to achieve. This introduces a high variability in the results where the curve is very steep since a small imprecision in the timing, can result in a large variation in the measured delay. We notice on the graph that the distribution

of values is very asymmetric, the median stays low, but the mean grows very quickly at the peaks, the 1^{st} and 3^{rd} quartiles are close, but the 0.05 and 0.95 quantiles show much more variability: there are few occurrences where the DL is very large but they have a very big impact. Also, these occurrences are catastrophic for the user experience, we reached a worst case scanner discovery latency of 255 s in our experiments, more than 4 min!

Once the model has been validated we have proceeded to validate the parameter optimization method. We have repeated the test process presented in Figures 4.3 and 4.4, 100 times for each of the configurations contained in Tables 3.1 and 3.2 from Chapter 3. As it can be seen from Tables 4.2 and 4.3 results are very accurate with a maximum error of $\approx 3\%$.

Table 4.2: Parameter Optimization respect to advertiser CL

CL	T_{SI}	T_{SW}	T_{AI}	**Matlab**	**Walt**	**error**
200 ms	400 ms	300 ms	190 ms	198.1 ms	204.40 ms	3.08 %
300 ms	500 ms	400 ms	290 ms	298.1 ms	303.51 ms	1.78 %
400 ms	60 ms	50 ms	390 ms	398.1 ms	406.02 ms	1.95 %
500 ms	500 ms	500 ms	490 ms	498.1 ms	505.82 ms	1.53 %
600 ms	300 ms	300 ms	590 ms	598.1 ms	596.31 ms	0.30 %
700 ms	200 ms	200 ms	690 ms	698.1 ms	706.19 ms	1.15 %
800 ms	70 ms	60 ms	790 ms	798.1 ms	803.81 ms	0.70 %
900 ms	2.56 s	1.92 ms	890 ms	898.1 ms	905.61 ms	0.83 %
1 s	1.92 ms	1.28 ms	990 ms	998.1 ms	1.006 s	0.79 %

Table 4.3: Parameter Optimization respect to scanner CL

CL	T_{SI}	T_{SW}	T_{AI}	**Matlab**	**Walt**	**error**
200 ms	400 ms	300 ms	190 ms	204.2 ms	200.76 ms	1.68 %
300 ms	600 ms	400 ms	290 ms	304.2 ms	306.76 ms	0.83 %
400 ms	800 ms	500 ms	390 ms	404.2 ms	403.79 ms	0.10 %
500 ms	900 ms	600 ms	490 ms	504.2 ms	504.82 ms	0.12 %
600 ms	1.28 s	640 ms	590 ms	604.2 ms	601.83 ms	0.39 %
700 ms	2.56 s	1.28 s	690 ms	704.2 ms	688.89 ms	2.17 %
800 ms	3.2 s	1.28 s	790 ms	804.2 ms	801.84 ms	0.29 %
900 ms	3.84 s	1.28 s	890 ms	904.2 ms	892.81 ms	1.26 %
1 s	4.48 s	1.28 s	990 ms	1004.2 s	987.87 ms	1.63 %

4.5 Chapter Conclusion

In this Chapter we validated the accuracy of DL and thus energy consumption model presented in Chapter 2. For this we used Walt platform. We used two Raspberry Pi nodes, one configured as the advertiser and one configured as scanner. They both contain a set of bash scripts in charge the communication between the two nodes, where the main node is contained in the advertiser node. For setting $T_{SI} - T_{SW} - T_{AI}$ configurations and communication between the two nodes we used BlueZ which is the official Linux Bluetooth protocol stack.

We tested all the configurations considered in Chapter 3 for the parameter optimization. For each configuration, advertising packets are sent from the advertiser until its reception at the scanner side and the process is repeated 100 times for each configuration. Packet traffic is captured via tshark (a terminal oriented version of Wireshark) and stored in a log file which is transferred from the advertiser node to the Walt server. In the Walt server, the log is processed by a set of Python scripts in order to calculate the worst case for DL values and average DL for each configuration. For the worst case of the parameter optimization validation we obtained a maximum of ≈ 3 % error for both Algorithm 1 and 3. Results are shown in Tables 4.2 and 4.3. We also provided with the corresponding energy consumption for these parameters in Figure **??** which shows that the best performance is achieved at the user side (at advertiser side for Algorithm 1 and scanner side for Algorithm 3).

For validation of the average DL, we executed the test for $T_{SI} = 400ms$, $T_{SW} = 300ms$ and $T_{AI} = [20ms, 10.24s]$ in steps of $20ms$. For both Algorithm 1 and 3 resulted very accurate except for the peaks (where T_{AI} is a multiple of T_{SI}). For Algorithm 1, experimental results show lower value in the peaks compared to the Matlab simulations and for Algorithm 3 experimental results show in general higher values in the peaks compared to the Matlab simulations. Apart from the peaks, which are actually to avoid because of the high DL and energy consumption associated; results are very accurate which validates our simulations based on the parameter optimization method.

Chapter 5

Enhancing BLE with Wake-up Radio

For synchronous communication, a reference clock is used to assign time slots to every node in the network. As drawback, maintaining and distributing the clock can be a difficult task and furthermore, expensive in terms of energy consumption. BLE in particular implements packets of up to 29 Bytes in order to control a LL connection. Parameters such as T_{CI}, Slave Latency, T_{ST}, etc., are sent by the master to every slave so that both nodes can be awake at the same time and establish a communication, these parameters are updated numerous times along the connection, either by request of the slave or autonomously by the master to change scheduling of the connection due to other activities, thereby spending important amount of energy.

In order to reduce latency and energy consumption due to idle listening, an asynchronous scheme is a favored choice. A radio transceiver typically consumes the highest percentage of the total power in a sensor node, (around 75% [58]). By using asynchronous communication, higher-power communication radio is kept in sleep mode for long periods of time and woken up asynchronously only when needed. A technique used to achieve this is to implement a WuR. A WuR is a dedicated receiver with ultra low power consumption that continuously monitors the channel, listening for a wake-up signal from other nodes and activating the main receiver upon detection.

The aim of this Chapter is to provide with estimation of BLE energy consumption based on analytical models when including asynchronous schemes in BLE and compare its energy consumption against classical BLE, thus determining under which configurations and scenarios, WuR approach would reduce energy consumption. Based on simulations under Matlab environment, we introduce the WuR approach to substitute the typical scheme of ND phase, being the most critical one. We analyze the performance of this

approach under some use cases with respect to critical metrics such as energy consumption and latency. In this Chapter, we provide with the results to determine the feasibility and benefits of such approach for different kinds of applications.

5.1 What is the Wake-up Radio?

Even though WSN protocols implement a duty cycle, there's a great waste of energy in idle listening. Devices spend a lot of time waiting for packet's arrival, channel scanning for active devices and receiving/transmitting packets intended for update and synchronization activities; which means that devices do not exclusively connect for application data exchange. In order to palliate this issue, one solution is to implement a very low power radio receiver who is in charge of monitoring the communication channel waiting for a signal, while the main receiver, the one intended to actually receive the data, spends most of its time in sleep mode and thus, saving considerable amount of energy. This very low power radio which is used as a supplementary receiver, is better known as WuR.

In general, the idea is that each node within a network is equipped with two radios: a main radio which is mostly powered off and is used to receive application data, and a second radio that operates with poorer sensitivity compared to the main one and operates with ultra-low power. This is the WuR who is able to wake up the main radio in the same node, when there is a reception of a packet which is directed to that specific node.

Figure 5.1 shows the principle of the WuR having a transmitting and a receiving node.

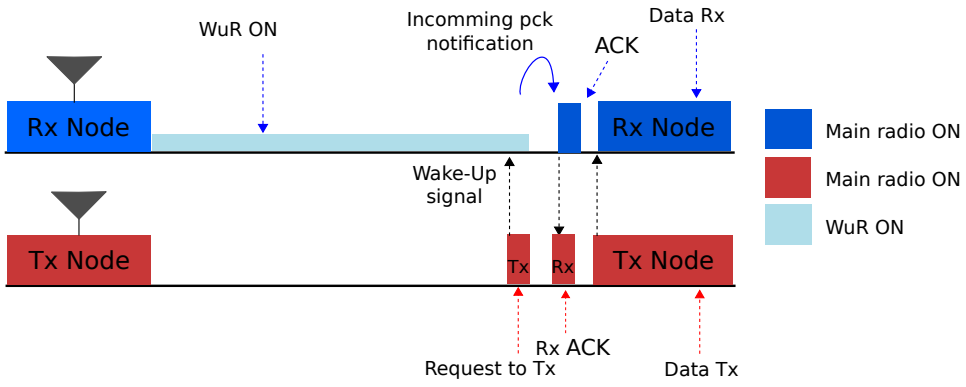


Figure 5.1: Wake-Up radio principle

The receiving node is monitoring the channel using its ultra-low power radio, the WuR; waiting for an incoming signal while the main radio, which consumes a considerably higher amount of energy, remains in sleep mode. When the transmitting node has data to send, it first sends a request to transmit (also known as wake-up signal) to the receiving node. Once the WuR on the receiving node detects the wake-up signal, it wakes up the main radio who sends and ACK to the transmitting node, indicating it is ready to receive the data. Then, the data packet containing application data is finally sent. After successful data packet reception, the main radio of both nodes go back to sleep mode and the WuR starts monitoring the communication channel again, waiting for a new wake-up signal.

Moreover, there is also the possibility to implement a duty cycle during the wake-up signal monitoring. Even though the WuR consumes very little power, implementing a duty cycle, would save important amount of energy on the receiving node side in a long term, thus extending its battery lifetime. Figure 5.2 shows the case when a WuR operates usign a duty cycle.

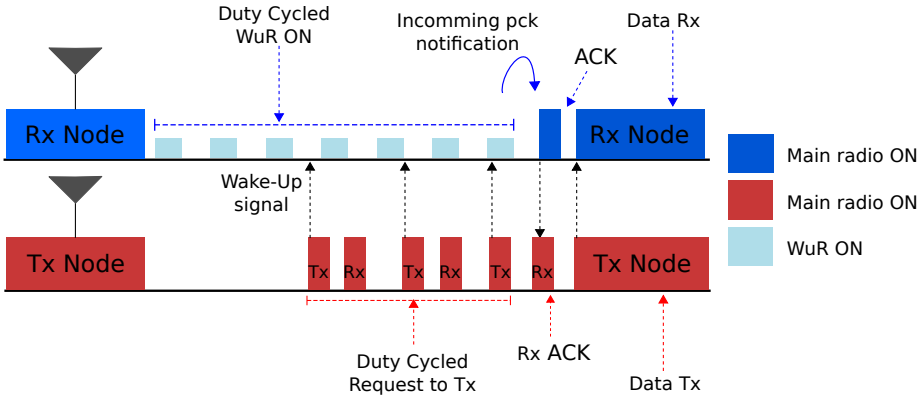


Figure 5.2: Duty cycled Wake-Up radio principle

5.2 Wake-up Radio Architecture

In order to familiarize the reader with WuR state of the art, in this section we present an overview of the different WuR architecture. Before entering into receivers architecture description, it is noteworthy that there are two types of receivers: active and passive receivers. The difference between them lies in the type of components which are used in its circuitry: passive or active. Active components are those able to control the current flow of the circuits and usually can inject power into a circuit (produce gains). They are mainly

electric generators and certain semiconductor components. The latter, in general, have a non-linear behavior, meaning that the relation between the applied voltage and the demanded current is not linear. Active components include amplifying components such as transistors, diodes, triacs, operational amplifiers, etc. On the other hand, passive components can't introduce net energy into the circuit. They can't amplify (increase the power of a signal), although they may increase a voltage or current (such as is done by a transformer or resonant circuit). Passive components include two-terminal components such as resistors, capacitors, inductors, and transformers.

Active receivers are in general very complex since they implement active components to achieve considerably higher sensitivity and data throughput, compared to what is possible to achieve with passive components based receivers. As drawback, these active components based receivers need to consume a greater amount of energy for their operation. Sensitivity is a very important figure of merit for these circuits, since depending on the sensitivity, a greater or lesser range is achieved. This is, the higher the sensitivity, the higher the distance in meters between the transmitting and the receiving node for which it is still possible for the receiving node to detect the transmitted signal. For this reason, implementing a circuit with a good trade-off between consumed power and performance is of capital importance.

A common technique for passive receivers is the implementation of very simple Radio Frequency Identification (RFID) systems, which do not even need for an energy source and for this reason, these are most simple and low energy type of wireless receivers. RFID is a remote data storage and retrieval system that uses devices called RFID tags. The purpose of RFID technology is to transmit the identity of an object using RF signals. A RFID tag can be either active or passive. Passive tags do not need internal power supply, while active ones do. For passive tags, an incoming signal induces a small and sufficient electrical current to operate the integrated circuit of the tag, so that it can generate and transmit a response. Although this would make RFID based receivers a formidable candidate for its implementation for WuR, the poor sensitivity of a receiver of this type, brings about the need of using high output power at the transmitting node side or much improved sensitivity tags [59], in order to increase the range which is the distance between transmitting and receiving node. Some recent investigations focused efforts on the trade-off between tag sensitivity and range as described in next paragraph.

Authors in [60], have presented a passive RFID tag that achieves a sensitivity of -21.2 dBm on the 866.4/925 MHz band with a data rate between 53.3 Kbps and 106.7 kbps. The solution achieves a maximum range of 19.6 meters for an Effective Isotropic Radiated Power (EIRP) of 36 dBm on the transmitter side, and having a power consumption of $3.244\mu W$ measured at

the 925 MHz port of the tag. Authors in [61] presented a passive RFID tag having a sensitivity of −18 dBm for a 40.5 dBm of EIRP on the transmitter side at 912.5 MHz, and the measured maximum read range of the proposed tag is 16.7 meters. Also, as an alternative to high cost and complexity added to WuR receivers based on RFID tags, authors in [62] proposed an RF energy harvesting wireless sensor node that can achieve a range of 1.16 meters with 13 dBm transmit power and can achieve a range of 9 meters. This solution implements a data rate of 33.33 kbps. Bringing BLE into this context, as an example of common BLE current solutions, the TI CC2541 device used in this work for evaluation has a maximum output power of 0 dBm on Tx and a sensitivity of −94 dBm (at 1 Mbps) with a data rate going from 250 kbps up to 2 Mbps, whereas ST BlueNRG device has a maximum outpout power of 8 dBm, a sensitivity of −88 dBm and a data rate of 1 Mbps, and both devices work on the 2.4 GHz band. Meaning that, such described tags are not viable solutions for existing BLE transceivers.

With respect to active receivers, different high level architectures are used. Architectures are categorize according with the type of frequency conversion implemented in the circuitry. In general, a receiver must extract the original information-bearing signal (also known as baseband signal) from a modulated carrier wave. The baseband signal describes the state of the signal before the modulation and it has a very narrow and near-zero frequency range. In order to recover the information content from the modulated carrier wave which is usually at very high frequency, a receiver will filter or separate a very small current, which is generated in the antenna and then selectively amplify it. In order to extract the baseband signal from the modulated carrier wave, the input signal is shifted to (usually) lower frequency to ease implementation of signal processing blocks such as gain and filtering. For this, receivers implement different architectures which can be called frequency conversion architectures.

Typical frequency conversion architectures which are adopted for WuR design are: direct down conversion or low-IF [63], also known as homodyne, or zero-IF receiver, the incoming signal is demodulated using synchronous detection driven by a local oscillator whose frequency is identical to, or very close to the carrier frequency. Envelope detection (tuned-RF) architecture [64], takes the high-frequency incoming signal and provides an output which is the envelope of the original signal, also known as AM demodulator. Uncertain-IF architecture [65], is similar to a heterodyne receiver architecture, the difference is that the intermediate frequency (IF) is not fixed. It may change within some range defined by the less accurate digitally tunable ring oscillator of the circuit. Super-regenerative architecture [66], a regenerative circuit employs an amount of positive feedback (which is also known

as regeneration). Part of the output is fed back to the input without phase inversion, to reinforce the signal. This allows a signal to be amplified many times by the same active device. A receiver that uses larger amounts of regeneration in a more complicated way to achieve even higher amplification, is called superregenerative receiver.

On the other hand, for WuR, the most common used modulation technique is On-Off Keying (OOK). It is the simplest form of amplitude-shift keying (ASK) modulation that represents digital data at the presence or absence of a carrier wave. It is the preferred option due to the low power consumption that can be achieved with its implementation.

5.3 Wake-up Radio in Wireless Sensor Network

WuR represents a viable solution for WSN as long as its power consumption remains limited to 50 μW [59, 67, 68], which offers good compromise between low power consumption and high sensitivity. Recent research [69] shows that implementing WuR benefits WSN by providing with substantial energy savings and demonstrates that switching from duty-cycled MAC-based networks to WuR-based networks represents the future of WSN.

We strongly consider that the road to break through barriers towards full implementation of BLE within IoT environments, includes the implementation of WuR as it provides with the possibility of optimizing nodes energy consumption without compromising the latency. In addition, implementation of BLE within the IoT increases the possibility of interoperability between devices which is, although not in the scope of this paper, one of the main challenges of the IoT; considering that BLE has significant presence within diverse areas such as: beacons and retail, consumer electronics, health and wellness, mobile telephony (two billion devices by 2018), computers, sports and fitness, home automation and wearable technology [70].

5.4 Is Wake-Up Radio Interesting to Enhance BLE?

Very recent research has been conducted in order to evaluate BLE performance when implementing a WuR. Authors in [71] focus on a BLE beacon use case used for providing location-based services based on the proximity to the BLE device. In particular authors focus on crowded situations with

several beacons within communication range. They evaluate the case where a beacon (peripheral- advertiser) device carries a WuR to upload data during a connection oriented data exchange as well as broadcast oriented communication. They calculated the time reduction during data download to be 40% in the case of connection oriented communication and by 60% in the case of broadcast oriented communication (only true for large amount of beacons (more than 100)).

5.4.1 BLE with Wake-up Radio for periodic low frequency scenarios

In this section we evaluate BLE performance with WuR implementation, for a much wider range of IoT applications, by combining the different operating modes presented in Chapter 3, together with the use of the WuR. We don't focus on a particular WuR architecture, instead we target the use of a WuR that meets the power consumption requirements mentioned in the previous section, this is $50\mu W$. Based on this, we have considered a 2.45 GHz WuR presented in [72], which has a power consumption of $50\mu W$ and voltage supply of $0.75V$ and is designed to receive data modulated with OOK. Accordingly, energy consumption and battery lifetime are estimated considering a WuR with a current consumption such that:

$$\mathrm{I}_{n_{th}state} = \frac{50\mu W}{supplied\ voltage} \quad (5.1)$$

Where $I_{n_{th}state}$ is the current consumption considered in our Matlab based simulations, during the WuR n_{th} reception state. And the *supplied voltage* is $0.75V$ as presented in [72]. Figure 5.3 shows the DC-BLE operating mode implementing WuR during ND process at scanner/master side.

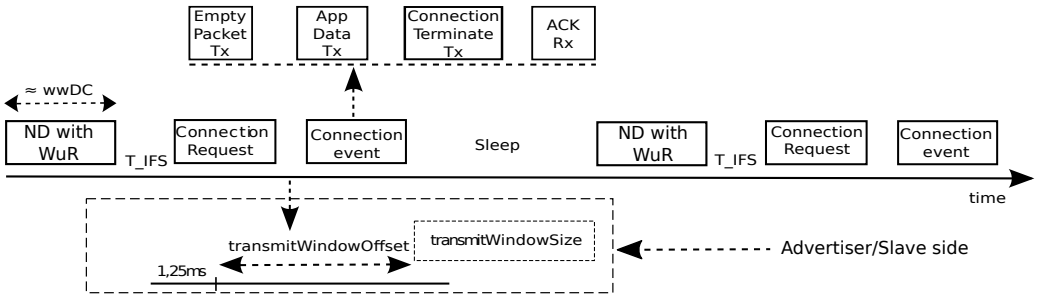


Figure 5.3: Master's perspective of proposed DC-BLE with Wake-Up radio

We have evaluated BLE performance for low duty cycle applications such as presented in Chapter 3, with the implementation of the mentioned WuR from [72]. When using the DC-BLE mode, the time a device will spend in ND will depend on the masterSCA and the communication period required for a specific use case. The higher the communication period, the lower the duty cycle and the higher the ratio between the time in ND and the time in CM. To give an idea of the proportion between the time that the devices spend in ND and the rest of the time of a cycle, here we provide with an example of devices behavior for a communication period of $1s$. The estimated duty cycle for both TI and STMicroelectronics devices is less than 2% and the time an advertiser device spends in ND represents $\approx 90\%$ of the total active time of the operating cycle. On the other hand for a scanner device, the duty cycle is slightly higher compared to the advertiser, and it is less than 3% and the time it spends in ND represents $\approx 94\%$ of the active time of the cycle. For this reason, by implementing the WuR on the scanner/master device during ND, a significant reduction of the energy consumption is expected, and thus extension of the battery lifetime.

Figures 5.4 and 5.5 (visible in the next page) illustrate battery lifetime results comparison for the TI and STMicroelectronics devices respectively when non duty cycled WuR scheme such as the one depicted in Figure 5.1 is implemented during ND.

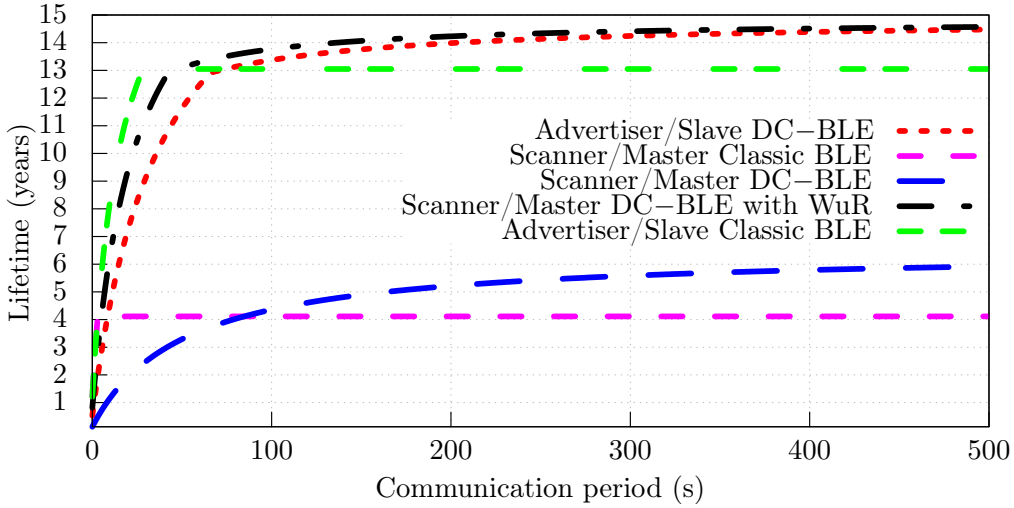


Figure 5.4: TI devices lifetime in DC-BLE mode with Wake-Up radio

Results correspond to a communication period between $1s$ and $500s$. In both cases results show that scanner lifetime using DC-BLE with WuR can be equal or even grater than advertiser lifetime using the DC-BLE mode. In

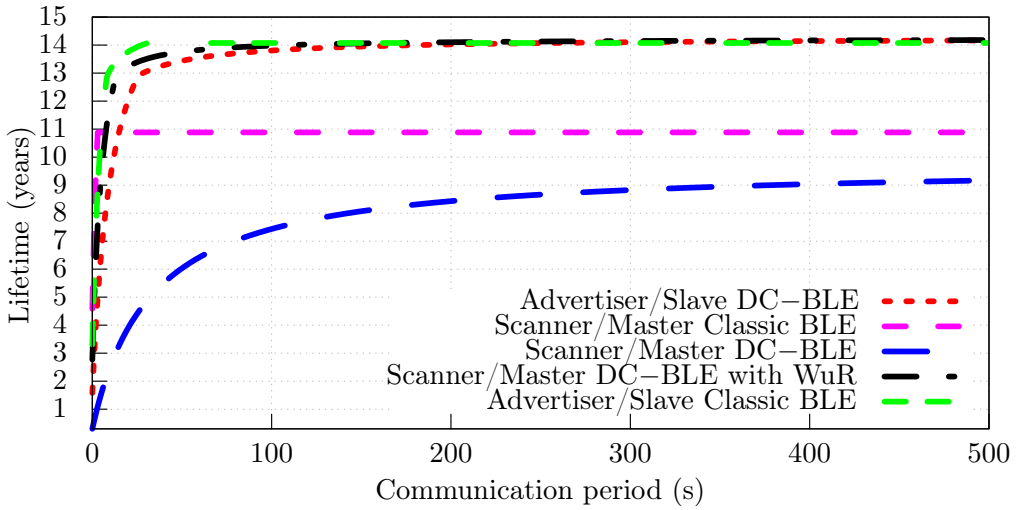


Figure 5.5: STMicroelectronics devices lifetime in DC-BLE mode with Wake-Up radio

Figure 5.4 it can be seen that, for TI devices resulting scanner lifetime using DC-BLE with WuR is improved respect to Classic BLE when the communication period is $\geq 7s$, which is ≈ 12 times better than DC-BLE without WuR, for which the communication period with better lifetime than Classic BLE is $\geq 86s$. According to our results, scanner lifetime using DC-BLE with WuR can be from ≈ 2.5 to ≈ 6 times higher compared to DC-BLE without WuR, for a communication period of $500s$ and $1s$ respectively. Whereas using DC-BLE with WuR can be from ≈ 1.1 to ≈ 3.5 times higher compared to Classic BLE, for a communication period of $7s$ and $500s$ respectively.

On the other hand, from Figure 5.5 it can be seen that for STMicroelectronics devices, scanner lifetime using DC-BLE with WuR is improved respect to Classic BLE when the communication period is $\geq 9s$, which is better than DC-BLE without WuR for all communication period, since the later is never better than Classic BLE. Similar as with TI device, we analyzed the ratio of DC-BLE with WuR lifetime with respect to Classic BLE and DC-BLE without WuR. With respect to Classic BLE, lifetime is from ≈ 1.02 to ≈ 1.3 times higher for a communication period of $1s$ and $500s$ respectively. And with respect to DC-BLE without WuR, lifetime is ≈ 1.5 to ≈ 9 times higher, for a communication period of $500s$ and $1s$ respectively. For STMicroelectronics scanner, lifetime improvements in not as significant as with TI device because lifetime results using Classic BLE are already better for the STMicroelectronics device compared to the TI device. However, the STMicroelectronics device shows a much more significant improvement

if implementing the WuR compared to the Classic BLE mode.

Figures 5.6 and 5.7 show lifetime results comparison for TI and STMicroelectronics devices respectively, when WuR is implemented during ND as depicted in Figure 5.2 (duty cycled WuR) vs. non duty cycled WuR.

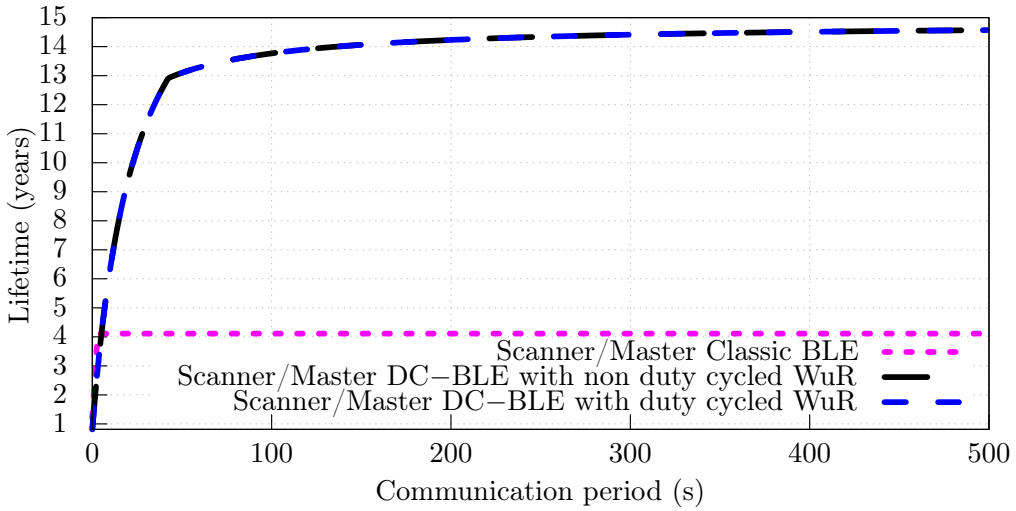


Figure 5.6: TI scanner/master device lifetime in DC-BLE mode with duty-cycled Wake-Up radio

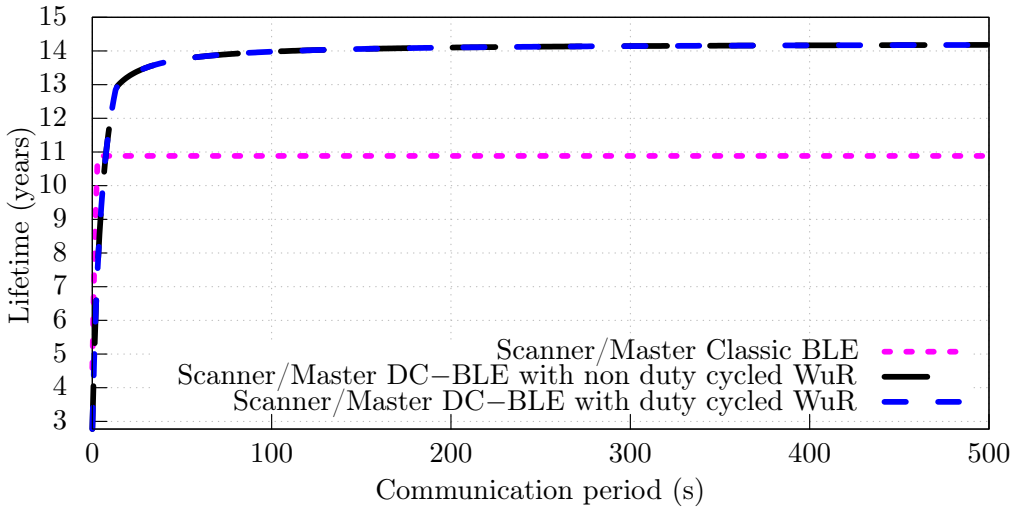


Figure 5.7: STMicroelectronics scanner/master device lifetime in DC-BLE mode with duty-cycled Wake-Up radio

It can be seen that there is no difference for this operating mode between

the duty cycled and the non duty cycled WuR. The time spent in ND, even though very high compared to the time spent in cm is not long enough to produce a important difference between the average current consumption during ND for this two modes, and as a result the same performance is obtained.

In conclusion, the interest of implementing a WuR when using the DC-BLE on the TI scanner/master device, occurs when the duty cycle is $\geq$ 0.36%, where WuR is implemented during ND which represents a 93.7% of the total active phase of the duty cycle. Whereas for the scanner/master STMicroelectronics device, the implementation of the WuR is advantageous for a duty cycle $\geq$ 0.26%, where the WuR utilization represents a 97.08% of the active part of the duty cycle. A non duty cycled WuR scheme can be implemented, which means continuous scanning during ND obtaining the same lifetime as with duty cycled wur scheme, and reducing DL and energy consumption at the advertiser side, since as it is already known, the minimum DL and energy consumption is obtained for the advertiser when a continuous scanning is performed.

5.4.2 BLE with Wake-up Radio for continuous high frequency scenarios

Different from classic BLE, we don't consider a connection oriented communication. Nevertheless, application data packets are sent/received through the data channel and standard payload size is considered according to the Bluetooth Core. The main difference lies in the fact that before using the data channel, we propose not to use the transmit window offset and transmit window size parameters (in order to reduce DL and energy consumption), since devices are intended to exchange data and to immediately go back to fully asynchronous mode. This is due to the fact that in this case, application data can not be sent within the advertising packet, since the scanning is performed with the WuR, so sending application data within advertising packets is not possible compared to the fully asynchronous mode as evaluated in Chapter 3.

This mode of operation is not included in the Bluetooth specification, so either a modification would be needed in order to use the data channel as previously exposed or either the possibility to send larger size payload within advertising packets such as in CM is required. Figure 5.8 visible in the next page, represents our proposed operating mode including the WuR for a TI device. The WuR listens for a wake-up signal using advertising channels, an acknowledgment packet is transmitted using advertising channels, and

application data reception is performed using BLE data channels.

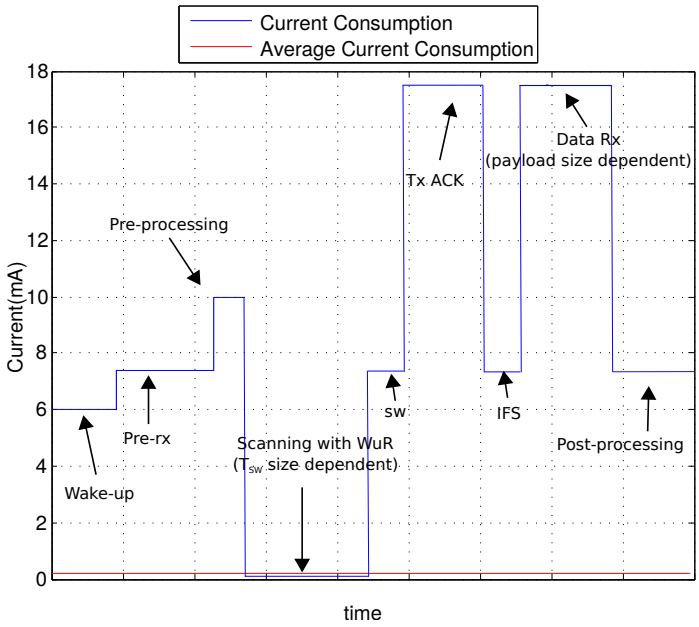


Figure 5.8: Scanner node states in fully asynchronous mode with Wake-Up radio for a TI device

In Figure 5.8 it is shown that, thanks to WuR implementation, average current consumption of the BLE node, is reduced considerably with respect to typical average BLE current consumption without WuR (for a TI device typically around $15mA$, although it varies depending on T_{SW} size). The average current remains in the order of μA, thus facilitating an extension of battery lifetime. The proposed mode would be similarly applied to any other BLE device manufacturer. In this section we provide results of lifetime estimation of a TI and a STMicroelectronics device when implementing the WuR.

We evaluated the implementation of BLE with WuR for fully asynchronous applications. WuR mentioned in Section 5.4.2 was implemented at scanner side during ND. The channel is monitored using the WuR during a time duration according to T_{SW} size. After successful reception of a wake-up signal, an acknowledge packet is sent to the advertiser in order to indicate it is ready for application data reception, for which we have considered an empty packet. Finally, in order to receive the application data, a reception state using the main radio is added after successful synchronization with the advertiser, as shown in Figure 5.8.

Scanner lifetime results for TI device have been estimated using $T_{SI} =$

$400ms$ with a duty cycle ranging from 5% to 100%, for which lifetime using the WuR is $\approx$ 20 and $\approx$ 238 times higher compared to fully asynchronous mode without WuR respectively, which is an improvement from 8.8 to 177.5 days of battery lifetime with a 5% duty cycle, and 0.3 to 81.7 days with a 100% duty cycle. Figure 5.9 shows the results for application data with a payload size of 2 Bytes since very low amount of data is considered for this kind of low duty cycle applications such as the temperature and humidity monitoring use case and the light switches use case considered in Chapter 3. However, we have evaluated the performance for different application data payload size (up to 37 Bytes). We have observed a lifetime reduction of $\approx$ 0.6% per extra byte sent for a 5% duty cycle, and a lifetime reduction of $\approx$ 0.3% per extra byte for 100% duty cycle when using the WuR. On the other hand, a lifetime reduction of $\approx$ 0.04% per byte sent is seen for a 5% duty cycle, whereas there is no lifetime variation for different data payload with a duty cycle of 100% when the WuR is not implemented. For both duty cycled and continuous scanning, BLE performance is better when implementing a WuR for channel monitoring.

in Figure 5.9.

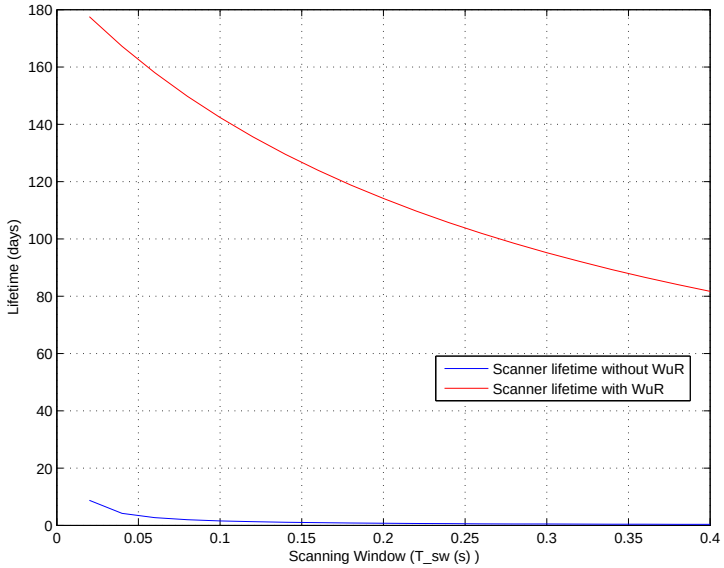


Figure 5.9: Scanner battery lifetime using fully asynchronous mode for TI device ($T_{SI} = 400ms$). Application data payload size of 2 Bytes

On the other hand, lifetime results for STMicroelectronics device are depicted in Figure 5.10 which is visible in the next page. Similarly as for TI device, scanner lifetime is estimated using $T_{SI} = 400ms$ with a duty cycle

ranging from 5% to 100% for a data payload of 2 Bytes. Battery lifetime using the WuR is ≈ 8 and ≈ 49 times higher compared to fully asynchronous mode without WuR, which is an improvement from 55.9 days to 472.2 days for a duty cycle of 5%, and 2.4 days to 114.5 days for a duty cycle of 100%. We have evaluated lifetime for different data payload size as well (up to 37 Bytes). Lifetime reduction with increase of 1 Byte is ≈ 0.5% for a duty cycle of 5% and 0.1% for a duty cycle of 100% when using the WuR. While a reduction per Byte increased is ≈ 0.05% for a duty cycle of 5% and no variation for a duty cycle of 100% when WuR is not implemented.

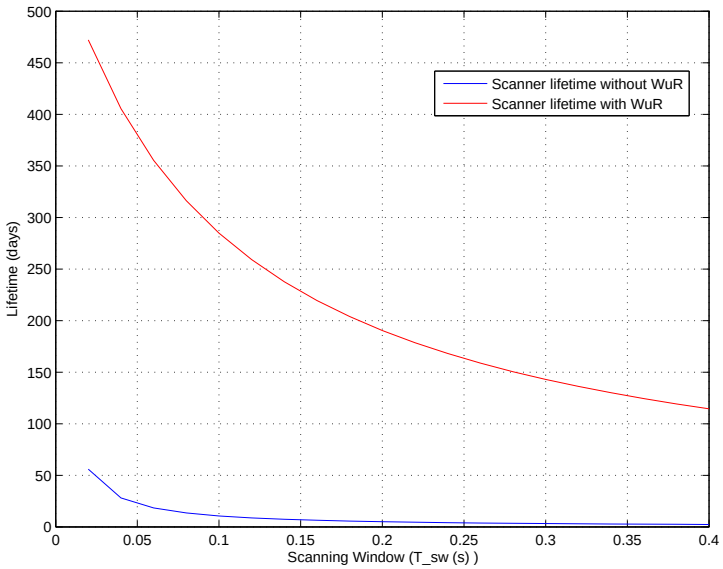


Figure 5.10: Scanner battery lifetime using fully asynchronous mode for ST device ($T_{SI} = 400ms$). Application data payload size of 2 Bytes

In conclusion, BLE scanner device performance in fully asynchronous mode is considerably much better when implementing a WuR, where the WuR is used for channel monitoring waiting for a wake-up signal from a BLE advertiser device. Although in proportion, TI device performance improvement when using the WuR is better compared to STMicroelectronics device; battery lifetime is much higher for STMicroelctronics device reaching up to 1.3 years of lifetime compared to 6 months for a TI scanner device. On the other hand, when WuR is not implemented, lifetime scarcely achieves 9 days and 1.9 months for TI and STMicroelectronics device respectively. Our results demonstrate the interest of using the WuR in order to extend battery lifetime for applications where location of the nodes hinders access to them, especially if implementing STMicroelectronics BLE device who achieves the

longest lifetime (although a battery with higher capacity is recommended in order to reduce maintenance costs).

5.5 Chapter Conclusion

In this chapter we introduced the concept of WuR to BLE. We have reviewed of WuR state of the art and have selected a WuR from [72] as it meets with power consumption requirements as presented in [59, 67, 68], that establishes that a $50\mu W$ WuR is required for integration within WSN, since it offers good compromise between low power consumption and high sensitivity. We have evaluated the impact of WuR implementation together with a BLE. Our results show the interest of using WuR for DC-BLE and fully asynchronous operating modes at scanner/master side during ND for periodic low frequency and continuous high frequency scenarios respectively. For fully asynchronous mode, we proposed a new operating mode when implementing WuR which would require a modification of the BLE specifications, whereas for DC-BLE with WuR a modification is not required.

With respect to periodic low frequency scenarios, the interest of implementing a WuR on the scanner/master side during ND is seen when the duty cycle is $\geq 0.36\%$ for a TI device (where ND represents 93.7% of the active part of the cycle), and when the duty cycle is $\geq 0.26\%$ for a STMicroelectronics device (where ND represents 97.08% of the active part of the cycle). DC-BLE with WuR performance (in terms of battery lifetime) for TI device, is better than Classic BLE, when the communication period is $\geq 7s$, which is ≈ 12 times better than DC-BLE without WuR, for which the communication period with better lifetime than Classic WuR is $\geq 86s$. For STMicroelectronics device, different from DC-BLE without WuR, for which there is no performance improvement compared to Classic BLE, DC-BLE with WuR reaches higher lifetime than Classic BLE when the communication period is $\geq 9s$.

Regarding continuous high frequency scenarios, we found that BLE scanner device performance in fully asynchronous mode is considerably much better when implementing the WuR, where the WuR is used for channel monitoring waiting for a wake-up signal from a BLE advertiser device. The best performance is obtained for the STMicroelectronics device who reaches up to 1.3 years of lifetime compared to 1.9 months using the same operating mode without WuR. Different from operating modes presented in Chapter 3, this operating mode would require modification of the BLE specifications since it establishes that, either application data is sent within advertising packets for non connected oriented communications, or either a pulling empty packet is

sent from the master after successful synchronization between scanner and advertiser and after a time which depends on the transmit window size, transmit window offset and devices clock accuracy. The later is done in order to allow battery recovery and thus optimizing its lifetime. We propose to avoid this waiting time as energy consumption is optimized with WuR implementation. This operating mode is proposed for continuous high frequency type of scenarios for which the CL is low and events can be either synchronous or asynchronous. Application data is not sent within advertising packets since they are received by the WuR. Instead, after successful reception of the wake-up signal (replacement for advertising packets), the main radio is immediately awakened in order to receive application data using BLE data channels. In addition, advertiser device must be able to send wake-up signals (in replacement for traditional advertising packets) using OOK modulation in concordance with the selected WuR, the later holds true for any type of scenario.

Conclusions and Perspective

The IoT allows to have more and more connected objects interacting with each other offering to improve daily life. Although, IoT has been around for many years now, it is still maturing. WSN are crucial for IoT and one of the key points that still need to be improved is the energy consumption of the nodes. A low energy consumption allows a lifetime extension for those IoT wireless devices running on batteries as wells as related costs reduction for those devices running on wired protocols.

In this work we focused on energy consumption reduction for wireless devices running on batteries. More precisely, we have concentrated on BLE as it has been proven to be more energy efficient compared to other technologies used for WSN. Our work proposed an energy reduction methodology for BLE. Our methodology includes a reduction of the energy consumption without modification of the BLE specification as presented in Chapter 3 as well as energy consumption reduction with modification of BLE PHY layer (by including a WuR) and the MAC layer since a mode of operation with a modification of the specification is required for our proposal as presented in Chapter 5. More precisely, our work contributed to the state of the art in 4 main aspects: energy model optimization, new battery lifetime estimation model, parameter optimization for ND phase and BLE enhancement with WuR implementation. Our conclusions in theses aspects are described as follows:

1. **Energy model optimization:** our energy optimization methodology builds upon an energy model from the state of the art [4]. We optimized the model in order to evaluate BLE performance at different levels. With our proposed model optimization, BLE performance is evaluated considering devices behavior at both communication and application level. At communication level, the scanner/master device plays the central role, whereas at application level it could play either the central or the peripheral role. Traditionally, constrained energy and DL has been considered at the advertiser/slave side since it usually plays the

peripheral role. Our methodology takes into account the case were the constrained energy and DL lies at the scanner side due to behavior at application level. Our model optimization was experimentally validated as shown in Chapter 4.

2. **Battery lifetime estimation model:** the impact of energy consumption over the battery lifetime was evaluated as nodes are typically battery operated. We have proposed a model to estimate coin cell batteries lifetime based on data provided by the manufacturer, Peukert's Law and discharge rate given by node behavior. Our model is intended to provide fast and realistic approximation of lifetime in order to give a more clearly and straightforward interpretation of the impact of our energy optimization methodology over BLE performance. Most of our results are given in terms of lifetime along Chapters 2,3 and 5. These results are based on the data sheet of a Panasonic CR2032 coin cell battery with a nominal capacity of 225 mAh.

3. **Parameter optimization of the ND phase:** the duration of this phase is non-deterministic but we can evaluate average and worst case values based on a suitable model. We proposed a parameter optimization based on our optimized energy model in order to provide with the set of $T_{AI} - T_{SI} - T_{SW}$ parameters that provides optimal DL and energy consumption during this phase. Using state of the art models to find a good trade-off between DL and energy consumption is not straightforward as results from this models are very general and need for large computation time. Our contribution for this aspect lies on the fact that a good trade-off between DL and energy consumption is obtained in less than $1s$ thanks to our Matlab based simulator, and it is obtained based on use case constraints such as maximum required latency, devices battery lifetime and devices behavior at both communication and application level. These kind of evaluations will help designing BLE networks with optimized energy consumption based on real use cases constraints. Our parameter optimization was experimentally validated as shown in Chapter 4. Additionally, thanks to our parameter optimization methodology, significant performance improvement is achieved compared to recommended Bluetooth SIG profiles.

4. **Overall energy consumption optimization:** it depends on parameters listed above when ND process represents a high percentage of the whole communication process, but it also depends on application requirements such as required latency, constrained energy availability and devices behavior at different levels. In order to optimize overall

energy consumption we proposed a scenario classification depending on application requirements such as maximum tolerable latency. Operating modes corresponding to each scenario type were proposed with no modification of the BLE specifications. Overall energy consumption optimization is based on the energy model and parameter optimization and results comparison with respect to Classic BLE are presented in terms of battery lifetime.

5. **BLE enhancement with WuR:** BLE performance was evaluated when implementing a WuR at scanner side. It was compared to BLE without WuR when using the different operating modes proposed in Chapter 3 whose choice depends on the scenario type (a scenario classification was also proposed in Chapter 3). According to our results, scanner lifetime can be significantly increased respect to all the operating modes previously evaluated without the use of the WuR and no modification of the specification as shown in Chapter 5. Evaluation of BLE performance using WuR is a new and still unexplored topic, very recently a research has been conducted in evaluating BLE with WuR at the advertiser side for a proximity application [71], while our work provides results for a wide variety of IoT applications going from periodic low frequency to continuous high frequency scenarios, in addition we considered duty cycled WuR. In this work, WuR was implemented at the scanner side as it typically has higher lifetime limitation. A new operating mode is proposed for implementing WuR for continuous high frequency scenarios that required a modification of the specifications. The advertiser device must be able to send wake-up signals (in replacement for traditional advertising packets) using OOK modulation in concordance with the selected WuR, the later holds true for any type of scenario. Our evaluation was based on a WuR from the state of the art [72] mainly based on its power consumption.

5.6 Future work

Our methodology can be applied in simulation during the design of a BLE network in order to ensure its optimal performance. However, some aspects still need to be taken into account such as single-master/several-slave start topology analysis, since our results are based on single-master/single-slave energy model. The impact of both packet collision and interference due to other technology using the same frequency band in the vicinity of BLE devices must be added to the evaluation, since our results are based on no-

collisions/no-interference environment. Additionally, experimental validation of our battery model is required in order to estimate the accuracy of the model.

On the other hand, a recent version on BLE core has been presented by the Bluetooth SIG which includes a mesh topology profile for BLE allowing many-to-many networking. An evaluation of the impact of using this new profile while using our proposed modes of operation with and without the WuR is a open challenge for new investigations. Additionally, for operating modes using the WuR a lifetime trade-off between scanner and advertiser can be evaluated such as in Section 3.4.3.3, in order to evaluate the impact of implementing the WuR only at master side or both master and slave.

Bibliography

[1] *Specification of the Bluetooth System Version 4.2*, Bluetooth SIG, 2014, www.bluetooth.org.

[2] S. Kamath and J. Lindh, "Measuring Bluetooth Low Energy Consumption," Texas Instruments, Tech. Rep. Application Note AN092, 2010.

[3] 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, 2012.

[4] 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

[5] Bluenrg current consumption estimation tool. STMicroelectronics. [Online]. Available: http://www.st.com/en/embedded-software/stsw-bnrg001.html

[6] J. Liu and C. Chen, "Energy analysis of neighbor discovery in bluetooth low energy networks," *Nokia.(nd)*, 2012.

[7] M. Ghamari, B. Janko, R. S. Sherratt, W. Harwin, R. Piechockic, and C. Soltanpur, "A survey on wireless body area networks for ehealthcare systems in residential environments," *Sensors*, vol. 16, no. 6, p. 831, 2016.

[8] Walt - wireless testbed. Université Grenoble Alpes, Grenoble INP / UJF. [Online]. Available: http://walt.forge.imag.fr/

[9] *Heart Rate Profile v1.0*, Bluetooth SIG, 2011, www.bluetooth.org.

[10] Ericsson, "Annual Report 2010," http://www.ericsson.com/res/investors/docs/2010/ericsson_ar_2010_en.pdf, 2010.

[11] F. J. Riggins and S. F. Wamba, "Research directions on the adoption, usage, and impact of the internet of things through the use of big data analytics," in *2015 48th Hawaii International Conference on System Sciences*, Jan 2015, pp. 1531–1540.

[12] Gartner says the internet of things installed base will grow to 26 billion units by 2020. Gartner. [Online]. Available: http://www.gartner.com/newsroom/id/2636073

[13] More than 30 billion devices will wirelessly connect to the internet of everything in 2020. ABI Research. [Online]. Available: https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/

[14] How many internet connections are in the world? right now. CISCO. [Online]. Available: https://blogs.cisco.com/news/cisco-connections-counter

[15] Morgan stanley: 75 billion devices will be connected to the internet of things by 2020. ABI Research. [Online]. Available: http://www.businessinsider.com/75-billion-devices-will-be-connected-to-the-internet-by-2020-2013-10?IR=T

[16] L. Tan and N. Wang, "Future internet: The internet of things," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 5. IEEE, 2010, pp. V5–376.

[17] S. Agrawal and M. L. Das, "Internet of Things: A Paradigm Shift of Future Internet Applications," in *Nirma University International Conference on Engineering (NUiCONE)*, December 2011, pp. 1–7.

[18] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?" in *International Workshop on the Security of the Internet of Things (SecIoT'10)*, 2010.

[19] Z. Ma, X. Shang, X. Fu, and F. Luo, "The Architecture and Key Technologies of Internet of Things in Logistics," in *International Conference on Cyberspace Technology (CCT)*, 2013, pp. 464–468.

[20] M. Lee, J. Hwang, and H. Yoe, "Agricultural Production System based on IoT," in *International Conference on Computational Science and Engineering*, 2013, pp. 833–837.

[21] I. Ganchev, Z. Ji, and M. O'Droma, "A Generic IoT Architecture for Smart Cities," in *Irish Signals and Systems Conference and China-Ireland International Conference on Information and Communications Technologies (ISSC/CIICT)*, June 2014, pp. 196–199.

[22] T. Kim, I. H. Kim, Y. Sun, and Z. Jin, "Physical layer and medium access control design in energy efficient sensor networks: An overview," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 2–15, 2015.

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

[24] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, "Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Node in a Cyclic Sleep Scenario," in *IEEE International Wireless Symposium (IWS)*, 2013, pp. 1–4.

[25] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4," in *Wireless Communications and Networking Conference Workshops (WCNCW)*, 2012, pp. 232–237.

[26] É. Morin, M. Maman, R. Guizzetti, and A. Duda, "Comparison of the device lifetime in wireless networks for the internet of things," *IEEE Access*, vol. 5, pp. 7097–7114, 2017.

[27] M. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of Things in the 5G Era: Enablers, Architecture and Business Models," *Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–17, 2016.

[28] K.-H. Chang, "Bluetooth: A Viable Solution for IoT?" *IEEE Wireless Communications*, vol. 21, pp. 6–7, 2014.

[29] B. SIG. Adopted bluetooth profiles, services, protocols and transports. [Online]. Available: https://www.bluetooth.com/specifications/adopted-specifications

[30] Z. He, B. Cui, W. Zhou, and S. Yokoi, "A proposal of interaction system between visitor and collection in museum hall by ibeacon," in *Computer*

Science & Education (ICCSE), 2015 10th International Conference on. IEEE, 2015, pp. 427–430.

[31] M. Koühne and J. Sieck, "Location-based services with ibeacon technology," in *Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on*. IEEE, 2014, pp. 315–321.

[32] T. Jorgensen and N. T. Johansen, "Z-wave as Home Control RF Platform," *Zensys A/S*, 2005.

[33] J. Decuir, "Introducing Bluetooth Smart: Part I: A Look at Both Classic and New Technologies," *IEEE Consumer Electronics Magazine*, vol. 3, no. 2, pp. 25–29, 2014.

[34] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: S*. John Wiley & Sons, 1998.

[35] K. Cho, W. Park, M. Hong, G. Park, W. Cho, J. Seo, and K. Han, "Analysis of latency performance of bluetooth low energy (ble) networks," *Sensors*, vol. 15, no. 1, pp. 59–78, 2014.

[36] 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, 2017.

[37] J. Rahme, N. Fourty, K. Al Agha, and A. Van den Bossche, "A recursive battery model for nodes lifetime estimation in wireless sensor networks," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[38] S. Park, A. Savvides, and M. Srivastava, "Battery capacity measurement and analysis using lithium coin cell battery," in *Proceedings of the 2001 international symposium on Low power electronics and design*. ACM, 2001, pp. 382–387.

[39] T. Simunic, L. Benini, and G. De Micheli, "Energy-efficient design of battery-powered embedded systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 1, pp. 15–28, 2001.

[40] L. M. Feeney, C. Rohner, P. Gunningberg, A. Lindgren, and L. Andersson, "How do the dynamics of battery discharge affect sensor lifetime?" in *Wireless On-demand Network Systems and Services (WONS), 2014 11th Annual Conference on*. IEEE, 2014, pp. 49–56.

[41] I. Baccouche, A. Mlayah, S. Jemmali, B. Manai, and N. E. B. Amara, "Implementation of a coulomb counting algorithm for soc estimation of li-ion battery for multimedia applications," in *Systems, Signals Devices (SSD), 2015 12th International Multi-Conference on*. IEEE, 2015, pp. 1–6.

[42] O. Mokrenko, M.-I. Vergara-Gallego, W. Lombardi, S. Lesecq, and C. Albea, "Wsn power management with battery capacity estimation," in *New Circuits and Systems Conference (NEWCAS), 2015 IEEE 13th International*. IEEE, 2015, pp. 1–4.

[43] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of the Electrochemical Society*, vol. 140, no. 6, pp. 1526–1533, 1993.

[44] R. Gu, P. Malysz, H. Yang, and A. Emadi, "On the suitability of electrochemical-based modeling for lithium-ion batteries," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 4, pp. 417–431, 2016.

[45] Y. Shi, K. Smith, R. Zane, and D. Anderson, "Life prediction of large lithium-ion battery packs with active and passive balancing," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 4704–4709.

[46] L. Zheng, J. Zhu, G. Wang, D. D.-C. Lu, and T. He, "Lithium-ion battery instantaneous available power prediction using surface lithium concentration of solid particles in a simplified electrochemical model," *IEEE Transactions on Power Electronics*, 2018.

[47] K. Furset and P. Hoffman, "High pulse drain impact on CR2032 coin cell battery capacity," Nordic Semiconductor and Energizer, Tech. Rep., 2011, technical Report. [Online]. Available: http://www.eetimes.com/ContentEETimes/Documents/Schweber\/C0924/C0924post.pdf

[48] W. Peukert, "About the dependence of the capacity of the discharge current magnitude and lead acid batteries," *Elektrotechnische Zeitschrift*, vol. 20, no. 1897, 1897.

[49] M. Tomasz. Long live coin cells: Getting years of battery life from small batteries in low power devices. [Online]. Available: http://www.edn.com/design/power-management/4421334/2/Long-Live-Coin-Cells--Getting-years-of-battery-life-from-small-batteries-in-\low-power-devices

[50] H. K. Fard, Y. Chen, and K. K. Son, "Indoor positioning of mobile devices with agile ibeacon deployment," in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*. IEEE, 2015, pp. 275–279.

[51] R. Knoblauch, M. Pietrucha, and M. Nitzburg, "Field studies of pedestrian walking speed and start-up time," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1538, pp. 27–38, 1996.

[52] R. W. Bohannon, "Comfortable and maximum walking speed of adults aged 20—79 years: reference values and determinants," *Age and ageing*, vol. 26, no. 1, pp. 15–19, 1997.

[53] D. Panescu, "Emerging technologies [wireless communication systems for implantable medical devices]," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 2, pp. 96–101, 2008.

[54] H. Burri and D. Senouf, "Remote monitoring and follow-up of pacemakers and implantable cardioverter defibrillators," *Europace*, vol. 11, no. 6, pp. 701–709, 2009.

[55] X. Zhang and B. Chang, "Research of temperature and humidity monitoring system based on WSN and fuzzy control," in *IEEE International Conference on Electronics and Optoelectronics (ICEOE)*, vol. 4, 2011.

[56] Fit iot-lab: Iot experimentation at a large scale. The FIT consortium. [Online]. Available: https://www.iot-lab.info/

[57] Wireless testbed based on raspberry pi. Grenoble INP Ensimag FabLab. [Online]. Available: http://fablab.ensimag.fr/index.php/Wireless_Tested_based_on_Raspberry_Pi

[58] C. Maity, A. Gupta, S. K. Panigrahi, and C. Garg, "Low power wake-up signalling in wireless network," in *Sensing Technology (ICST), 2012 Sixth International Conference on*. IEEE, 2012, pp. 439–443.

[59] N. M. Pletcher, *Ultra-low power wake-up receivers for wireless sensor networks*. ProQuest, 2008.

[60] C. Y. Yao and W. C. Hsia, "A x2013;21.2 -dbm dual-channel uhf passive cmos rfid tag design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1269–1279, April 2014.

[61] A. Sharma, A. T. Hoang, F. Nekoogar, F. U. Dowla, and M. S. Reynolds, "An electrically small, 16.7 m range, iso18000-6c uhf rfid tag for industrial radiation sources," *IEEE Journal of Radio Frequency Identification*, pp. 1–1, 2018.

[62] K. Kaushik, D. Mishra, S. De, K. R. Chowdhury, and W. Heinzelman, "Low-cost wake-up receiver for rf energy harvesting wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6270–6278, Aug 2016.

[63] C. Hambeck, S. Mahlknecht, and T. Herndl, "A 2.4 μw wake-up receiver for wireless sensor nodes with- 71dbm sensitivity," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on.* IEEE, 2011, pp. 534–537.

[64] D. C. Daly and A. P. Chandrakasan, "An energy-efficient ook transceiver for wireless sensor networks," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 5, pp. 1003–1011, 2007.

[65] N. M. Pletcher, S. Gambini, and J. M. Rabaey, "A 2GHz 52 μW Wake-Up Receiver with -72dBm Sensitivity Using Uncertain-IF Architecture," in *IEEE International Solid-State Circuits Conference, (ISSCC)*, 2008, pp. 524–633.

[66] B. Otis, Y. H. Chee, and J. Rabaey, "A 400 μw-rx, 1.6 mw-tx super-regenerative transceiver for wireless sensor networks," in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International.* IEEE, 2005, pp. 396–606.

[67] N. M. Pletcher, S. Gambini, and J. M. Rabaey, "A 2ghz 52 μw wake-up receiver with-72dbm sensitivity using uncertain-if architecture," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International.* IEEE, 2008, pp. 524–633.

[68] N. E. Roberts and D. D. Wentzloff, "A 98nW wake-up radio for wireless body area networks," in *IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, 2012, pp. 373–376.

[69] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled mac protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 674–687, 2016.

[70] Bluetooth SIG. [Online]. Available: www.bluetooth.com

[71] D. Giovanelli, B. Milosevic, D. Brunelli, and E. Farella, "Enhancing bluetooth low energy with wake-up radios for iot applications," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*. IEEE, 2017, pp. 1622–1627.

[72] C. Bryant and H. Sjöland, "A 2.45ghz, 50μw wake-up receiver front-end with 88dbm sensitivity and 250kbps data rate," in *Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014 - 40th European*. IEEE, 2014, pp. 235 – 238.

Publications

- A. Liendo, D. Morche, R. Guizzetti, and F. Rousseau, "Precise BLE Parameter Optimization for IoT Related Applications," in IEEE International Conference on Communications (ICC), 2018.
- A. Liendo, D. Morche, R. Guizzetti, and F. Rousseau, "Efficient Bluetooth Low Energy Operation for Low Duty Cycle Applications," in IEEE International Conference on Communications (ICC), 2018.

Acknowledgement

I want to thank all persons who helped me getting this far. I would like to start by conveying my special gratitude to my thesis advisors: Dominique Morche, Franck Rousseau, Roberto Guizzetti and also Didier Belot who was advisor during the first year of my research. For all of their contributions, guidance and patience, infinite thanks! I would also like to extend my gratitude to Andrzej Duda, head of the Drakkar group at LIG Laboratory for his receptivity and good advices.

The time spent on this project has been of immense professional and personal growth, so I would also like to thank STMicroelectronics, CEA-Leti and LIG laboratory for opening their doors and allowing me to be part of this wonderful universe of research.

I had the opportunity to share with several people in the different work environments, which had an influence on my work and/or with whom I spent countless good times which were also a source of motivation. Therefore, I would also like to thank: Matthieu Verdy, Alin Ratiu, Robert Polster, César Vásquez, Anthony Goavec, Gabriel Mugny, David Buffeteau, Elodie Morin, Henry-Joseph Audeoud, Timothy Claeys, Etienne Dublé, Pierre Brunisholz, just to name some of them. Thanks for all the science, the sports, the music and good conversations.

To the people who accompanied me during all this time, friends and family. A especial thank to my parents, my brothers and especially to my sister Adriana for always being close and being a source of inspiration and support for me.

And last but not least, Néstor Reyes who has been next to me during most of the time of the realization of this thesis, thanks for being there during the good and less good times and all the support.