POLITECNICO DI TORINO

DET Dipartimento di Elettronica e Telecomunicazioni

Master's Degree in Biomedical Engineering

Master's Degree Thesis

A comparative performance study of different low power wireless protocols for event-driven sEMG acquisition system

Supervisors

Prof. Danilo Demarchi

PhD Paolo Motto Ros

Candidate

Elia Pellegrino

March 2021

Abstract

Nowadays, wireless systems are more and more important in biomedical world, both to eliminate cables encumbrance as well as integration of embedded systems inside Internet of Things (IoT) world. In days where electronic health and fitness monitoring, as well as telemedicine are well-studied topics, the need for a fast, reliable and at low power consumption wireless protocol is of extreme importance.

This master thesis project aims to analyze and characterize performances of different wireless communication protocols, both already existing and not, to send sEMG (surface Electromyography) data acquired with custom PCB boards using classical front-end electronics together with *Average Threshold Crossing technique* (ATC). Every time muscular signal overcomes a fixed pre-defined threshold, a digital events is made available: in this way both power consumption, as well as circuitry complexity is reduced, due to lower data size acquired. Protocols studied, implemented, and tested are *Bluetooth Low Energy*, *IEEE 802.15.4*, and a custom protocol, tailor-made for this application.

The proposed system is composed of two *nRF-52840 Development Kits*, which simulate transmitter and receiver side: in particular, the former has the task to collect events inside a time window, assigning each of them with a relative timestamps, and at the end send the resulting packet to the receiver side, in which further processing (as gesture recognition) or ATC reconstruction could be performed.
After implementing them, protocols have been tested using lab instrumentation, to measure both power consumption, throughput and delay between transmission and reception: all protocols behave well in terms of transmission reliability using 1.5 m distance between transmitter and receiver. Features mentioned above, are used to compare protocols and choose the best one.
Considering throughput, all protocols show promising results, suitable for this application: *BLE* shows 215 kbit/s versus 116 kbit/s for *IEEE 802.15.4* and 322 kbit/s for custom protocol. The latency between transmission and reception is a key concept in order to allow for a real-time system: over all performed trials *BLE* shows a latency of 2.5 ms but with a high variability due to stack used, versus 4.5 ± 0.02 ms for *IEEE 802.15.4* and 800 ± 10 µs for custom protocol. Results from energy consumption, important for battery life, list here refer to one single transmission event: 38µC for *BLE* versus 125µC for *IEEE 802.15.4* and 32 µC for custom protocol. Considering the above results, if we exclude the need for standardization, in which case *BLE* shows better performance than *IEEE 802.15.4*, it is clear that the new proposed protocol is the best among all.

Summary

This thesis aims is to study the feasibility of wireless communication in a sEMG data acquisition and classification system, based on Average Threshold Crossing (ATC) technique. Different protocols have been reviewed, and the total performance is evaluated taking into account various parameters, such as transmission latency, packets lost rate and power consumption.

In Chapter 1, *Introduction*, the EMG signal and its acquisition system is introduced, as well as the different communication protocols implemented.

In Chapter 2, *State of art*, a brief overview of existing technology is shown.

In Chapter 3, *Materials and methods*, the hardware used as well as methods to collect and process data are listed.

In Chapter 4, *Firmware*, firmware written and implemented on boards is detailed.

In Chapter 5, *Results*, results for both power consumption and transmission characterization are listed.

In chapter 6, *Conclusion and future works*, a final overview with the most important results as well as possible better technology are presented.

,

Contents

List of Tables VI

List of Figures VII

1 Introduction 1

The Skeletal muscle . . . 1

ElectroMyoGraphy . . . 5

Wireless Protocol . . . 11

Bluetooth Low Energy . . . 12

IEEE 802.15.4 . . . 23

Nordic© Proprietary Protocol . . . 31

Ultra Wide Band Technology . . . 33

State of art 37

Matherials and Methods 41

Transmitting/Receiving board . . . 41

Supporting hardware . . . 42

Test methods . . . 43

Testing procedures . . . 44

Firmware and custom protocol development 49

Bluetooth© Low Energy . . . 49

Transmitter . . . 49

Receiver . . . 52

Communication workflow . . . 54

IEEE 802.15.4 . . . 58

Transmitter . . . 58

Receiver . . . 61

Communication Workflow . . . 63

Proprietary Protocol . . . 65

Transmitter . . . 66

Receiver . . . 69

Communication Workflow . . . 71

Results 73
Power Consumption . . . 73
Bluetooth$^{©}$ Low Energy . . . 73
IEEE 802.15.4 . . . 84
Proprietary Protocol . . . 89
Comparison . . . 94
Throughput, Latency, Jitter and Reliability . . . 98
IEEE 802.15.4 . . . 98
Proprietary . . . 108
Bluetooth$^{©}$ Low Energy . . . 122
Comparison . . . 133

Conclusion and future works 139

Bibliography 141

List of Tables

1.1 Different type of Physical layer 26
1.2 Radio electrical specification 33
1.3 Current consumption measured in different states of BLE device. [6, Table 4] 38
1.4 Comparison between different wireless protocol. 2 throughput are given: the physical one takes into account also the mandatory protocol's overhead. [7, Table 1] 38
1.5 GAP parameters 51
1.6 Scanning parameter 54
1.7 *Bluetooth*$^{©}$ *Low Energy* roles associated with the 2 boards. 55
1.8 Parameter used during communication. The TX Power can be set from -40 to 8 dBm: the tuning has been done in testing mode, to evaluate the influence on the overall power consumption. The frequency channel can be converted into physical frequency using formula inside Figure 1.23 60
1.9 Parameter used during communication. The TX Power can be set from -40 to 8 dBm: however, for the receiver board, the choice of this parameter is not so important, because it transmits only the ACK for the Notification ON/OFF packet 63
1.10 Parameters used in configuration.The TX Power can be set from -40 to 8 dBm. 1Mbit/s bit rate is also available 69
1.11 Results in real-scenario for transmitter side. Connection interval is set to 7ms. Idle mode for BLE is intended in a connection with master device. The maximum payload for *IEEE 802.15.4* is 100B to comply with the standard. 96
1.12 IEEE 802.15.4 with CCA results 103
1.13 As expected, 1 MBPS latency is twice than 2 MBPS 116
1.14 Mean value reflects the BLE behavior over entire transmission and power level. As one can see in charts above, when the packet is correctly sent at first shot, the latency is around 2ms for both modulation schemes 129
1.15 Total latency is calculated as $connection\_interval + latency$, referred to total latency of one event between transmitter and receiver. High value of jitter for both BLE implementations reflects the non-total control over the transmission of data: in real world the total latency for this technology is a multiple of connection interval 138

List of Figures

1.1 In *(a)*, the muscle structure is presented. In b, the muscle functional unity is shown 2
1.2 **a)** Neuromuscular junction and α-motor neuron, **b)** Sliding filament theory and myofibril structure. 3
1.3 a: Twitch, b: Summation, tetanus 4
1.4 Action potential mechanism [11]. In order to "fire", the voltage must pass the critical threshold around -55mV 5
1.5 a) Tripole approximation of Intracellualr Action potential along (Z-axis) muscle fiber. b)Tripole movement in 2D along muscle fiber; endplate represents the innervation zone, where axon reach muscle fiber.[19] 6
1.6 **a**) Electrical model of tissue and electrode; **b**) EMG signal acquired at different electrodes placement 7
1.7 Typical *sEMG* acquisition system 8
1.8 Example of *single* (1,2) and *double* (3) differential acquisition system [27] . 9
1.9 10
1.10 Typical Wireless Body Area Network. Figure taken from [20, Figure 2] . . 11
1.11 Fundamental blocks of *BLE* architecture. **Host** and **LE Controller** communicates using HCI Interface. 13
1.12 14
1.13 Physical channels. Colored channels are used only for advertising events. Total bandwidth occupied by BLE technology is 80 MHz 15
1.14 On-air packet format 15
1.15 Simplified version of [1, Page 203, Figure 2.1] 17
1.16 Some examples of possible network topology with BLE. The important thing to notice is that a device can be both peripheral and central at the same time (as K or O), but one peripheral can be connected only to one central at a time. More information here [1, p.257] 18
1.17 GATT Architecture. Figure taken from [42] 20
1.18 The advertising ahd scanning events are represented: they occurs only in Channel 37,38,39. The associated parameters are chosen considering both power consumption and latency between advertising and connection event. 21
1.19 Connection events. 22
1.20 Figure taken from [18, Figure 1] 24
1.21 IEEE 802.15.4 stack 25
1.22 Figure taken from [18, Figure 14] 26

1.23 IEEE Physical channels. The formula to determine center frequency is $F_c = 2405 + 5 * \mathrm{k}$ . . . 27
1.24 PHY layer's architecture . . . 28
1.25 MAC layer's architecture . . . 28
1.26 On-air packet format . . . 29
1.27 Beacon enabled mode. Black rectangle represents the start of each transmission . . . 29
1.28 Radio driver implemented by Nordic© . . . 31
1.29 Packet format . . . 32
1.30 State diagram of Radio peripheral. [46, Figure 110] . . . 32
1.31 Fractional Bandwidth definition . . . 34
1.32 IR-UWB On-Off Keying, In synchronized- OOK, a previously assigned timeslot is used to expect an information bit: then a sort of OOK modulation is assigned in each slot. . . . 34
1.33 UWB Technology compared to other common RF-system (as IEEE 802.11 which can operates also in 5GHz center frequency). The PSD limit imposed by FCC is in reality with a more complicated shape, but for sake of clarity has been shown in a simpler manner. [2, Fig.1] . . . 35
1.34 Figure taken from [18, Version 2011, Figure 85]. 1 symbol is capable of carrying 2 bits of information: changing value of above parameters, bit rate is tuned accordingly . . . 35
1.35 Center frequency and correspondent bandwidth for the different channels defined by *IEEE 802.15.4* standard. Figure taken from [18, Version 2011, Figure 94] . . . 36
1.36 Histogram of the measured latency of a train pulses at 800Hz, with 10000 occurences. The connection interval was set to 7.5ms. [6, Figure 11] . . . 37
1.37 [7, Figure 7] . . . 39
1.38 **a**) The developed electrical circuit by the authors. **b**) Current consumption from wake-up to transmitted data. . . . 40
1.39 nRF-52840-DK . . . 42
1.40 Additional hardware used . . . 43
1.41 General idea to transmit ATC signal. The reconstruction of the signal is made by the receiver board: if the trailing edge timestamp is not available, the signal is keep at digital 1 for 300µs. In this figure, the transmission latency is not shown (for simplicity reason) . . . 44
1.42 Current measurement scheme. The oscilloscope used is a *Rigol MSO5104* . 45
1.43 Experimental setup to estimate throughput and latency between packets. Black: ATC simulation, Red: transmitter, Yellow: collection board, Blue: receiver . . . 46
1.44 **a)** Similarity between number of events in correspondent time-windows using both cosine and ruzicka index. Threshold to calculate error is also shown. **b)** Similarity between inter-events distance of same events using both cosine and ruzicka index. Threshold to calculate error is also shown. . . . 47
1.45 Transmitter FSM.The WFE in CONN state, means that the board is "ready" to keep some events, previously assigned during configuration. . . . 50
1.46 Configuration process. At the end, the board passes in ADV state . . . 50

1.47 Real Advertising packet. It has been captured by *Wireshark*, the most used network analyzer, together with *nRF Sniffer for Bluetooth LE*, a pre-built nordic application used as a plug-in in Wireshark to be able to sniff (using a Nordic board) *BLE* packets over the air 52

1.48 Receiver FSM. The WFE in CONN state, means that the board is "ready" to keep some events, previously assigned during configuration 53

1.49 Configuration process in ON state 53

1.50 In *(a)*, advertising and connection is explained; in *(b)* and *(c)*, start and stop of ATC data is presented; in *(d)*, the connection is lost. These figures have been taken from *WireShark* software 56

1.51 Communication workflow. Each arrow represents a packet exchanged between transmitter and receiver: after *Notification ON* packet is sent and ACKed by transmitter, it will start to send *ATC packets*, until *Notification OFF* packet is sent. 57

1.52 Real on air packet. The dashed arrows represent the logical connection between different layers. In transmission mode, the path is from higher to lower layers while in reception mode is the opposite. The figure refers to a 1-channel acquisition system, although can be scalable to multi-channels ones. 58

1.53 Transmitter FSM.The WFE in WORK state, means that the board is "ready" to keep some events, previously assigned during configuration. 59

1.54 Configuration process. At the end, the board is ready to transmit data . . 59

1.55 On-air packet format 60

1.56 Transmitter FSM.The WFE in WORK state, means that the board is "ready" to keep some events, previously assigned during configuration. 62

1.57 Configuration process. At the end, the board is ready to transmit data . . 62

1.58 Network created using *IEEE 802.15.4* standard 64

1.59 Communication workflow. Each arrow represents a packet exchanged between transmitter and receiver: after *Notification ON* packet is sent and ACKed by receiver, transmitter will start to send *ATC packets*, until *Notification OFF* packet is sent and ACKed. 64

1.60 In *(a)* and *(b)* start and stop of communication are shown; in *(c)*, the Notification ON and its ACK are shown. These figures have been taken from *WireShark* software 65

1.61 State diagram of transmitter board. Dashed rectangle represents the functioning of antenna every time a transmission is issued (every connection interval): the text in ***bold and italic*** indicates the events generated by hardware, while the text above the arrows represents tasks triggered by the user. 67

1.62 Configuration stage 68

1.63 Real on-air packet in custom protocol. *Preamble* contains information about modulation used, while *Length* indicates the length of the *Payload*. The overall overhead (also *CRC*) is 9B. *Access-Address* is set to 0x89ABCDEFA4 69

1.64 State diagram of receiver board. Inside dashed rectangle antenna states are shown: with respect to transmission board, in which the antenna always return in *DISABLED* state, here it remains always in *RX* state, to listen for **Notification ON** packet, keeping power consumption at high level. The protocol could be optimized in order to synchronize transmitter and receiver. 70
1.65 Configuration stage. It is similar to the one of transmitter board (Table 1.10): it's really important that both *frequency channel* and *Access-Address* are equal to the one configured in transmitter side 70
1.66 Communication workflow. Each arrow represents a packet exchanged between transmitter and receiver: after *Notification ON* packet is sent, transmitter will start to send *ATC packets*, until *Notification OFF* packet is exchanged. For the first implementation, no ACK packets have been implemented. 71
1.67 **a**): Notification ON, **b**): Notification OFF 71
1.68 Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 22ms, where advertising starts (**c**). In idle mode (**d**), the current consumption is around 0.2mA 74
1.69 Advertising events 75
1.70 Antenna power level impact on overall power consumption 76
1.71 Antenna power level impact on overall power consumption during 1 connection events, for 1Mbit/s (**b**) and 2Mbit/s (**c**) physical layer. 77
1.72 Different connection events and antenna power level impact on overall power consumption during 1 connection events, for 1Mbit/s (**a**) and 2Mbit/s (**b**) physical layer. 78
1.73 Time needed to send packet is roughly the double for 1Mbit/s phy. 79
1.74 2Mbit/s offers less power consumption 80
1.75 Impact of payload with costant antenna power 81
1.76 Current ((b) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 8.5ms, where scanning starts (**c**). 82
1.77 Scanning event 83
1.78 Connection event 84
1.79 Current (**a**) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 400ms, where transmitter is ready to send ATC data. . . 85
1.80 Current consumption in idle state. A mean operation has been performed on current signal to clean raw data. 85
1.81 Current absorbed during transmission at different antenna power level, with (**b**) and without (**a**) cca; as expected, current increase with antenna power level 86
1.82 Current absorbed during transmission at different antenna power level, with and without cca(**a**); in **b** and **c** some energy considerations are made . . . 87
1.83 Impact of payload size. More time needed for transmit data bring higher consumption 88

1.84 Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 400ms, where antenna stays on in order to receive available data 88
1.85 Receiving state 89
1.86 Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 8.5ms, where antenna stays on in order to receive available data 90
1.87 Current and power consumption in idle state 90
1.88 Current profiles of different modulation scheme (**b**,**c**) and comparison between them (**a**) 91
1.89 Total charge and maximum current in one transmission event. Higher bit rate of 2Mbit/s modulation scheme results in low transmission time which brings benefit in energy consumption 92
1.90 Impact of payload size 92
1.91 Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 8.5ms, where antenna stays on in order to receive available data 93
1.92 93
1.93 Current (*a*) and power consumption (*b*) results for receiving side. 1Mbit/s modulation scheme shows lower consumption 93
1.94 Current profile for different protocols during one transmission event. Maximum payload size used: 100B for *IEEE 802.15.4* and 200B for *BLE and Proprietary*. Lower bit-rate reflects in higher time needed to encode and send the entire packet. 94
1.95 Results of different protocols in worst energy consumption scenario. In **a)** maximum current value reaches during one transmission is shown, while in **b** and **c** total charge (for one transmission) and the resulting estimate of lifetime are given. 95
1.96 For BLE, connection interval is set to 7.5 ms 97
1.97 Latency and Jitter results for −20dBm and −12dBm 99
1.98 Latency and jitter results for −8dBm and −4dBm 100
1.99 Latency and jitter results for 0dBm and 2dBm 101
1.100Latency and jitter results for 4dBm and 8dBm 102
1.101**a)**Latency and **b)** Jitter over different trials at different antenna power level 103
1.102Throughput at different antenna power level 104
1.103Distribution of events for different antenna power levels 105
1.104Frequency deviation between transmitter and receiver 106
1.105Similarity between distance of consequent events 106
1.106Similarity between number of events in corresponding time windows 107
1.107Packets lost and events lost ratio 107
1.108Results for −20dBm and −12dBm using 1 Mbit/s physical layer 108
1.109Results for −8dBm and −4dBm using 1 Mbit/s physical layer 109
1.110Results for 0dBm and 2dBm using 1 Mbit/s physical layer 110

1.111Results for 4dBm and 8dBm using 1 Mbit/s physical layer 111
1.112Results for −20dBm and −12dBm using 2 Mbit/s physical layer 112
1.113Results for −8dBm and −4dBm using 2 Mbit/s physical layer 113
1.114Results for 0dBm and 2dBm using 2 Mbit/s physical layer 114
1.115Results for 4dBm and 8dBm using 2 Mbit/s physical layer 115
1.116Latency and jitter results for 1 and 2 Mbit/s physical layer 116
1.117 117
1.118Events distribution for 1Mbit/s 118
1.119Events distribution for 2Mbit/s 119
1.120Results are closed to zero except for -20 dBm, in which the high volume of events lost reflects in higher deviation 120
1.121Similarity between distance of consequent events 121
1.122Similarity between number of events inside corresponding time windows . . 121
1.123Reliability for Proprietary protocol 122
1.124Latency and jitter for 2Mbit/s 123
1.125Latency and jitter for 2Mbit/s 124
1.126Latency and jitter for 2Mbit/s 125
1.127Latency and jitter for 2Mbit/s 126
1.128Latency and jitter for 2Mbit/s 127
1.129Latency and jitter for 1Mbit/s 128
1.130Bluetooth© Low Energy results 129
1.131Events distribution for 2Mbit/s. **f)** chart refers to 1Mbit/s modulation . . 130
1.132Frequency deviation for both modulation schemes. The fact that some packets are sent some connection intervals later, slightly distorts frequency content of ATC signal 131
1.133Similarity between distance of consequent events over all trials. For both indexes, high values of error are reached 131
1.134Similarity between number of events in corresponding time windows over all trials 132
1.135Latency and jitter comparison between protocols 134
1.136Throughput and transmission error comparison between protocols 135
1.137Frequency Deviation between transmitted and received ATC signal. All protocols behave well in terms of median frequency, except for *Proprietary-2MBps*, where the high number of packets lost reflects in higher deviation. For mean frequency, *BLE* presents the worst performance for the non-total control over transmission, as explained in Section 1. For all protocols, however, the frequency content remains more or less stable, except in that case where lot of packets are lost. 136
1.138ATC similarity comparison between protocols 137

Chapter 1

Introduction

The Skeletal muscle

Skeletal muscles, which provide the final output from the *motor system*, are not only the main actors involved in human movements but also play a key role in body health and energy homeostasis [8].
As one can see in Figure 1.1, they are attached to the bones by means of a tough band of connective tissue, *tendons*, fundamental viscoelastic structures, which allow muscles to transmit power towards the bones, producing human movements. The structure of a muscle is hierarchical: the outer layer, protected by a connective tissue called *epimysium*, is composed of bundles of muscle fibers, *fascicles*, surrounded and protected by a sheath of connective tissue, *perimysium*. Each muscular fiber, in turn, consists of many filaments, *myofibrils*, supported by *sarcoplasmic reticulum*: this membrane has not only a supportive role, but it also stores calcium ions (Ca^{++}), fundamental in muscle contraction as explained later. Inside each myofibril, little contractile units called *sarcomeres* are located: they are responsible for force generation due to two fundamentals filamentous proteins, *actin* and *myosin*, the latter thicker than the former. The spatial position of the two proteins gives the skeletal muscle its typical striated look. The physiology of the sarcomere is studied under the electron microscope, where different areas can be seen:

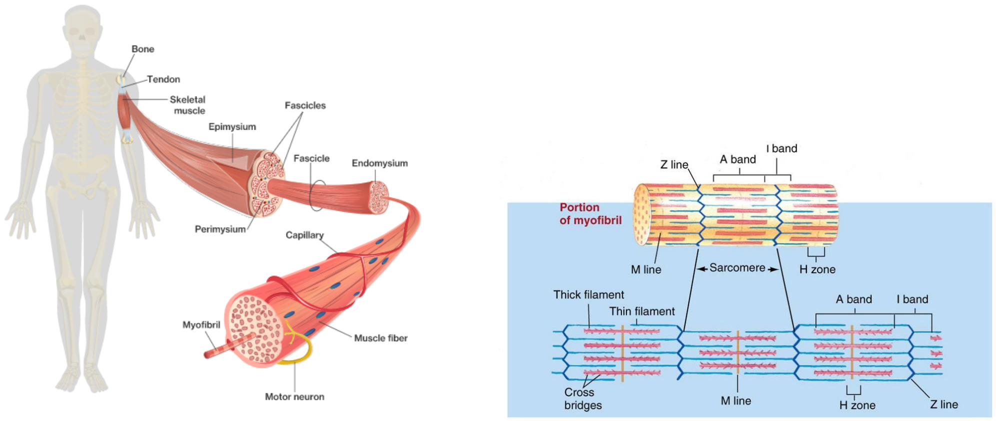
- **Z-line**: it is the anchor point between adjacent actin filaments. The force is generated between different Z-lines, which move one towards the other during muscle contraction.

- **M-line**: it is the anchor point between adjacent myosin filaments. During contraction, the M-lines stay in the same position.

- **I-band**: it is the zone surrounding the Z-line. In this band, only actin filaments are found, hence the name I, from *isotropic*.

- **H-zone**: it is the zone where only myosin filaments are found.

- **A-band**: it is the band in which the entire myosin filament is placed, even if some actin filaments are found, hence the name A, from *anisotropic*



(a) Muscle organization [29]

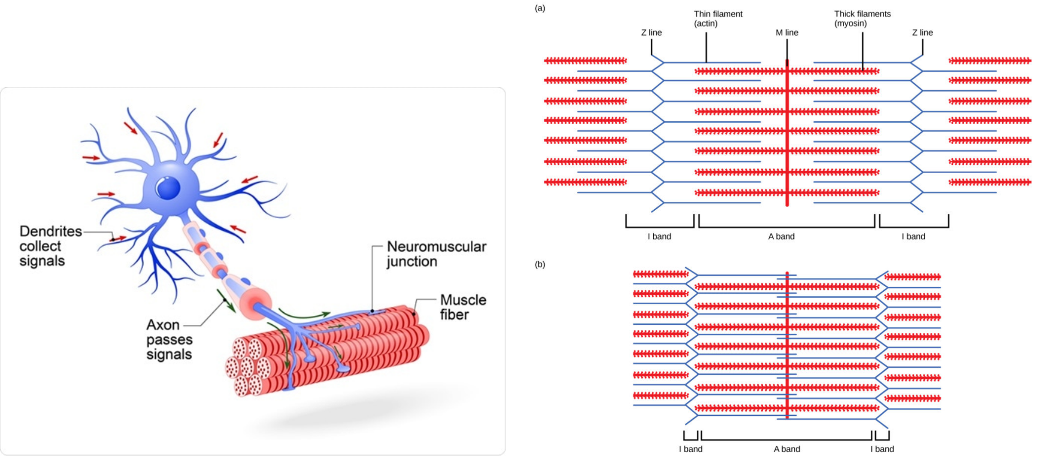
(b) Myofibril structure [12]

Figure 1.1: In *(a)*, the muscle structure is presented. In *b*, the muscle functional unity is shown

Muscle contraction

In order to understand sEMG signal, it is very important to know how muscle contraction works.

Muscles, like every other organ in the human body, are innervated by neural cells, in this case called *motor neurons*: what is important to notice is that one neuron innervate **more than one** muscle fiber, creating the smallest functional unit called the *motor unit*. Different type of motor neurons exist, but the most important are α ones, responsible for the generation and propagation of electrical stimulus from CNS towards skeletal muscles.



(a) Motor neuron and neuromuscular junction [24]

(b) Muscle contraction. Figure taken from [26]

Figure 1.2: **a)** Neuromuscular junction and α-motor neuron, **b)** Sliding filament theory and myofibril structure.

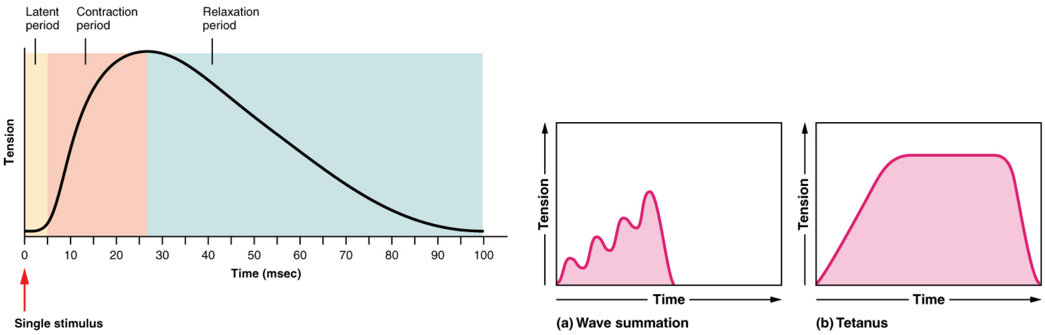
When an action potential is propagated along its axon and reach the *neuromuscular junction*, the chemical synapse between motor neurons and muscle fiber, it induces the opening of membrane channels, selective to calcium ions. The increase in Ca^{++}, induced the release of *acetylcholine* in the pre-synaptic space: the neurotransmitter will then bind to nicotinic receptor on post-synaptic membrane, causing action potential to propagate in muscle fiber along all directions.

When the action potential reaches myofibrils, the sarcoplasmic reticulum is excited to release its inner calcium: this will bring to a cascade of biochemical reaction (that goes beyond the scope of the thesis), also involving the *ATP*, the "energy molecule," that will bring the actin to slide on the myosin (*Sliding filament theory*), shortening the Z-line and producing muscle contraction and force(Figure 1.2).

The force produced by muscle depends on two main factors:

- **Type of MU recruited**: *Motor units* are divided into 3 big classes [3]:
  - Slow (S): They are small and can generate small force, but they are rich of mitochondria (and so can "burn" a lot of oxygen), thus resistant to fatigue, and the first to be recruited in human movement.
  - Fast Fatigable (FF): In contrast with S-type, these motor units are bigger and faster to twitch, thus generating a higher force. They have fewer mitochondria, thus consuming potentially less oxygen, and are therefore easily fatigued. They are recruited when more tension must be produced.
  - Fast Fatigue-resistant (FR): They are a mix of previous motor units
- **Firing rate**: this is the AP's motor-neuron frequency reaching neuromuscular junction. When an AP reaches a muscle fiber, it generates a *twitch* (Figure 1.3), a single

contraction-relaxation cycle: in any case, it is not sufficient to produce enough force. If another AP reach the same muscle fiber, it will produce another twitch, producing a *summation* phenomenon: the muscle will increase its output force, until a plateau (*tetanus*) is reached, around $35 - -40$Hz



(a) Tension generated in a single stimulus. [48]

(b) Summation and tetanus phenomenons [49]

Figure 1.3: a: Twitch, b: Summation, tetanus

Action Potential

Action potential (AP) plays a key role in cell to cell communication: it is a temporary depolarization of membrane potential, occurring in a certain type of cells, like muscle ones and neurons. In resting condition, the cells' membrane shows an electric unbalance, due to differences in sodium and potassium ions concentration, with cytoplasm more negative than extracellular space; this potential difference is maintained thanks to the physiological mechanism. In some condition (as neurotransmitter presence or electric stimulus), channel selective to sodium ions (Na^+) opens: in order to balance both Na^+ concentration and potential difference, positive ions start flowing inside the cells, until depolarization occurs (if a certain threshold is passed in the initial stage). Around $+30 - -40$ mV, voltage-gated(sensitive to the change of potential) ion channels, selective to potassium ion (K^+), opens: following concentration and electrical gradient, positive ions flows outside the cell, repolarizing the potential membrane. At this moment, the cell goes in a refractory state, where it can not *fire* another AP. This period is of crucial importance in the propagation of action potential towards muscle: the *electric wave* can move only in one direction. The excited cell will return in resting condition, ready to be depolarized again, after the activation of Na^+/K^+*-ATPase*, a special **energy consumer** molecule that actively pumps 3 Na^+ outside and 2 K^+ inside the cell, against gradient. This mechanism is well explained looking at Figure 1.4

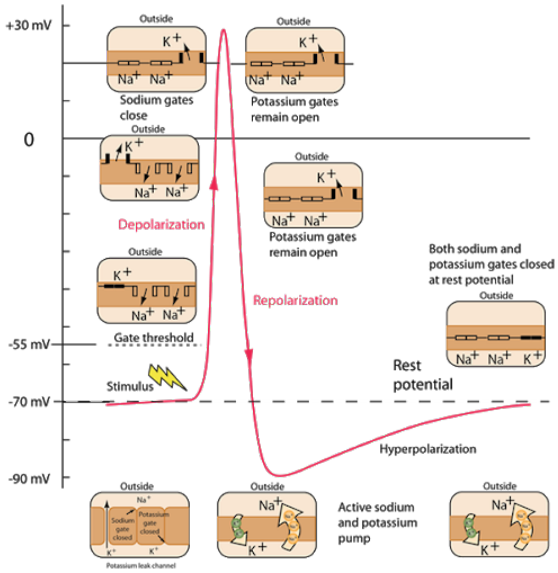


Figure 1.4: Action potential mechanism [11]. In order to "fire", the voltage must pass the critical threshold around −55mV

ElectroMyoGraphy

Electromyography (**EMG**) is an electrodiagnostic technique used for evaluating and recording muscle activity. Two different types of recording are available, depending both on the final scope and type of electrodes:

- **DEEP EMG**
- **SURFACE EMG**

In both condition, a small electrical current, generated by muscle fibers due to the exchange of ions across muscle fiber membranes, is recorded.

DEEP EMG

Deep or Intramuscular EMG is the most invasive technique because some needles must be inserted inside muscle for recording. Usage of tiny needle electrode placed very close to muscle, decrease detection volume and noise sources, thus permitting the recording of few *motor units*; with decomposition technique, single MUAP (*motor unit action potential*) can be extrapolated. Even if intramuscular EMG is the gold standard for the physiological study of motor units, it is rarely applied in clinical routine, both for invasiveness of needles and difficulty in interpreting results from decomposition technique. [30]

SURFACE EMG

Surface EMG, here *sEMG*, is the most used technique in different fields:

- Biomechanics and movement analysis: used in the identification of muscle activation intervals and levels, studying muscle coordination
- Muscle fatigue: used in monitoring myoelectric manifestations of fatigue
- Rehabilitation and sports medicine: assessment of effectiveness rehabilitation treatments
- Prosthetic control: gesture recognition on *sEMG* for driving prosthetic parts.

It is recorded placing *dry/wet* electrodes on human skin, above the muscle under control: the placement of the electrodes plays a key role in the acquisition of high-level quality *sEMG* signal. [31] When an AP reach neuromuscular junction, it propagates along muscle fibers, towards tendons, in all directions, with a velocity called (*conduction velocity (CV)*, Figure 1.5): the sEMG is the sum of all APs of all the MUs recruited (N) during movement under analysis, depending both on time (t) and force level required (F). (1.1) The AP shape, referred in (1.1) as u_i, can be approximated by simple functions, such as the analytical expressions provided by Rosenfalck or triangular approximation [19]. It derives from an electric field generated by a moving tripole (Figure 1.5).

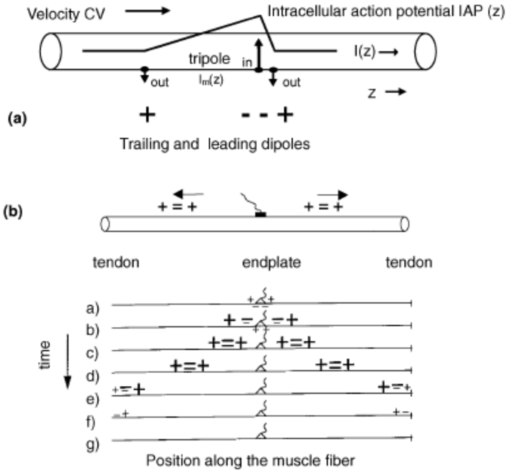
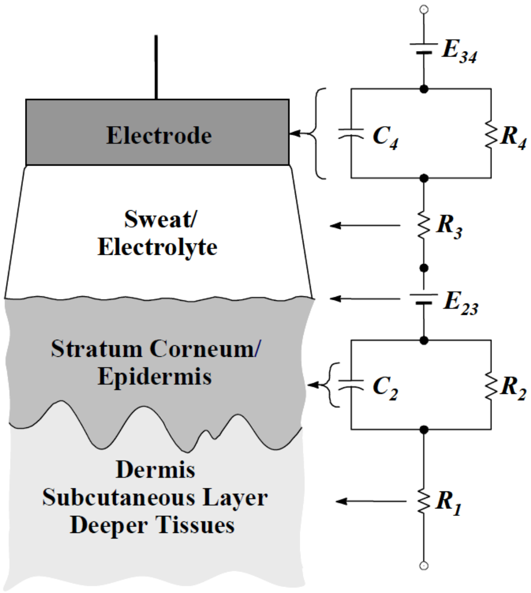


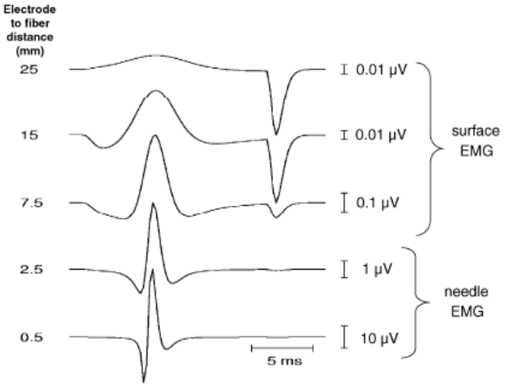
Figure 1.5: a) Tripole approximation of Intracellualr Action potential along (Z-axis) muscle fiber. b)Tripole movement in 2D along muscle fiber; endplate represents the innervation zone, where axon reach muscle fiber.[19]

$$sEMG(t, F) = \sum_{i=1}^{N} u_i(t, F) \tag{1.1}$$

The signal passes through different layers of human tissues moving towards the surface, which acts as an intrinsic low-pass filter: the cut-off frequency depends on the thickness of the tissue between electrodes and innervation zone. So, the deeper the muscle, the less the frequency content of the EMG signal. [23] This effect is well explained in Figure 1.6



(a) Skin-electrode interface. The electrical model of the electrode is also shown: *non-polarizable* electrodes, ideally, are resistance (R_4), while *polarizable* ones are capacitor (C_4). Figure taken from [4]



(b) Effects of tissue between muscle and electrodes [19]. The low-pass filter effect can be appreciated.

Figure 1.6: **a**) Electrical model of tissue and electrode; **b**) EMG signal acquired at different electrodes placement

Typical characteristic of sEMG signal are [32]:

- Amplitude: 50-1000 µV
- Bandwidth: 0.1-400 Hz

sEMG ACQUISITION SYSTEM

Typical sEMG acquisition systems include electrodes placed on the skin to sense electric potential, an analog front-end, comprising at least a high-pass filter, to remove motion noise artifact (caused by cable and electrodes), and a low pass filter, to remove high-frequency noise, and an instrumentation amplifier, used to amplify the bio-electric signal (Figure 1.7). An optional notch filter to remove 50/60Hz power line noise can be inserted. The amplified signal can then be digitalized with ADC or feed into a voltage comparator, to obtain a quasi-digital signal (*ATC technique*, explained in Section 1)

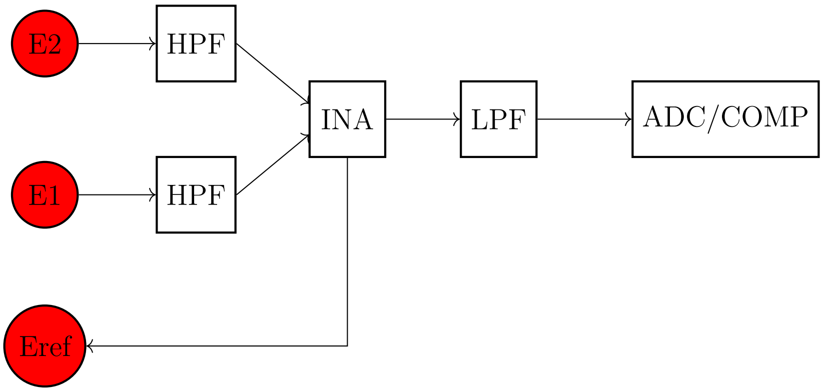


Figure 1.7: Typical *sEMG* acquisition system

Below, a brief explanation of each block in Figure 1.7 is presented:

- E1/E2/Eref: these are electrodes. They can be divided in 2 large groups, *polarizable* and *non-polarizable*: the former block the charge, while the latter allows charges to move freely across the interface. So *non-polarizable* electrodes are preferred: real electrodes, however, have a behavior in between the two extreme. As already mentioned, the position of electrodes is crucial for having a high-quality signal. Usually, the reference electrode is placed in a non-conductive area, for example, on bone's prominence. The two working electrodes must assure good contact with the skin to reduce motion artifact. Conductive gel, if wet electrodes are used, is often used to better interface ionic current (from the tissue) and electrical current (from electrodes). [32]
- HPF: High Pass Filter. The cutoff frequency of this HPF can vary among different scopes. Usually, cable motion artifact has a frequency range of $1 - 50$Hz, while electrode motion artifact is up to 20Hz. For example, in myoelectric prostheses, it is usually preferred filter above $50 - 60$Hz, for improving the control stabilty.
- INA: Instrumentational Amplifier. This is a differential amplifier used to record the difference between potential at E1 and E2. The key concept of this electronic block is input impedance: it must be very high in order to get a good electrical measurement, minimizing the effect of electrode-skin interface impedance. The CMRR (*Common Mode Rejection Ratio*), the ratio between differential and common gain, is usually used to evaluate the performance of INAs. More INAs and electrodes can be combined together to obtain *spatial filters*(Figure 1.8). Moreover, high gain and specific bandwidth must be taken into account when choosing amplifier.

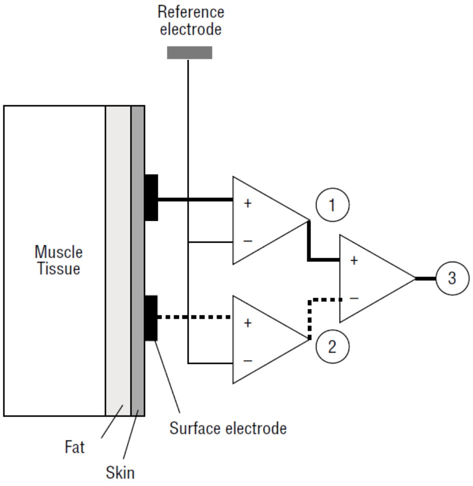


Figure 1.8: Example of *single* (1,2) and *double* (3) differential acquisition system [27]

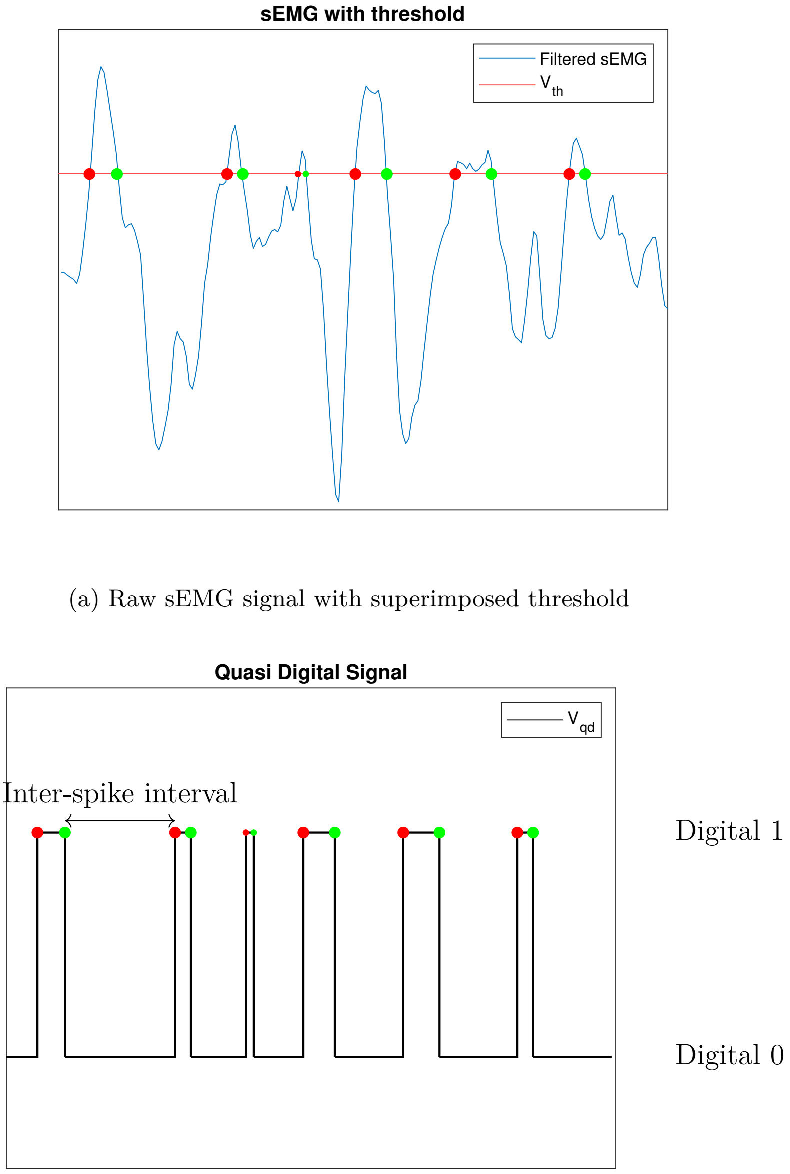
- LPF: Low Pass Filter used to reject high-frequency noise. Usually, the cutoff frequency is set to $400 - 500$Hz, cutting outside EMG bandwidth
- ADC: Analog to Digital Converter. Usually, EMG signal is sampled, with sampling frequency at least equal to 1kHz to avoid aliasing and sent to some processor unit for classification/evaluation purpose.
- COMP: Voltage Comparator. Explained in Section 1

ATC Technique

In a classical approach, *sEMG* is acquired, filtered, and then sampled and transmitted (wired/wireless) to a processing unit to further manipulate data: however, this technique leads to bigger PCB boards with higher power consumption and a lot of data acquired (sometimes not really useful), not suitable for wearable and low-power devices. A novel technique, based on voltage comparator, has been proposed in [21], [44]: the basic idea is to compare the input signal to a pre-chosen threshold and creating a quasi digital signal, following (1.2). The mechanism is well explained in Figure 1.9.

$$V_{qd}{}^{1} = \begin{cases} 1 & \text{if } sEMG \geq V_{Th} \\ 0 & \text{if } sEMG < V_{Th} \end{cases} \tag{1.2}$$

[1] In this case V_{qd} refers to a digital signal: 1 means 3.3V, while 0 means 0V.



(a) Raw sEMG signal with superimposed threshold

(b) Quasi digital signal

Figure 1.9

In this way, the main source of power consumption (ADC) is not used, improving battery life, PCB difficulty, and data dimensionality. The quasi-digital signal is highly correlated with *sEMG*, [41], but with a lower data dimension: in this way, the wireless transmission can be further improved.
The information is here conveyed as the time between different spikes: one can expect an inter-spike interval correlated with sEMG bandwidth. From a biomedical point of view,

as explained in Section 1, more spikes represent higher activation of motor neurons, thus higher force produced. In contrast, each spike cannot be seen as an activation of a single motor unit, but a bunch of them, due to higher conductor volume in the *sEMG* acquisition system.

Wireless Protocol

Advancement in wireless communication systems had a huge impact on everyday human life. Among the most promising technologies, WBAN (*Wireless Body Area Network*) is the first candidate (Figure 1.10). It consists of sensors placed on the body, which communicate (*intra-communication*) to a central node (smartphone, tablet, custom embedded system): from this, all the data collected could be sent over the internet to the different stakeholders.

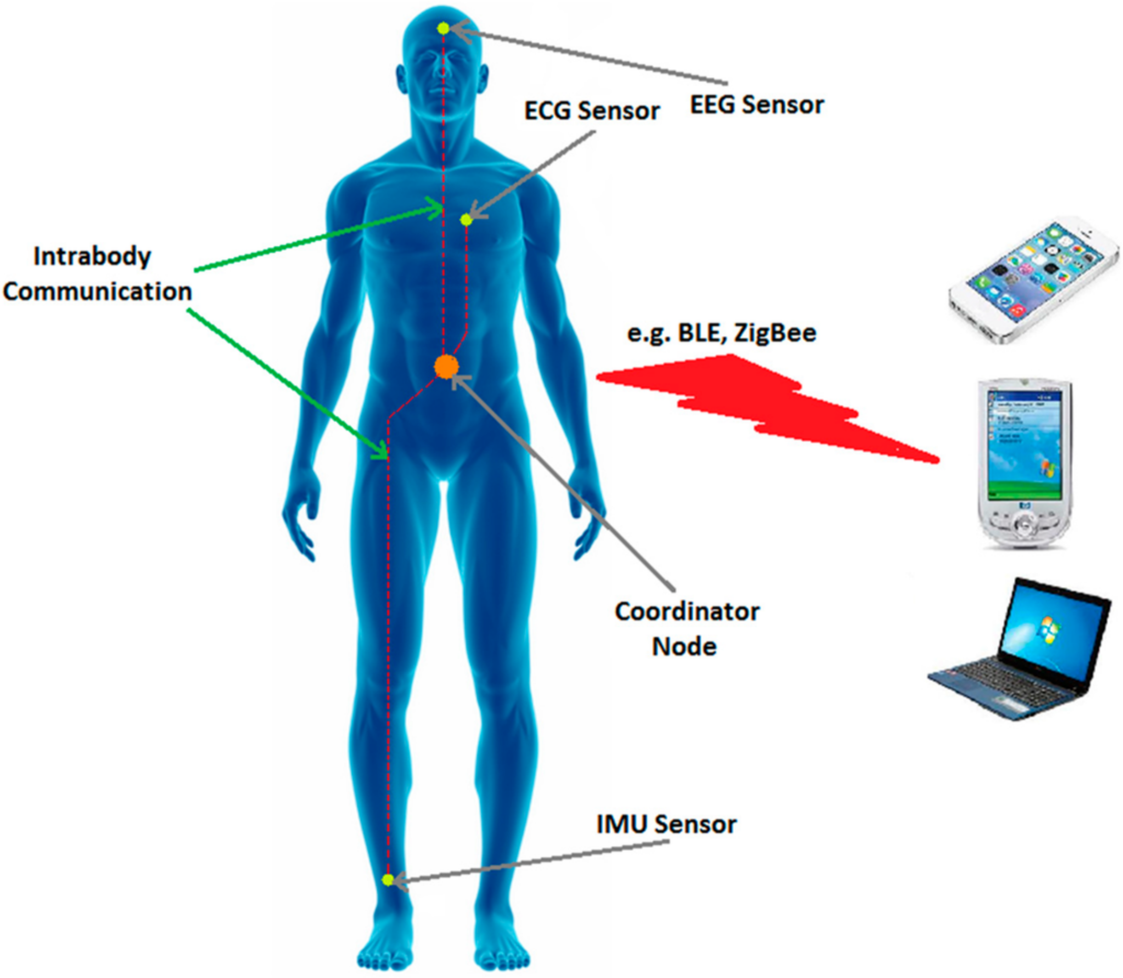


Figure 1.10: Typical Wireless Body Area Network. Figure taken from [20, Figure 2]

The focus of this thesis is the wireless intra-node communication: different protocols have been tested, taking into account:

- Power consumption/Battery life
- Transmission latency: defined as the time between transmission and reception
- Jitter: latency variability over entire communication
- Throughput: it must not be confused with bit rate. The throughput represents the *effective* capacity of a communication channel. In few words, this number *doesn't* consider overhead or footer bits (only user-configurable payload) and other practical problems related to implementation of wireless protocol.

In this section, the investigated protocols will be explained.

Bluetooth Low Energy

Bluetooth© is one of the main short-range communication protocol used in WBAN technology, for its cost, robustness, and power-consumption; a lot of features can be used, allowing for product differentiation. The entire material used to write this subsection is taken from [1]. Bluetooth comes in three different forms:

- Low Energy (LE): up to 2Mbit/s, used in application where power consumption is crucial.
- Basic Rate (BR): 721kbit/s
- Enhanced Data Rate (EDR): 2.1Mbit/s

The BR/EDR modes describe different protocols not used in this work.
Bluetooth© Low Energy was created by Bluetooth Special Interest Group (SIG) as a part of 4.0 Core Specification: the focus for *BLE*, as the name suggests, was to create a wireless protocol optimize for low power, low cost, low complexity, and low bandwidth ([52]). It operates in the unlicensed 2.4 GHz Industrial Scientific and Medic (ISM) band, with a frequency hopping transceiver and many FHSS carriers. To reduce interference and allowing multiple access to the same medium, both Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) are used.

BLE architecture

The BLE architecture (shown in Figure 1.11) consists of 3 big layers stacked on top of the other:

- LE controller: it implements physical and link layer
- HOST: it implements l2cap and service layer
- Application: API used by firmware designer to implement his own stack.

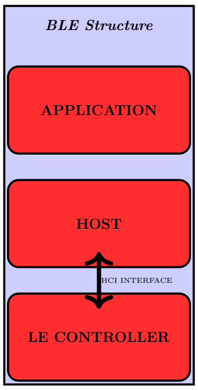


Figure 1.11: Fundamental blocks of *BLE* architecture. **Host** and **LE Controller** communicates using HCI Interface.

A minimal implementation of the BLE core covers the four lower layers as well as two common service layer protocols. [52, p.202]

Le controller

LE Controller implements the lowest levels of the entire stack. The main actors are here presented, for more details, please look at [1, Vol.6].

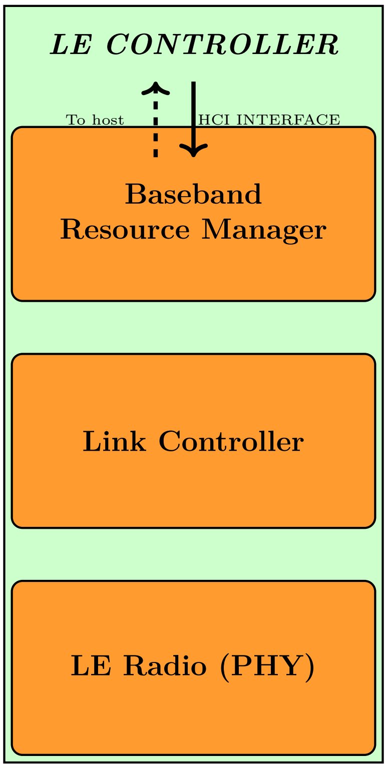


Figure 1.12

- Baseband Resource Manager (BRM): It is responsible for all access to the radio medium: on one side, BRM negotiates "access contract" with those entities interested in physical channels, while on the other side it grants them the previously agreed time on physical medium.

- LE Radio (PHY): The LE system operates in 2.4GHz, with 40 RF channels, 3 of which are used only as advertising channels, with $frequency_{hop} = 2\text{MHz}$ (Figure 1.13). Gaussian Frequency Shift Keying(GFSK) together with Frequency Hopped Spread Spectrum (FHSS), to minimize interference over the air, are used as modulation technique, allowing a maximum ***theoretical*** bit rate equal to 2Mbit/s. In practical application, however, some limitations are imposed, like hardware/software limits, BLE stack implementation or other constraints. Time Division Duplex (TDD) scheme is used, providing the effect of a full-duplex communication.

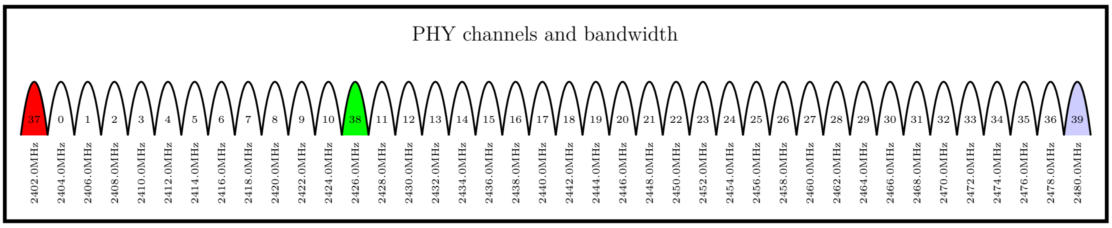


Figure 1.13: Physical channels. Colored channels are used only for advertising events. Total bandwidth occupied by BLE technology is 80 MHz

- Link Controller: it is responsible for the creation and modification of logical links, as well as updating physical connection parameters between devices. It works as a finite state machine, where the most important roles are [1, p.2587]:
  - Scanning: the device listens for some *advertiser* to initiate a connection procedure. It is called ***scanner***.
  - Advertising: the device sends advertising packets, to be discovered from scanner device.It is called ***advertiser***.
  - Connection: connection procedure is completed: from now on the advertiser and scanner are bound together until a DISCONNECTION procedure occurs or signal is lost between the two devices.

Each device is identified by a device address, a 48-bit number, which can be unique among all existing devices (Public device address) or generate at runtime (Random Device Address): this address is sent over the air during advertising events.
Link manager also controls the packet format ([1, p.2865]): for LE Uncoded PHYs, the general on-air packet is described in Figure 1.14. Access Address (AA) must not be confused with Device Address: AA is used to prevent collision and interference, from other sources, during connection: in fact, Device Address is only exchanged during advertising and not during connection.

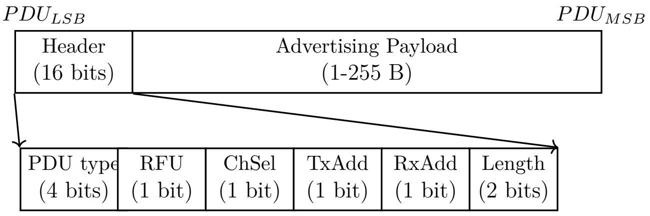
LSB MSB

Preamble (1 or 2 B)	Access-Address (4 B)	PDU (2-258 B)	CRC (3 B)

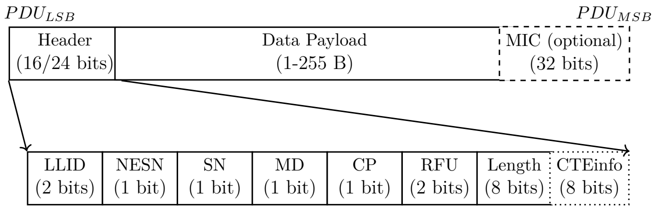
Figure 1.14: On-air packet format

The **PDU (Physical Data Unit)** changes, according to the state (advertising or connection) of the device: in few words, Advertising PDU is different from Data PDU, although the overall packet remains the same.

- Advertising PDU [1, p.2871]: Different type of *Headers* are available, each of them describing different advertising events, via *PDU type* field, constraining the length and content of Advertising Payload.



– Data PDU [1, p.2892]: It contains a mandatory header, where LLID field describes the *scope/aim* of the packet and the Data payload associated. When exchanging data, the LLID is equal to $0b01$ or $0b10$, and Data payload is described by higher layers, as explained later. NESN/SN fields are used for data flow and assuring that no packet is lost at Link Layer (this mechanism allows for re-transmission of packets).



Host

Host implements the higher layers of Bluetooth protocol stack. It consists of 5 main layers, some of them further divided.

- L2CAP [1, Vol. 3, Part A]: The Bluetooth Logical Link Control and Adaptation Protocol(L2CAP) supports higher-levels protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information. Its functioning is based on L2CAP channels, created, managed, and closed by L2CAP channel manager, used for transport of service protocols and application data. The ordering of transmission of PDU from higher layers to LE controller is instead controlled by L2CAP resource manager.

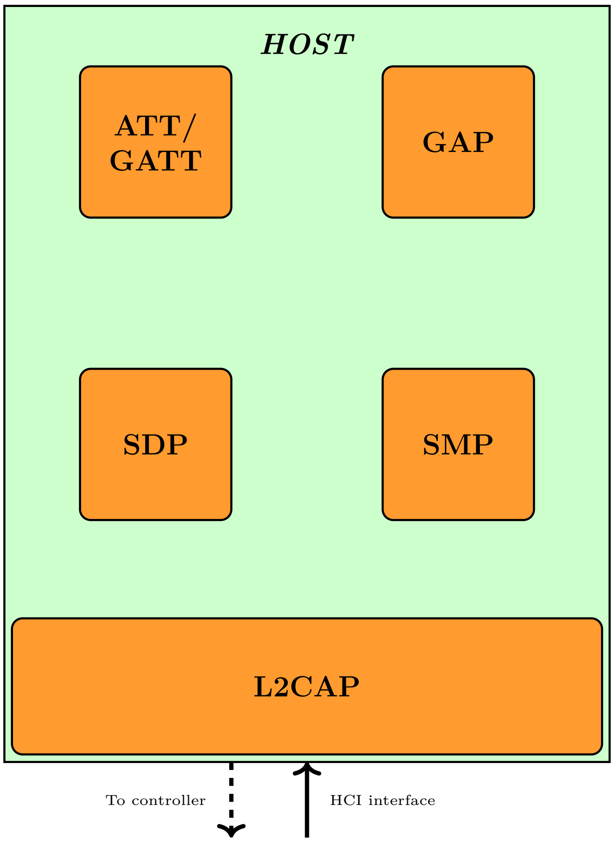


Figure 1.15: Simplified version of [1, Page 203, Figure 2.1]

- Security Manager Protocol (SMP) [1, Vol. 3, Part H]: SMP is the peer-to-peer protocol used to generate and store encryption and identity keys. It communicates directly with LE controller, but operates only over a dedicated fixed L2CAP channel.

- Service Discovery Protocol (SDP) [1, Vol. 3, Part B]:SDP provides tools for applications to discover which services and correspondent characteristics are available on peer-device. Services and characteristics are the way in which BLE stores his data, as explained in ATT/GATT section.

- Generic Access Profile (GAP) [1, Vol. 3, Part C]: It introduces definitions, recommendations and common requirements related to modes and access procedures that are to be used by transport and application *profiles*: moreover, it describes behavior of devices in standby and connecting state, in order to guarantee the reliability and establishing connection between different Bluetooth devices. A *profile* defines a selection of messages and procedures from the Bluetooth SIG specifications, in order that all Bluetooth devices (also from different manufacturers) behave and act in the same manner and can understand each other without problems.
  GAP specifies different *roles* (as Link Layer, but they ***must not be confused***, and they ***are not related at all!***) [1, p. 1283, Table 2.1]:

- Broadcaster [1, p. 1285]
- Observer [1, p. 1285]
- Peripheral [1, p. 1285]: Any device that accepts the establishment of LE active physical links: it corresponds to the Link Layer Slave role. In "standby state", it is an advertiser who can be discovered by scanners in proximity. The BLE protocol has been developed in order to require few resources, optimizing power consumption and memory usage, perfect for embedded low-power systems.
- Central [1, p. 1285]: Any device that initiates the establishment of LE active physical link: it corresponds to the Link Layer Master role. It is responsible for establishing the connection, so in idle state it *must* be a scanner (although it could also behave as an advertiser). The BLE protocol is asymmetric, meaning the Central role needs more resources to work.

In Figure 1.16, an example of existing network topologies is shown.

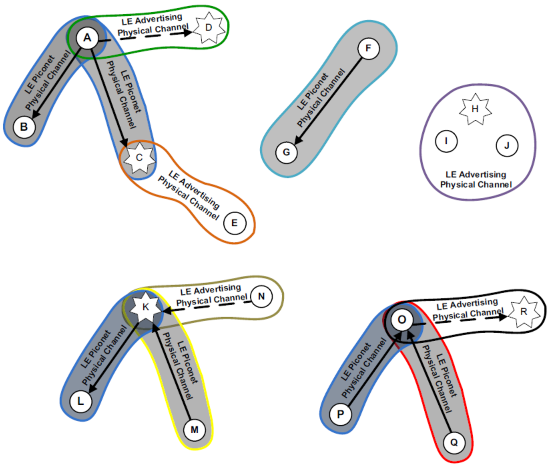


Figure 1.16: Some examples of possible network topology with BLE. The important thing to notice is that a device can be both peripheral and central at the same time (as K or O), but one peripheral can be connected only to one central at a time. More information here [1, p.257]

GAP also describes different *modes* [1, p.1294]: a mode can be described as a well defined temporary state, in which a connected peer can perform some *procedures* or to achieve a particular goal. These modes can be programmed by a firmware designer or chosen by the final user via GUI (Graphical User Interface). For example, when a device is in *discoverable* mode, it can be "seen" by another Bluetooth device, and some *procedure* can be performed on it, as connection or *service discovery*.
In addition, GAP, together with SMP, manages all the aspects of a secure and encrypted communication, a primary design goal in BLE technology.
Inside GAP's core specification, other two additional parts are available, used in particular to customize BLE device:

  - Advertising data format [52, Chapter 3]: It describes how the advertising packet is composed, with the possibility to add or to hide some fields, according to the final use case
  - GAP service [52, Chapter 3]: This is a mandatory *service*, explained in 1, describing some important features of a BLE device, as the *device name*.

- Attribute Protocol (ATT) [1, p.1476]: ATT describes how 2 devices can exchange data: in particular, an ATT *client* request some data (here called *attributes*)to an ATT *server*. As already stated above, ATT *client and server* must not be confused with all the other roles described above, as *peripheral, central, master, etc.*. Each attribute (i.e., piece of information) is associated with three properties:

  - Universal Unique IDentifier (UUID): it is a 32 or 128-bit value, used to identify the attribute. The probability of 2 attributes having the same UUID is almost zero.
  - Attribute handle: it is a 16-bit value assigned by each server to refer to a particular attribute during a connection.
  - Attribute permissions: They describe the possible operations that could be performed on an attribute. An attribute could be readable and/or writeable.

  Some ATT operations can be performed, as changing the Maximum Transmission Unit (MTU, length of ATT payload), read/write operations, and other [52, ATT]. The maximum size of custom data to be exchange (without header) is equal to 244B, which forms a total ATT Payload of 251B (considering mandatory header).

- Generic Attribute Profile (GATT) [1, p.1532]: It works on top of ATT and defines a framework used to exchange data and profiles over BLE connection. Data is organized hierarchically in *services*, which contained user pieces of data or meta-data, called *characteristics*. A *characteristic* is always described by at least two attributes: a *declaration*, which contains some information about the characteristic, and a *value*, the piece of information itself (Figure 1.17).

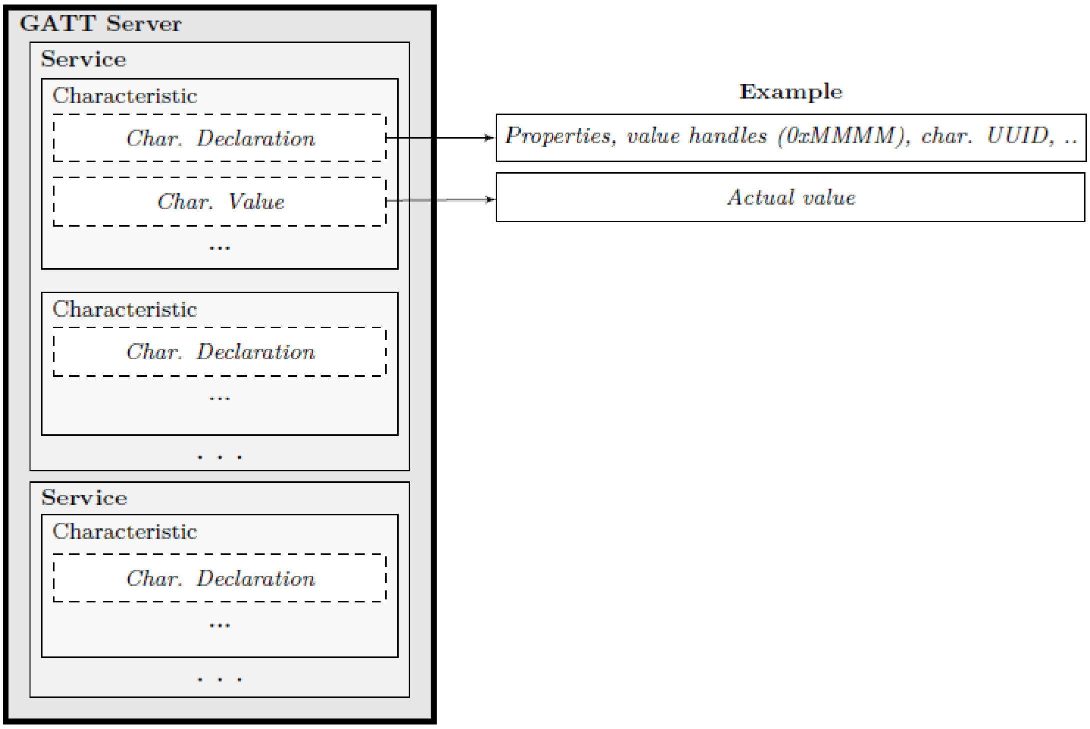


Figure 1.17: GATT Architecture. Figure taken from [42]

When data must be exchanged between 2 BLE devices, some procedures must be followed in order to achieve the transmission [52, p.96-97]. In our application, *Characteristic Value Notification* has been chosen: this operation permits the server to send *notification* packets, without the need of acknowledgment from the client(s), permitting the best throughput and the minimum power consumption.

Application

Application layer is the highest layer, interfacing directly with software/firmware engineer. Different types of stack are available, depending on use case and embedded hardware used in developing BLE©. In this work, chosen hardware is nRF-52840DK by Nordic Semiconductor©, with its powerful and well-documented software development kit *nRF5 SDK v. 17.00.*([37])

Advertising, scanning and connection events

In Figure 1.18, the advertising and scanning procedure is explained. Advertising packets are only sent on channels 37, 38, 39: the most important parameter is the *advertising interval*, the time between different advertising events: in this period, three advertising packets (one for each channel) are sent over the air. The scanner most important parameters are

scanner scan interval and *scanner scan window*, respectively, time between different scanning events and duration of a single scanning events. During scanning events, the scanner device switches on the antenna in the correspondent channel: if it answers to advertising packets, active scanning is performed.
If a scanner is in proximity and listens in the same physical channel, a connection (depending on advertising packet), can occur.

Advertising and Scanning events

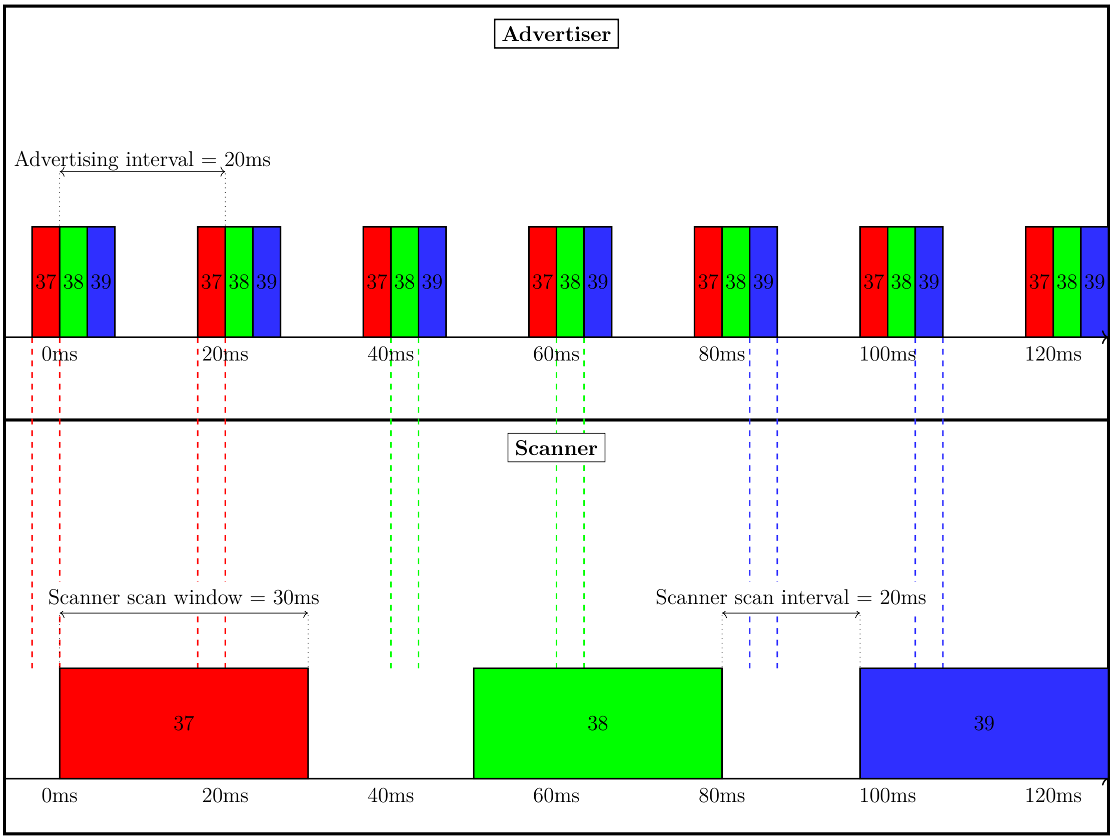


Figure 1.18: The advertising ahd scanning events are represented: they occurs only in Channel 37,38,39. The associated parameters are chosen considering both power consumption and latency between advertising and connection event.

It is described by three important parameters:

- Connection interval [1, p. 2981]: It is defined as the time between 2 consecutive connection events. It should be between 7.5 ms (achieving the maximum possible throughput but consuming more power) to 4 s.

- Slave latency [1, p. 2981]: It is defined as the number of connection events that a Link Layer Slave can decide to skip before the connection is lost.

- Connection supervision timeout [1, p. 2981]: It's defined as the maximum time between 2 packets before considering the connection lost.

In Figure 1.19, mode of communication during connection is explained. Once it is established, every *connection event*, the master (in transmission mode) sends a message to the slave (in reception mode) to inform him that it is ready to receive some data: if the slave must exchange some pieces of information, then it will send them, otherwise it will exchange a small message to keep the connection alive.

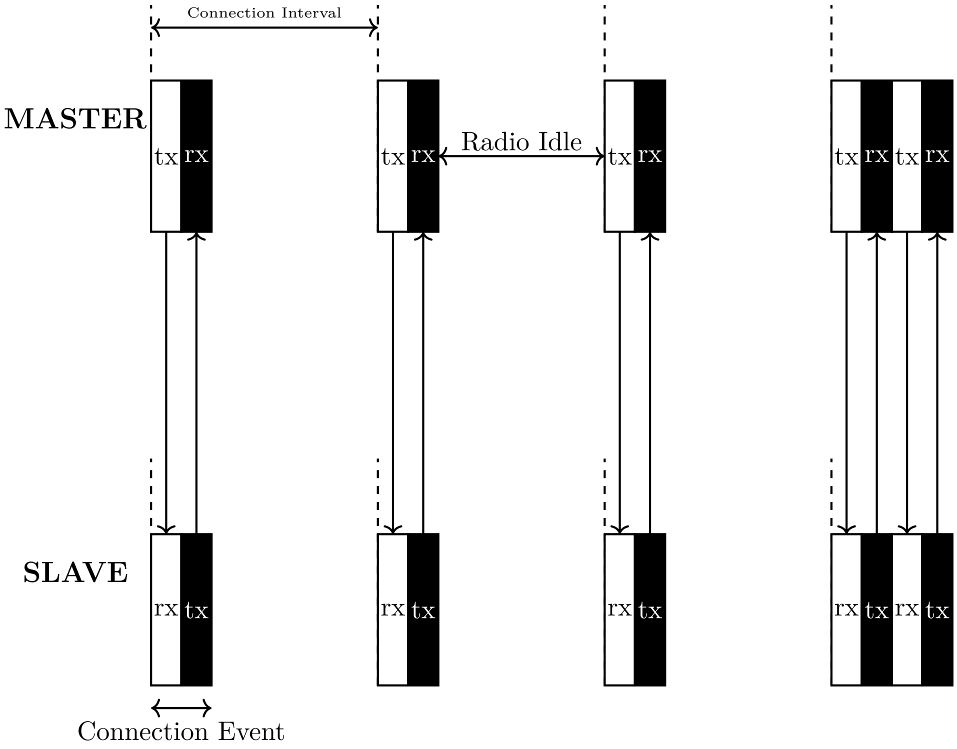


Figure 1.19: Connection events.

BLE Considerations

The aim of the present work is to exploit different protocols in order to achieve the maximum throughput with minimum power consumption and transmission latency. Here, some theoretical considerations are made.

BLE Throughput From BLE Core Specification [1], the maximum reachable throughput is 2Mbit/s. However, in practical implementation, some constraints that limit the throughput are present. As stated above, in fact, BLE devices in a connection can communicate only during predefined connection events, separated in time by *connection interval.* Moreover, the ATT maximum size, if *Data Length Extension* is enabled [33], is equal to 244B. Adding the fact that during connection events, the total number of packets sent is dependent both by the stack used both by the physical time needed to send one packet, the theoretical value is never reached. Assuming a minimum value of 7.5ms and 5 packets for each connection interval:

$$BLE\_throughput = \frac{1000[\text{ms}]}{conn\_interval[\text{ms}]} \cdot ATT\_SIZE[\text{B}] \cdot 8 \cdot number\_packets =$$

$$= \frac{1000}{7.5} \cdot 244 \cdot 8 \cdot 5 = 1.301\text{Mbit/s}$$

Transmission Latency Transmission latency is the time between the dispatch and the reception of the message. The problem, in a real implementation, is the identification of dispatch and reception time: not always the stack used permits to know exactly when the packet is sent/received, and so the time used is the call to the *sending routine* and the notification when the message is received. In a theoretical world, the time needed for a 244-bytes packet would be:

$$Latency^{2} = \frac{Packet\_size}{bit\_rate} = \begin{cases} 976\mu\text{s} & \text{bit\_rate} = 2\text{Mbit/s} \\ 1952\mu\text{s} & \text{bit\_rate}= 1\text{ Mbit/s} \end{cases}$$

This time is used to determine the maximum number of packets inside a connection event.

Power Consumption Power consumption and battery life are strictly dependent on the stack and hardware used. The main source of consumption is achieved in transmission when the radio is on: usually, values between $-40dBm$ and $20dBm$ are used considering both power consumption and distance, according to use case.

IEEE 802.15.4

IEEE 802.15.4 [3] is the communication protocol implemented by the Institute of Electrical and Electronics Engineers (IEEE) for Low-Rate Wide Personal Area Network (LR-WPAN), focused on limited power and flexible throughput. [18] It implements only the lowest level of ISO/OSI stack, the *PHY* and *MAC* layer, leaving higher layers to firmware engineers. *Zigbee*, *Thread*, and *6LoWPAN* are implemented above this standard: their performance could be comparable (or worst) to the one achieved with 802.15.4 due to higher processing of data, and for this reason the will not be further processed. In this section, a brief overview of the standard is given: for further reading, please have a look at [18] and [25]

Network Topologies

Inside an IEEE 802.15.4 network, two different types of devices can be found:

[2] This value takes into account the time needed to encode and decode data, as a function of bit rate: the physical time needed for the signal to travel from A to B is negligibile

[3] In this thesis the 2006 version is used because of the hardware used, although the 2020 version is available on line

- Full Function Device (FFD): A device that can operate as a *coordinator*
- Reduced Function Device (RFD): A device that implements only a minimal implementation of the stack, used in simple applications, like light switch. It cannot operate as a coordinator.

Depending on the use case, two available network topologies are available, *star* and *peer-to-peer*. As seen in Figure 1.20, both of them include a PAN *coordinator*, used to begin, terminate and route the communication over the network (together with other use case applications): it is responsible also for choosing the PAN *identifier*, a 16-bit number used to identify that particular network, as described in [18, Sec. 7.5.5]. In this way, devices from different PAN can communicate between them. Each peer inside a PAN is identified by a 64-bit *extended address*, that could be shortened during communication (*short address*). Usually the coordinator is a main-powered device, while all the other peers use batteries as energy storage.

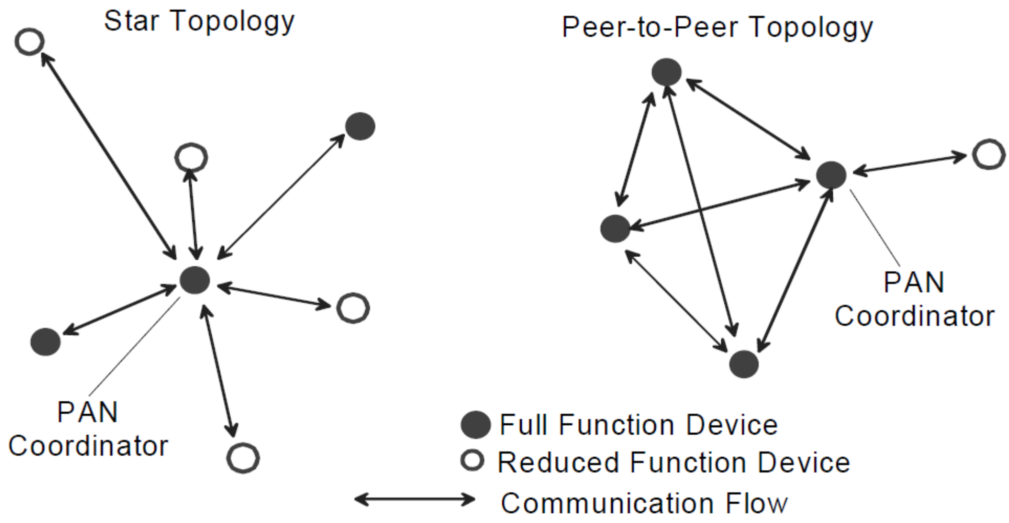


Figure 1.20: Figure taken from [18, Figure 1]

Architecture

As already stated, IEEE 802.15.4 architecture is composed of two main layers, each of them sub-divided into little entities that perform some operations: a *PHY* layer used to transmit and receive packets from air medium and perform some *Energy Detection* and other operations and a *MAC* layer that permits transmission of information between physical layer and higher layers (Figure 1.21).

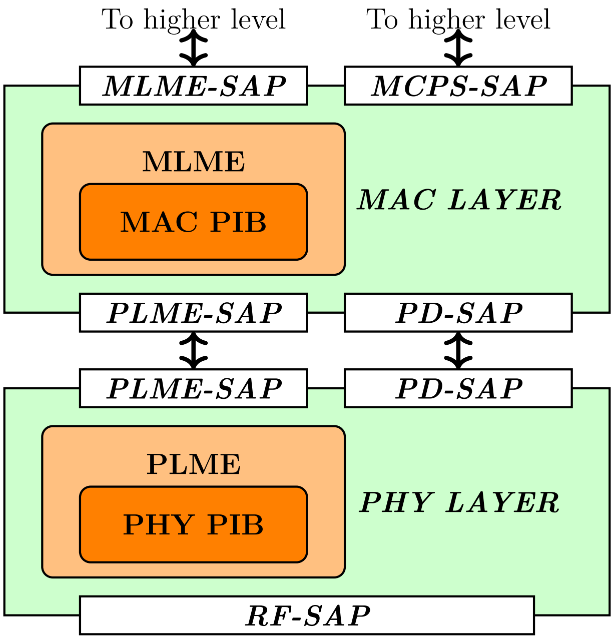


Figure 1.21: IEEE 802.15.4 stack

Without going too deep into the protocol's details, the communication between user and layers is achieved through used of *primitives* [18, Section 5.6], well explained in Figure 1.22:

- Request: A request for service is passed from the user to the correspondent layer
- Confirm: It contains the result of a previous request
- Indication: It contains the result of an internal event (i.e. packet received), a sort of notification for the user
- Response: It is the event generated by the user after an indication primitive has been received.

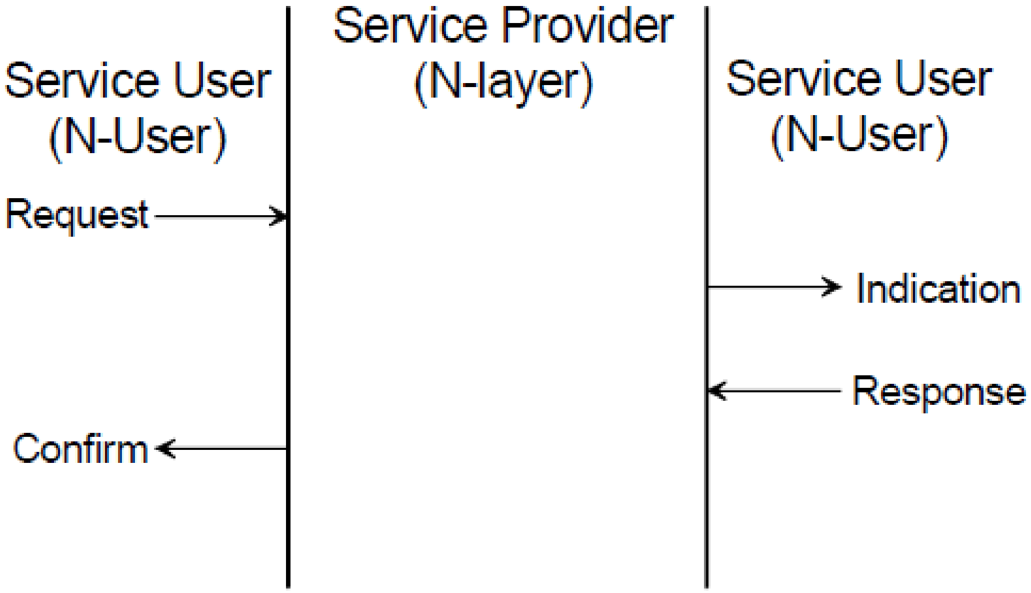


Figure 1.22: Figure taken from [18, Figure 14]

PHY layer In its lower layer, the standard specifies three different available frequency bandwidth, with different modulation schemes, as seen in Table 1.1. Spread Spectrum techniques are used in order to reduce interference over the air. In this application, ISM band (2400 MHz) is used, with 16 available channels, separated by 5MHz (Figure 1.23). When transmitting and receiving data, *MAC* layer need a finite amount of time to process the packet: in order to avoid some problems, like antenna overheat, two successive transmitted packets must be separated by at least an InterFrame Spacing (IFS) period: this is different according to the length of the frame. [18, Section 7.5.1.3]

Table 1.1: Different type of Physical layer

Frequency Band (MHz)	Chip rate (kchip1/s)	Modulation	Bit rate (kbit1/s)	Symbols
868-868.6	300	BPSK	20	Binary
902-928	600	BPSK	20	Binary
2400-2483.5	2000	O-QPSK	250	16-ary Orthogonal

PHY layer is responsible for different tasks, each of them accessible through *primitives* approach:

- Activation/Deactivation of transceiver
- Energy Detection (ED) [18, Section 6.9.7]: it is an indication of the energy received

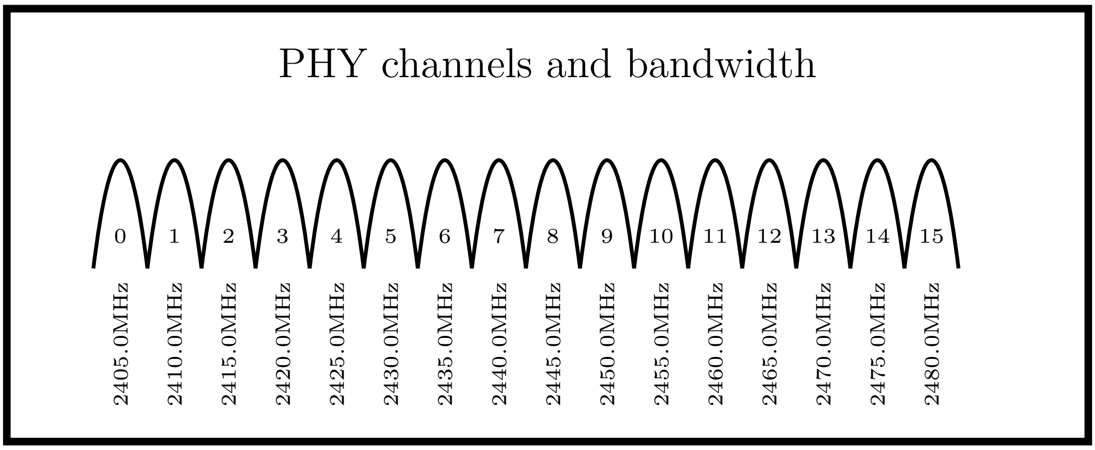


Figure 1.23: IEEE Physical channels. The formula to determine center frequency is $F_c = 2405 + 5 * \mathrm{k}$

in a specific channel. Usually, it is performed to choose the "cleanest" channel, maybe prior to the initialization of a PAN or before peer-to-peer communication. No decoding is applied to the received signal.

- Link Quality Indicator (LQI) [18, Section 6.9.8]: It is performed to characterize the quality of the received frame: it may be used by higher layers to choose the best channels.
- Clear Channel Assessment (CCA) [18, Section 6.9.9]: it is a mechanism for determining whether the medium is busy or not. In few words, before sending a frame, a CCA is performed: if the medium is busy (according to a fixed threshold), the packet will be retransmitted later.
- Channel frequency selection: the ability to tune the transceiver to a specific channel, requested by higher layer.

As one can see in Figure 1.24, the phy layer is composed by:

- PHYsical Layer Management Entity (PLME) [18, Section 6.2.2]: It manages the access to the services described in 1. It is accessible from higher layer, always using *primitives*, through *PLME-Service Access Point (SAP)*, which provides meaning of communication and configuration to the services. Some attributes and constants are store inside *PHY PAN Information Base (PIB)*, accessible and writable (in some cases) using PLME-GET.request *primitive*.
- PHY Data (PD) - Service Access Point (SAP) [18, Section 6.2.1]: It allows the flow of data with higher layer, using *primitives*.
- Radio Frequency (RF)-SAP: It manages and configures RF hardware and firmware.

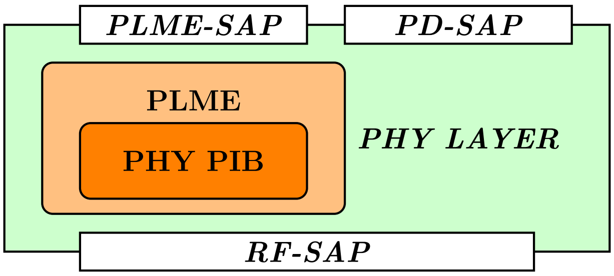


Figure 1.24: PHY layer's architecture

Medium Access Control (MAC) layer . *MAC* layer allows communication between higher layers and physical one, as well as many other services, like reliable communication between peers, device security, supporting PANs starting and maintaining, and others. [18, Section 7].

Its architecture is shown in Figure 1.25 Some SAPs are present, allowing communication with other layers: the working is similar to the one explained in 1, with different services activated. For more details, see [18, Section 7.1].

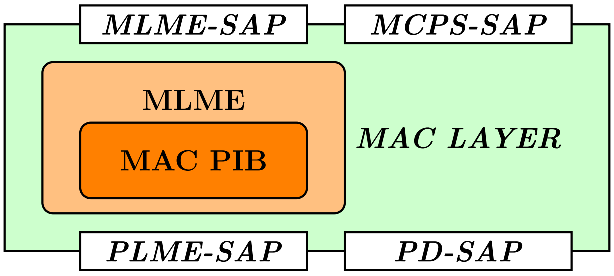


Figure 1.25: MAC layer's architecture

The payload described by IEEE 802.15.4 MAC layer is shown in Figure 1.26. To understand details off all fields, please refer to [18, Section 7.2.1]. The important thing to notice is that the *maximum configurable payload* is 118 B, as stated in [18, Table 85].

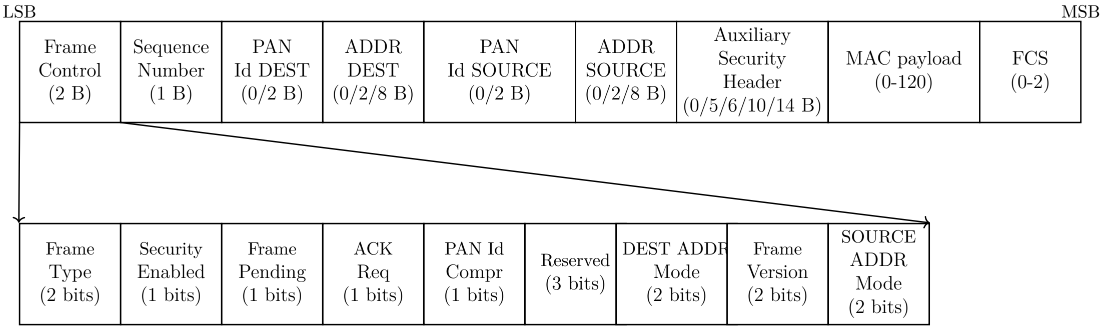


Figure 1.26: On-air packet format

Frame Control [18, Section 7.2.1.1] field controls some parameters of communication, type of message *Frame Type*, security aspect, as well as the behavior of other variable fields. For example, if *ACK Request* bit is set, the message will be ACKnowledge by the receiver. Sequence number [18, Section 7.2.1.2] is an incremental 1 octet, identifying the current frame. Destination/Source PAN Identifier [18, Section 7.2.1.3] represents variable-length identifier (depending on *Frame control* field) of PAN receiver/transmitter devices. In this way devices from different PAN can communicate (if in range). Destination/Source Address [18, Section 7.2.1.4] contains addresses, within PAN, of receiver/transmitter devices. Auxiliary Security Header [18, Section 7.2.1.7] controls security aspects of communication. FCS [18, Section 7.2.1.9] field is a 16-bit CRC, used in error checking.
Devices communicate, exchanging previously discussed packets, in two different ways:

- Beacon-Enabled Mode [18, Section 7.2.1.1]: In a beacon-enabled network (Figure 1.27), the coordinator periodically sends a beacon frame (black rectangle), containing rules to medium access, to synchronize all connected peers. After beacon is sent, Contention Access Period (CAP) starts: during this time, each connected peer can communicate with coordinator only in its previously assigned slot. A variable length Inactive Period follows CAP.

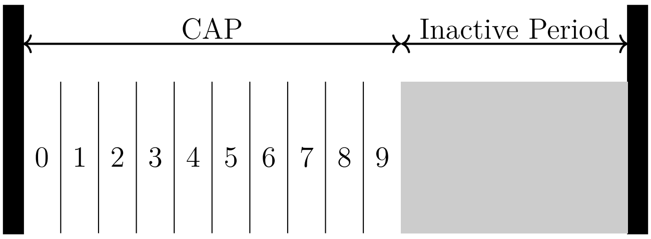


Figure 1.27: Beacon enabled mode. Black rectangle represents the start of each transmission

- Non beacon-enabled mode: In this mode, no beacon or finer control is used. A transmitter can send data whenever he wants after performing unslotted CSMA-CA algorithm.

Retransmission [18, Section 7.5.6.4.3] of data is allowed if ACK bit of *Frame Control* is set and occurs if an ACK frame is not received within *macAckWaitDuration* from the end of previously transmitted frame.

IEEE 802.15.4 Considerations

The aim of the present work is to exploit different protocols in order to achieve the maximum throughput with minimum power consumption and transmission latency. Here, some theoretical considerations are made.

Transmission Latency Considering 2.45GHz band, the maximum theoretical bit rate should be 250kbit/s. Using the *maximum configurable payload size* of 118B (corresponding to a total packet length of 127B, due to mandatory header), leads to:

$$Latency^{4} = \frac{Payloadsize}{bitrate} = \frac{127 \cdot 8}{250} = 4064\mu\text{s}$$

Throughput From IEEE 802.15.4 specification [18, Section 6], the maximum theoretical bit rate is 250kbit/s. As already stated, between 2 different transmitted frames, certain fixed amount of time must pass: from specification [18, Section 6.1.3], the maximum LIFS (Long Interframe Spacing) period is equal to 40 symbols, corresponding to 640 µs

$$\text{LIFS} = \frac{\#symbols}{\text{k}symbols\text{1/s}} = \frac{40}{62500} = 640\mu\text{s}$$

Summation of *Latency* and *LIFS*, lead to maximum achievable throughput (or minimum permitted connection interval in analogy with BLE technology) of (here, 118B is used because it is the configurable payload):

$$Connection\_Interval = Latency + LIFS = 4064 + 640 = 4704\mu\text{s}$$

$$Throughput = \frac{1000[\text{ms}]}{conn\_interval[\text{ms}]} \cdot Payload\_Size[\text{B}] =$$

$$= \frac{1000}{4.7} \cdot 118 \cdot 8 = 200.9\text{kbit/s}$$

Power Consumption Here, the same considerations made in Section 1 are valid. However, IEEE 802.15.4 is a protocol developed to favor battery-powered device, so the overall power consumption should be less than BLE technology.

[4] This value takes into account the time needed to encode and decode data, as a function of bit rate: the physical time needed for the signal to travel from A to B is negligibile

Nordic© Proprietary Protocol

As explained later, the hardware chosen for this work is owned by Nordic© Semiconductor. For *nNRF 52840-DK* used here, some drivers and Software Development Kit (SDK) developed by Nordic's engineers already exist. [46]
In [46, Section 6.20], radio drivers are presented. The multidomain 2.4GHz transceiver supports Frequency-Shift Keying (FSK) modulation, allowing implementation of both *Bluetooth© low energy* 1Mbit/s and 2Mbit/s, *IEEE 802.15.4* 250kbit/s and proprietary protocol (in this case, only *physical layer* is implemented, leaving higher layer to firmware engineers). The latter is explained in this section.
The radio peripheral is easily configurable thanks to automatic packet assembler/disassembler, CRC generator/checker, and optional data whitening/de-whitening. IFS Controller is used to configuring the Interframe Spacing, the minimum amount of time between 2 different transmitted and/or received packets. RAM peripheral is directly connected to packet assembler/disassembler (after performing address matching) through EasyDMA (Figure 1.28). This module can be used only in half-duplex mode.

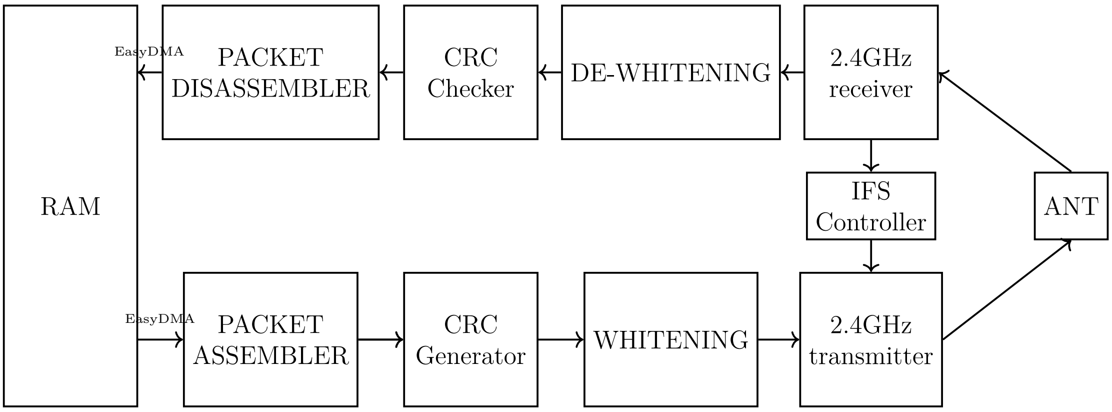


Figure 1.28: Radio driver implemented by Nordic©

The configurable packet, stored inside RAM region, is shown in Figure 1.29; different fields are present:

- Preamble: used to identify the protocol used
- CI, TERM1, TERM2: used in certain features of Bluetooth Low Energy long range mode
- Length: Dimension in bytes of payload
- Payload: The size in byte of $PAYLOAD + S0 + S1 + LENGTH$ must be less than **258B**
- ADDR BASE and PREFIX: These two fields together form the on-air address. The transmitted address must match one of the enabled received addresses implemented on the receiver side. These addresses ***are not*** used to identify different boards in a network, but refers to physical channel on which the antenna listens. [46, Section 6.20.2]

LSB MSB

PREAMBLE 1B	ADDR BASE	ADDR PREFIX	CI	T1	S0	LENGTH	S1	PAYLOAD	CRC	T2

Figure 1.29: Packet format

The entire module has been developed using *state machine* approach. The antenna can be only in one state at a time: transitions between them are possible thanks to some *tasks* (es. START, STOP, RXEN, DISABLE, etc...) triggered by user or *events* triggered by the system itself (es. reception of a frame, etc...)

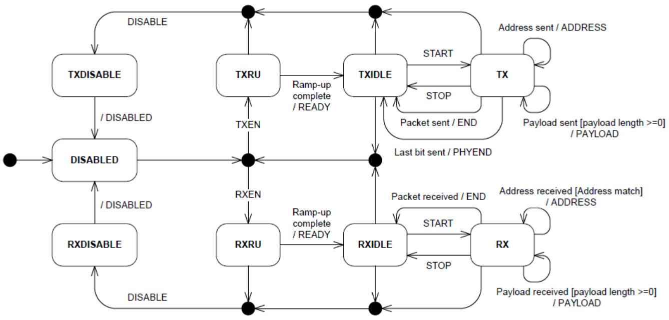


Figure 1.30: State diagram of Radio peripheral. [46, Figure 110]

Different registers are used to control both antenna configuration (as the modulation scheme, transmit power, frequency channels) , on-air packet configuration and firmware implementation (triggering certain *task*, adding integration with other peripherals, setting proper Interframe Spacing and more). [46, Section 6.20.14]

Considerations

Using the lower level of an entire stack directly allows the implementation of a simpler protocol, tailor-made for this use case but not as standardized as *IEEE 802.15.4* or *Bluetooth©* *Low Energy*.

Transmission Latency As already stated, transceiver supports both 1Mbit/s and 2Mbit/s, impacting on transmission latency. Assuming a 260B payload:

$$Latency^{5} = \frac{Packet\_size}{bit\_rate} = \begin{cases} 1040\mu\text{s} & \text{bit\_rate=2Mbit/s} \\ 2080\mu\text{s} & \text{bit\_rate=1Mbit/s} \end{cases}$$

In this case, performances are equal to *Bluetooth©* *Low Energy* technology, for the same bit rate reached.

[5] This value takes into account the time needed to encode and decode data, as a function of bit rate: the physical time needed for the signal to travel from A to B is negligibile

Throughput The throughput reachable with this technology is strictly dependent on the firmware written and its optimization. The Interframe Spacing can be manually set: here the calculation are made considering $TIFS$ = 1000µs and $Payload\_Size$ = 250B. Proceeding as already done in Section 1:

$$Connection\_Interval = Latency + TIFS = \begin{cases} 2040\text{µs} & \text{bit\_rate=2Mbit/s} \\ 3080\text{µs} & \text{bit\_rate=1Mbit/s} \end{cases}$$

$$Throughput = \frac{1000[\text{ms}]}{conn\_interval[\text{ms}]} \cdot Payload\_Size[\text{B}] = \begin{cases} 980.4\text{kbit/s} & \text{bit\_rate=2Mbit/s} \\ 649.4\text{kbit/s} & \text{bit\_rate=1Mbit/s} \end{cases}$$

From the calculation, the throughput is not so closed to the theoretical bit rate: however, optimizing firmware, the Interframe Spacing can be further reduced to values in the order of tens of µs.

Power Consumption *NORDIC*© provides some antenna electrical specifications ([46, Section 6.20.15]). However, the real consumption can be significantly increased by other external factors, as peripheral used, voltage regulator, or firmware optimization. Here, some antenna characteristics are presented.

Description	Value	Units
Operating Frequencies	2360-2500	MHz
PLL Channel Spacing	1	MHz
TX Only Run Current (@+8dBm)	32.7	mA
TX Only Run Current (@+4dBm)	21.4	mA
TX Only Run Current (@ 0dBm)	10.6	mA
TX Only Run Current (@-4dBm)	8.1	mA
TX Only Run Current (@-8dBm)	7.2	mA
TX Only Run Current (@-12dBm)	6.4	mA
TX Only Run Current (@-16dBm)	6.0	mA
TX Only Run Current (@-20dBm)	5.6	mA
TX Only Run Current (@-40dBm)	4.6	mA
TX start up current (@+4dBm)	11.0	mA
RX Only Run Current (1Mbit/s)	9.9	mA
RX Only Run Current (2Mbit/s)	11.1	mA
RX Start up current (1Mbit/s)	6.7	mA
RX sensitivity 1Mbit/s	-93	dBm
RX sensitivity 2Mbit/s	-89	dBm

Table 1.2: Radio electrical specification

It has to be noted that both *Bluetooth*© *Low Energy* and *IEEE 802.15.4* are built on top of the presented radio driver: so, antenna consumption should be the same across the different protocols.

Ultra Wide Band Technology

Ultra Wide Band (UWB) technology is a recent technique for ultra low-power, short-range systems, suitable both for low and high data rate. Firstly, it was used for radar application and indoor positioning, although recently medical applications (for telemonitoring system) have been investigated. The FCC (Federal Communications Commission) allocates a frequency bandwidth of 7.5GHz, from 3.1GHz to 10.6GHz in America, while in Europe, it is formed by two bands: one in range 3.4-4.8GHz, the other in 6-8.5GHz

The main characteristic for UWB communication is the usage of a larger bandwidth, obtained by exchanging very short impulse in the order of ns(although other techniques to achieve a larger bandwidth are suitable, as frequency hopping, OFDM or direct-sequence CDMA [16]), using On-Off Keying modulation(shown in Figure 1.32), Burst Position Modulation, or other modulation techniques: this technology is called *Impulse-Radio(IR) UWB*. In order to standardize it, a minimum bandwidth of 500MHz or *fractional bandwidth (FB)* greater than 0.2 is required. The *FB* is defined as the bandwidth at -3dB (w.r.t maximum power achieved), divided by *center frequency* (Figure 1.31).

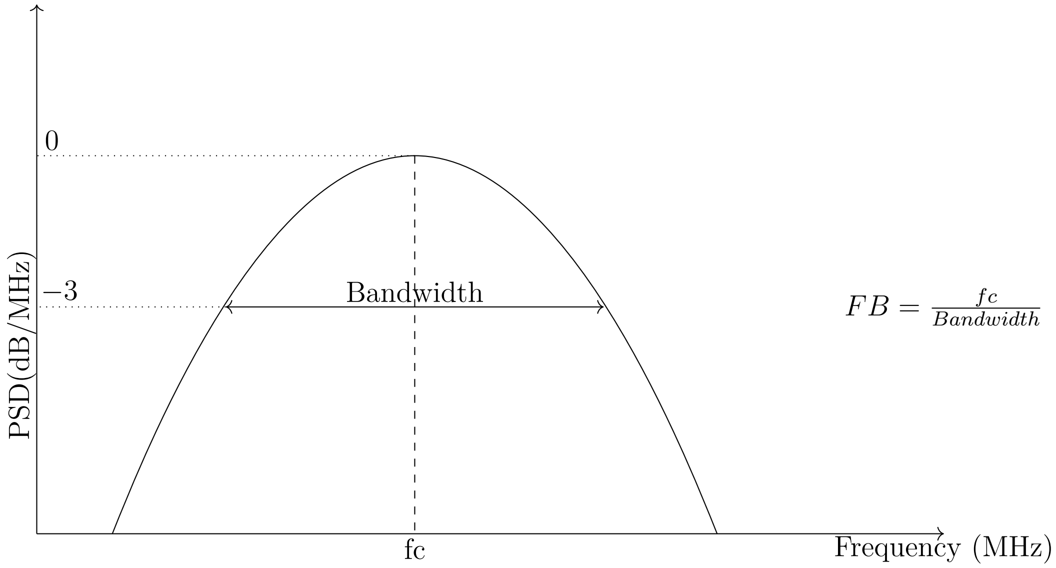


Figure 1.31: Fractional Bandwidth definition

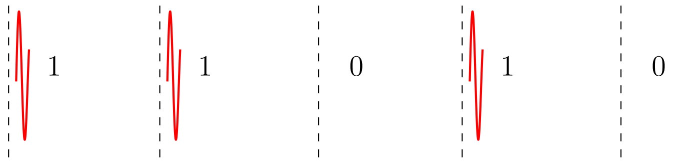


Figure 1.32: IR-UWB On-Off Keying, In synchronized- OOK, a previously assigned time-slot is used to expect an information bit: then a sort of OOK modulation is assigned in each slot.

In order to allow the sharing of the same medium by multiple users, it is often combined with Time Hopping Multiple Access, in which each user can communicate only in a previously assigned time slot. The advantages of UWB technology lies in the fact that it can be used as an overlay to existing RF-systems: this can be understood by looking at Figure 1.33. The RF-system, working in a narrow bandwidth, will see the UWB signal as noise (the red area inside green area), thus not interfering in the communication. Moreover, the power consumption is kept low, due to the spreading of energy in a much larger bandwidth, with respect to RF-Narrow Band systems where a decrease in antenna power drastically reduce the overall performance transmission.

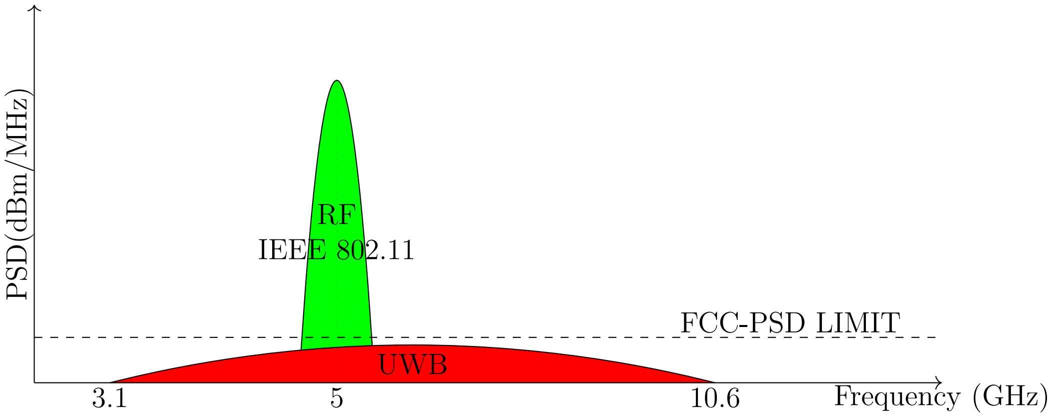


Figure 1.33: UWB Technology compared to other common RF-system (as IEEE 802.11 which can operates also in 5GHz center frequency). The PSD limit imposed by FCC is in reality with a more complicated shape, but for sake of clarity has been shown in a simpler manner. [2, Fig.1]

The characteristics of an UWB communication protocol is highly correlated to its implementation and the modulation used: medium power consumption is in the order of some mW, reaching very high data rate([16, Table 2-3]) and latency (for one single pulse) depending on distance but less than 200ns([14]).
The increasing interest in UWB technology led the *IEEE* to employ it as a possible physical layer in definition of *IEEE 802.15.4* standard in 2007 ([18, Version 2011, Section 14]). A mix of Burst Position Modulation (BPM) and binary phase-shift keying (BPSK) is used (Figure 1.34), allowing both coherent (with phase recovery) and non-coherent (without phase recovery) receivers, reaching a maximum data rate of 27.24Mbit/s with a maximum Pulse Repetition Frequency (PRF, indicating the maximum number of UWB pulses per second) of 499.2MHz. Bandwidth and center frequency described by the standard are shown in Figure 1.35.

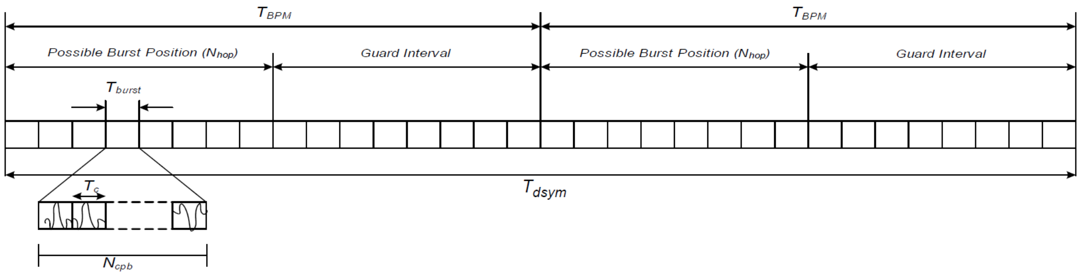


Figure 1.34: Figure taken from [18, Version 2011, Figure 85]. 1 symbol is capable of carrying 2 bits of information: changing value of above parameters, bit rate is tuned accordingly

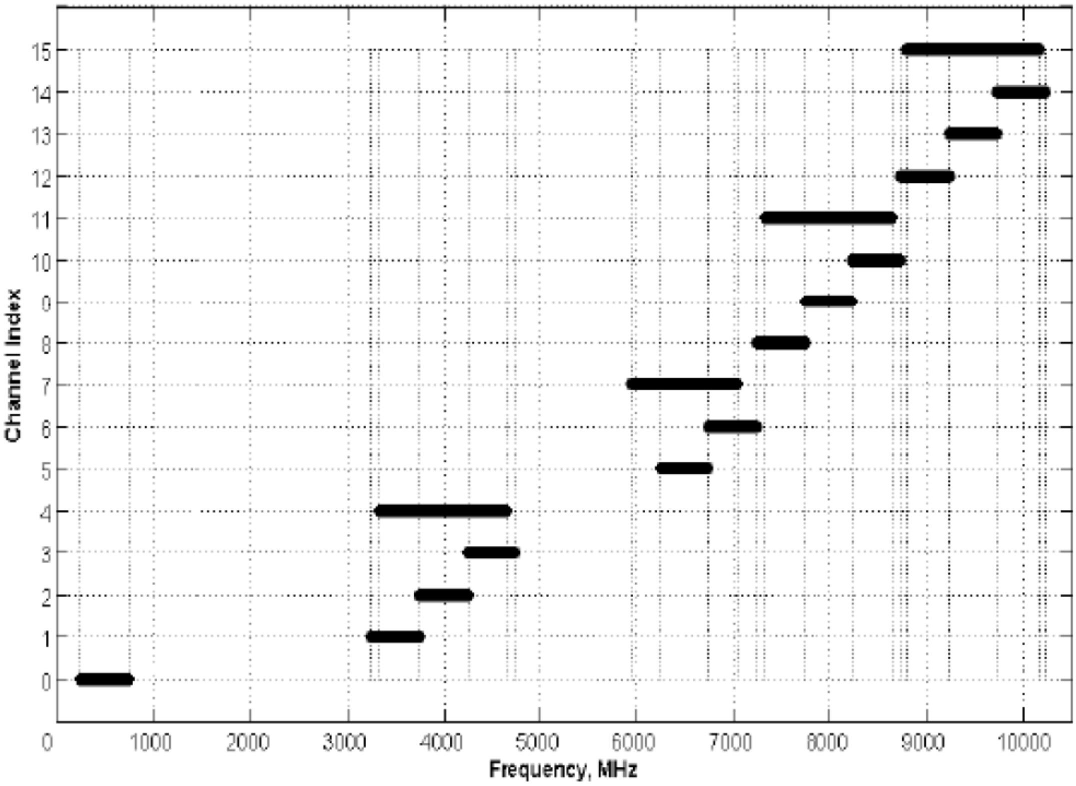


Figure 1.35: Center frequency and correspondent bandwidth for the different channels defined by *IEEE 802.15.4* standard. Figure taken from [18, Version 2011, Figure 94]

State of art

In recent years, wireless communication has increased its importance in the biomedical field. The increasing interest in e-health system, IoT, and big data for medical applications need a huge amount of data from different patients: the use of wireless communication became of fundamental importance to avoid cable encumbrance. Moreover, the need for construction of Body Area Network, in order to place different sensors on a patient's body, fits perfectly inside the wireless world.
This thesis is placed under a bigger project, well described in [43], [45] and [34]. Authors showed a maximum throughput (using *Bluetooth*© *Low Energy*) of 1kB/s transmitting raw emg, and 28B/s of ATC data. The power consumption for acquiring and transmitting raw sEMG was 23.47mW, while in the ATC case 20.23mW. The low throughput can be explained by the need for a higher *connection interval*: in fact, for this use case, it was set to 130ms.

In [22] and [35], authors present the need for wireless body area networks, exploiting different communication protocols used (with their pros and cons). Some critical aspects to keep in mind when developing a BAN are presented, as well as already existing applications.

In [6], authors propose a wireless acquisition system of multiple bio-signals, using *Bluetooth*© *Low Energy*. A comprehensive study has been made, taking into account both latency transmission and current consumption. The hardware used is a custom-made board, using *nrf52840-SoC*. The measured latency is shown in Figure 1.36, while current consumption is listed in Table 1.3.

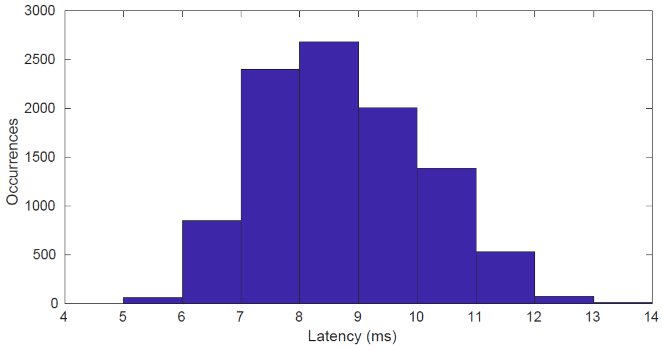


Figure 1.36: Histogram of the measured latency of a train pulses at 800Hz, with 10000 occurences. The connection interval was set to 7.5ms. [6, Figure 11]

States	Current(µA)	Description
ADVERTISING	200	Bluetooth LE enabled
CONNECTED	1200	Connection established
STREAMING	4600	3 EMG channels streaming (TX POWER = 0dBm)

Table 1.3: Current consumption measured in different states of BLE device. [6, Table 4]

In [7], Brunelli et al. exploit both *BLE* and *Low Power Wifi* for a multichannel sEMG acquisition system in prosthetic hand control. Firstly, they compare (Table 1.4) the most used wireless protocols in order to choose the best protocols in terms of power consumption, number of possible acquisition channels, and latency. *BLE* and *LP-WiFi* are then exploited. The performance are shown in Figure 1.37. They managed to implement an 8-channels acquisition system at 500Hz, corresponding to 64kbit/s, consuming 17.5mW for *BLE* and 70.37mW for *LP-WiFi*.

Protocol	Throughput$_{\text{application}}$(kbit/s)	Throughput$_{\text{physical layer}}$(Mbit/s)	Power Consumption
BLE	305	1	low (∼ mW)
Zig Bee	115.2	0.250	low (∼ mW)
6LoWPAN	72	0.250	low (∼ mW)
ANT	20	0.06	ultra low (∼ µW)
LpWiFi	16000	54	high (∼ W)

Table 1.4: Comparison between different wireless protocol. 2 throughput are given: the physical one takes into account also the mandatory protocol's overhead. [7, Table 1]

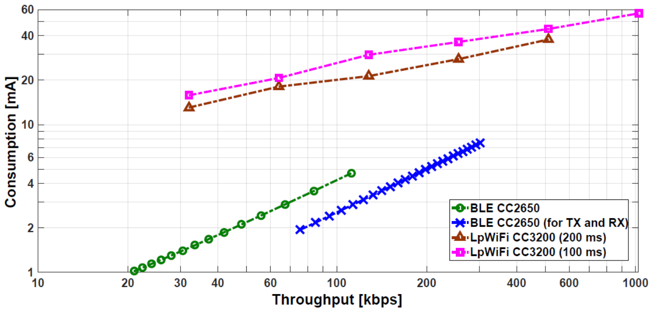


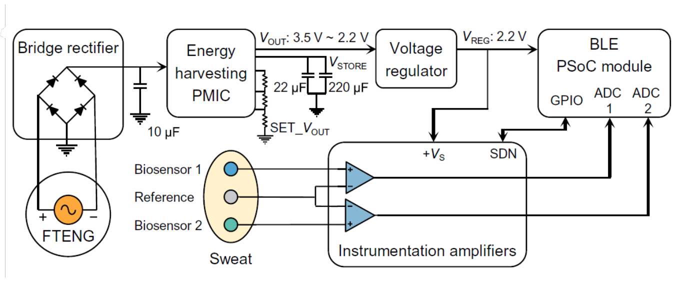
Figure 1.37: [7, Figure 7]

In [10]-[9], authors proposed a high-density wearable sEMG acquisition system, powered by a 3.7V, 600mA h LiPo battery, assuring a lifetime in transmitting mode of around 5h. The wireless protocol used is *WiFi*, for the need of a high throughput: the maximum data rate achieved was of 9Mbit/s, with an average latency of 12ms. Current consumption was pretty high due to the number of channels and wireless communication: 119mA in transmitting mode, corresponding to roughly 440mW. In this work, the focus was set on a high number of channels and throughput, discarding the power consumption.

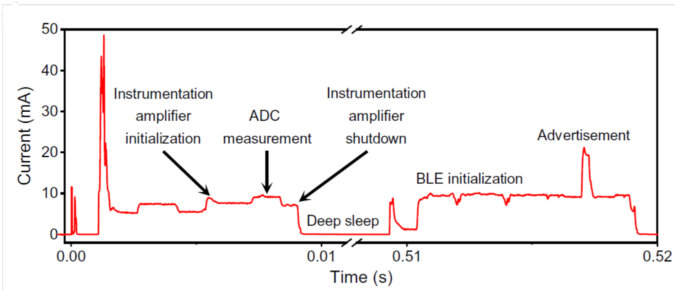
In [50], *IEEE 802.15.4* is investigated. Packet delivery ratio is studied, changing both the inter-arrival time and the packet size, in a different testing scenario (Body Area Network with 1/2/3 different nodes). They showed that increasing the packet size decreases the correct delivery ratio: the same effect has been obtained, decreasing the inter-arrival time.

In [5], the authors proposed a custom acquisition board with a proprietary *IEEE 802.15.4*. In the first part of the study, they compare different previous works: then, they decided to implement their own wireless protocol, reaching a maximum throughput of 180kbit/s, with a synchronization time of 1µs and a low (∼ mW) power consumption, reaching an autonomy of 8h with a standard coin cell battery (LIR2450 Li-ion).

From a futuristic perspective, the work described in [47] is really promising. They managed to implement a little BAN, composed of only 1 transmitting node (sweat bio-sensor), using *Bluetooth© Low Energy*, powered by human motion (so virtually battery-free) using *triboelectric nanogenerators (TENGs)*. They are a novel technique for energy harvesting, in which mechanical energy is converted into electrical energy, exploiting both triboelectric and inductive phenomenons. A capacitor is then charged and discharged every time there is need for some current. In Figure 1.38, the electrical circuit and the current provided are shown.



(a) [47, Figure 4]



(b) [47, Figure 4]

Figure 1.38: **a**) The developed electrical circuit by the authors. **b**) Current consumption from wake-up to transmitted data.

Wireless transmission of ATC data has already been accomplished using IR-UWB together with synchronized-OOK in [41], where ATC is used to estimate force. Transmitter and receiver are well explained in [15] and [13]: transmitter side manages to consume 32pJ$/pulse$ with $center frequency$ = 4GHz maintaining receiver power consumption less than 5mW

Inside [28], an UWB system is proposed using Integral Pulse Frequency Modulation (IPFM), where a spike is produced every time the integral of a modulating signal exceed a certain threshold: this spike triggers UWB pulse-generator, which will produce 2 UWB-pulses. Total power consumption is 40.48mW, spending 80pJ$/pulse$ allowing for data rate equal to 0.5Mbit/s. This system could be applied in situations where IPFM of the biological signal is significant, as Heart Rate Variability (HRV).

In [17], authors proposed a custom OOK IR-UWB for biomedical devices; a brief review of existing technology is made. Data rate reaches is around 250Mbit/s, while the energy required to receive 1bit is equal to 72pJ

Matherials and Methods

This thesis' aim is the development and comparison of different wireless protocols in a really simple BAN, including only one transmitting and receiving node. In this section, an overview of the hardware used and methods developed to acquire, transmit and process ATC data is given.

Transmitting/Receiving board

The hardware chosen to implement the transmitting and receiving node is the commercial *nRF-52840-DK* ([37]), a really powerful development kit (based on *nRF-52840 SoC* [46]), compatible both with *Bluetooth*© *Low Energy*, *IEEE 802.15.4*, *Zigbee*, *Thread*, *ANT*, and proprietary 2.4GHz protocol. As one can see in Figure 1.39, the main components of the board are:

- nRF-52840 SoC: Microprocessor (and all the components needed) mounted on the board (red).
- PCA 10056: Onboard debugger, in order to flash microcontroller without the need of other external hardware, but only plugging in an USB cable (green).
- General Purpose Input/Output (orange)
- Programmable buttons and LEDs (white)
- 5 different power sources (black):
  - 1: Li-Po connector
  - 2: External power supply at 3.3V
  - 3: USB connector, both working as power supply for the entire board, as well as loading firmware directly from the laptop
  - 4: nRF-52840 SoC power supply, in order to switch on only the microprocessor
  - 5 (not shown in the picture): coin cell battery connector, on the bottom side of the board.
- PCB antenna (light blue)

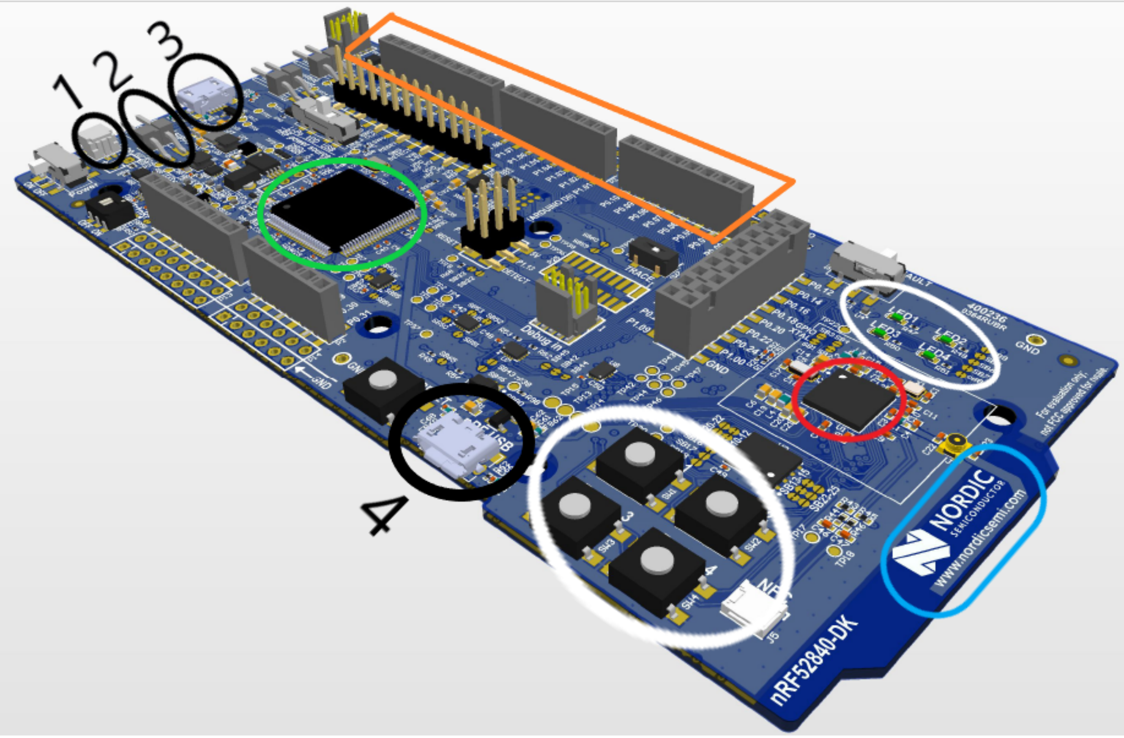


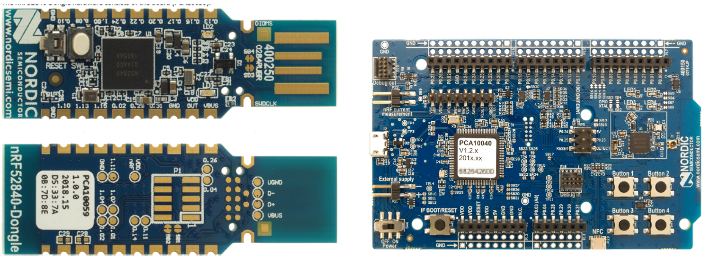
Figure 1.39: nRF-52840-DK

The SoC mounted on board supports different peripherals, all listed in [46, Section 6]. In this work, only some of them are used:

- GPIO/GPIOTE [46, Section 6.9-6.10]: This module allows the user to read/write the 32 pins independently, even if some of them are used by other peripherals. It also gives the possibility to connect events to some tasks (for example, triggering a specific task (i.e led on) after a specific events occur, as passage of a pin from low to high).
- Radio [46, Section 6.20]: It gives information on how to access the radio transceiver
- Timer [46, Section 6.30]: This module allows easy access to timer/counter, which runs on the high-frequency clock source (already mounted).
- Universal Asynchronous receiver/transmitter (UART) [46, Section 6.33]: It allows communication between different UART peripherals. It is normally used to transmit some data to a laptop for further processing.

Supporting hardware

During the entire development of firmware, some other hardware (for debugging purpose, collection of data, or other) has been used. In particular, in the first part of *BLE* development, the *nRF-52840-Dongle* ([38]) has been used to check the correct functioning of the developed transmitter. The dongle is a "small" version of the development kit, without the on-board debugger (even if with some tricks, it can be programmed externally). It has been used with *nRF Connect*, a software application developed by *Nordic* (Figure 1.40). In order to collect data for latency measurement (time between transmission and reception), an additional board (nRF-52 DK [39]) has been used (Figure 1.40). It is really similar to the one used for the transmitter/receiver, but with a different on board debugger and a different layout. It has also been used as simulation of ATC signals, in order to check the reliability of transmission.



(a) nRF52840 Dongle

(b) nRF52-DK

Figure 1.40: Additional hardware used

Test methods

The main purpose of the study is to find the right wireless protocol for ATC transmission (that will be used for classification system, as explained in [34]), as well as the best technique to keep all the information contained in the signal, with low power consumption, latency, and high throughput. The overall system has been tested with only one channel, although more channels could be added with small changes to the firmware.

BLE, IEEE 802.15.4, Custom protocol

In Figure 1.41 the general idea developed is presented. Taking into account the idea of a low power device, it has been decided to transmit only in a pre-fixed amount of time (not only in *BLE* technology, but also with *IEEE 802.15.4* and custom protocol), while in the period when the radio is off, a *relative* (relative to the start of a new window) timestamp is associated at each event (only the leading edge[6]). In this way, each event can be reconstructed knowing the *relative* timestamp and the *window* in which occurs. The timestamp is accomplished by starting a new *timer* at the beginnning of each window: to keep the payload's size low, an **8-bit**,**31250**Hz timer is used (allowing for a maximum window length of 8.19ms). In this way, the maximum theoretical latency will be roughly equal to the window length plus transmission latency (more details will be made available later).

[6] In the first part of the study, a timestamp was associated also to the trailing edge, but this idea was abandoned for redundancy of data

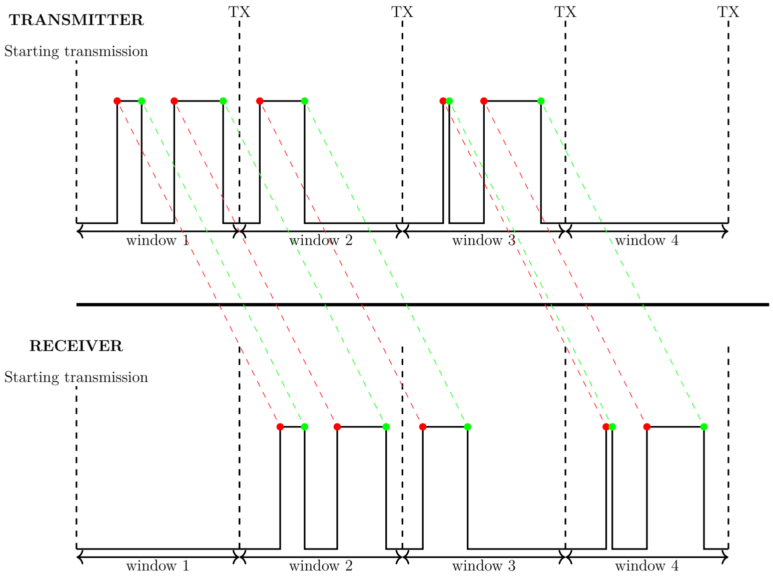


Figure 1.41: General idea to transmit ATC signal. The reconstruction of the signal is made by the receiver board: if the trailing edge timestamp is not available, the signal is keep at digital 1 for 300µs. In this figure, the transmission latency is not shown (for simplicity reason)

Testing procedures

Different testing procedures were performed in other to debug and provide results on power consumption, transmission delay, and throughput. In this section, each of them is explained.

Power Consumption

In order to evaluate the power consumption of the board, the procedures listed in [40, Section 9] were performed. In particular, an external 10Ω resistor is put in series to the nRF-52840DK power supply: in this way, the current absorbed by the chip is evaluated recording the voltage drop on the resistor, amplified by a fixed gain using *Instrumentation Amplifier*. An *INA126* has been used, with no gain resistor, bringing a gain of 5: the scheme is shown in Figure 1.42. An oscilloscope has been used to record and acquire the output: sampling frequency and internal ADC resolution are tuned to have low size data but with good accuracy, avoiding aliasing.

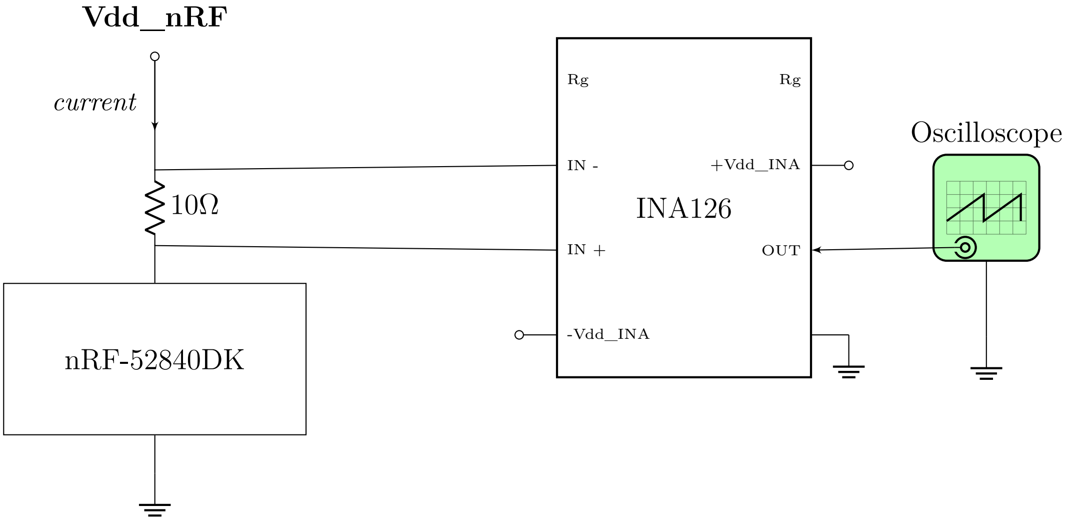


Figure 1.42: Current measurement scheme. The oscilloscope used is a *Rigol MSO5104*

The current drained by the hardware is measured both in the initialization phase as well as working state in order to make some considerations on battery life.

Throughput, Latency, Similarity, and Reliability

Throughput and latency have been estimated using the setup shown in Figure 1.43. Distance between boards is constant during different trials and set to 1.5m: during experiments, the user body was put between transmitter and receiver to simulate a real scenario. ATC data was simulated using one board, using real data obtained in another separate research ([51]). Central board is used to estimate the latency between transmission and reception: every time one packet is put in the queue for transmission (tx) and available to process at the receiver side (rx), one pin on both boards is toggled: on collecting board, this toggling triggers a capture on a running timer and a serial transmission of data to a personal laptop. In this way, both throughput and latency can be estimated.
Transmitter and receiver send to PC the tx/rx packet, in order to study the **reliability** of transmission. In particular, a percentage error is calculated, intended both as packet lost and events lost: the former is used to characterize the transmission while the latter to quantify the reliability of ATC reconstruction (not all packets lost could contain some events)
Jitter, intended as the variability of latency, is also estimated. A good protocol should have low latency and jitter as well as high throughput that should be related only to payload size and connection interval set.

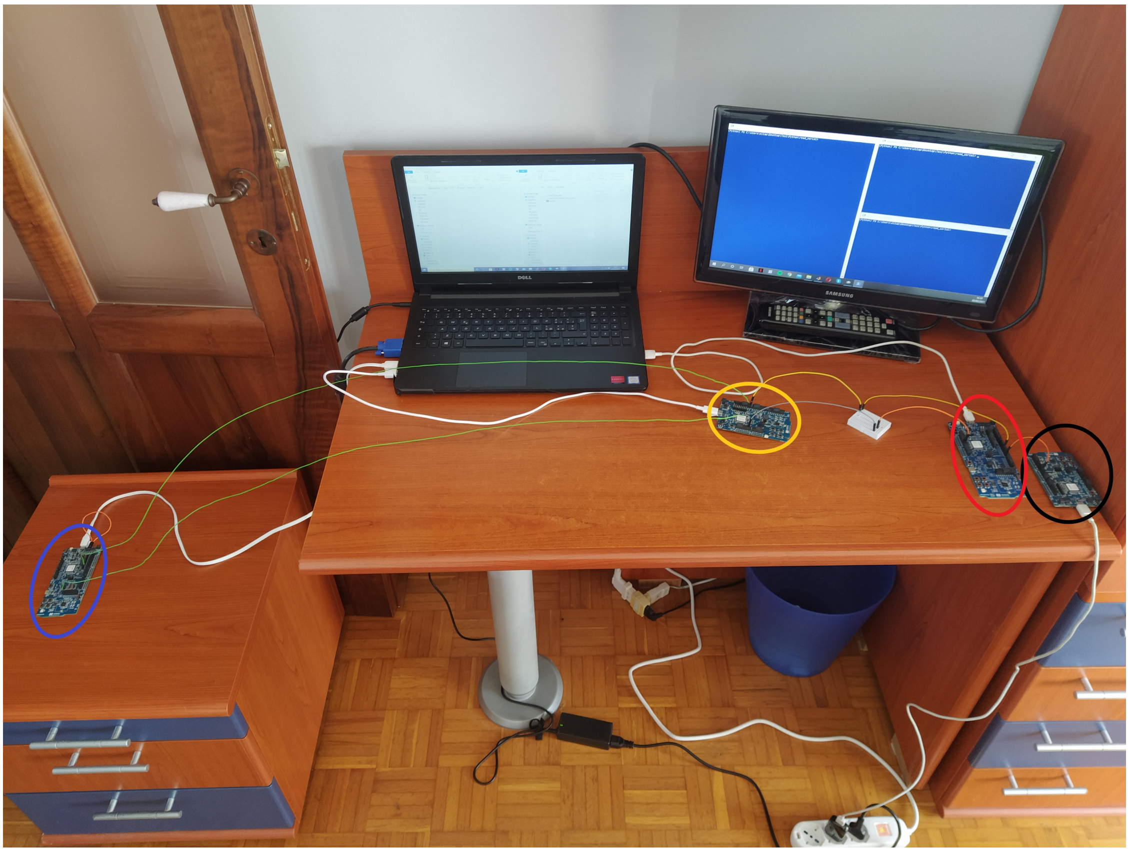


Figure 1.43: Experimental setup to estimate throughput and latency between packets. Black: ATC simulation, Red: transmitter, Yellow: collection board, Blue: receiver

On the laptop, *Python* scripts (3 blue screens on television) run to collect serial data from transmitter,receiver, and collection board, which will be post-processed using *Matlab*©. Moreover, some considerations on ATC signal have also been performed: the idea is to assess the **similarity** between transmitted and rebuilt signal. To accomplish this, two different similarity measures, ranging between 0 (total inequality) and 1 (total equality) have been used:

- Cosine Similarity: It is defined as the cosine of the angle between 2 different non-zero vectors. In order to compute the index, the Formula 1.3 has been used ([54]).

$$cos(A,B) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \quad where\ A = (A_1, A_2, \ldots, A_n), B = (B_1, B_2, \ldots, B_n) \tag{1.3}$$

- Ruzicka/Jaccard index: It measures the similarity between finite positive vectors, computing the area of intersection divided by the area of union. In digital processing, the Jaccard index (also known as Ruzicka similarity) is computed using Formula 1.4 ([55]).

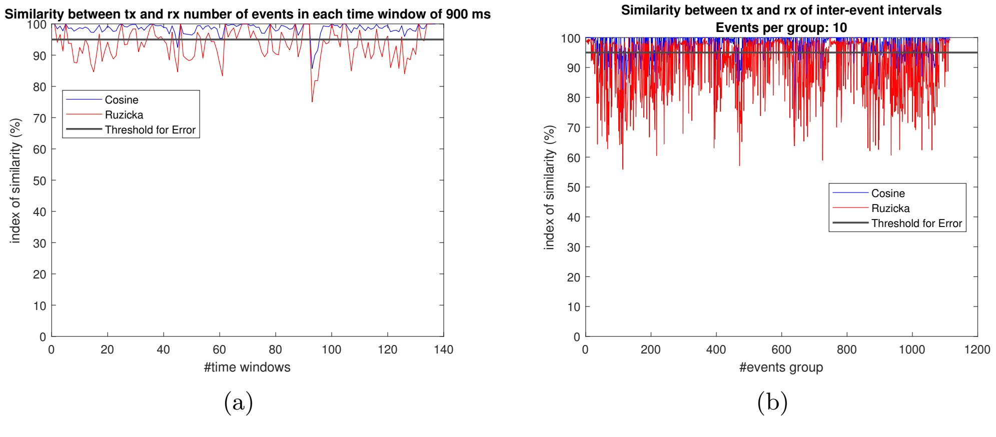
$$J(A,B) = \frac{\sum_{i=1}^{n} \min(A_i, B_i)}{\sum_{i=1}^{n} \max(A_i, B_i)} \quad where\ A = (A_1, A_2, \dots, A_n), B = (B_1, B_2, \dots, B_n) \tag{1.4}$$

These statistics have been performed on 2 different features:

- *Number of events inside corresponding windows*: firstly, the total transmission is divided into smaller epochs (around 15–30 ms), in which the number of events is counted. In this way, the total ATC transmission is characterized by a long array, containing the number of events in each epoch. Due to the high number of data, in order to catch inequality between tx and rx during transmission, this long array is reshaped inside a matrix: in this way, the similarity is assessed between shorter epochs and not on all the transmission

- *Distance between events*: firstly, a sort of event rate is performed, calculating the distance in time between $event_n$ and $event_{n+1}$. Then, for the same reason as above, the similarity is performed on shorter number of correspondent events

The obtained values are mean together in order to estimate the overall similarity during one transmission: to take into account possible anomalies during transmission, an error is estimated normalizing the number of times in which similarity goes below 95%. See Figure 1.44 for details



(a)

(b)

Figure 1.44: **a)** Similarity between number of events in correspondent time-windows using both cosine and ruzicka index. Threshold to calculate error is also shown. **b)** Similarity between inter-events distance of same events using both cosine and ruzicka index. Threshold to calculate error is also shown.

To compare different protocols between them, the above estimates are used together to perform ATC similarity as follow (2 indexes, one for number of events in same window and the other for distance between events)

$$similarity_{nevents} = \Big(\frac{similarity_{cosine} + similarity_{ruzicka}}{2}\Big) - \Big(\frac{error_{cosine} + error_{ruzicka}}{2}\Big)$$

$$similarity_{distevents} = \Big(\frac{similarity_{cosine} + similarity_{ruzicka}}{2}\Big) - \Big(\frac{error_{cosine} + error_{ruzicka}}{2}\Big)$$

In this way, both similarity and error are considered together to calculate only one number to describe reliability: indexes would be in the range $0 - -1$, with higher value meaning higher ATC similarity. In cases where error is too big, negative values could be possible, meaning a total dissimilarity. At the end $similarity_{ATC}$ is calculated as a mean of the above indexes, having only one number (at fixed antenna power level) to characterize ATC similarity.

In addition to this, the **mean and median frequency** of events distribution is performed and compared.

Firmware and custom protocol development

In this chapter, the firmware, and stack implementation for the different tested protocols are shown.

Bluetooth© Low Energy

The framework used to implement *Bluetooth© Low Energy* is a pre-compiled stack, implemented by *Nordic©* ([36]). It uses different peripherals (as timers, real timer counters, etc.), which are not accessible by user. Moreover, only high level functions are available to be used, while low level ones are already compiled: in this way, the behavior of transmitter and receiver board is not totally controllable, and the performance are strictly correlated with stack implementation provided by *Nordic©*.
Two boards have been developed using *Bluetooth©*, one transmitter, and one receiver side. In the following, each board is explained.

Transmitter

The purpose of the transmitter board is to timestamp each ATC event and send it to the receiver board for post-processing of data, as explained in Section 1. The functioning of the board can be seen as a Finite State Machine (FSM), as seen in Figure 1.45. The first state is the **OFF** one (in which the entire board is off), and no current is drained from the battery. With a mechanical switch, the board goes in **ON** state, where a configuration is performed: in particular, the needed peripherals are configured according to the use case, power management is performed (registering the useful event handler (WFE)), and the entire *BLE* stack is initialized. Once the configuration is finished, the device enters in **ADV**ertising state, in which it starts broadcasting advertising packets, as explained in Figure 1.18, waiting for a connection procedure. If a scanner is in proximity, a connection can be achieved, and the board passes in **CONN**ection state: it will stay inside it until the central board activates the notification for the custom characteristic (explained later), triggering the passage in TX state. Here, every time the configured pin for ATC signal senses a leading and/or trailing edge, it adds to a buffer that timestamp(s) (always relative to the start of a new window): the array will be sent in the next possible connection event (Section 1) until the central switch off the notifications.

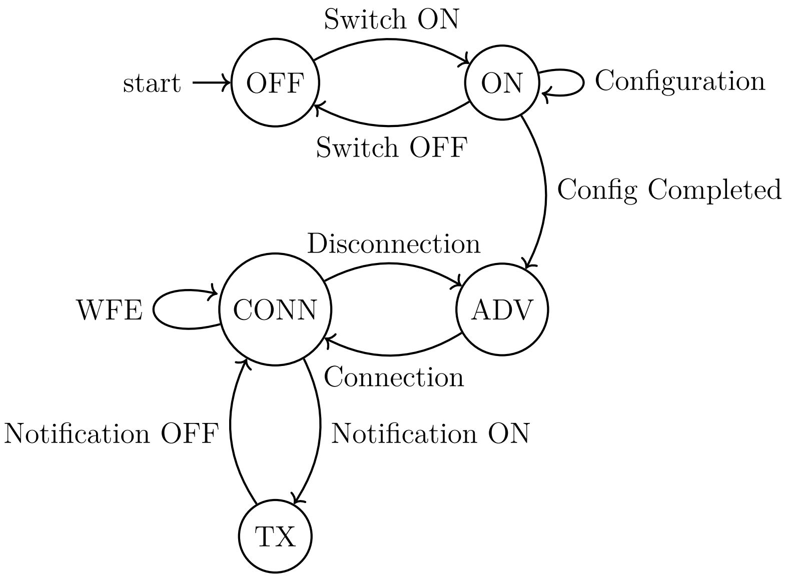


Figure 1.45: Transmitter FSM.The WFE in CONN state, means that the board is "ready" to keep some events, previously assigned during configuration.

Configuration

In Figure 1.46, the operations performed are shown.

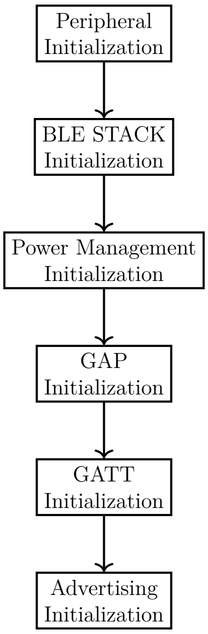

Peripheral Initialization
BLE STACK Initialization
Power Management Initialization
GAP Initialization
GATT Initialization
Advertising Initialization

Figure 1.46: Configuration process. At the end, the board passes in ADV state

In *peripheral initialization*, the explained peripherals are configured:

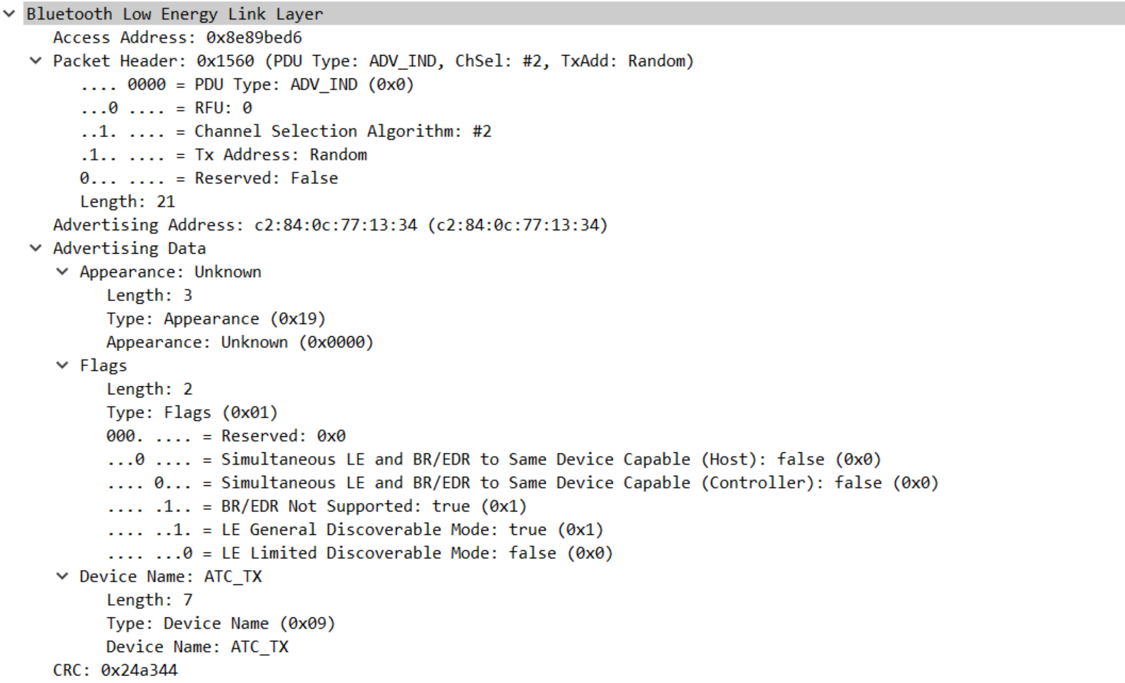
- Timer1: Used for general timestamp(for debugging purpose and for synchronization of transmission)
- Timer2: Used for ATC transmission: it works at 31250Hz with 8-bit
- UART: used for PC communication and debugging. The baud-rate is set to 1000000 without hardware flow control (HWFC)
- Real Time Counter (RTC): used for synchronizing transmitter and receiver antenna, to minimize total power consumption.
- LEDs: indication of current FSM states
- GPIO: used for ATC signal and debugging purpose

During *BLE stack initialization*, the needed resources for *BLE* functioning are allocated.
Inside *Power Management Initialization*, events handlers are registered.
In *GAP Initialization*, some important parameters of *BLE* are set (Table 1.5). In *GATT Initialization*, the module for handling GATT operations as well as the custom serivce is initialized. In order to register to the stack a custom service, a 128-bit UUID (Universally Unique IDentificator) must be assigned([53]): a 16-bit UUID must also be assigned to all the characteristics inside the service. The service and characteristic *handle* (the number identifying that particular service/characteristic during a particular connection) is assigned automatically by the used stack.

Parameter	Value
Security	None
Min Connection Interval	7.5ms
Max Connection Interval	7.5ms
Connection Supervision Timeout	4s
Slave Latency	0
Device Name	ATC_TX

Table 1.5: GAP parameters

Inside *Advertising Initialization*, the advertiser module is initialized, with an advertising interval equal to 40ms, until a connection occurs. Inside the advertising packet the device name and its appearance (what type of device it is) have been inserted. The device has been put in *General Discoverable mode*([1, p.1297]) in order to be discoverable from all the scanners in proximity. The real advertising packet is shown in Figure 1.47.



```
⌄ Bluetooth Low Energy Link Layer
    Access Address: 0x8e89bed6
  ⌄ Packet Header: 0x1560 (PDU Type: ADV_IND, ChSel: #2, TxAdd: Random)
      .... 0000 = PDU Type: ADV_IND (0x0)
      ...0 .... = RFU: 0
      ..1. .... = Channel Selection Algorithm: #2
      .1.. .... = Tx Address: Random
      0... .... = Reserved: False
      Length: 21
    Advertising Address: c2:84:0c:77:13:34 (c2:84:0c:77:13:34)
  ⌄ Advertising Data
    ⌄ Appearance: Unknown
        Length: 3
        Type: Appearance (0x19)
        Appearance: Unknown (0x0000)
    ⌄ Flags
        Length: 2
        Type: Flags (0x01)
        000. .... = Reserved: 0x0
        ...0 .... = Simultaneous LE and BR/EDR to Same Device Capable (Host): false (0x0)
        .... 0... = Simultaneous LE and BR/EDR to Same Device Capable (Controller): false (0x0)
        .... .1.. = BR/EDR Not Supported: true (0x1)
        .... ..1. = LE General Discoverable Mode: true (0x1)
        .... ...0 = LE Limited Discoverable Mode: false (0x0)
    ⌄ Device Name: ATC_TX
        Length: 7
        Type: Device Name (0x09)
        Device Name: ATC_TX
    CRC: 0x24a344
```

Figure 1.47: Real Advertising packet. It has been captured by *Wireshark*, the most used network analyzer, together with *nRF Sniffer for Bluetooth LE*, a pre-built nordic application used as a plug-in in Wireshark to be able to sniff (using a Nordic board) *BLE* packets over the air

Receiver

The purpose of the receiver board is to collect the incoming stream of data from the transmitter for post-processing data or to reconstruction of the ATC signal. Its functioning, as the transmitter board, can be seen as a FSM, with states shown in Figure 1.48. With a mechanical switch the board passes from **OFF** state to **ON** state, where the configuration automatically starts. Once it is completed, the board enters in **SCAN**ning state, where it "listens" for advertising packets. A *whitelist* is implemented: in this way, the receiver board can only connect to device which name is inside the list (of course, the name must be set accordingly to transmitter one). If the transmitter is in proximity (and it is advertising), a connection (with parameters set in Table 1.5) occurs. The receiver board will decide when activate the notification for the custom characteristic, and it will go in **RX** state, where it will receive ATC packet.

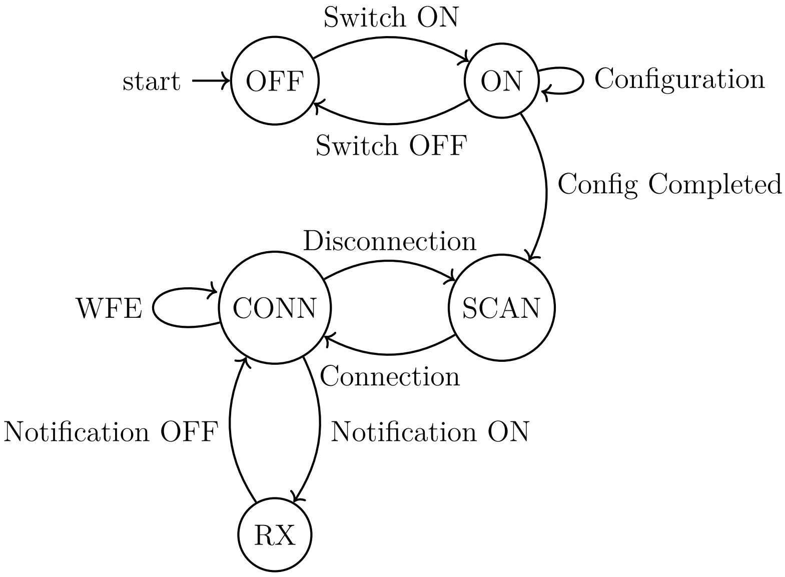


Figure 1.48: Receiver FSM. The WFE in CONN state, means that the board is "ready" to keep some events, previously assigned during configuration

Configuration

The configuration process is shown in Figure 1.49.

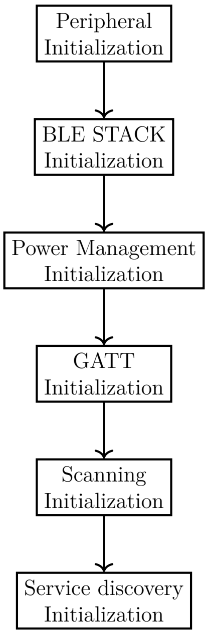

Peripheral Initialization
BLE STACK Initialization
Power Management Initialization
GATT Initialization
Scanning Initialization
Service discovery Initialization

Figure 1.49: Configuration process in ON state

Inside *Peripheral Initialization*, the peripheral used are initialized with their parameters:

- TIMER1: it is a 8-bit, 31250Hz timer used for reconstruction of ATC signal
- TIMER2: it is used for debugging purpose
- Real Timer Counter (RTC): it is used for reconstruction of ATC signal
- LEDs: indication of different states
- BUTTONs: used for activating/deactivating notifications on transmitter board
- GPIO PINs: 1 PIN is used as output for ATC reconstruction, while other pins are used as debugging
- UART: used for debugging purposes. The baud-rate is set to 1000000 without Hardware Flow Control.

During *BLE stack initialization*, the needed resources for *BLE* functioning are allocated.
Inside *Power Management Initialization*, events handlers are registered.
In *GATT Initialization*, the module used for handling GATT operations is initialized.
Scanning Initialization is responsible for configuring the scanning module following Table 1.6.

Parameter	Value
Scan Interval	100ms
Scan Window	50ms
Active Scan	NO
Physical Link	1Mbit/s
Scan Duration	∞

Table 1.6: Scanning parameter

Inside *Service Discovery Initialization*, the service discovery protocol, well explained in [1, Vol.3, Part B], is initialized. This module provides a method for the central board to discover different services stored inside the peers. Here, when the custom service is found, the service handle and the characteristic handle is stored, in order to perform different operations later.

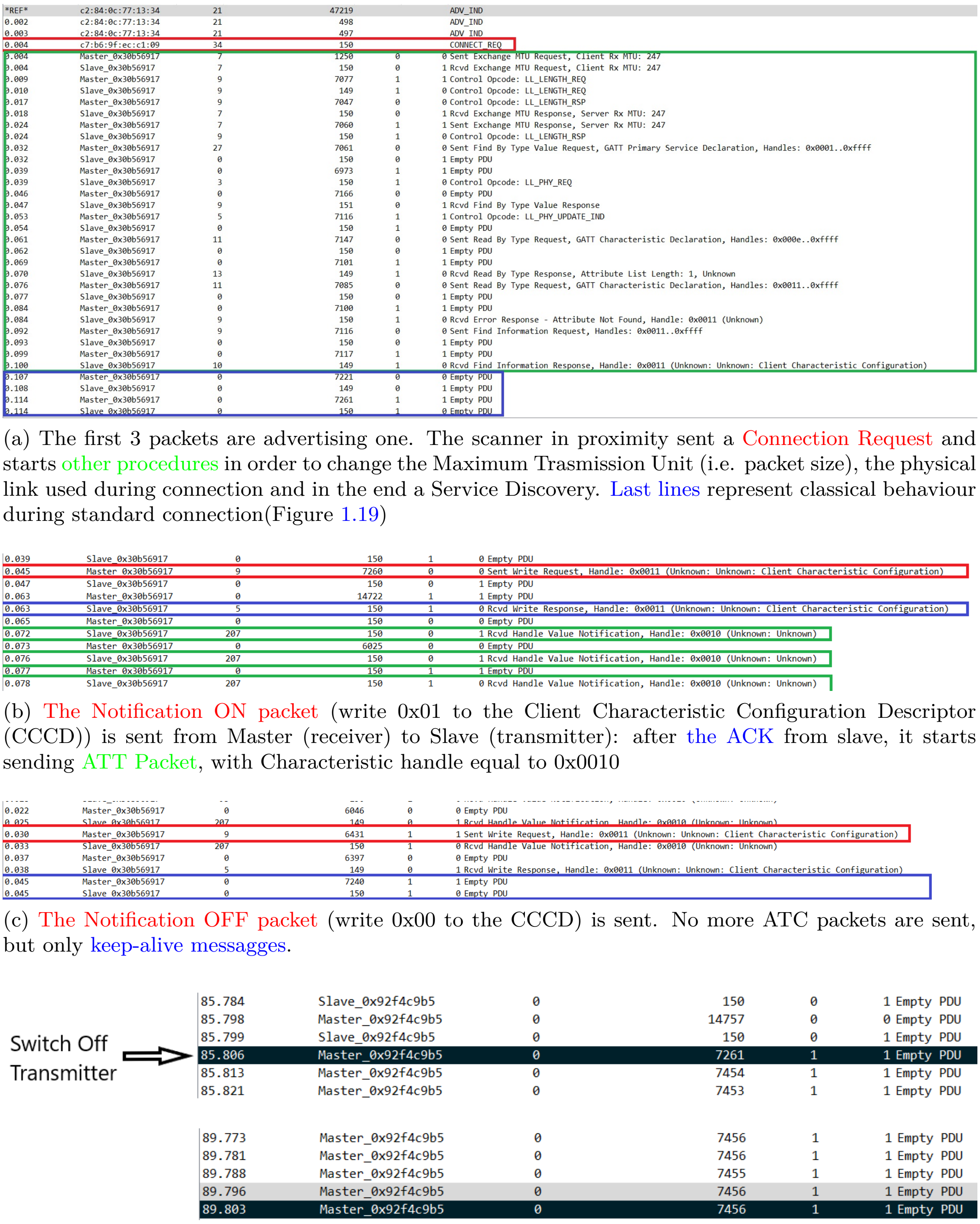
Communication workflow

Inside Table 1.7, different roles associated with transmitter and receiver are shown in order to better understand the communication workflow between the boards. It's important to notice that the transmitter board works as *GATT Server*, because it stores ATC packets, while the receiver board as *GATT Client*, because it "asks" the server for the needed data.

BOARD	GAP Role	GATT Role	LL Role(Initiator)	LL Role(Connection)
Transmitter	Peripheral	Server	Advertiser	Slave
Receiver	Central	Client	Scanner	Master

Table 1.7: *Bluetooth*$^{\copyright}$ *Low Energy* roles associated with the 2 boards.

The communication workflow, outlined in Figure 1.51, is a mix of *asynchronous* and *synchronous* paradigms: the *Notification ON* packet, in fact, must be ACKed by the transmitter board, before starting to send, every connection interval, a *GATT Notification*, associated to the custom characteristic: the notification, as already explained, it is not ACKed from the receiver. The real behavior of advertising, scanning, connection and notification, is shown and explained in Figure 1.50, where each type of packet is explained.



```
*REF*       c2:84:0c:77:13:34      21         47219          ADV_IND
0.002       c2:84:0c:77:13:34      21           498          ADV_IND
0.003       c2:84:0c:77:13:34      21           497          ADV_IND
0.004       c7:b6:9f:ec:c1:09      34           150          CONNECT_REQ
0.004       Master_0x30b56917       7          1250    0    0 Sent Exchange MTU Request, Client Rx MTU: 247
0.004       Slave_0x30b56917        7           150    0    1 Rcvd Exchange MTU Request, Client Rx MTU: 247
0.009       Master_0x30b56917       9          7077    1    1 Control Opcode: LL_LENGTH_REQ
0.010       Slave_0x30b56917        9           149    1    0 Control Opcode: LL_LENGTH_REQ
0.017       Master_0x30b56917       9          7047    0    0 Control Opcode: LL_LENGTH_RSP
0.018       Slave_0x30b56917        7           150    0    1 Rcvd Exchange MTU Response, Server Rx MTU: 247
0.024       Master_0x30b56917       7          7060    1    1 Sent Exchange MTU Response, Server Rx MTU: 247
0.024       Slave_0x30b56917        9           150    1    0 Control Opcode: LL_LENGTH_RSP
0.032       Master_0x30b56917      27          7061    0    0 Sent Find By Type Value Request, GATT Primary Service Declaration, Handles: 0x0001..0xffff
0.032       Slave_0x30b56917        0           150    0    1 Empty PDU
0.039       Master_0x30b56917       0          6973    1    1 Empty PDU
0.039       Slave_0x30b56917        3           150    1    0 Control Opcode: LL_PHY_REQ
0.046       Master_0x30b56917       0          7166    0    0 Empty PDU
0.047       Slave_0x30b56917        9           151    0    1 Rcvd Find By Type Value Response
0.053       Master_0x30b56917       5          7116    1    1 Control Opcode: LL_PHY_UPDATE_IND
0.054       Slave_0x30b56917        0           150    1    0 Empty PDU
0.061       Master_0x30b56917      11          7147    0    0 Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x000e..0xffff
0.062       Slave_0x30b56917        0           150    0    1 Empty PDU
0.069       Master_0x30b56917       0          7101    1    1 Empty PDU
0.070       Slave_0x30b56917       13           149    1    0 Rcvd Read By Type Response, Attribute List Length: 1, Unknown
0.076       Master_0x30b56917      11          7085    0    0 Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x0011..0xffff
0.077       Slave_0x30b56917        0           150    0    1 Empty PDU
0.084       Master_0x30b56917       0          7100    1    1 Empty PDU
0.084       Slave_0x30b56917        9           150    1    0 Rcvd Error Response - Attribute Not Found, Handle: 0x0011 (Unknown)
0.092       Master_0x30b56917       9          7116    0    0 Sent Find Information Request, Handles: 0x0011..0xffff
0.093       Slave_0x30b56917        0           150    0    1 Empty PDU
0.099       Master_0x30b56917       0          7117    1    1 Empty PDU
0.100       Slave_0x30b56917       10           149    1    0 Rcvd Find Information Response, Handle: 0x0011 (Unknown: Unknown: Client Characteristic Configuration)
0.107       Master_0x30b56917       0          7221    0    0 Empty PDU
0.108       Slave_0x30b56917        0           149    0    1 Empty PDU
0.114       Master_0x30b56917       0          7261    1    1 Empty PDU
0.114       Slave_0x30b56917        0           150    1    0 Empty PDU
```

(a) The first 3 packets are advertising one. The scanner in proximity sent a Connection Request and starts other procedures in order to change the Maximum Trasmission Unit (i.e. packet size), the physical link used during connection and in the end a Service Discovery. Last lines represent classical behaviour during standard connection(Figure 1.19)

```
0.039       Slave_0x30b56917        0           150    1    0 Empty PDU
0.045       Master_0x30b56917       9          7260    0    0 Sent Write Request, Handle: 0x0011 (Unknown: Unknown: Client Characteristic Configuration)
0.047       Slave_0x30b56917        0           150    0    1 Empty PDU
0.063       Master_0x30b56917       0         14722    1    1 Empty PDU
0.063       Slave_0x30b56917        5           150    1    0 Rcvd Write Response, Handle: 0x0011 (Unknown: Unknown: Client Characteristic Configuration)
0.065       Master_0x30b56917       0           150    0    0 Empty PDU
0.072       Slave_0x30b56917      207           150    0    1 Rcvd Handle Value Notification, Handle: 0x0010 (Unknown: Unknown)
0.073       Master_0x30b56917       0          6025    0    0 Empty PDU
0.076       Slave_0x30b56917      207           150    0    1 Rcvd Handle Value Notification, Handle: 0x0010 (Unknown: Unknown)
0.077       Master_0x30b56917       0           150    1    1 Empty PDU
0.078       Slave_0x30b56917      207           150    1    0 Rcvd Handle Value Notification, Handle: 0x0010 (Unknown: Unknown)
```

(b) The Notification ON packet (write 0x01 to the Client Characteristic Configuration Descriptor (CCCD)) is sent from Master (receiver) to Slave (transmitter): after the ACK from slave, it starts sending ATT Packet, with Characteristic handle equal to 0x0010

```
0.022       Master_0x30b56917       0          6046    0    0 Empty PDU
0.025       Slave_0x30b56917      207           149    0    1 Rcvd Handle Value Notification, Handle: 0x0010 (Unknown: Unknown)
0.030       Master_0x30b56917       9          6431    1    1 Sent Write Request, Handle: 0x0011 (Unknown: Unknown: Client Characteristic Configuration)
0.033       Slave_0x30b56917      207           150    1    0 Rcvd Handle Value Notification, Handle: 0x0010 (Unknown: Unknown)
0.037       Master_0x30b56917       0          6397    0    0 Empty PDU
0.038       Slave_0x30b56917        5           149    0    1 Rcvd Write Response, Handle: 0x0011 (Unknown: Unknown: Client Characteristic Configuration)
0.045       Master_0x30b56917       0          7240    1    1 Empty PDU
0.045       Slave_0x30b56917        0           150    1    0 Empty PDU
```

(c) The Notification OFF packet (write 0x00 to the CCCD) is sent. No more ATC packets are sent, but only keep-alive messagges.

```
85.784      Slave_0x92f4c9b5        0           150      0      1 Empty PDU
85.798      Master_0x92f4c9b5       0         14757      0      0 Empty PDU
85.799      Slave_0x92f4c9b5        0           150      0      1 Empty PDU
85.806      Master_0x92f4c9b5       0          7261      1      1 Empty PDU
85.813      Master_0x92f4c9b5       0          7454      1      1 Empty PDU
85.821      Master_0x92f4c9b5       0          7453      1      1 Empty PDU

89.773      Master_0x92f4c9b5       0          7456      1      1 Empty PDU
89.781      Master_0x92f4c9b5       0          7456      1      1 Empty PDU
89.788      Master_0x92f4c9b5       0          7455      1      1 Empty PDU
89.796      Master_0x92f4c9b5       0          7456      1      1 Empty PDU
89.803      Master_0x92f4c9b5       0          7456      1      1 Empty PDU
```

(d) The transmitter is switched OFF: after Connection Supervision Timeout (expressed by first column in second), the connection is definitely lost and master stops sending keep-alive messagges.

Figure 1.50: In *(a)*, advertising and connection is explained; in *(b)* and *(c)*, start and stop of ATC data is presented; in *(d)*, the connection is lost. These figures have been taken from *WireShark* software

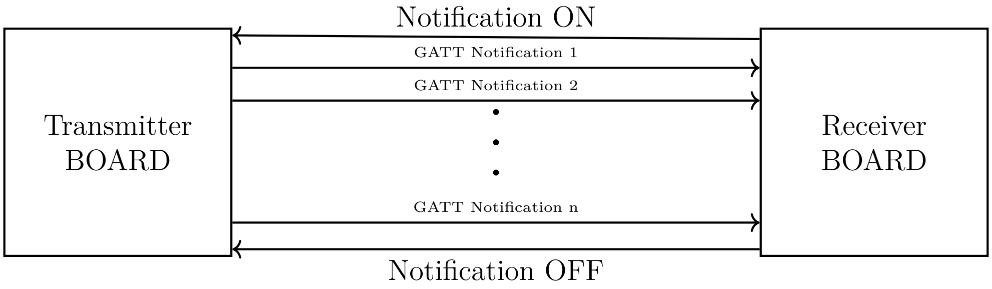


Figure 1.51: Communication workflow. Each arrow represents a packet exchanged between transmitter and receiver: after *Notification ON* packet is sent and ACKed by transmitter, it will start to send *ATC packets*, until *Notification OFF* packet is sent.

Each *GATT Notification* packet, show in Figure 1.52 is formed by (excluding mandatory header and footer):

- **Counter**: represent an increasing 8-bit counter for a simple correction error and debugging/testing purpose. It can be seen as an identifier for different connection events.
- **Ch ID**: represent an Id for different channels. It has been added for future development of a multi-channel system
- **#EV**: represent the total number of ATC events in that connection interval/window.
- **ATC_Data**: it is a variable-length array, up to MAX_Ch_Data bytes, containing the events' timestamp during acquisition window (in *BLE* case during connection interval)

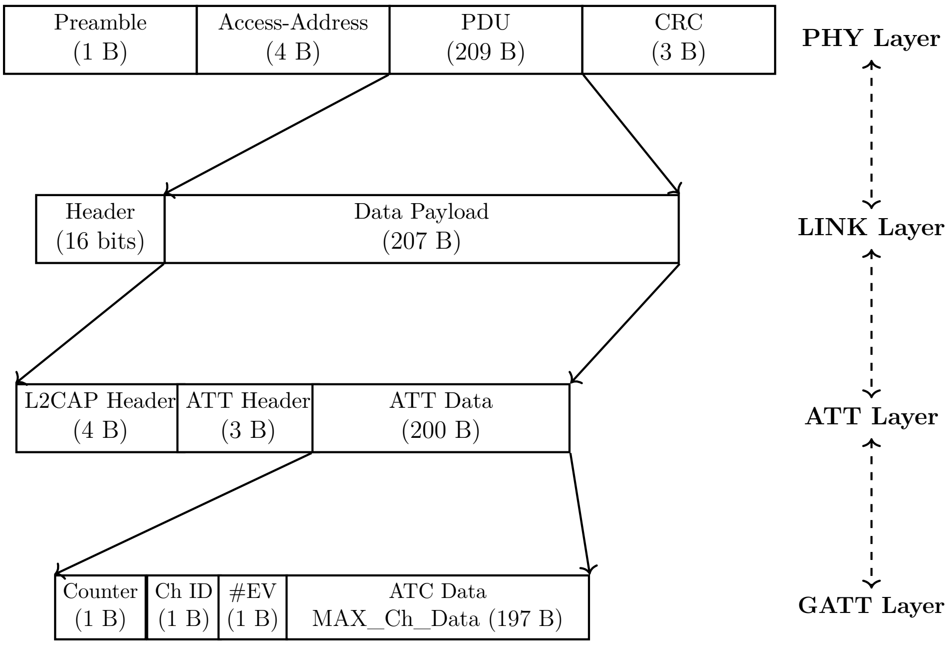


Figure 1.52: Real on air packet. The dashed arrows represent the logical connection between different layers. In transmission mode, the path is from higher to lower layers while in reception mode is the opposite. The figure refers to a 1-channel acquisition system, although can be scalable to multi-channels ones.

IEEE 802.15.4

The *IEEE 802.15.4* is the second protocol implemented and tested for ATC transmission: a custom implementation has been made, although a *Nordic*© one is available (but the performance was very low, and it was discarded). In the following, both transmitter and receiver sides are explained.

Transmitter

The transmitter board, as already explained, is in charge of transmitting ATC events at every connection interval (inside the standard, there is no concept of connection interval, but in order to make a comparison with *BLE* standard, the same idea is taken). Its functioning can also be seen as a FSM (Figure 1.53), simpler than *BLE* one. The board starts in **OFF** state, passing in **ON** state with a mechanical switch: here, a configuration process begins, setting the proper parameters and initializing the needed peripherals. Once the configuration is completed, the board enters in **WORK**ing state: here, when the user wants to start sending ATC data (pushing the proper) button, it will go in **TX** state transmitting data each connection interval. The overall working of this board is much simpler than *BLE* one, because neither advertising nor connection must happen. It has been decided to keep low the total complexity, so a ***non-beacon-enabled network*** (the transmitter can send data whenever it wants) is implemented.

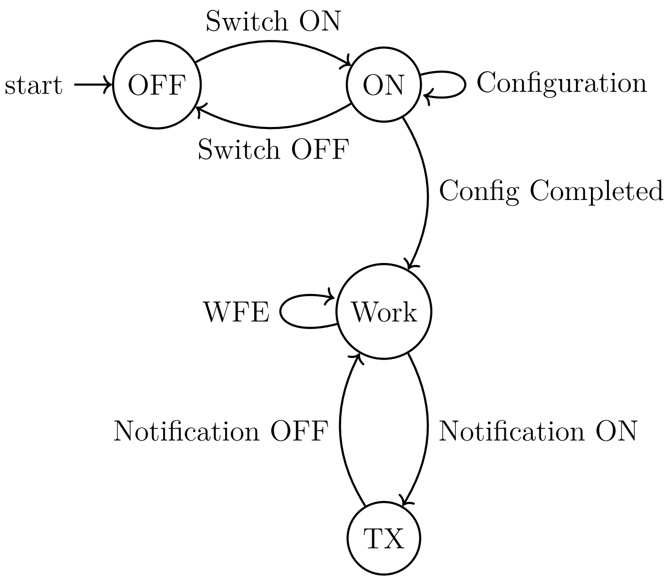


Figure 1.53: Transmitter FSM.The WFE in WORK state, means that the board is "ready" to keep some events, previously assigned during configuration.

Configuration

The configuration process is a crucial aspect to have reliable communication between nodes: in the following, each block of Figure 1.54 is explained.

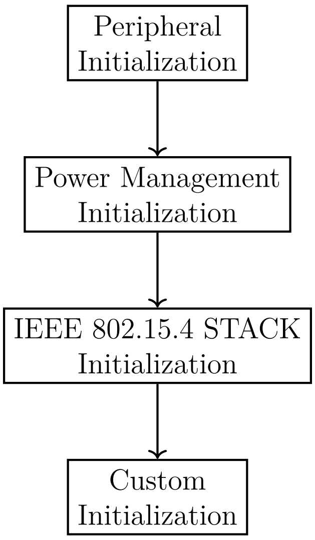

Peripheral Initialization
Power Management Initialization
IEEE 802.15.4 STACK Initialization
Custom Initialization

Figure 1.54: Configuration process. At the end, the board is ready to transmit data

In *Peripheral Initialization*, the needed hardware is configured

- Timer1: Used for general timestamp(for debugging purpose and for synchronization of transmission)

- Timer2: Used for ATC transmission: it works at 31250Hz with 8-bit
- UART: used for PC communication and debugging. The baud-rate is set to 1000000 without hardware flow control (HWFC)
- LEDs: indication of current FSM states
- GPIO: used for ATC signal and debugging purpose
- BUTTONs: used for activation/deactivation of Notification

Inside *Power Management Initialization*, some power aspects are handled (in order to optimize power consumption) as well as registering handlers for different events.
During *IEEE 802.15.4 STACK Initialization*, the modules, and hardware needed for the correct working of the entire standard (for example a dedicated timer for handling ACK automatically) are configured.
Inside *Custom Initialization*, the IEEE 802.15.4 *PAN-Id, Source Address, and TX Power*, as well as the *packet* to be exchanged are configured; their value is shown in Table 1.8 and Figure 1.55.

Parameter	Value
PAN_Id	0xBABA
Short Address	0xDEDA
Extended Address	0xBABA10203040DEDA
TX Power	0dBm
Frequency Channel	15
Connection Interval	7 ms

Table 1.8: Parameter used during communication. The TX Power can be set from −40 to 8 dBm: the tuning has been done in testing mode, to evaluate the influence on the overall power consumption. The frequency channel can be converted into physical frequency using formula inside Figure 1.23

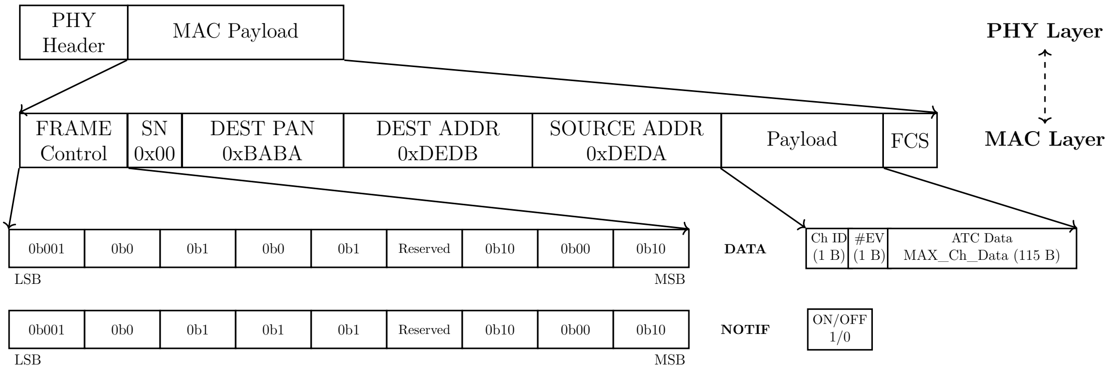


Figure 1.55: On-air packet format

The payload is similar to *ATT Data of BLE* (except for the *Counter Field*, here not implemented for the presence of the Sequence Number (SN) field, which works in the same way), in order to update the entire system for more than only 1 channel. The *Frame Control* field(Figure 1.26), is set in order to maximize the *Payload* size (not including the source PAN-Id and using short addresses) without any kind of security; moreover, the ACK is disabled writing in the proper field the correct value(more information in Section 1). In order to start a communication, before ATC packets are started being sent, a Notification ON packet (that must be ACKed before the real starting) is transmitted to the receiver board (in the same way when the user wants to stop transmitting data, a Notification OFF packet is sent): in this way, studying a better protocol, the power consumption of receiver board could also be optimized.

Receiver

The receiver board, as already stated in *BLE* section, has the task to collect the received packet containing ATC timestamps inside one connection interval, to post-process data, or to reconstruct the entire signal. As expected, it can be described by a FSM, showed in Figure 1.56. The initial state is the **OFF** one: it passes in **ON** state with a mechanical switch; here, a configuration process begins. Once finished, the board is ready to receive data (**WORK**ing state, in which the radio is in reception mode): it will enter in **RX** state, when a Notification ON packet (from the transmitter) is received, while it will return in **WORK** state at the reception of Notification OFF packet. As explained before, to keep the implementation simple, a ***non-beacon enabled network*** has been developed: the two boards will manage to communicate as long as the transmitted power is sufficient to reach the receiver side, without worry about higher finer control ([18, Section 7.1]). The fact that a Notification ON/OFF packet is exchanged for starting/stopping a stream of data, could optimize the receiver's power consumption, for example, keeping on the radio only in a certain amount of time.

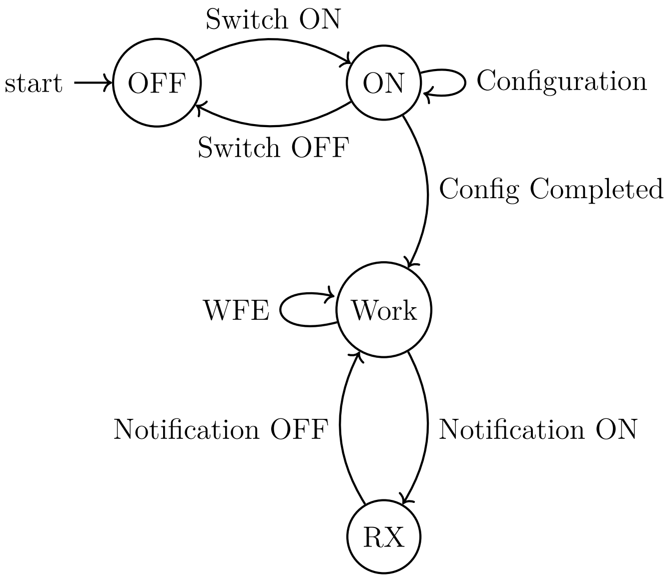


Figure 1.56: Transmitter FSM.The WFE in WORK state, means that the board is "ready" to keep some events, previously assigned during configuration.

Configuration

The configuration process is always used to prepare the board to work compliant to the *IEEE 802.15.4* standard and to initialize the needed peripherals. It is shown in Figure 1.57

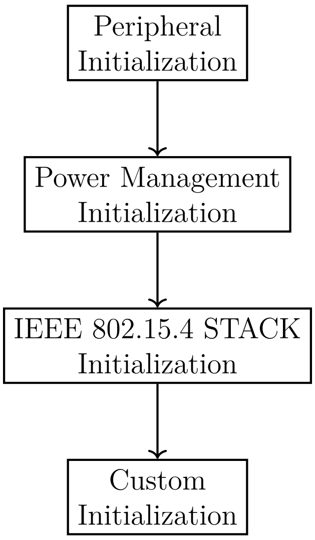

Peripheral Initialization
Power Management Initialization
IEEE 802.15.4 STACK Initialization
Custom Initialization

Figure 1.57: Configuration process. At the end, the board is ready to transmit data

- Timer1: Used for general timestamp(for debugging purpose and for synchronization of transmission)

- Timer2: Used for ATC reconstruction: it works at 31250Hz with 8-bit
- UART: used for PC communication and debugging. The baud-rate is set to 1000000 without hardware flow control (HWFC)
- Real Time Counter (RTC): used for ATC reconstruction.
- LEDs: indication of current FSM states
- GPIO: used for the reconstructed ATC signal and debugging purpose

Inside *Power Management Initialization*, some power aspects are handled (in order to optimize power consumption) as well as registering handlers for different events.
During *IEEE 802.15.4 STACK Initialization*, the modules and hardware needed for the correct working of the entire standard (for example, a dedicated timer for handling ACK automatically) are configured.
Inside *Custom Initialization*, the IEEE 802.15.4 *PAN-Id and Source Address*, as well as the right *frequency channel* (of course, it must be equal to the one set inside transmitter board.)

Parameter	Value
PAN_Id	0xBABA
Short Address	0xDEDB
Extended Address	0xBABA10203040DEDB
TX Power	0 dBm
Frequency Channel	15

Table 1.9: Parameter used during communication. The TX Power can be set from −40 to 8 dBm: however, for the receiver board, the choice of this parameter is not so important, because it transmits only the ACK for the Notification ON/OFF packet

Communication Workflow

The communication workflow, as in *BLE* standard, is a mix between asynchronous and synchronous: the former is used to transmit data, while the latter is employed in the exchanging of Notification ON and OFF packets (Figure 1.59).
As already stated, the network implemented is a ***non-beacon enabled***: in this case, no association, PAN discovery, or other finer controls are implemented, keeping the total complexity pretty low. Transmitter and receiver are placed in the same Personal Area Network (PAN): a short and extended address has been assigned (and of course, it is different) to each board (Figure 1.58). The communication will be possible as long as the transmitted power is sufficient to reach the receiver side. What is important to notice is that the two boards could be placed in two different PAN, but the overall payload would be at least 2B bigger, for the need to put in the header the destination PAN. Another important

thing to point out, with respect to *BLE* implementation, is that here the transmitter board decides when to start the communication: it is no more a client-server paradigm but nearer to publish/subscribe mechanism.

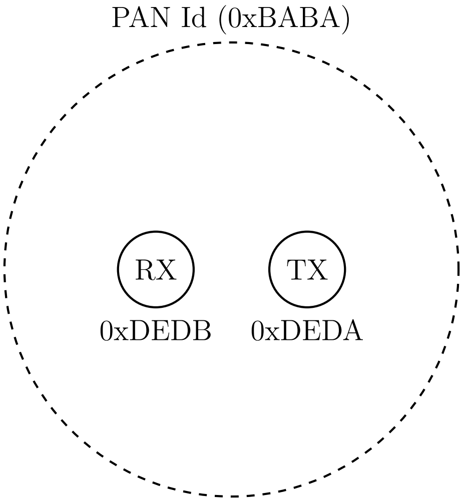


Figure 1.58: Network created using *IEEE 802.15.4* standard

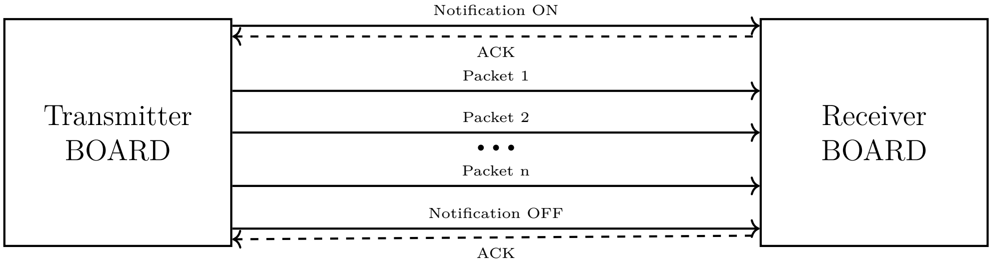
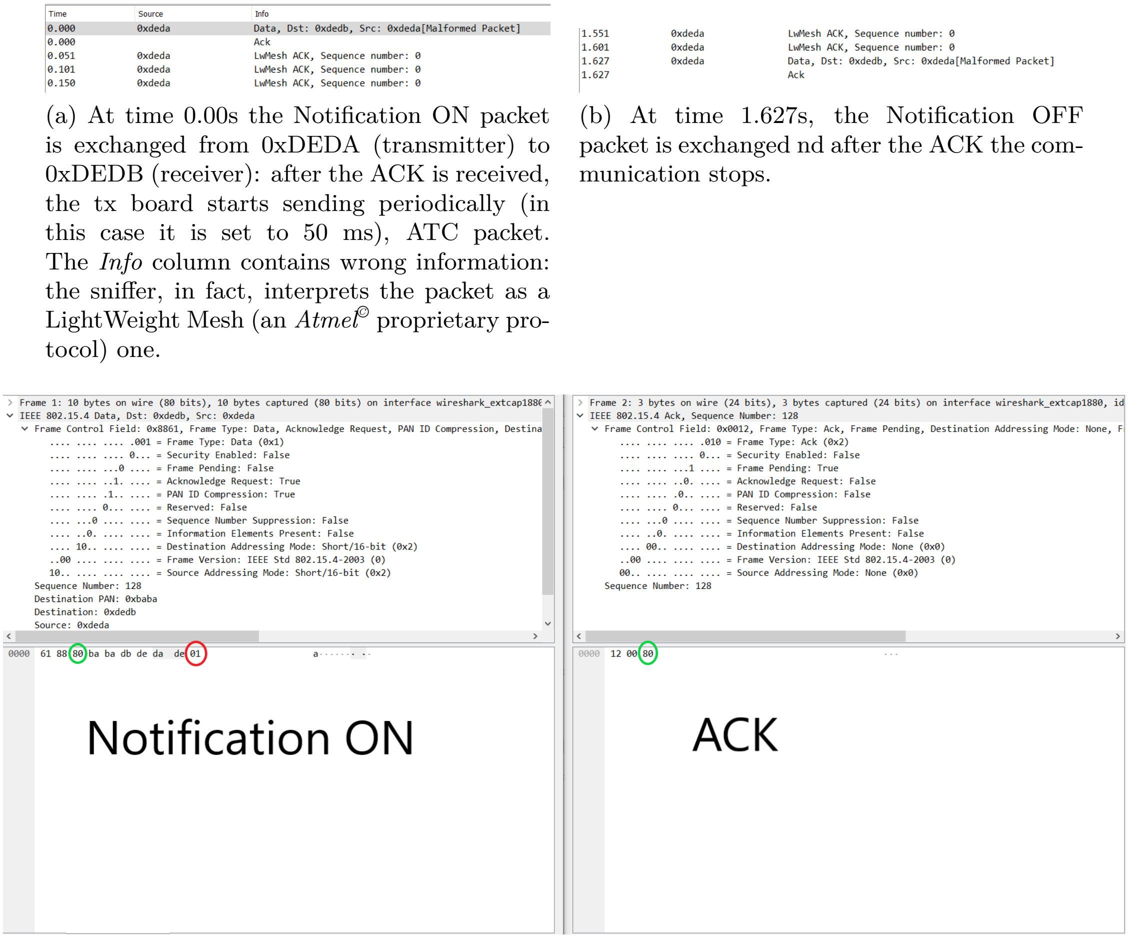


Figure 1.59: Communication workflow. Each arrow represents a packet exchanged between transmitter and receiver: after *Notification ON* packet is sent and ACKed by receiver, transmitter will start to send *ATC packets*, until *Notification OFF* packet is sent and ACKed.

Inside *IEEE 802.15.4* standard, in order to avoid collision with other users, a Direct Sequence Spread Spectrum is used (with respect to *BLE* where FH-SS was used): it's really important that both receiver and transmitter are tuned on the same frequency channel, although the communication will never be possible. In order to assess if the medium is idle or not, a Clear Channel Assessment (CCA [18, Section 6.9.9]) can be performed prior to transmission of data: in few words, it is a mechanism to understand whether or not a medium is busy or not. If the CCA failed, the decision is left to higher levels. This mechanism is also used inside the CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) algorithm, a finer method to assure the dispatch of data whenever the medium is free, as described in [18, Section 5.5.4.1]: however the integration and usage of it has a great impact on the power consumption. In this implementation, the impact of CCA is also studied.



```
Time        Source      Info
0.000       0xdeda      Data, Dst: 0xdedb, Src: 0xdeda[Malformed Packet]
0.000                   Ack
0.051       0xdeda      LwMesh ACK, Sequence number: 0
0.101       0xdeda      LwMesh ACK, Sequence number: 0
0.150       0xdeda      LwMesh ACK, Sequence number: 0
```

(a) At time 0.00s the Notification ON packet is exchanged from 0xDEDA (transmitter) to 0xDEDB (receiver): after the ACK is received, the tx board starts sending periodically (in this case it is set to 50 ms), ATC packet. The *Info* column contains wrong information: the sniffer, in fact, interprets the packet as a LightWeight Mesh (an *Atmel*© proprietary protocol) one.

```
1.551       0xdeda      LwMesh ACK, Sequence number: 0
1.601       0xdeda      LwMesh ACK, Sequence number: 0
1.627       0xdeda      Data, Dst: 0xdedb, Src: 0xdeda[Malformed Packet]
1.627                   Ack
```

(b) At time 1.627s, the Notification OFF packet is exchanged nd after the ACK the communication stops.

```
> Frame 1: 10 bytes on wire (80 bits), 10 bytes captured (80 bits) on interface wireshark_extcap188(
v IEEE 802.15.4 Data, Dst: 0xdedb, Src: 0xdeda
  v Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, PAN ID Compression, Destina
      .... .... .... .001 = Frame Type: Data (0x1)
      .... .... .... 0... = Security Enabled: False
      .... .... ...0 .... = Frame Pending: False
      .... .... ..1. .... = Acknowledge Request: True
      .... .... .1.. .... = PAN ID Compression: True
      .... .... 0... .... = Reserved: False
      .... ...0 .... .... = Sequence Number Suppression: False
      .... ..0. .... .... = Information Elements Present: False
      .... 10.. .... .... = Destination Addressing Mode: Short/16-bit (0x2)
      ..00 .... .... .... = Frame Version: IEEE Std 802.15.4-2003 (0)
      10.. .... .... .... = Source Addressing Mode: Short/16-bit (0x2)
    Sequence Number: 128
    Destination PAN: 0xbaba
    Destination: 0xdedb
    Source: 0xdeda

0000  61 88 80 ba ba db de da  de 01                     a·······.·
```

```
> Frame 2: 3 bytes on wire (24 bits), 3 bytes captured (24 bits) on interface wireshark_extcap1880, id
v IEEE 802.15.4 Ack, Sequence Number: 128
  v Frame Control Field: 0x0012, Frame Type: Ack, Frame Pending, Destination Addressing Mode: None, F
      .... .... .... .010 = Frame Type: Ack (0x2)
      .... .... .... 0... = Security Enabled: False
      .... .... ...1 .... = Frame Pending: True
      .... .... ..0. .... = Acknowledge Request: False
      .... .... .0.. .... = PAN ID Compression: False
      .... .... 0... .... = Reserved: False
      .... ...0 .... .... = Sequence Number Suppression: False
      .... ..0. .... .... = Information Elements Present: False
      .... 00.. .... .... = Destination Addressing Mode: None (0x0)
      ..00 .... .... .... = Frame Version: IEEE Std 802.15.4-2003 (0)
      00.. .... .... .... = Source Addressing Mode: None (0x0)
    Sequence Number: 128

0000  12 00 80                                           ···
```

(c) The Notification ON and its ACK are shown. The red circle indicates the configurable payload (only 9B of MAC header plus 1B of PHY header, not shown in figure), while the green circle indicates the mechanism of the ACK: the Sequence Number (SN) is sent back to the transmitter in order to acknowledge the correct reception

Figure 1.60: In *(a)* and *(b)* start and stop of communication are shown; in *(c)*, the Notification ON and its ACK are shown. These figures have been taken from *WireShark* software

Proprietary Protocol

The custom protocol implemented used the *radio drivers* of *nRF52840*, found in [46, Section 6.20]: it has been implemented writing directly to radio registers, to have a total control on the behavior of boards. It is worth noting that the same radio drivers are used in the development of both *BLE* and *IEEE 802.15.4*, using, in that case, the SDK developed by *Nordic*©.

Some features, accessible directly by the hardware and configurable by the user, are available:

- Data Whitening/De-Whitening [46, Section 6.20.3]: It is used in order to avoid some problems during communication, as DC-biased and clock recovery. If the transmitter side enables the whitening, it's important that the receiver side enables the de-whitening, although the message will be garbage.
- CRC Generation [46, Section 6.20.4]: The CRC is automatically appended at the end of the "incoming" packet: the polynomial function is completely configurable by the user. It is important that CRC polynomial is equal to both sides.
- Interframe Spacing [46, Section 6.20.9]: It is defined as the time, in µs, from the end of the last bit of the previous packet received and to the start of the first bit of the subsequent packet that is transmitted.
- Access-Address Configuration [46, Section 6.20.2]: It is the most important part; it defines which access-address to use in the transmission and reception of messages. In order to exchange message, it's important that the *TXADDRESS*(on transmitter) and *RXADDRESS*(on receiver) are equal. It should be set to some alternating value, avoiding redundancy of bit (as 0xFFFF or 0b1111..), otherwise a lot of garbage is collected (and some could also pass the CRC check).

Transmitter

The functioning of the transmitter is shown in Figure 1.61: it is the same of the one found in *IEEE 802.15.4*. The real states of the antenna during transmission are shown in Figure 1.61, in the little dashed rectangle; the states, tasks, and events dealing with the radio are accessible through registers, allowing for a huge customizability. During each transmission, the antenna follows the path represented: when a transmission is finished, it returns to the lower possible power consumption state, *DISABLED*. The time needed for the antenna to be ready to transmit is up to 140µs, totally acceptable for the application. Particular caution must be made in the configuration stage, in particular the antenna one.

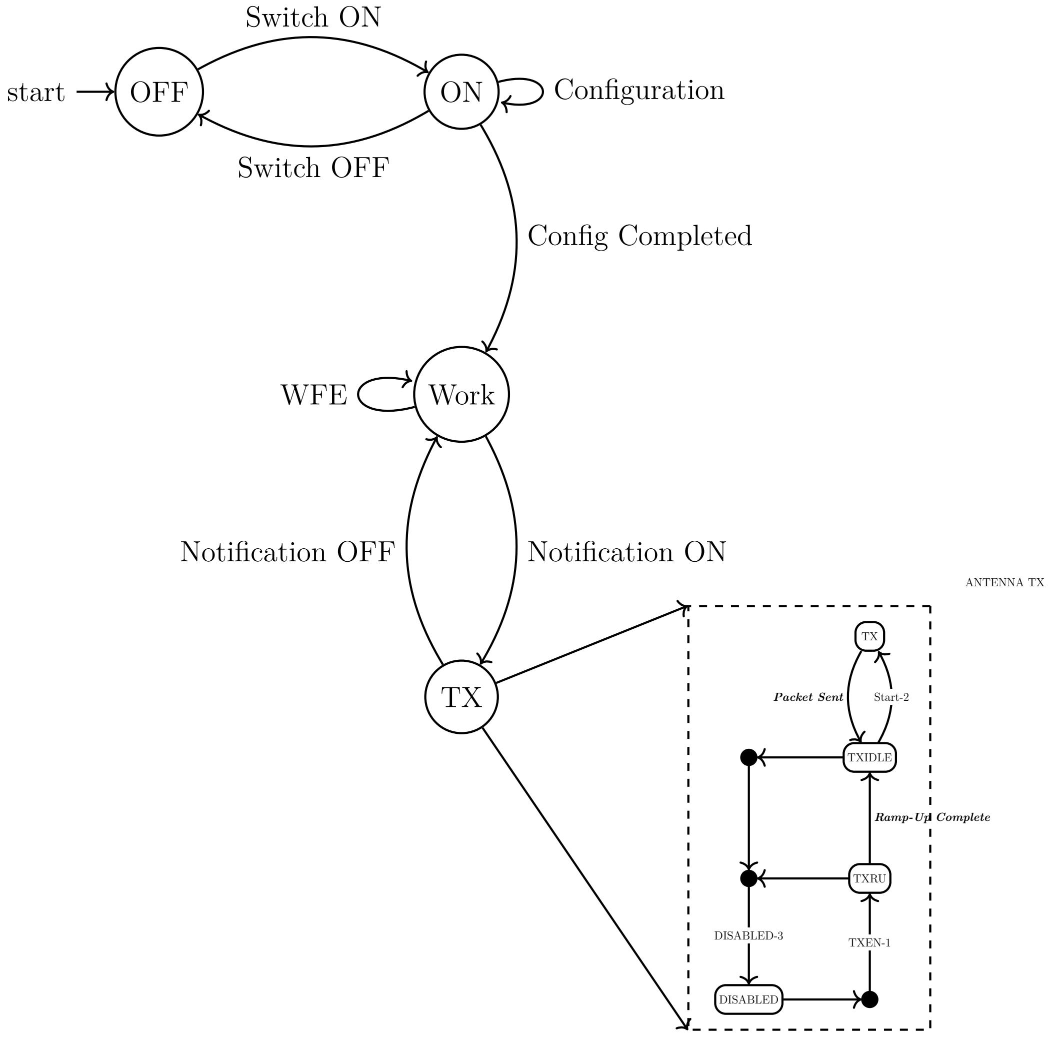


Figure 1.61: State diagram of transmitter board. Dashed rectangle represents the functioning of antenna every time a transmission is issued (every connection interval): the text in ***bold and italic*** indicates the events generated by hardware, while the text above the arrows represents tasks triggered by the user.

Configuration

The configuration phase begins with a *Peripheral Initialization*, where the needed peripherals are configured. Notice that this phase is similar for the three implemented protocols: in this way, a coherent and homogeneous comparison can be made.

- Timer1: Used for general timestamp(for debugging purpose and for synchronization of transmission)
- Timer2: Used for ATC transmission: it works at 31250Hz with 8-bit
- UART: used for PC communication and debugging. The baud-rate is set to 1000000

without hardware flow control (HWFC)

- LEDs: indication of current FSM states
- GPIO: used for ATC signal and debugging purpose
- BUTTONs: used for start and stop transmission

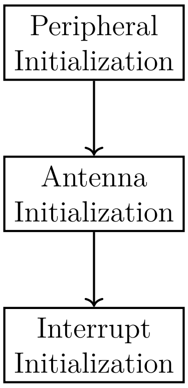


Figure 1.62: Configuration stage

During *Antenna Initialization*, the antenna, its functioning, and the overall packet format are configured. The parameters used are listed in Table 1.10. The real on-air packet format is shown in Figure 1.63: the size of the payload is the maximum allowed and can be adjusted according to the use case (for example, considering the maximum number of channels). As one can see, two different packet types are available: one is used to exchange **ATC data** (the data payload is equal to the one found in *BLE GATT Layer*, Figure 1.52), while the other represents the **Notification ON/OFF**: the two can be distinguished by the receiver by the *Frame Type* field. Inside **Notification**, *Frame Control* field is used to exchange important information between transmitter and receiver:

- $b_0 \ldots b_3$: 4 bits to indicate the current number of channels (maximum available $2^4 = 16$)
- $b_4 \ldots b_9$: 6 bits to indicate the maximum possible ATC data for each channel and acquisition window (maximum available $2^6 = 64$)
- $b_{10} \ldots b_{15}$: 6 bits to indicate the acquisition window size in ms(maximum available $2^6 = 64$ms)

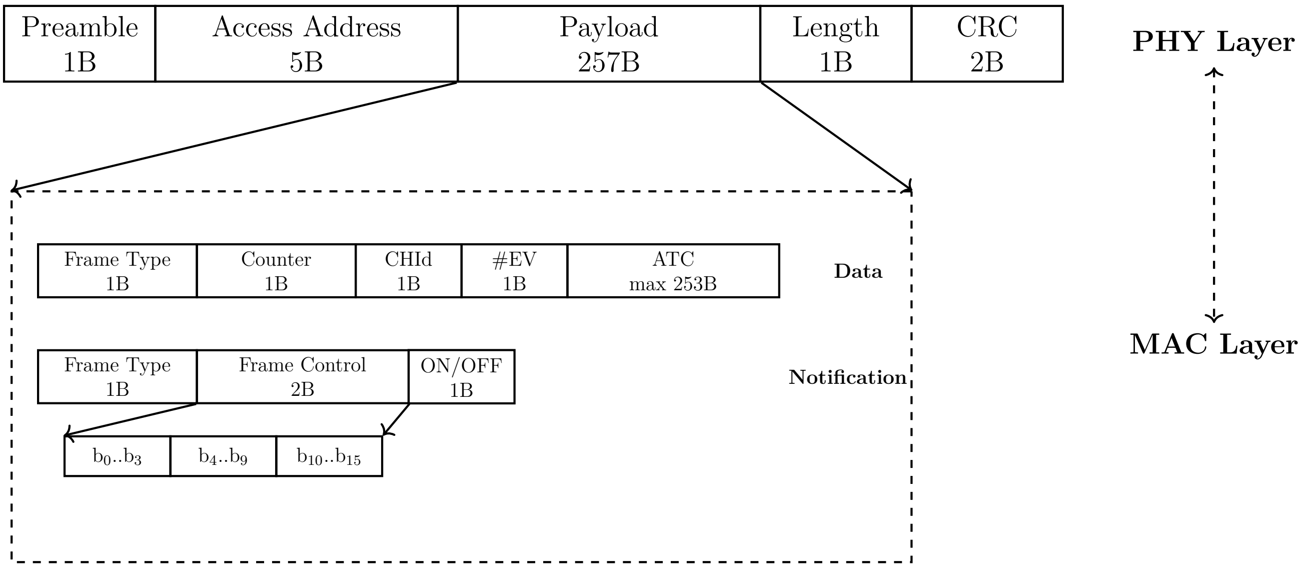


Figure 1.63: Real on-air packet in custom protocol. *Preamble* contains information about modulation used, while *Length* indicates the length of the *Payload*. The overall overhead (also *CRC*) is 9B. *Access-Address* is set to 0x89ABCDEFA4

Parameter	Value
Frequency Channel	7
Power	0 dBm
CRC Size	2B
CRC Polynomial	$x^{16} + x^{12} * x^5 + 1$
Access Address (Base)	0x89ABCDEF
Access Address (Prefix)	0xA4
Bit Rate	2 Mbit/s
Modulation	FSK
IFS	250µs
Whitening	Yes

Table 1.10: Parameters used in configuration.The TX Power can be set from -40 to 8 dBm. 1Mbit/s bit rate is also available

Inside *Interrupt Initialization* some event handlers are registered, in order to "catch" events from antenna or other peripherals.

Receiver

The Finite State Machine for the receiver board is shown in Figure 1.64. It is similar to all the other ones seen above. It is worth noting that the antenna (the main source of power consumption) always stays on, in order to listen for packets in an asynchronous way.

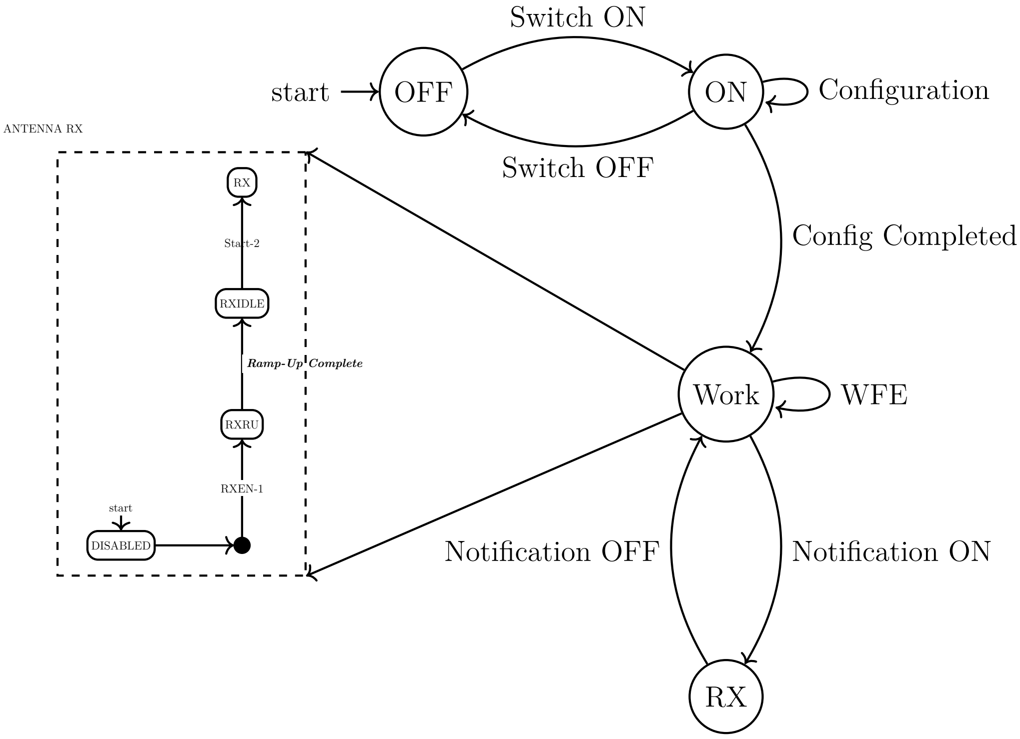


Figure 1.64: State diagram of receiver board. Inside dashed rectangle antenna states are shown: with respect to transmission board, in which the antenna always return in *DISABLED* state, here it remains always in *RX* state, to listen for **Notification ON** packet, keeping power consumption at high level. The protocol could be optimized in order to synchronize transmitter and receiver.

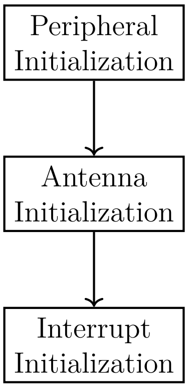


Figure 1.65: Configuration stage. It is similar to the one of transmitter board (Table 1.10): it's really important that both *frequency channel* and *Access-Address* are equal to the one configured in transmitter side

Communication Workflow

The ease of implementation of custom protocol was the main goal when developing it: the approach is totally asynchronous, with the transmitter's antenna only used in transmission mode, while receiver's one in reception mode (*simplex communication*). However, the use of a 2B long *CRC* as well as custom *Access-Address* made the communication reliable. The use of a *Notification ON/OFF* mechanism (Figure 1.66)to inform the receiver of the start and stop of ATC data has been adopted both for further development as well as comparison with other protocols. Unfortunately, in contrast to other explained protocols, it was not possible to validate the communication with Wireshark©, due to a lack of a properly sniffer: the debug terminal of the IDE used to program the development kits are shown in Figure 1.67

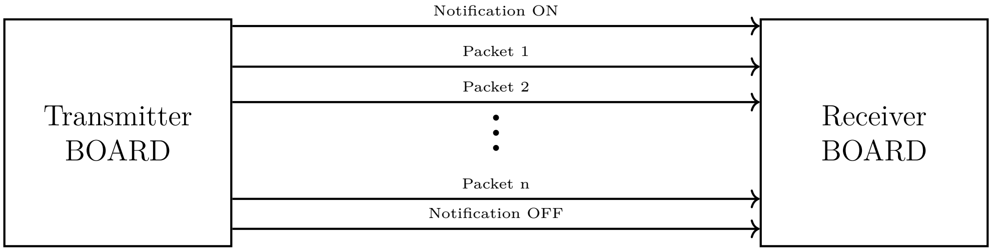
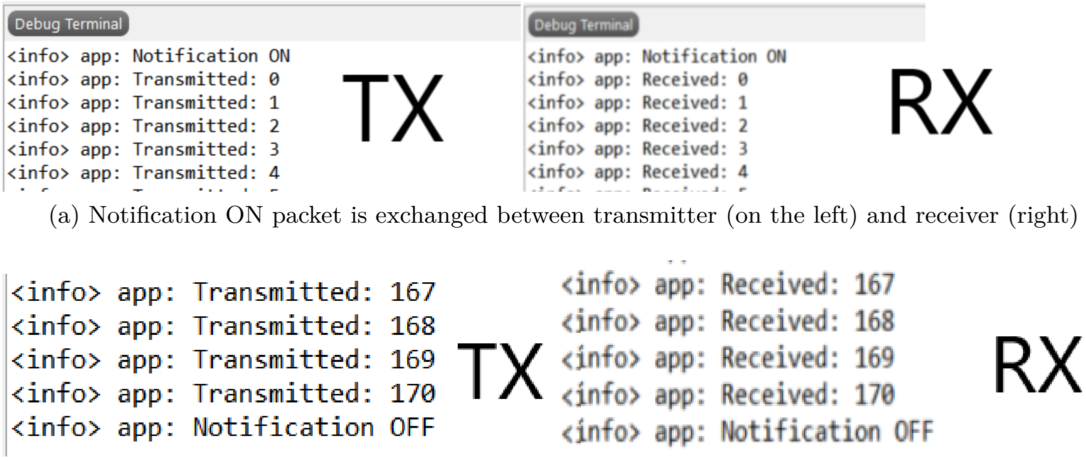


Figure 1.66: Communication workflow. Each arrow represents a packet exchanged between transmitter and receiver: after *Notification ON* packet is sent, transmitter will start to send *ATC packets*, until *Notification OFF* packet is exchanged. For the first implementation, no ACK packets have been implemented.



```
<info> app: Notification ON
<info> app: Transmitted: 0
<info> app: Transmitted: 1
<info> app: Transmitted: 2
<info> app: Transmitted: 3
<info> app: Transmitted: 4
```

```
<info> app: Notification ON
<info> app: Received: 0
<info> app: Received: 1
<info> app: Received: 2
<info> app: Received: 3
<info> app: Received: 4
```

(a) Notification ON packet is exchanged between transmitter (on the left) and receiver (right)

```
<info> app: Transmitted: 167
<info> app: Transmitted: 168
<info> app: Transmitted: 169
<info> app: Transmitted: 170
<info> app: Notification OFF
```

```
<info> app: Received: 167
<info> app: Received: 168
<info> app: Received: 169
<info> app: Received: 170
<info> app: Notification OFF
```

(b) Notification OFF packet, after a while, is exchanged between boards. ATC data flow immediately stops

Figure 1.67: **a**): Notification ON, **b**): Notification OFF

Results

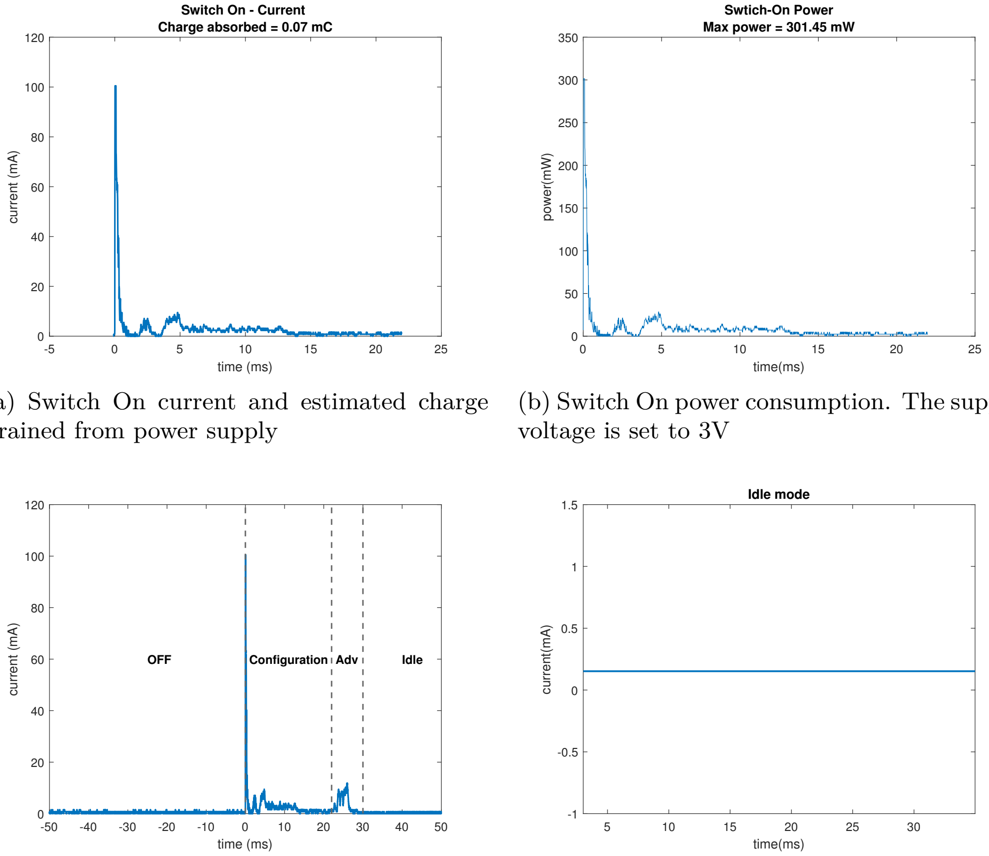
Power Consumption

Bluetooth$^{©}$ Low Energy

The *BLE* results are divided into transmitter and receiver parts, with the main focus on transmitter side: in fact, the receiver could be represented by a laptop, smartphone, or other device where battery lifetime is not a primary problem.

Transmitter

In order to completely characterize the transmitter, the current consumption is monitored from the beginning phase, where the board is switched on and the configuration initiates. The profile of both current and power consumption is shown in Figure 1.68.



(a) Switch On current and estimated charge drained from power supply

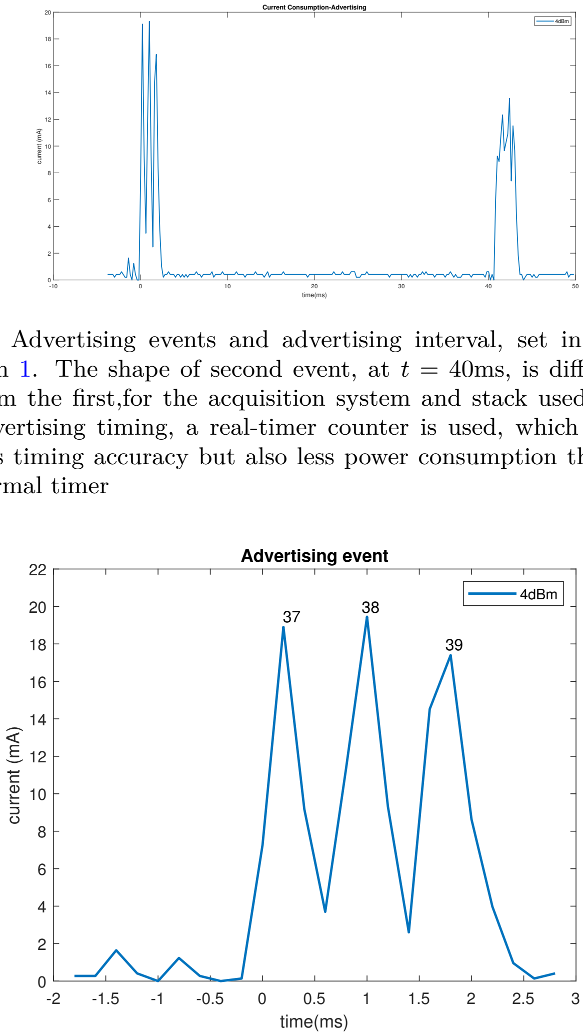
(b) Switch On power consumption. The supply voltage is set to 3V

(c) Board's life cycle

(d) Idle mode estimated after advertising event ends

Figure 1.68: Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 22ms, where advertising starts (**c**). In idle mode (**d**), the current consumption is around 0.2mA

Advertising events are shown in Figure 1.69: advertising interval and advertising on 3 dedicated channels, as explained in Figure 1.18, can be appreciated.

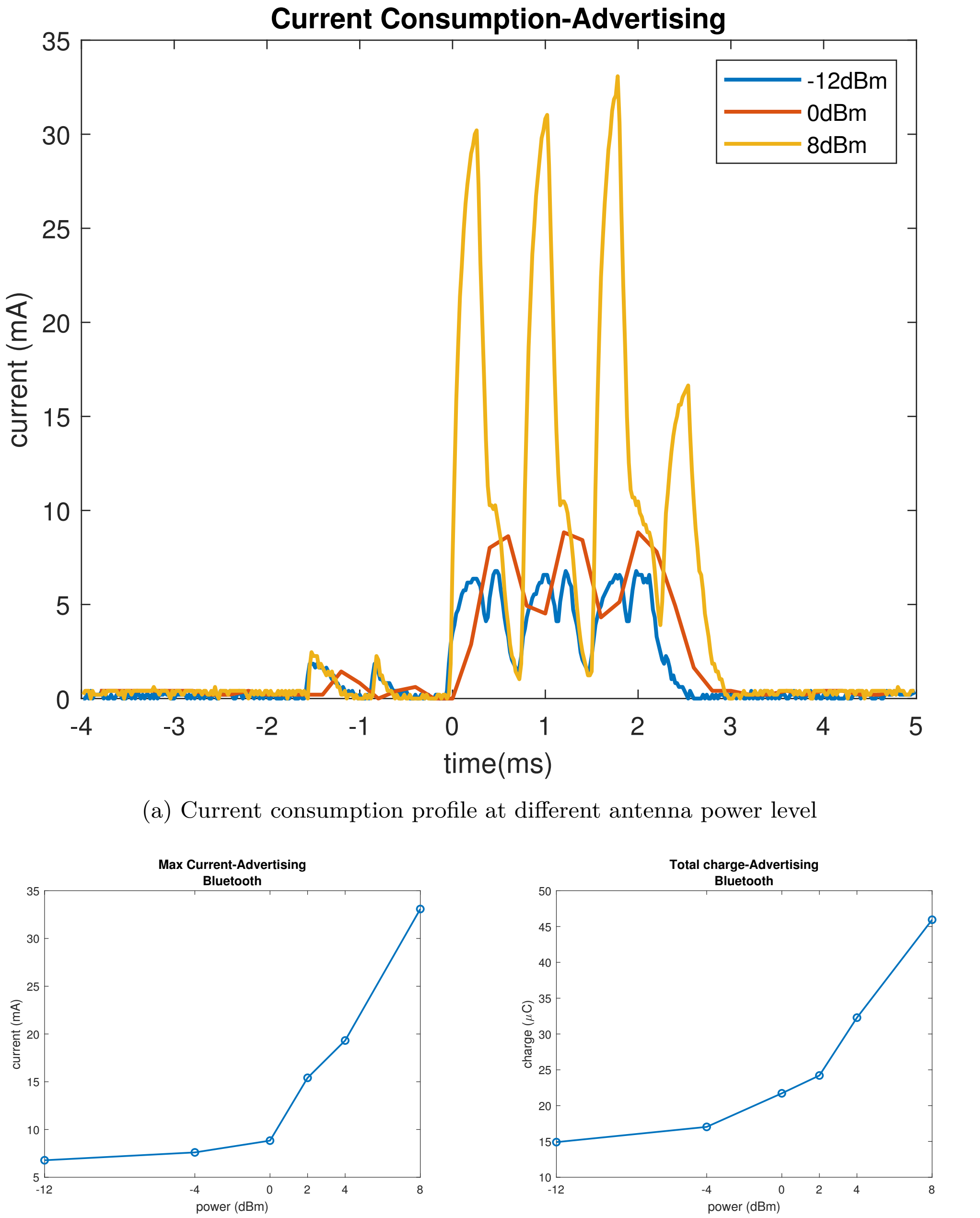


(a) Advertising events and advertising interval, set in Section 1. The shape of second event, at $t = 40$ms, is different from the first,for the acquisition system and stack used: for advertising timing, a real-timer counter is used, which have less timing accuracy but also less power consumption than a normal timer

(b) Advertising event on channel 37,38 and 39

Figure 1.69: Advertising events

In order to optimize power consumption, different antenna power levels have been tested: lesser the power, lesser the distance between the board and a hypothetical scanner in order to initiate a connection procedure. In Figure 1.70, the impact of RF output power on the overall consumption during an advertising events is shown.



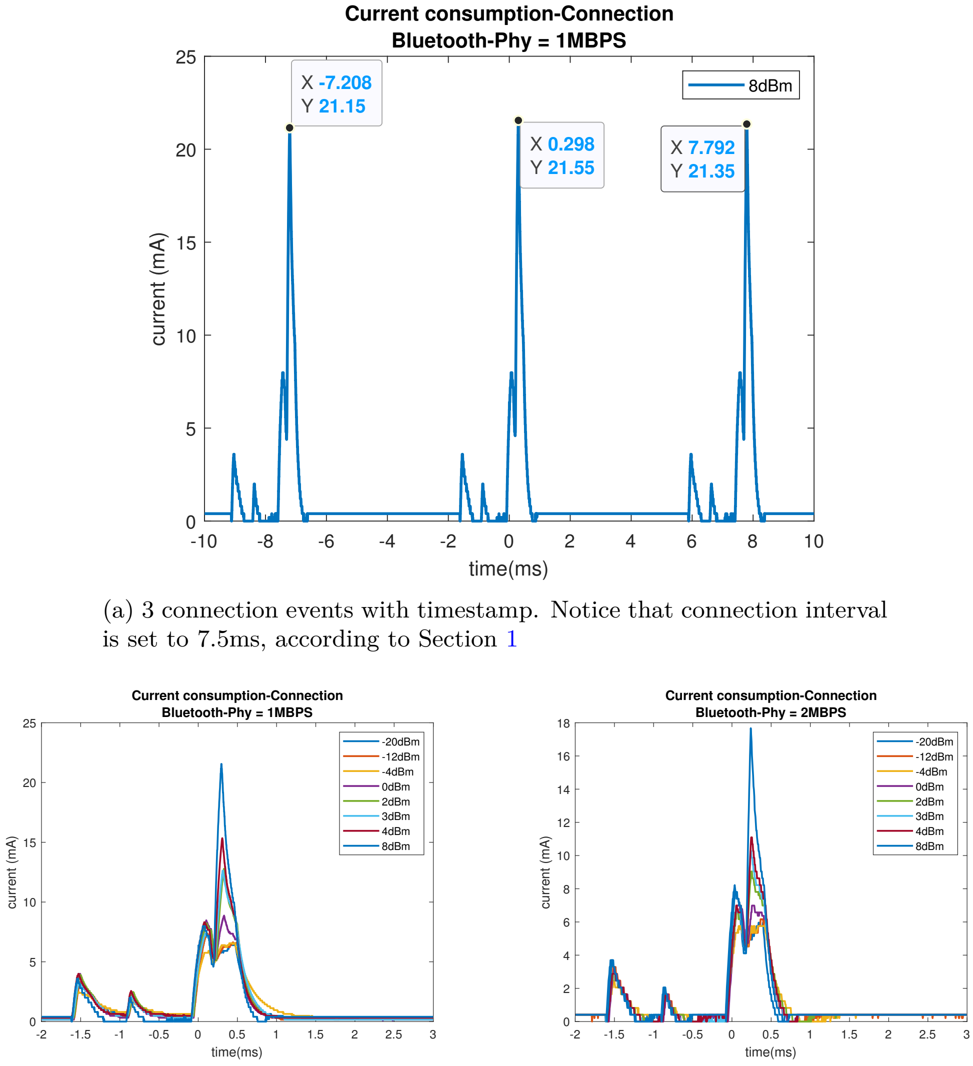
(a) Current consumption profile at different antenna power level

(b) Maximum current during advertising at different antenna power level. As expected the current increase with antenna power level.

(c) Charge drained from battery during advertising events (in a 9msinterval, as in sub-figure **a**))

Figure 1.70: Antenna power level impact on overall power consumption

When a connection occurs, every *connection interval* the antenna must be switched on both in reception (from master) and transmission (for sending packets). The implementation of *BLE* protocol supports both 1Mbit/s as well as 2Mbit/s, both tested and evaluated, in order to find the optimal one. For advertising events, only 1Mbit/s physical layer is used, so the comparison is not possible.



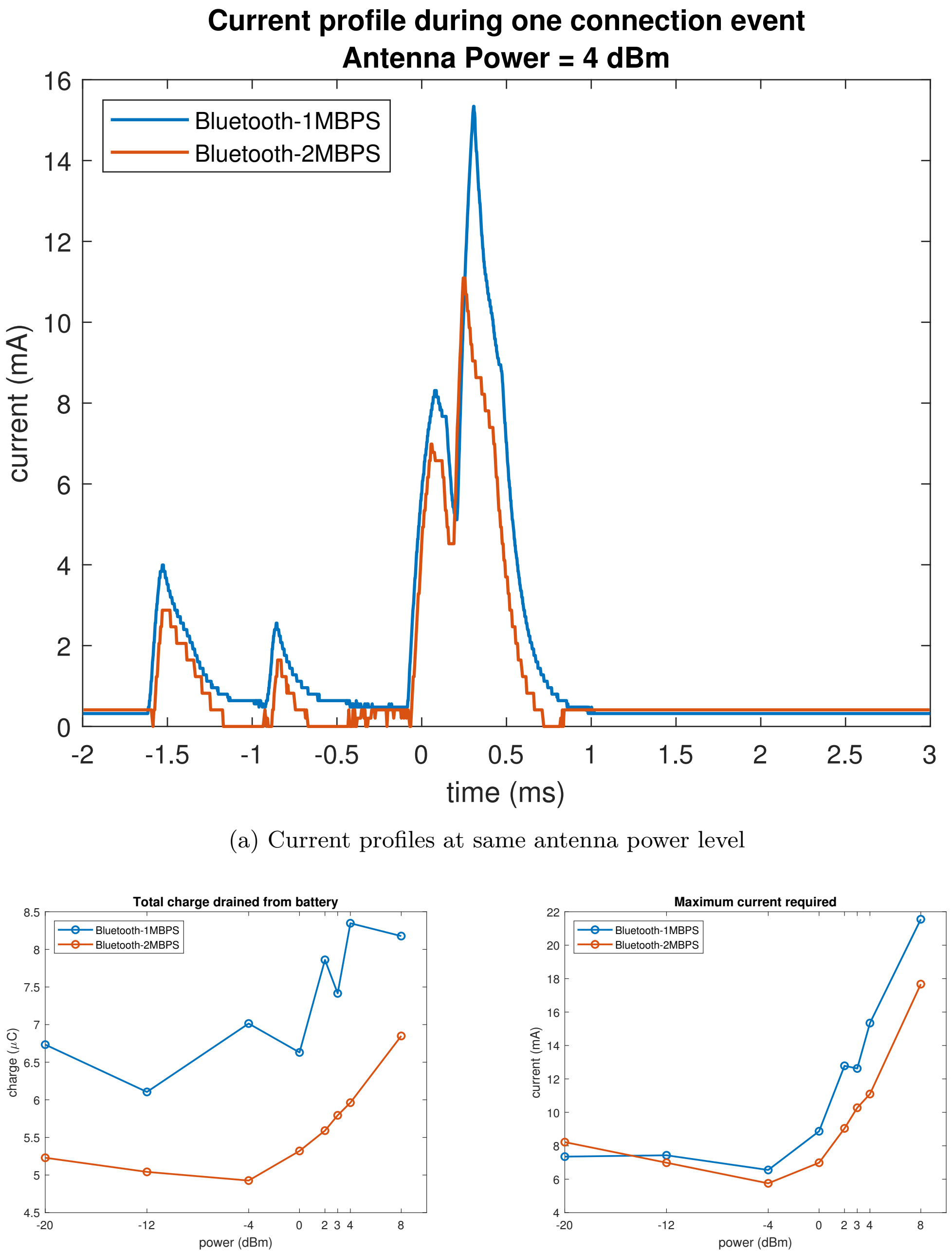
(a) 3 connection events with timestamp. Notice that connection interval is set to 7.5ms, according to Section 1

(b) 1Mbit s^{-1} PHY layer

(c) 2Mbit s^{-1} PHY layer

Figure 1.71: Antenna power level impact on overall power consumption during 1 connection events, for 1Mbit/s (**b**) and 2Mbit/s (**c**) physical layer.

In order to make comparison, it's useful to visualize the maximum current and total charge for the different physical layer, varying the antenna power level, as shown in Figure 1.72



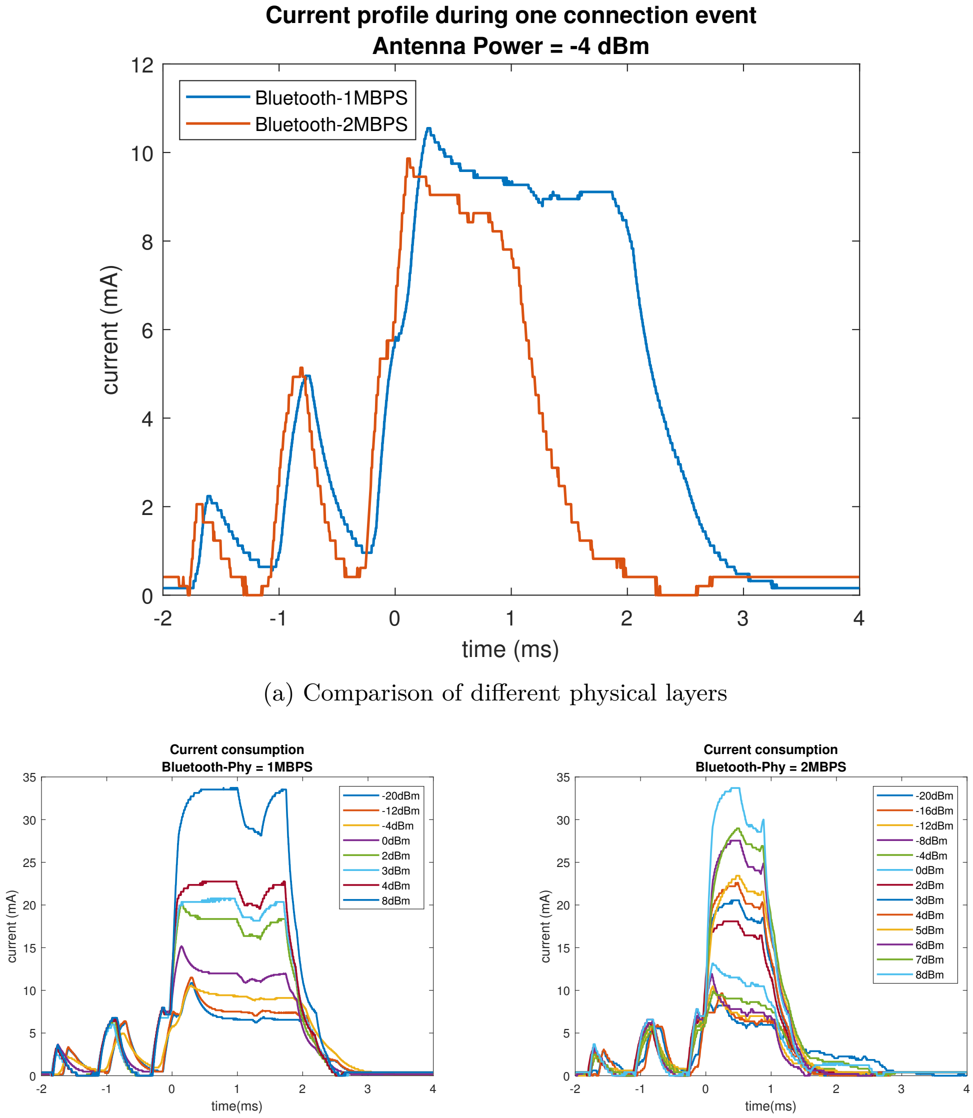
(a) Current profiles at same antenna power level

(b) The difference in charge drained between physical layers lies in the fact that with 2Mbit s^{-1} PHY, the modulation permits twice bit rate and thus half time needed to transmit the connection packet

(c) Maximum current required. For low antenna power level the maximum current is not increasing but stay costant: this is due to the ramp-up current, needed to switch on antenna, higher than transmission current.

Figure 1.72: Different connection events and antenna power level impact on overall power consumption during 1 connection events, for 1Mbit/s (**a**) and 2Mbit/s (**b**) physical layer.

When transmitter starts sending ATC data, the overall current profile is similar to connection one but here, the difference between modulation schemes is more appreciable due to increased payload size(set to 200B in Figure 1.73).



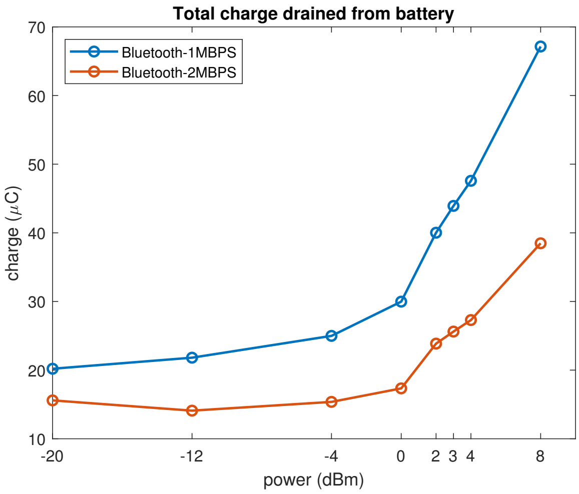
(a) Comparison of different physical layers

(b) Current profile with physical layer set to 1Mbit s^{-1}

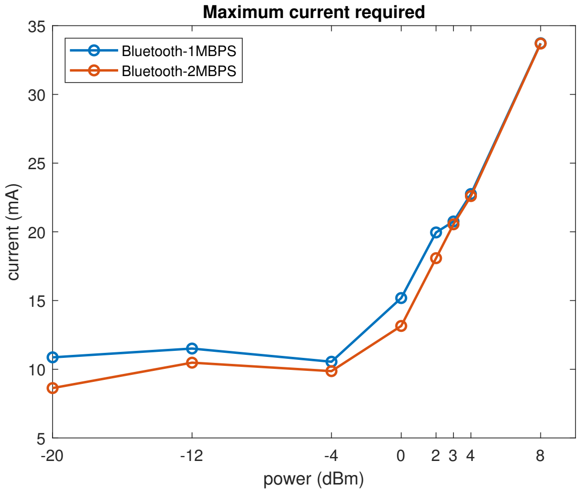
(c) Current profile with physical layer set to 2Mbit s^{-1}

Figure 1.73: Time needed to send packet is roughly the double for 1Mbit/s phy.

Maximum current and transmission time has a direct influence on charge consumed for one single connection event, as explained in Figure 1.74



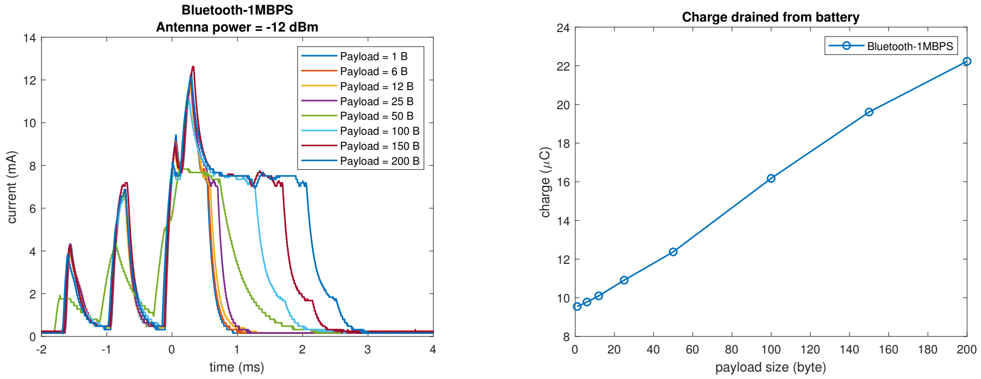
(a) Charge drained from battery, during 6ms window (Figure 1.73). The difference is in the increased transmission time for 1Mbit s^{-1}.



(b) Maximum current required

Figure 1.74: 2Mbit/s offers less power consumption

Payload size is another key parameter in determining battery life. Bigger payload brings to higher energy consumption, decreasing battery life; it must be chosen according to use case. Here, it is set 200B, in order to achieve maximum throughput, with an eye towards multi-channels acquisition system. The impact of it on 1Mbit/s physical layer is shown in Figure 1.75



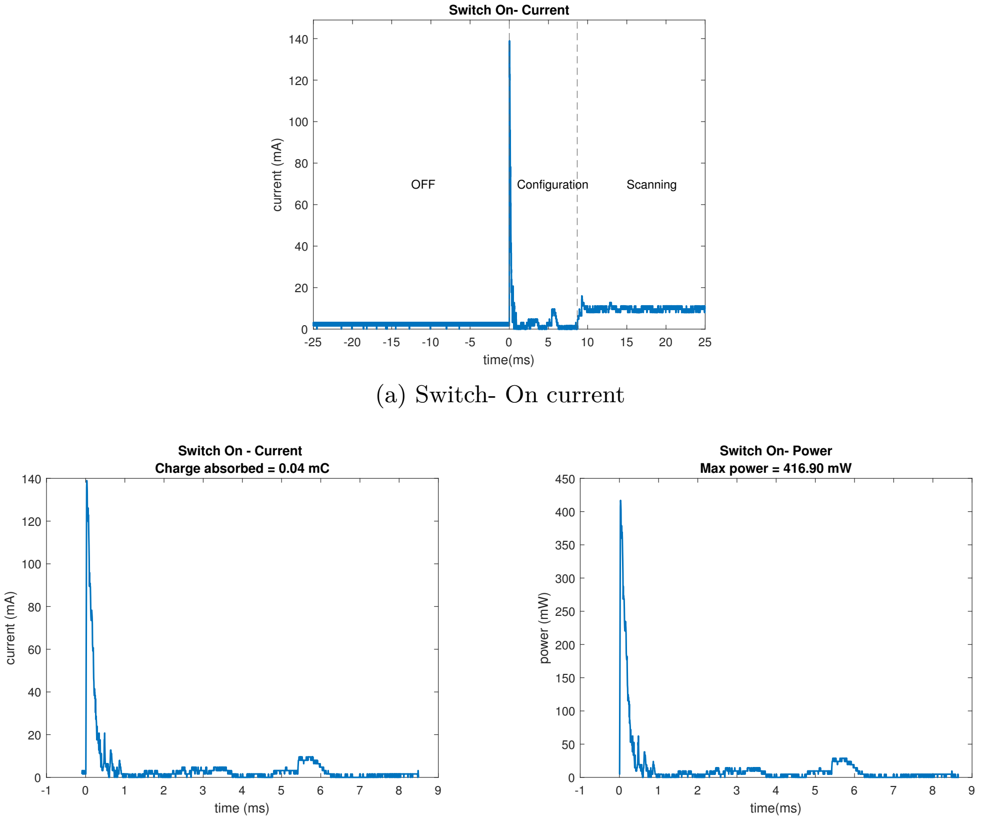
(a) Current profile with different payload size (excluding header and footer)

(b) Difference in total required channel

Figure 1.75: Impact of payload with costant antenna power

Receiver

Receiver board has been tested as transmitter one: switch-on current and configuration phase are shown in Figure 1.76.



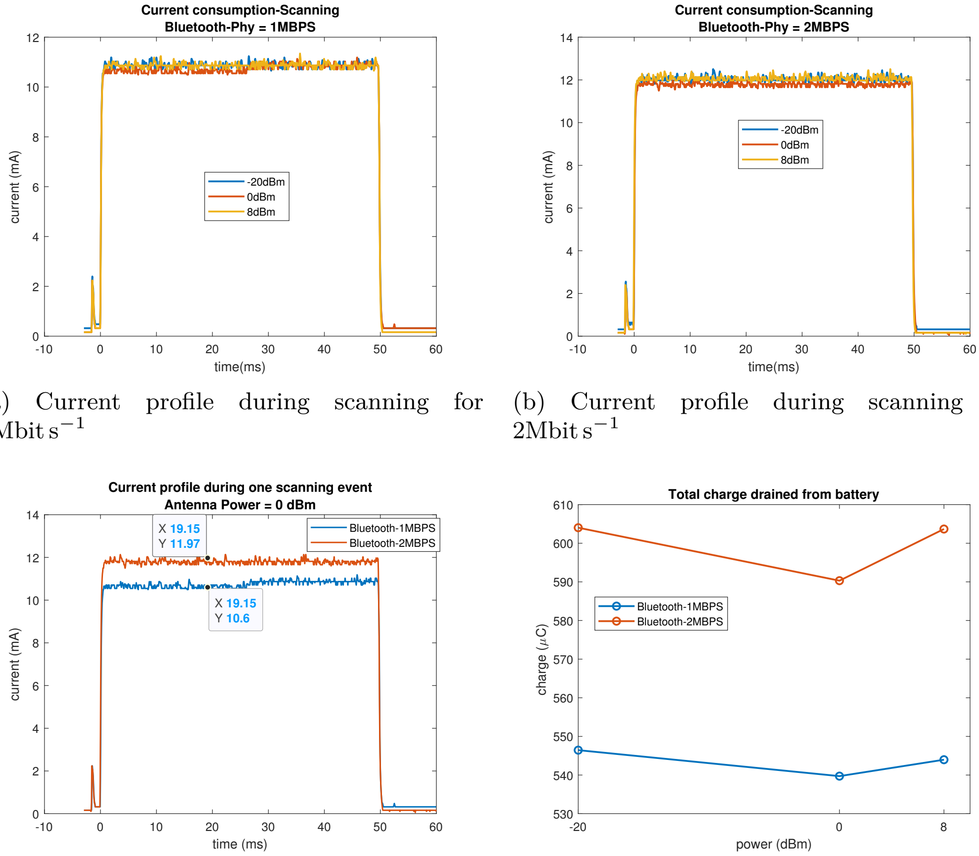
(a) Switch- On current

(b) Charge drained

(c) Power consumption, with $V_{dd} = 3\text{V}$

Figure 1.76: Current ((b) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 8.5ms, where scanning starts (**c**).

At the end, the board enters in scanning mode, ready to initiate a connection with in proximity advertiser. The antenna must be tuned, according to the modulation scheme used, which determines the total power consumption. Results are shown in Figure 1.77



(a) Current profile during scanning for 1Mbit s^{-1}

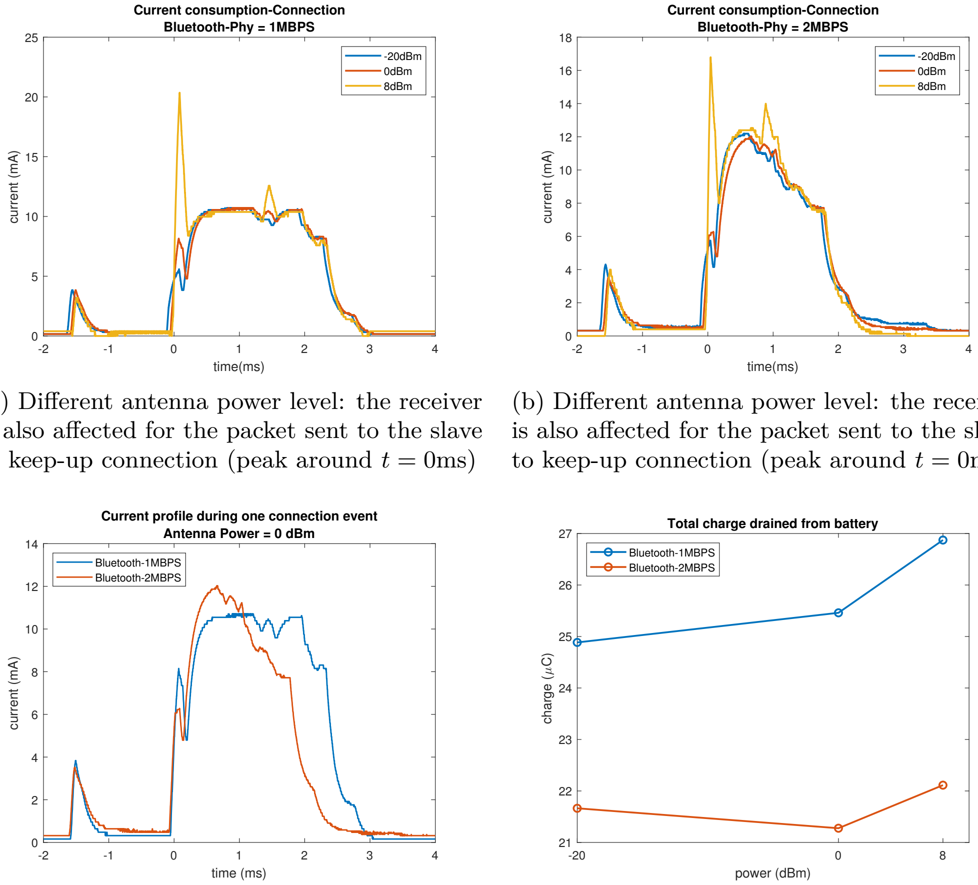
(b) Current profile during scanning for 2Mbit s^{-1}

(c) Comparison of different modulation schemes at same antenna power level. In this case, the difference arises from different modulation rather than antenna power level, which affects transmitter part

(d) Charge required for one scanning events for different modulation schemes. It is more or less constant within same modulation scheme: difference, in this case, is due to quantization and acquisition from oscilloscope's ADC

Figure 1.77: Scanning event

After a connection occurs, boards are ready to exchange ATC data, whenever master is ready to accept it: in the following, comparison of 1 and 2 Mbit/s are made during one exchange of data (during one single connection the behavior is the same, but with less time needed due to the payload size of connection packet)



(a) Different antenna power level: the receiver is also affected for the packet sent to the slave to keep-up connection (peak around $t = 0$ms)

(b) Different antenna power level: the receiver is also affected for the packet sent to the slave to keep-up connection (peak around $t = 0$ms

(c) Comparison at same antenna power level of different modulations: 2Mbit s^{-1} stays on less time for higher bit rate used

(d) Total charge absorbed during one connection event: bit rate plays a key role in energy consumption.

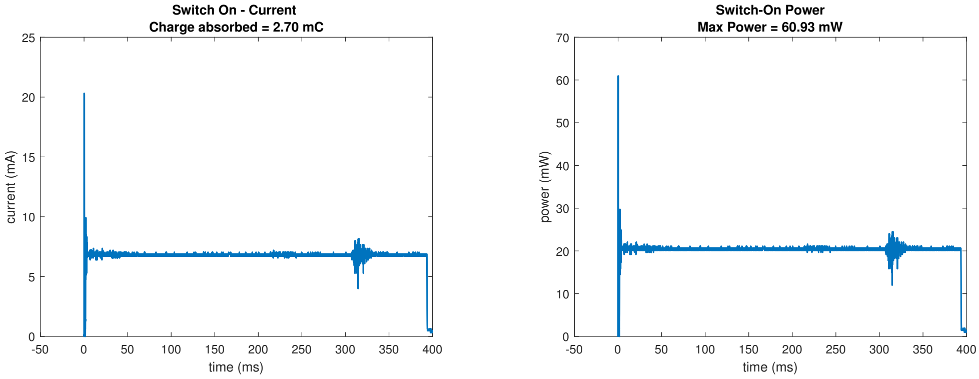
Figure 1.78: Connection event

IEEE 802.15.4

The characterization of transmitter and receiver implementing *IEEE 802.15.4* was equal to the one for *BLE* technology, but here more simpler because there aren't advertising and connection states (tx) nor scanning state (rx).

Transmitter

Current and power absorbed during switch on and configuration phase are shown in Figure 1.79: peak current is small, but the total time needed to set up the board is high, leading to a big energy consumption



(a) Switch- On current

(b) Switch-On Power

Figure 1.79: Current (**a**) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 400ms, where transmitter is ready to send ATC data.

After initialization, the board enters in idle state, where it waits for external input to trigger the transmission of ATC data: here, the consumption is minimum. The antenna is off and only source of power dissipation are external peripherals and MCU.

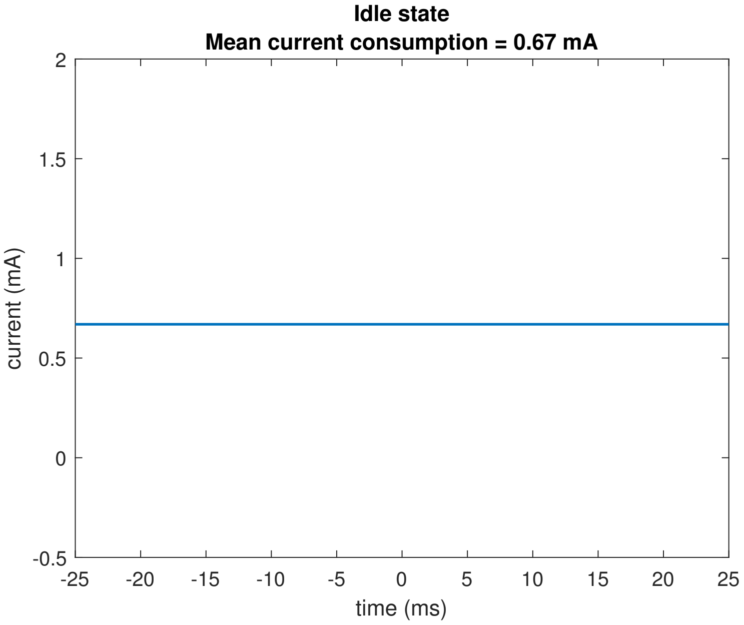
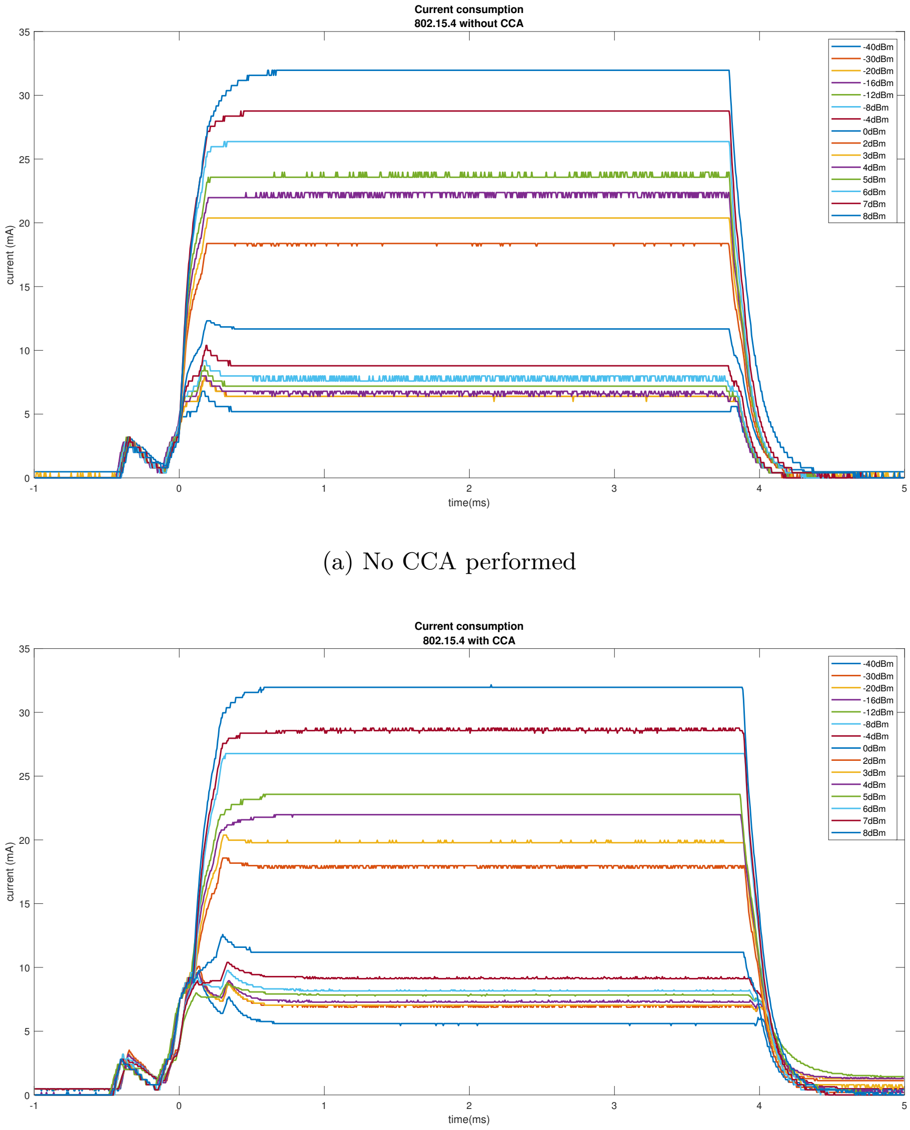


Figure 1.80: Current consumption in idle state. A mean operation has been performed on current signal to clean raw data.

When the user presses the appropriate button, transmitter starts sending ATC data: optionally, it could perform Clear Channel Assesment (CCA) before the real transmission;

the impact of it has been also studied.

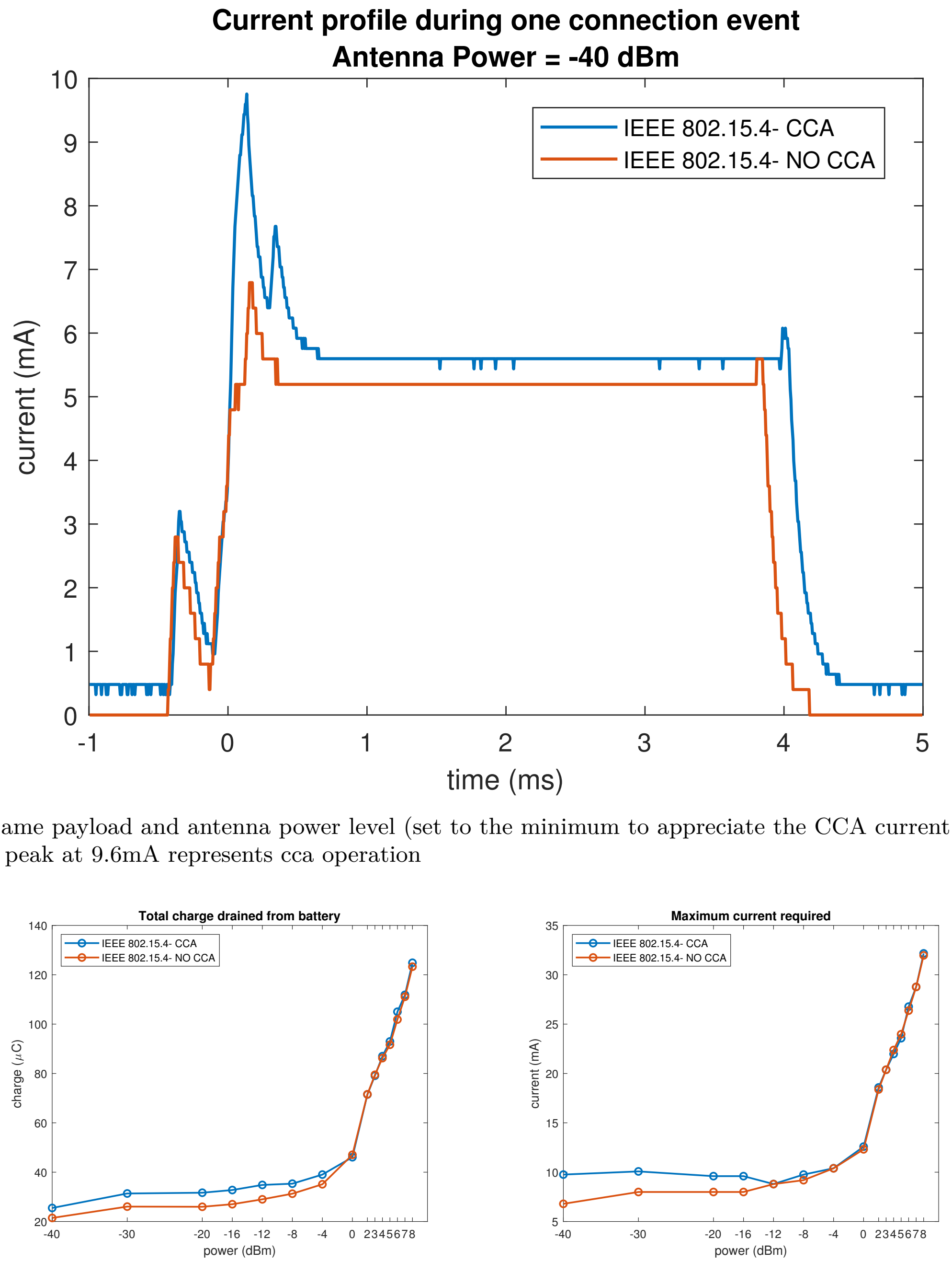


(a) No CCA performed

(b) CCA performed

Figure 1.81: Current absorbed during transmission at different antenna power level, with (**b**) and without (**a**) cca; as expected, current increase with antenna power level

In order to understand differences between the different implementation, some charts are plotted and explained in Figure 1.82



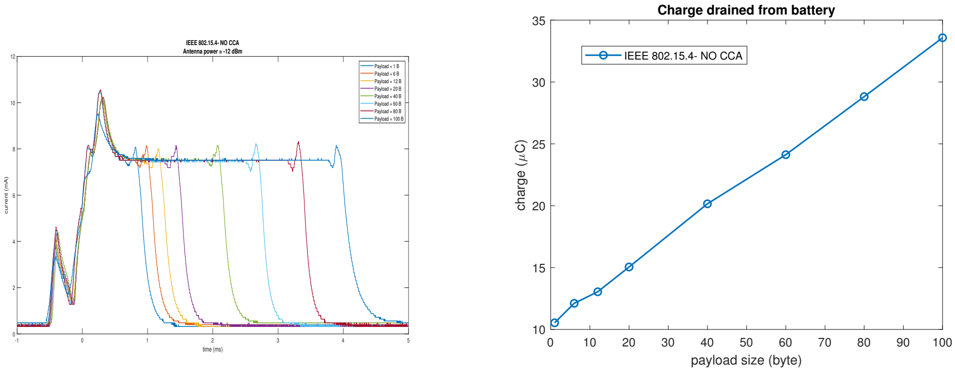
(a) Same payload and antenna power level (set to the minimum to appreciate the CCA current peak). Blue peak at 9.6mA represents cca operation

(b) Total charge absorbed for 1 single transmission. The CCA has an impact for low antenna power level: for higher one, the current transmission overshadow the cca one

(c) Maximum current in transmission: same considerations made in **b** are valid

Figure 1.82: Current absorbed during transmission at different antenna power level, with and without cca(**a**); in **b** and **c** some energy considerations are made

Payload size has a direct impact on energy consumption: measures above are all taken using payload size equal to 100B, excluding header and footer. In order to understand payload's impact, some measures have been taken, fixing antenna power level (without cca) and varying packet's size.



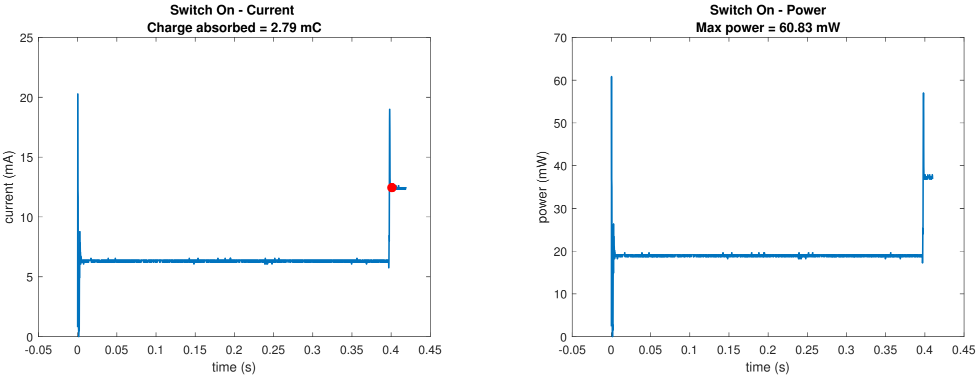
(a) Current profile varying payload size; as expected, antenna stays in on state more time when payload size is high

(b) Total charge against payload size.

Figure 1.83: Impact of payload size. More time needed for transmit data bring higher consumption

Receiver

The current needed to switch on and configuration is shown in Figure 1.84

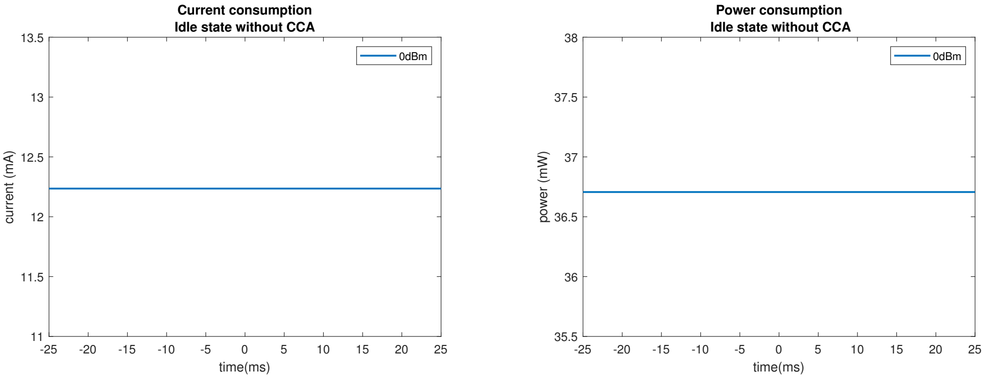


(a) Switch- On current. Red point indicates the moment in which antenna is ready to receive data

(b) Switch- On Power

Figure 1.84: Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 400ms, where antenna stays on in order to receive available data

As soon as the configuration is completed, the antenna remains in RX state, where it can listen for all the packets sent compliant to *IEEE 802.15.4*: the current consumption is expected to be constant till the board is switched off. In a preliminary phase, some other measures were taken varying the transmitted power, but, as expected, the result didn't change: in fact, the board is in a receiving state, where the only difference in power consumption arise in different modulation scheme. The tx power affects only the ACK packets sent at the start and stop of communication and it is negligible on the overall consumption. In Figure 1.85, the current consumption is shown: it remains constant over the time (as expected) and it is high for the need of antenna to be in a receiving state, so with all the analog front-end switched on and working.



(a) Current consumption

(b) Power consumption

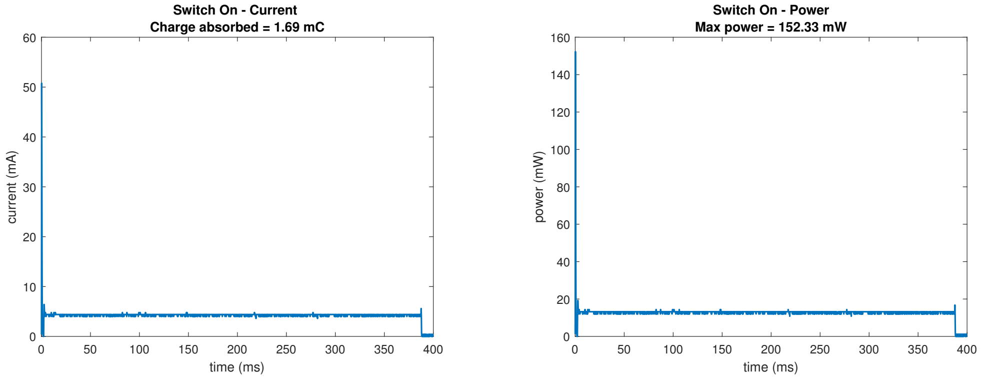
Figure 1.85: Receiving state

Proprietary Protocol

The characterization of transmitter and receiver implementing the new protocol was equal to the one for *IEEE 802.15.4* technology. In the following transmitter and receiver results are shown.

Transmitter

The current needed to switch on and configuration is shown in Figure 1.86

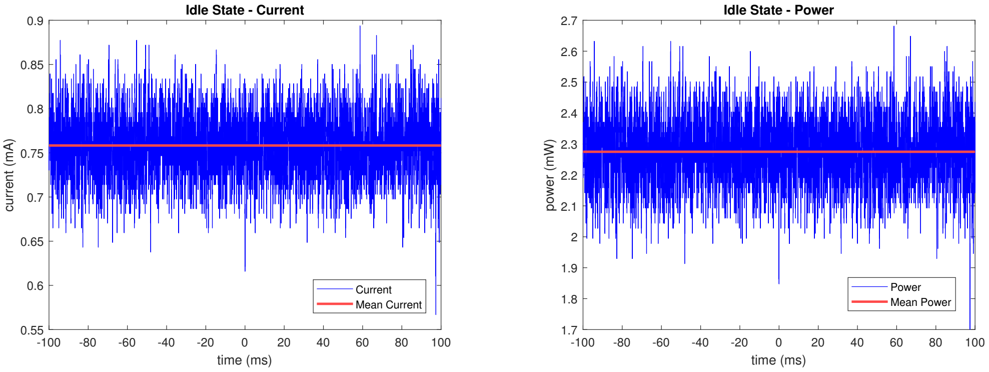


(a) Switch- On current

(b) Switch- On Power

Figure 1.86: Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 8.5ms, where antenna stays on in order to receive available data

Once the board is configured, it is ready to send data, whenever the user press the right button: there is no concept of advertising and connection. The transmitter enters in an idle state, where it waits for some events: current and power consumption are shown in Figure 1.90

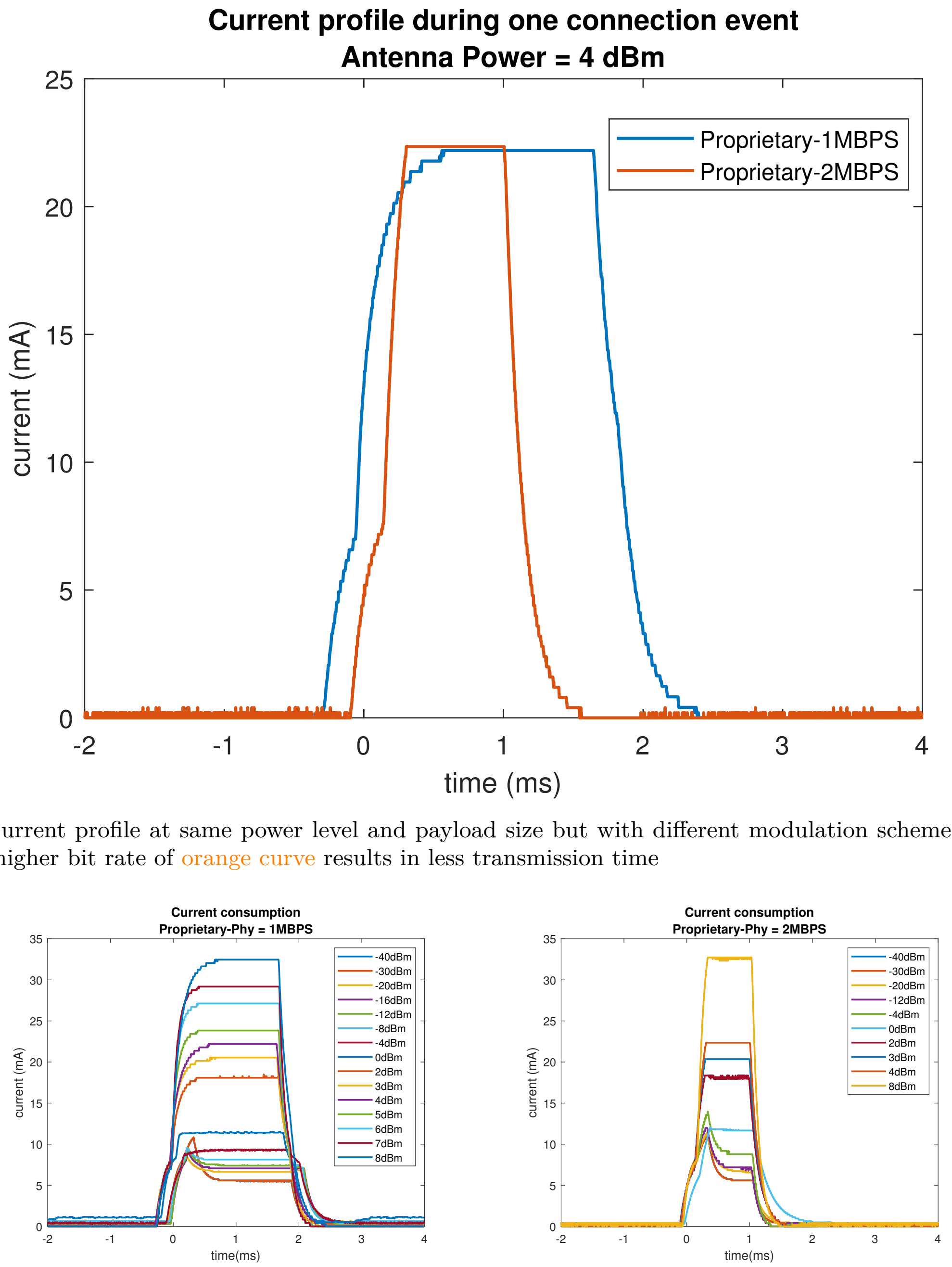


(a) Current consumption

(b) Power consumption

Figure 1.87: Current and power consumption in idle state

The only thing that this board can do is to send data (in a future implementation, a receiver functionality could be added, in order to wait for ACK before the real start of data streaming). Two implementation are available: one exploit 1Mbit/s modulation scheme, while other implements 2Mbit/s one.



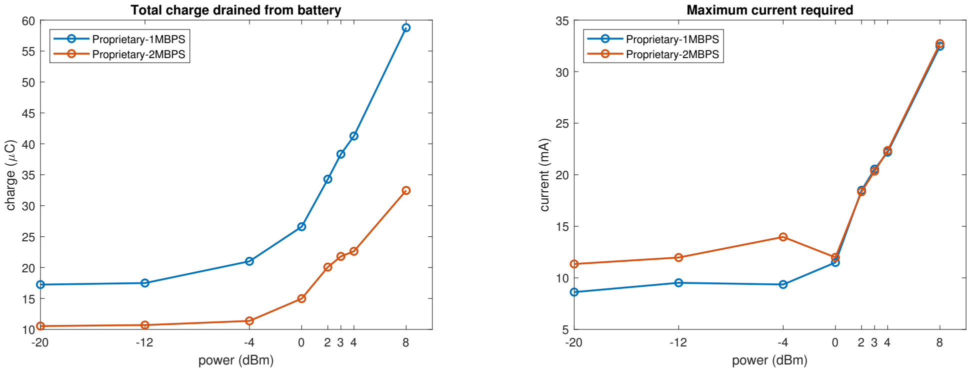
(a) Current profile at same power level and payload size but with different modulation schemes used. The higher bit rate of orange curve results in less transmission time

(b) Current profile for 1Mbit s^{-1} modulation scheme varying antenna power level

(c) Current profile for 2Mbit s^{-1} modulation scheme varying antenna power level

Figure 1.88: Current profiles of different modulation scheme (**b**,**c**) and comparison between them (**a**)

In order to compare the two modulation schemes used, different trials at different antenna power level have been collected. Results are shown in Figure 1.89

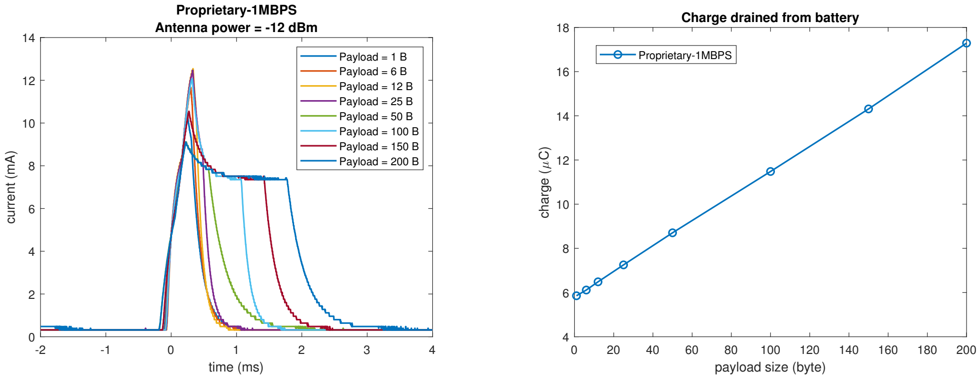


(a) Charge absorbed

(b) Peak current. At low antenna power level, the current needed to swtich on antenna represents the main current consumption, while at higher levels the transmission current prevails

Figure 1.89: Total charge and maximum current in one transmission event. Higher bit rate of 2Mbit/s modulation scheme results in low transmission time which brings benefit in energy consumption

The impact of payload size has also been studied. In the charts plotted above, the length of GATT data (configurable one) is set to 200B.



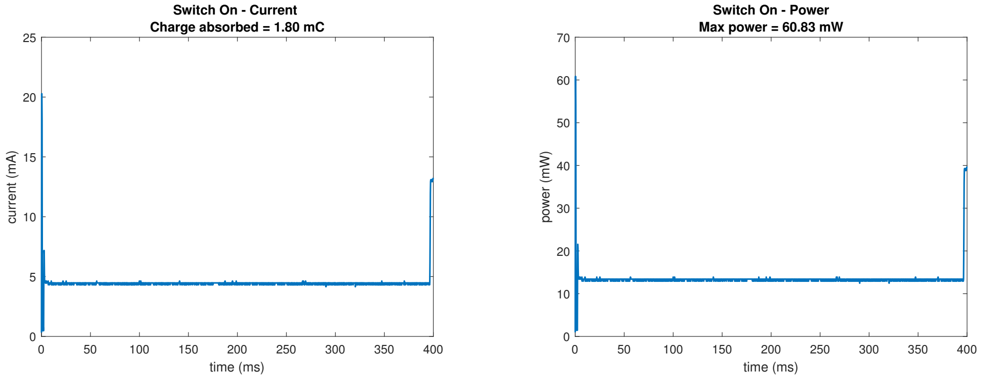
(a) Current profile varying payload size. The transmission time increase with incresing payload length

(b) Charge against payload size. As expected charge drained increase with length

Figure 1.90: Impact of payload size

Receiver

The current needed to switch on and configuration is shown in Figure 1.91

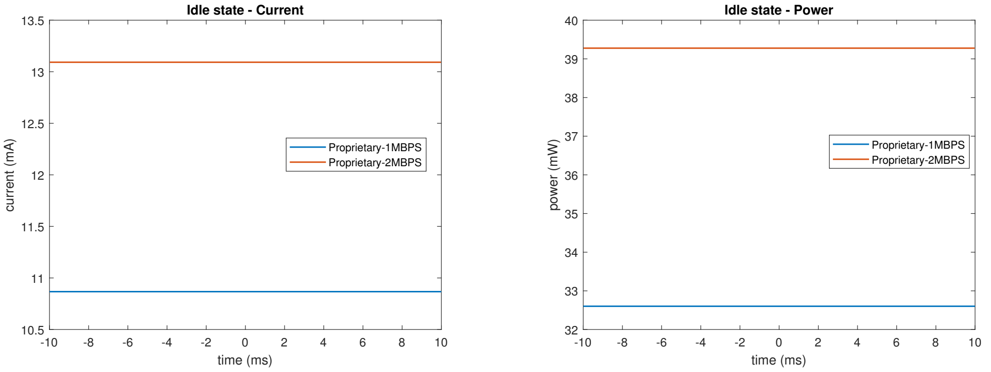


(a) Switch- On current

(b) Switch- On Power

Figure 1.91: Current ((a) and power consumption (**b**) during switch on and configuration phase. The board is mechanically switch on at t = 0 ms: the configuration phase last till t = 8.5ms, where antenna stays on in order to receive available data

As soon as the configuration is completed, the antenna remains in RX state, where it can listen for all the packets sent compliant to modulation schemes provided by *Nordic*©: the current consumption is expected to be constant till the board is switched off. In a preliminary phase, some other measures were taken varying the transmitted power, but, as expected, the result didn't change: in fact, the board is in a receiving state, where the only difference in power consumption arise in different modulation scheme(1Mbit/s vs 2Mbit/s). Results are shown in Figure 1.92: for both modulation schemes, the consumption is constant and high for the need to have analog front-end always switched on and working.



(a) Current consumption

(b) Power consumption

Figure 1.92

Figure 1.93: Current (a) and power consumption (b) results for receiving side. 1Mbit/s modulation scheme shows lower consumption

Comparison

The choice of the best protocol is carried out considering in particular transmitter side, even if some considerations are made also for receiver side.

Transmitter

Comparison between protocols and power is shown in Figure 1.94 and 1.95: best protocol in terms of energy consumption is the **Proprietary 2** Mbit/s. Indeed, considering the charge needed for one single transmission, *Proprietary 2* Mbit/s shows great results, while *IEEE 802.15.4* is the worst one (in addition to the fact that maximum payload achievable is only 100 B). An estimate of lifetime (in transmission mode) is given in Figure 1.95, considering a *CR2032 lithium battery* with *capacity* equal to 220mA h = 792C.

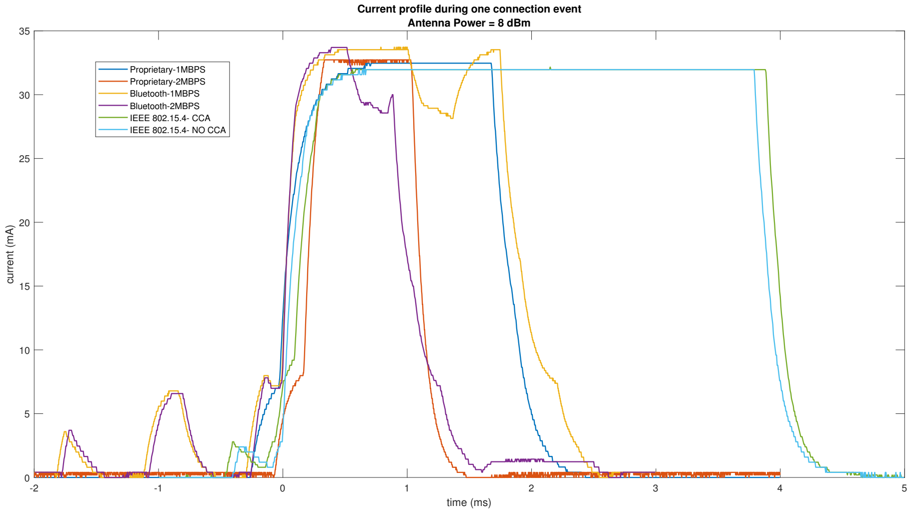
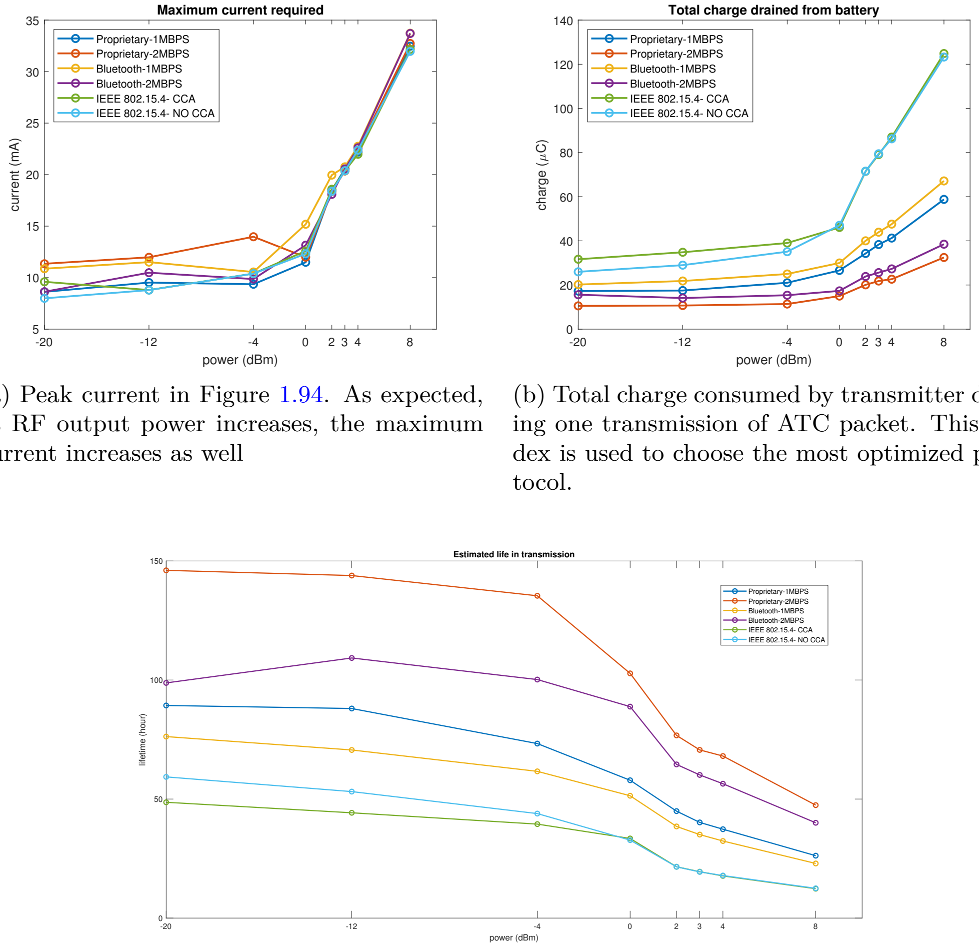


Figure 1.94: Current profile for different protocols during one transmission event. Maximum payload size used: 100B for *IEEE 802.15.4* and 200B for *BLE and Proprietary*. Lower bit-rate reflects in higher time needed to encode and send the entire packet.



(a) Peak current in Figure 1.94. As expected, as RF output power increases, the maximum current increases as well

(b) Total charge consumed by transmitter during one transmission of ATC packet. This index is used to choose the most optimized protocol.

(c) Estimate of lifetime in transmission mode: the entire battery capacity is used when calculating these estimates.

Figure 1.95: Results of different protocols in worst energy consumption scenario. In **a)** maximum current value reaches during one transmission is shown, while in **b** and **c** total charge (for one transmission) and the resulting estimate of lifetime are given.

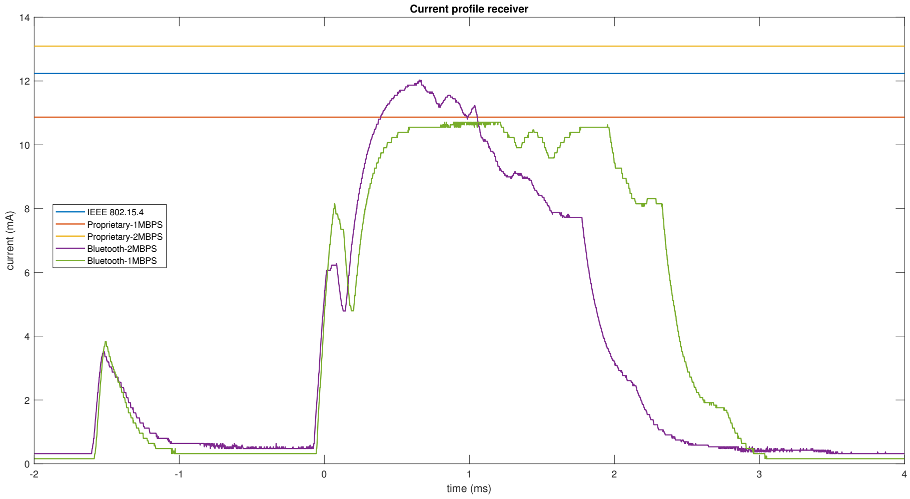
The results listed below in Table 1.11 are given for worst scenario in same conditions, i.e. considering higher throughput (maximum payload size) with minimum connection interval and higher antenna power level (that allow for maximum reliability at fixed distance). Highlighted row is the one which gives best performance in transmission condition.

Protocol	Payload [B]	Maximum current [mA]	Charge [µC]	Life (TX) [h]	Life (Idle) [h]
Proprietary 2 MBPS	200	32.73	32.47	47.43	293
Proprietary 1 MBPS	200	32.47	58.77	26.20	293
Bluetooth 2 MBPS	200	33.70	38.48	40.02	240
Bluetooth 1 MBPS	200	33.73	62.14	22.94	201
IEEE 802.15.4 CCA	100	32.16	124.85	12.33	328
IEEE 802.15.4	100	31.96	123.27	12.49	328

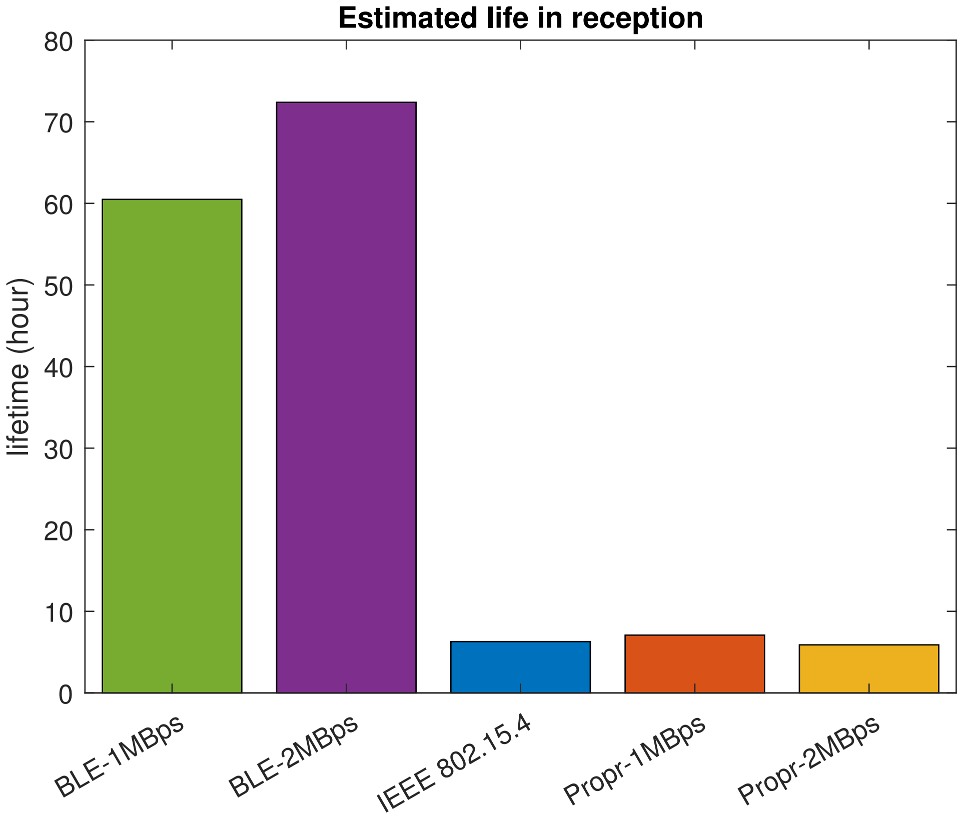
Table 1.11: Results in real-scenario for transmitter side. Connection interval is set to 7ms. Idle mode for BLE is intended in a connection with master device. The maximum payload for *IEEE 802.15.4* is 100B to comply with the standard.

Receiver

Results are shown below. In this case best protocol is *Bluetooth Low Energy* for its synchronization during transmission, a feature applicable also to proprietary protocol (in future works). In fact, for *BLE* technology transmitter and receiver side are synchronized together: both of them know when they can exchange data (every connection interval) and they will switch on RF output only in that amount of time: this behavior is reflected in Figure 1.96.



(a) Receiver current profile during one connection event. Only *BLE* technology shows a synchronization with transmitter, which drastically reduced energy consumption. All the other protocols, must have always RF output power enabled, increasing charge consumption



(b) Estimated lifetime in reception mode. The synchronizing mechanism between transmitter and receiver in *BLE* drastically impacted on estimated lifetime.

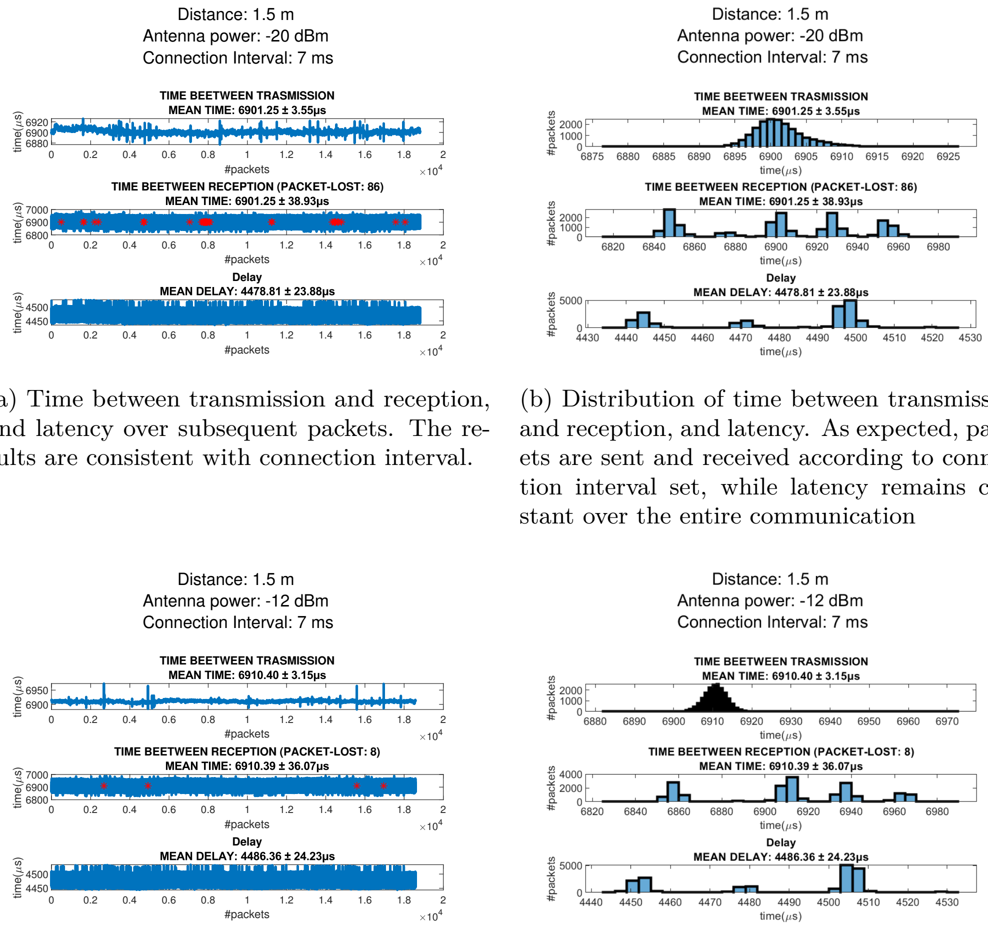
Figure 1.96: For BLE, connection interval is set to 7.5 ms

Throughput, Latency, Jitter and Reliability

The methods of data collection for different measures (latency, latency. ecc..) have been estimated using procedures explained in Section 1, working with transmitter and receiver boards, which implements firmware explained in Chapter 1. In this section, results for different tested protocols are shown.

IEEE 802.15.4

IEEE 802.15.4 uses a direct sequence spread spectrum technique together with O-QPSK modulation scheme, allowing for a bit rate up to 250kbit/s: in this way both throughput and latency are decrease in favor of reliability. Different trials have been performed varying antenna power level (at constant distance), in order to study reliability and correlation with parameters explained above. **Latency and Jitter** are shown in charts below, each for different antenna power level. *Left* charts shows time between different transmission and reception (y-axis) against subsequent packets, while *right* charts plots distribution. Results are more or less constant over different antenna power levels.



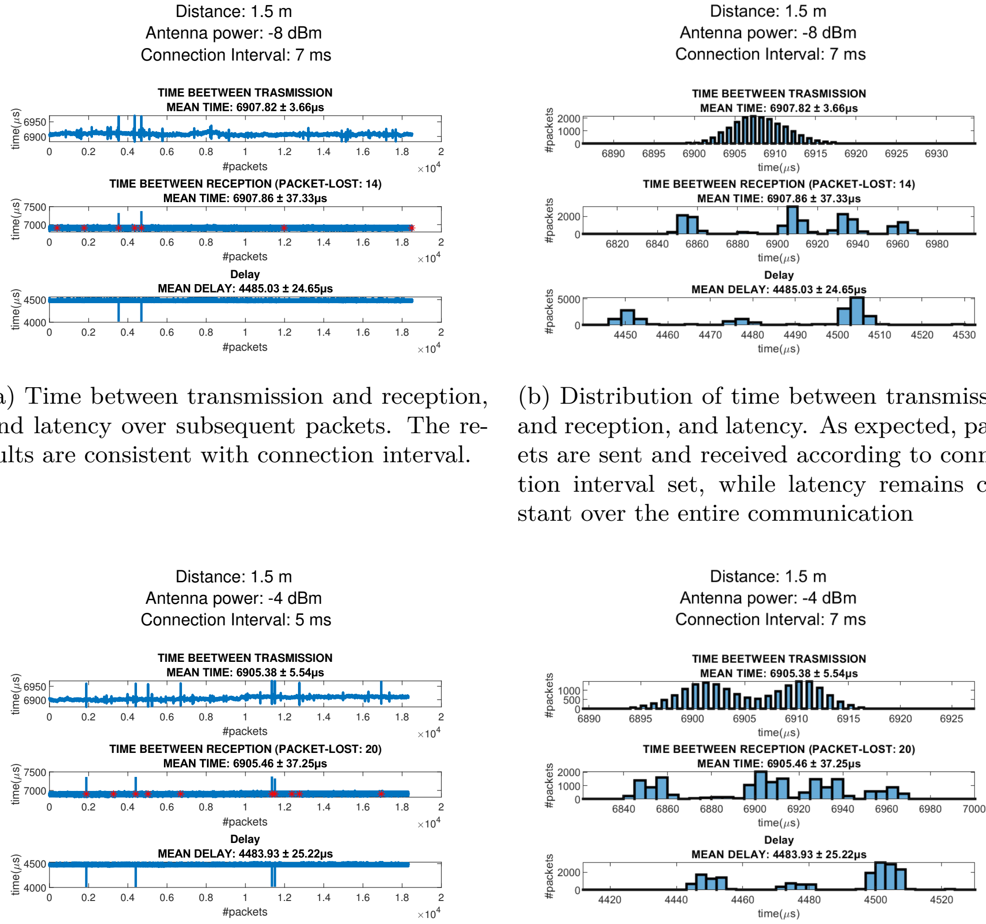
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.97: Latency and Jitter results for −20dBm and −12dBm



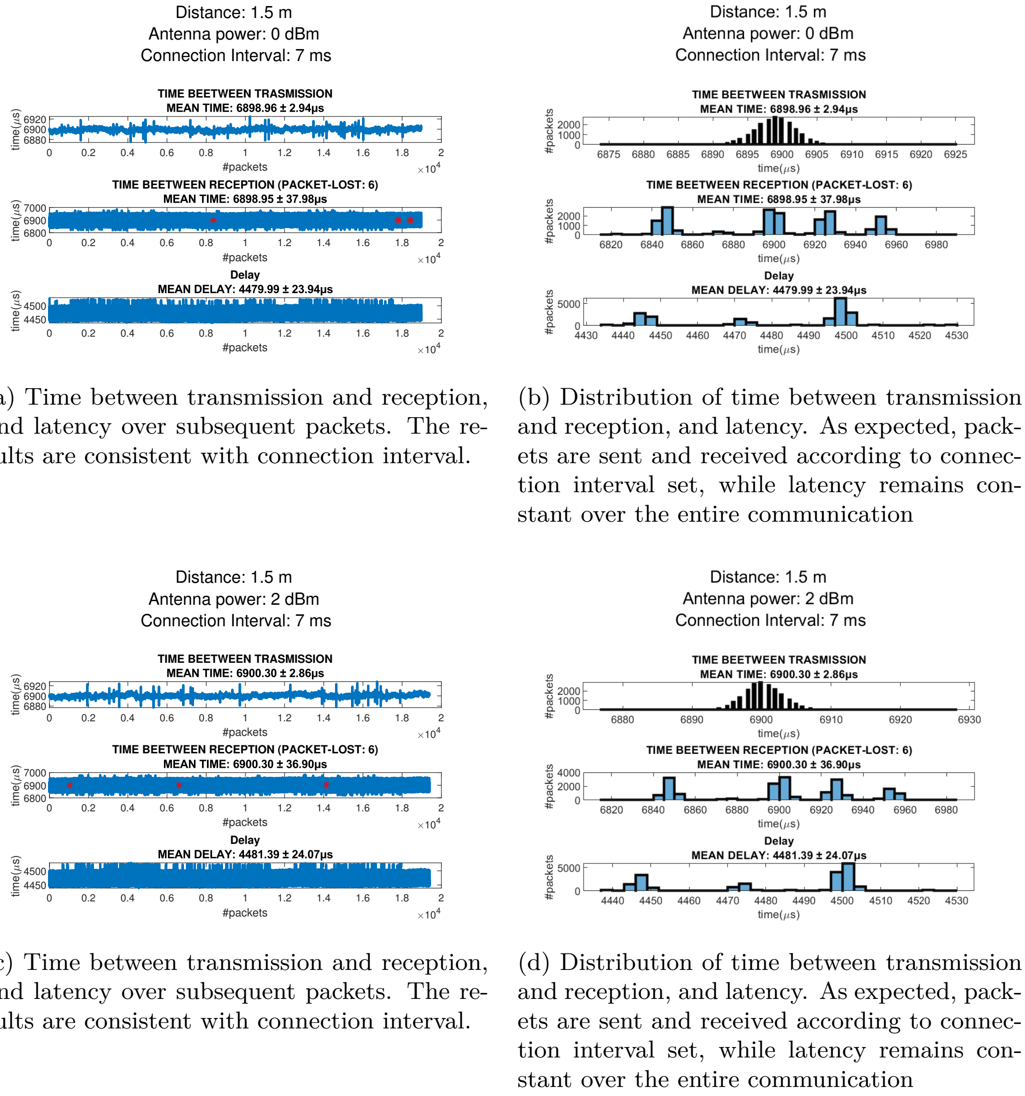
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.98: Latency and jitter results for −8dBm and −4dBm



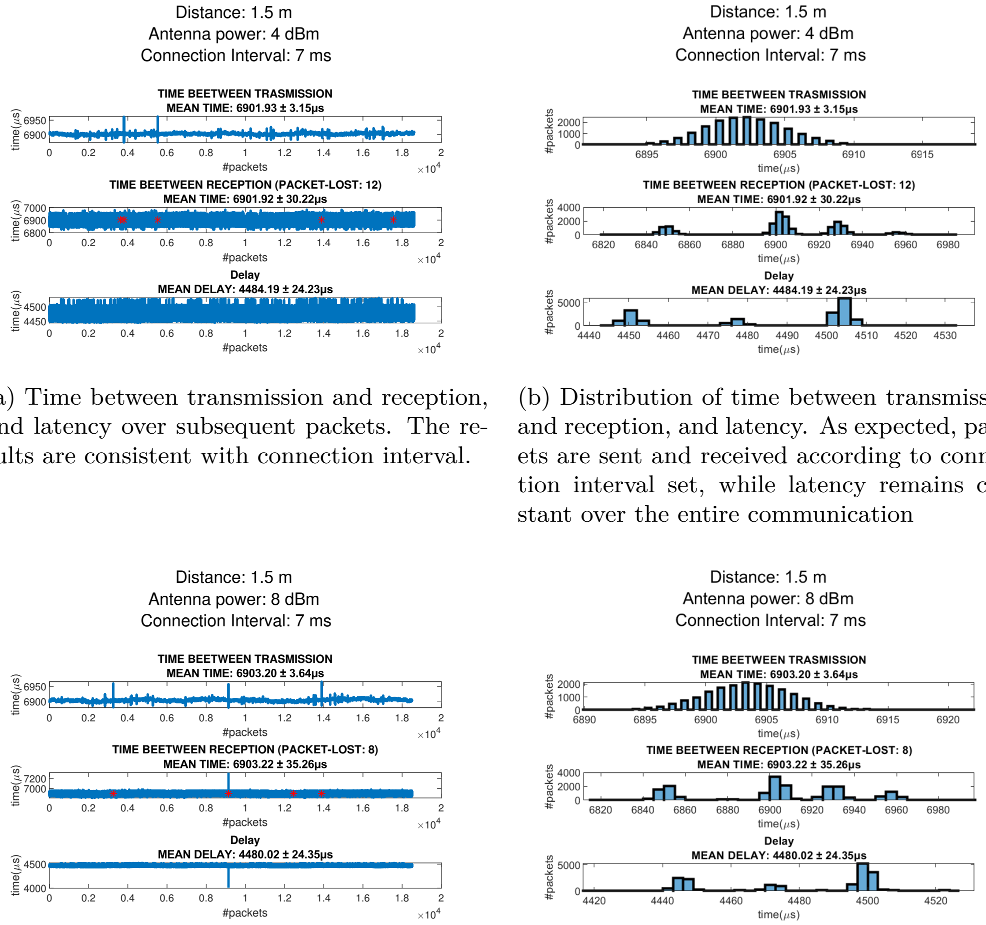
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.99: Latency and jitter results for 0dBm and 2dBm



(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

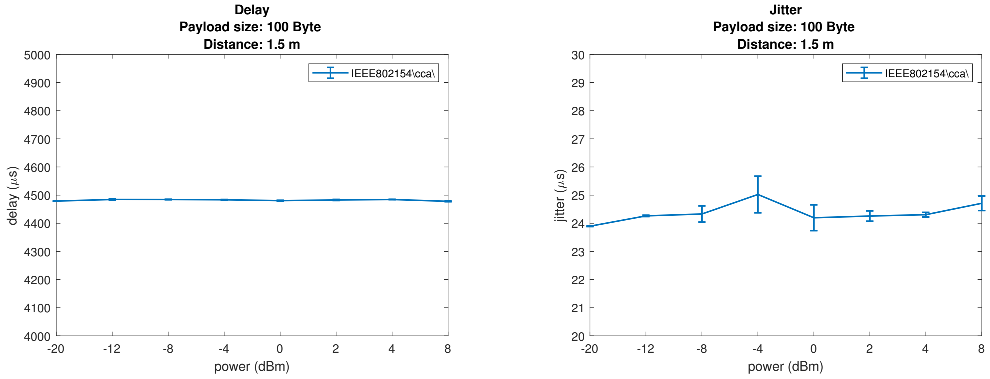
(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.100: Latency and jitter results for 4dBm and 8dBm

Latency and jitter mean value are obtained for each antenna power level over different trials: the overall results are shown in Figure 1.101. It can be noticed that both latency and jitter remain constant varying antenna power level: in this way, one single value can be extrapolated in order to characterize the transmission. The transmission latency and jitter (with given parameters given in Figure 1.101) is listed in Table 1.12

Protocol	Latency [µs]	Jitter [µs]
IEEE 802.15.4 CCA	4482	24

Table 1.12: IEEE 802.15.4 with CCA results



(a)

(b)

Figure 1.101: **a)**Latency and **b)** Jitter over different trials at different antenna power level

Throughput has been estimated considering the effective number of packets transmitted during elapsed time: different trials at different varying antenna power levels have been collected. Results are shown in Figure 1.102. As expected, throughput doesn't vary among different antenna power levels: results show that mean throughput is 115.90 ± 0.06 kbit/s

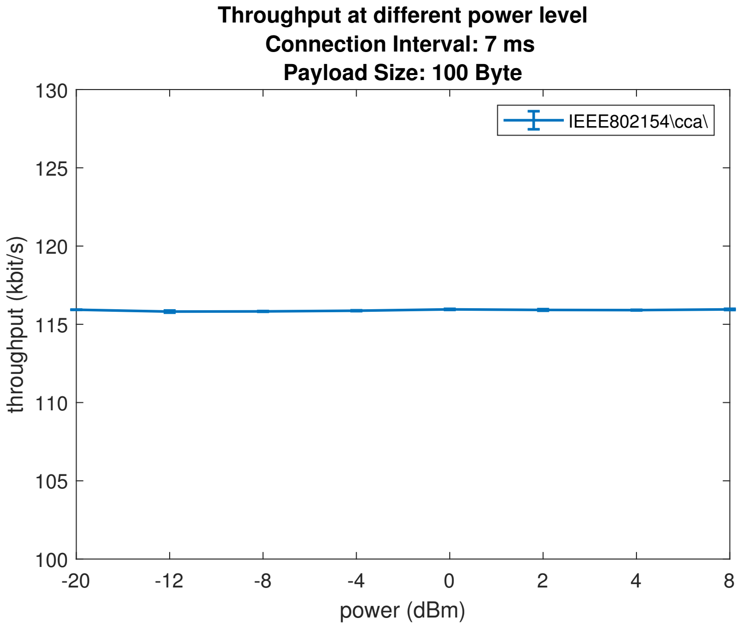
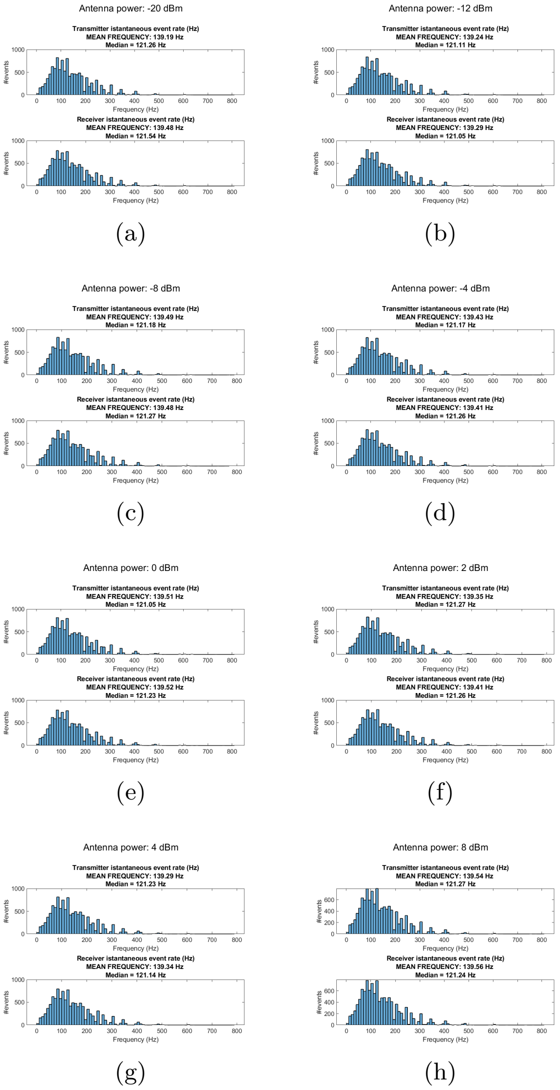


Figure 1.102: Throughput at different antenna power level

Similarity between ATC events has been measured using data plotted in Figure 1.103, which shows events distribution at different antenna power levels. In this way, a physiological study is conducted, trying to understand if received ATC signal is not distorted by transmission. If latency was perfectly constant over the entire communication, transmitter and receiver distribution would be equal: however, in real scenario, jitter exists, which brings tiny difference in events distribution.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1.103: Distribution of events for different antenna power levels

Mean frequency and median frequency deviation between tx and rx of transmitted ATC signal, over all trials at fixed power level, is plotted in Figure 1.104, while **cosine and ruzicka similarity and error** of the 2 metrics explained in Chapter 1 are shown in Figure 1.105 and Figure 1.106

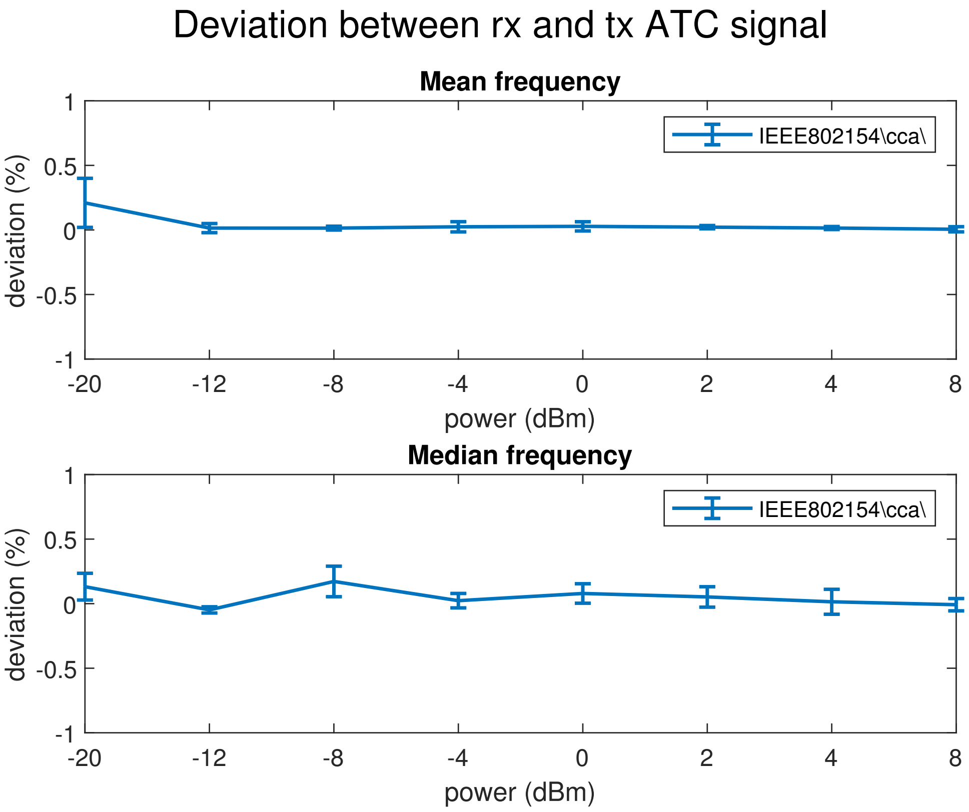
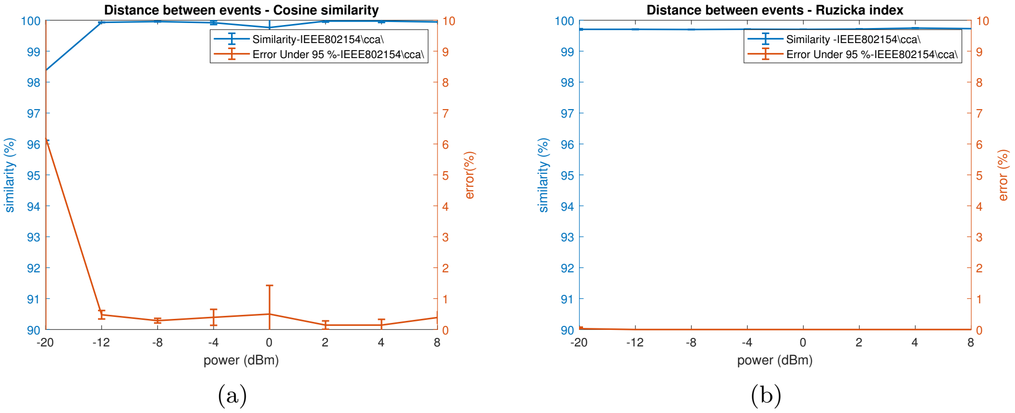


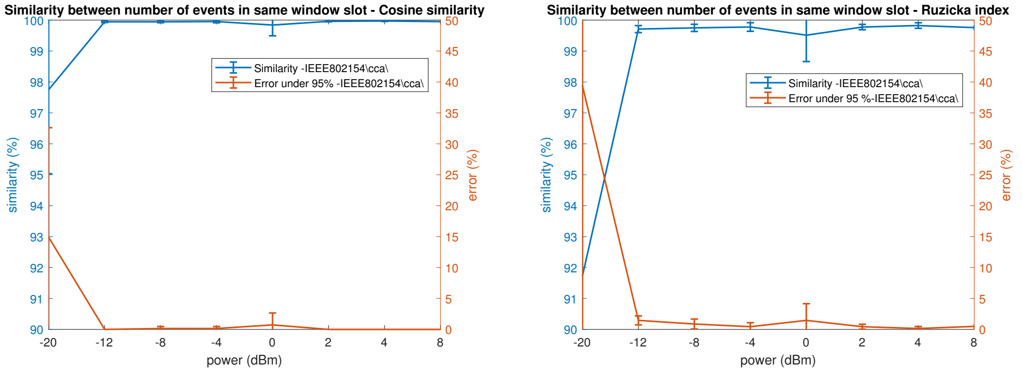
Figure 1.104: Frequency deviation between transmitter and receiver



(a)

(b)

Figure 1.105: Similarity between distance of consequent events



(a) (b)

Figure 1.106: Similarity between number of events in corresponding time windows

Reliability is shown in Figure 1.107. The direct sequence spread spectrum technique used, together with Clear Channel Assessment (CCA), allows for a minimum transmission error also at low antenna power levels, at the expense of throughput (lower bit rate).

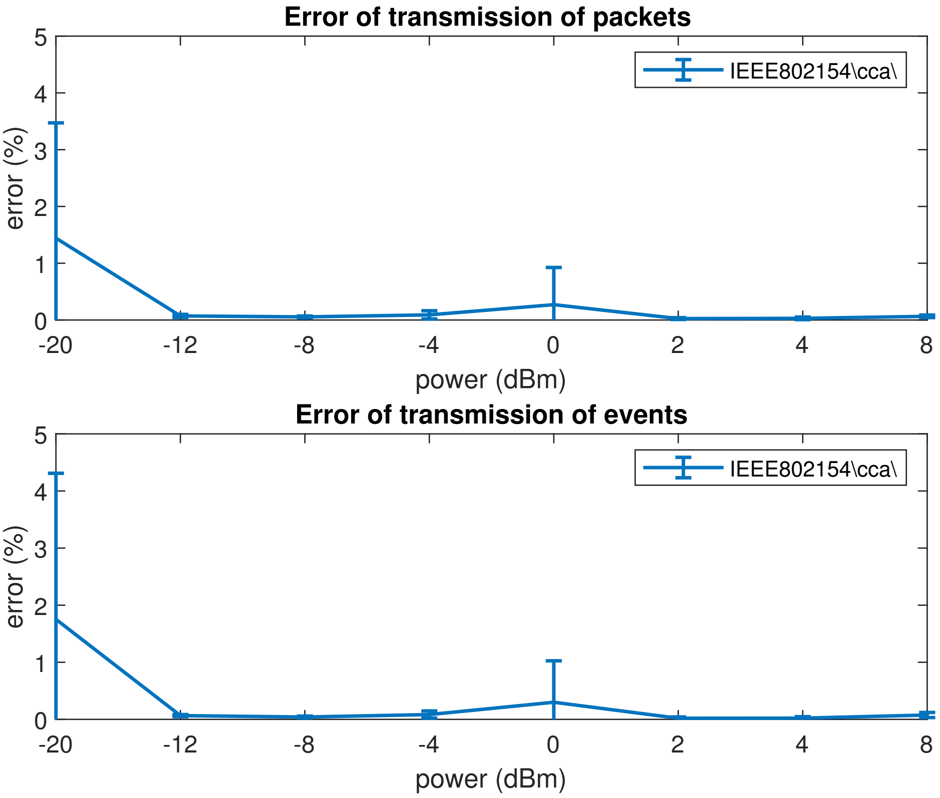
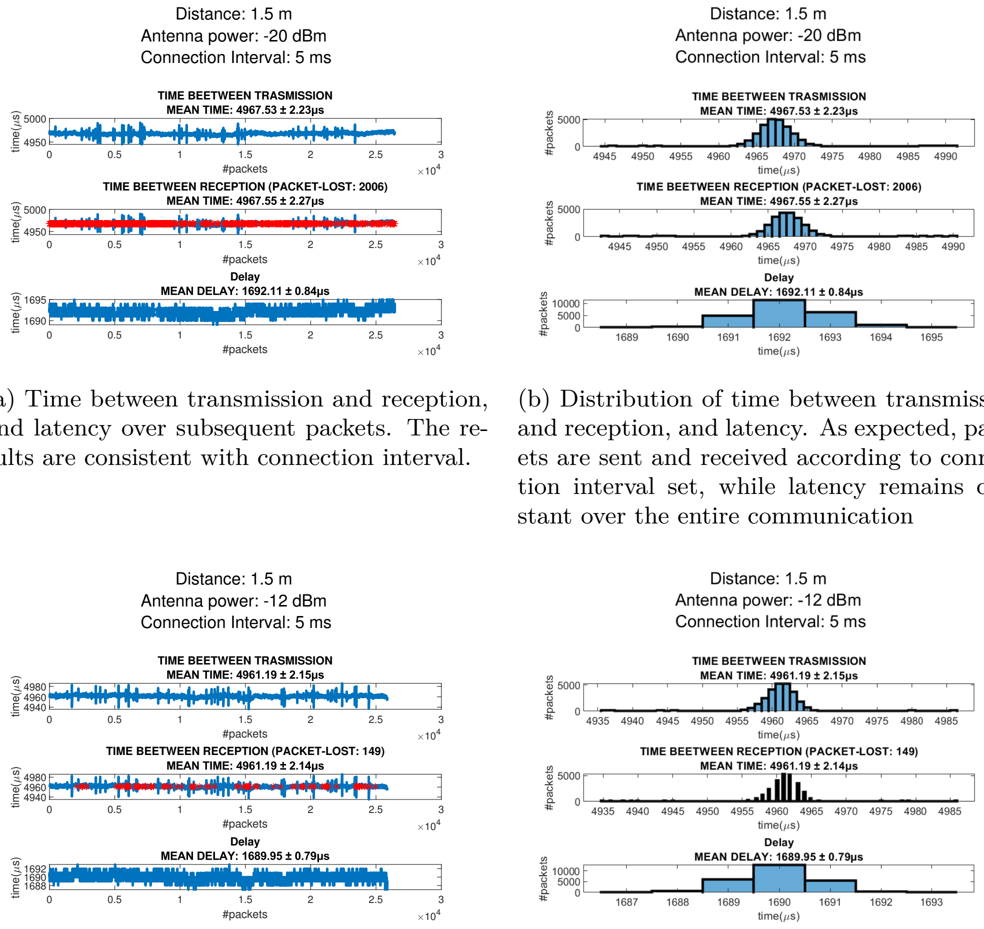


Figure 1.107: Packets lost and events lost ratio

Proprietary

Proprietary protocol, implemented on top of Nordic proprietary protocol, used a Frequency Shift Keying modulation, without any use of spread spectrum technique. The reliability of lower power is expected to be lower with respect to *IEEE 802.15.4*, which implements DS-SS.

Latency and Jitter at different antenna power levels for 1Mbit/s with payload size equal to 200B are shown in charts below



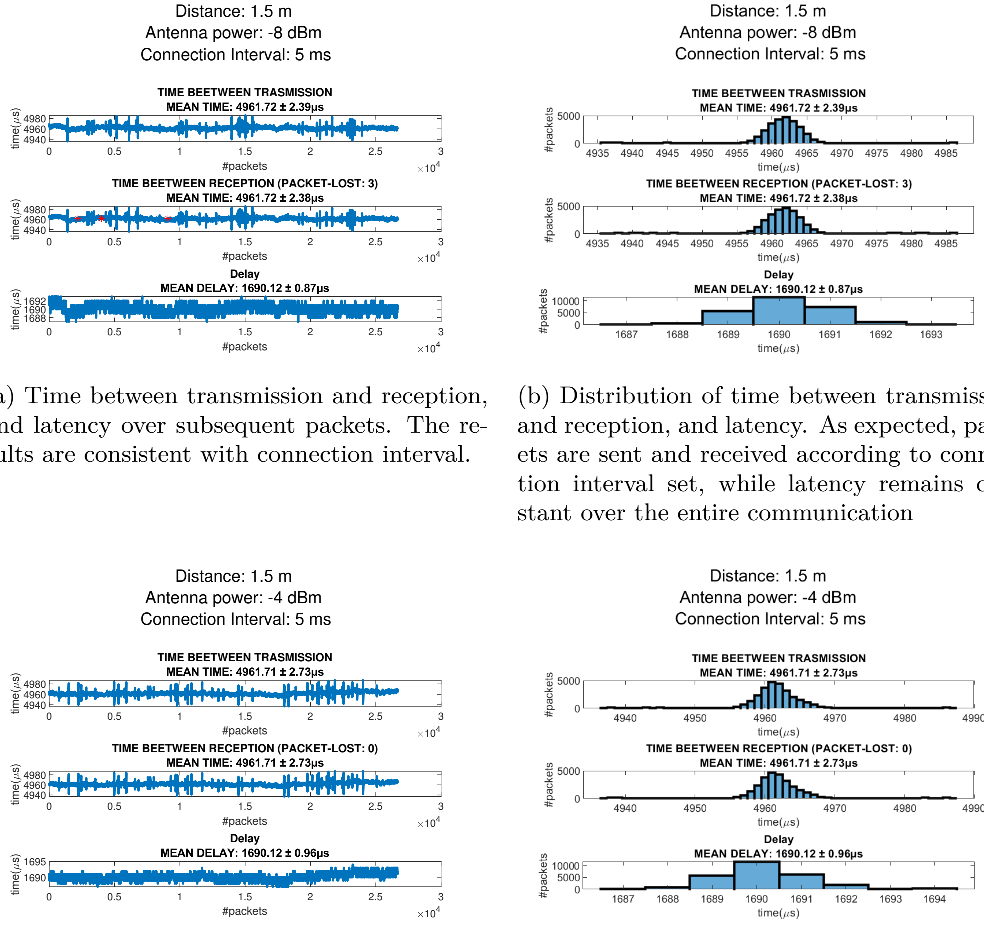
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.108: Results for −20dBm and −12dBm using 1 Mbit/s physical layer



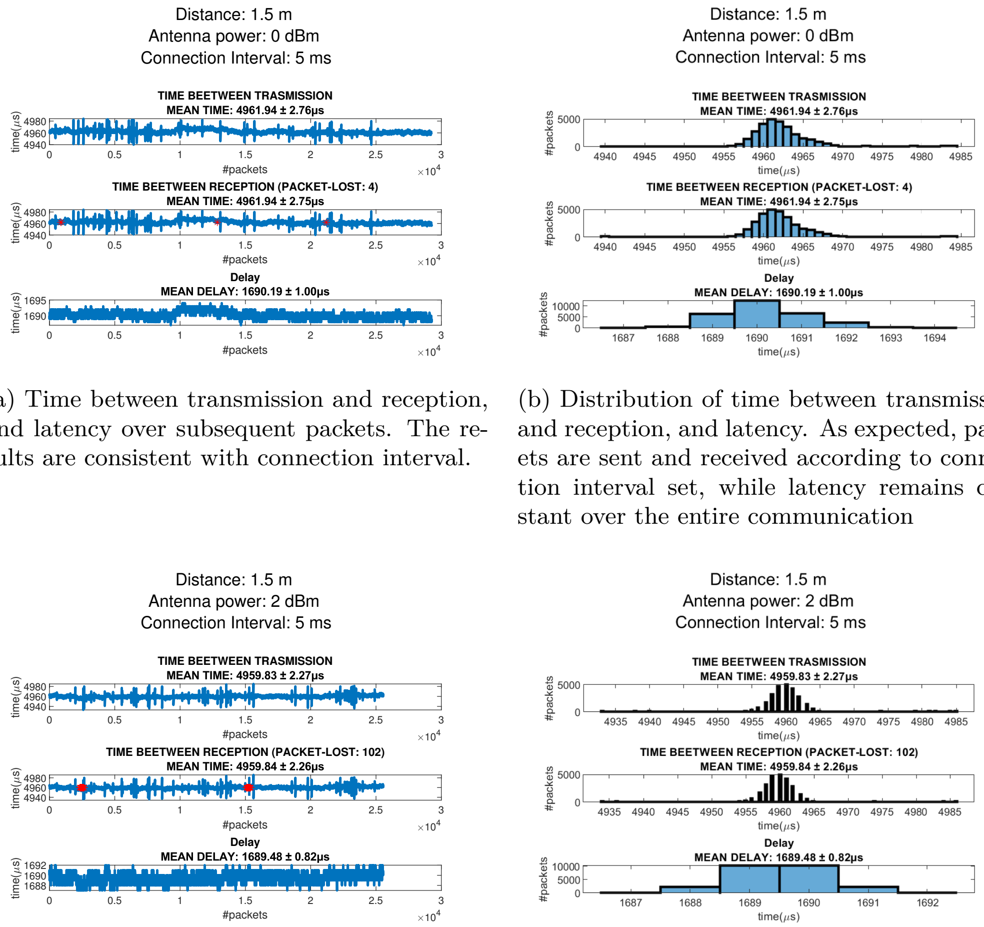
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.109: Results for -8dBm and -4dBm using 1 Mbit/s physical layer



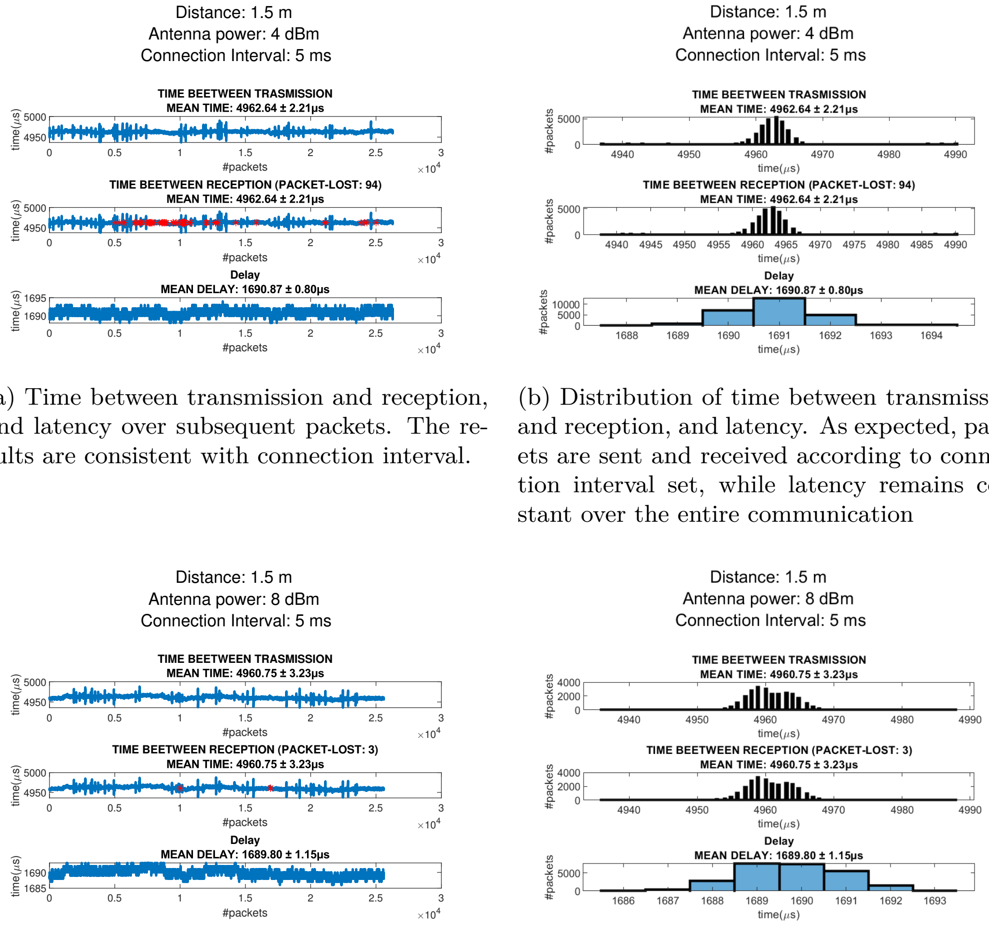
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.110: Results for 0dBm and 2dBm using 1 Mbit/s physical layer



(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

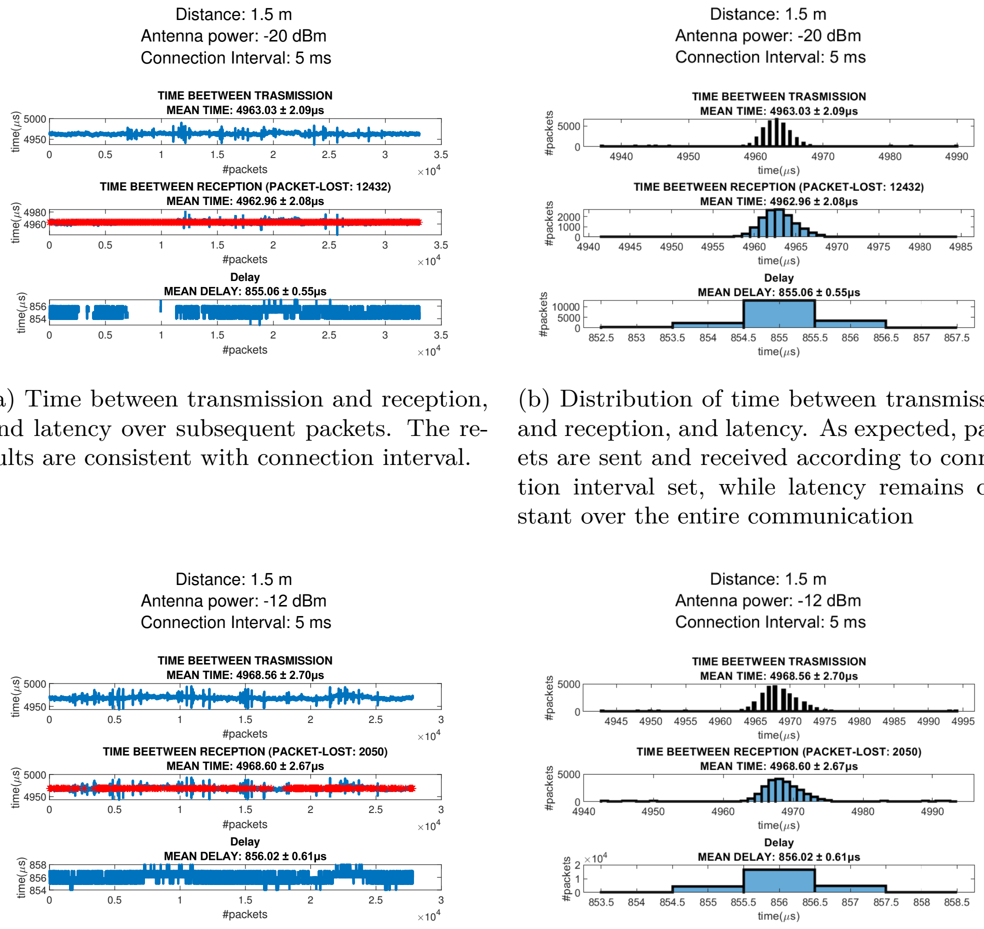
(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.111: Results for 4dBm and 8dBm using 1 Mbit/s physical layer

Latency and Jitter at different antenna power levels for 2 Mbit/s with payload size equal to 200B are shown in charts below



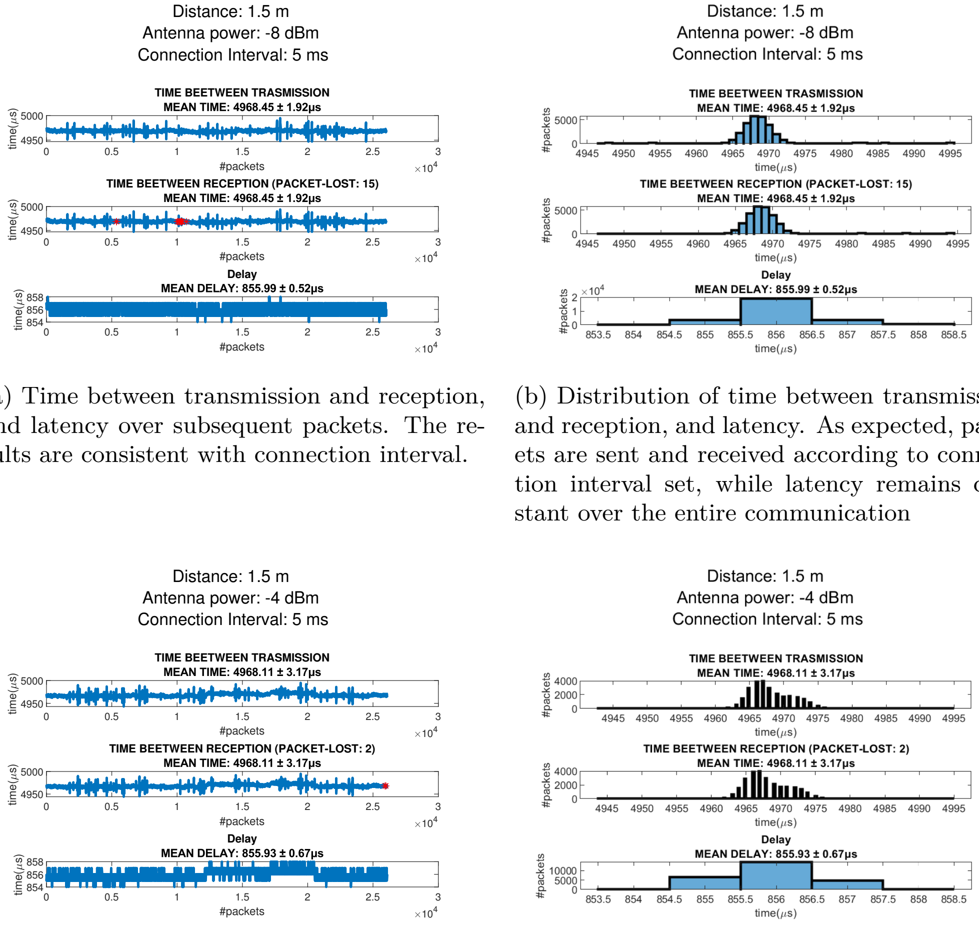
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.112: Results for -20dBm and -12dBm using 2 Mbit/s physical layer



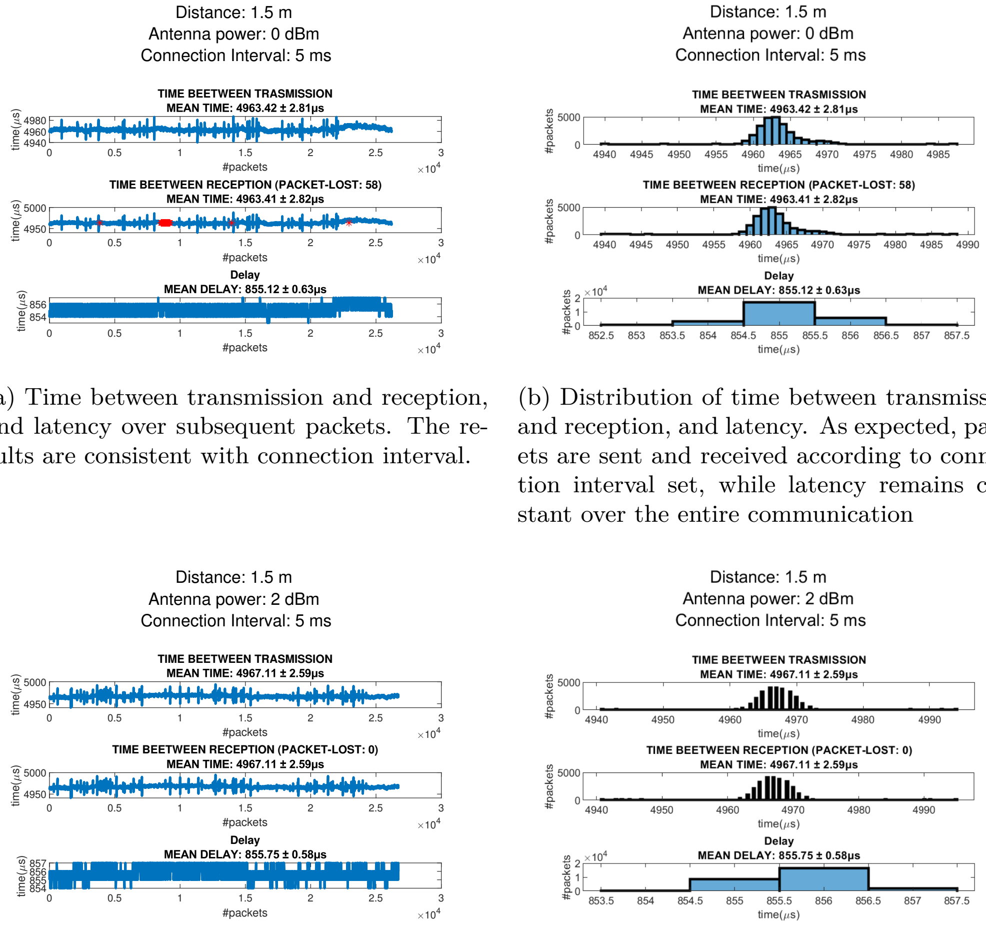
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.113: Results for −8dBm and −4dBm using 2 Mbit/s physical layer



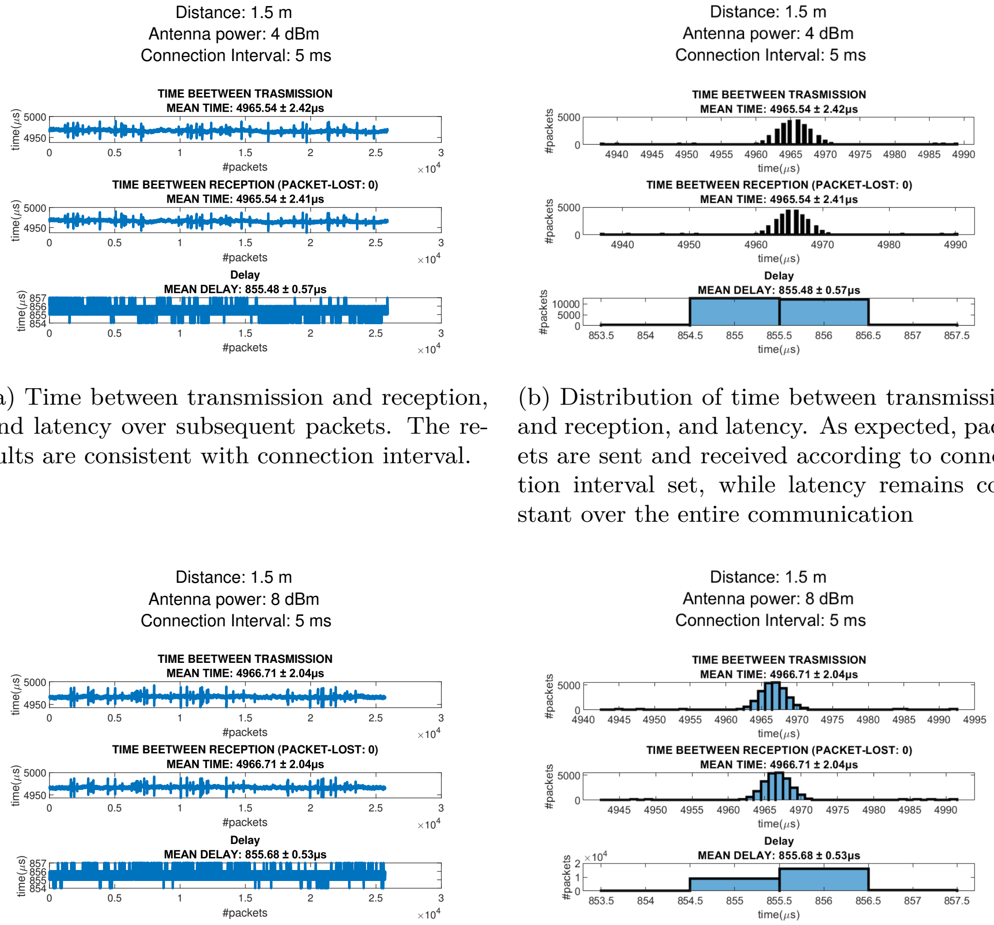
(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.114: Results for 0dBm and 2dBm using 2 Mbit/s physical layer



(a) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

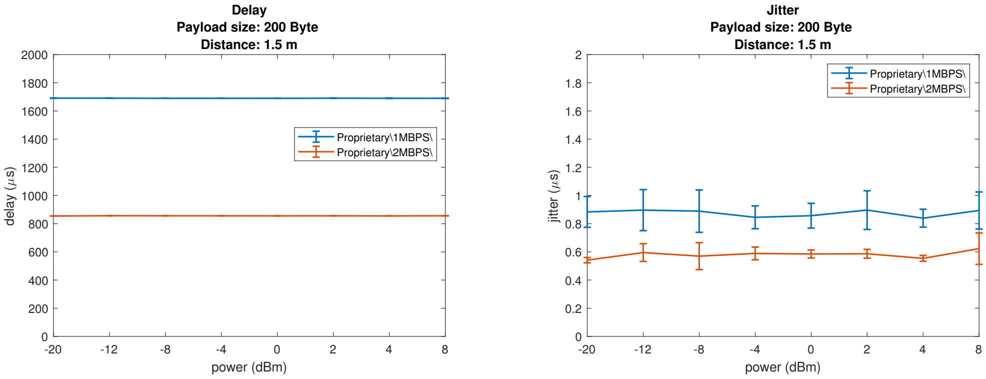
(b) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

(c) Time between transmission and reception, and latency over subsequent packets. The results are consistent with connection interval.

(d) Distribution of time between transmission and reception, and latency. As expected, packets are sent and received according to connection interval set, while latency remains constant over the entire communication

Figure 1.115: Results for 4dBm and 8dBm using 2 Mbit/s physical layer

Latency and jitter stay constant over all trials and antenna power level: overall results are shown in Figure 1.116 and summarize (considering all antenna power levels) in Table 1.13



(a)

(b)

Figure 1.116: Latency and jitter results for 1 and 2 Mbit/s physical layer

Bit rate	Latency [µs]	Jitter [µs]
1 MBPS	1690.20	0.88
2 MBPS	855.71	0.58

Table 1.13: As expected, 1 MBPS latency is twice than 2 MBPS

Throughput has been estimated over different trials at different antenna power levels: results are shown in Figure 1.117

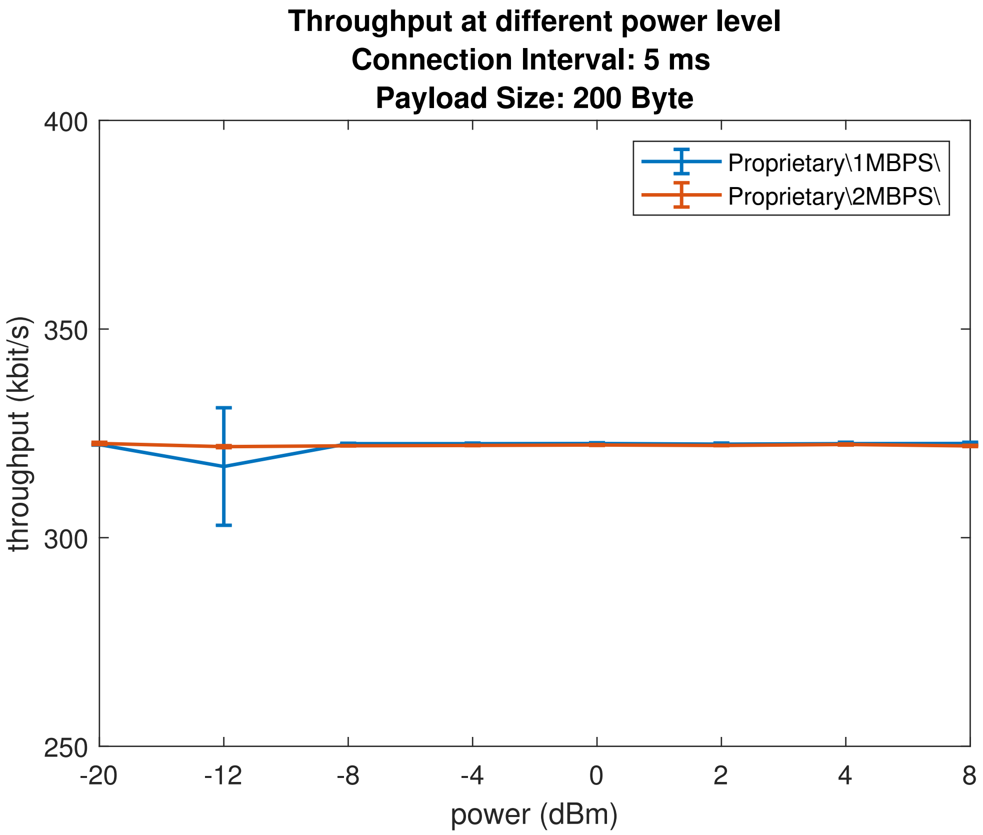
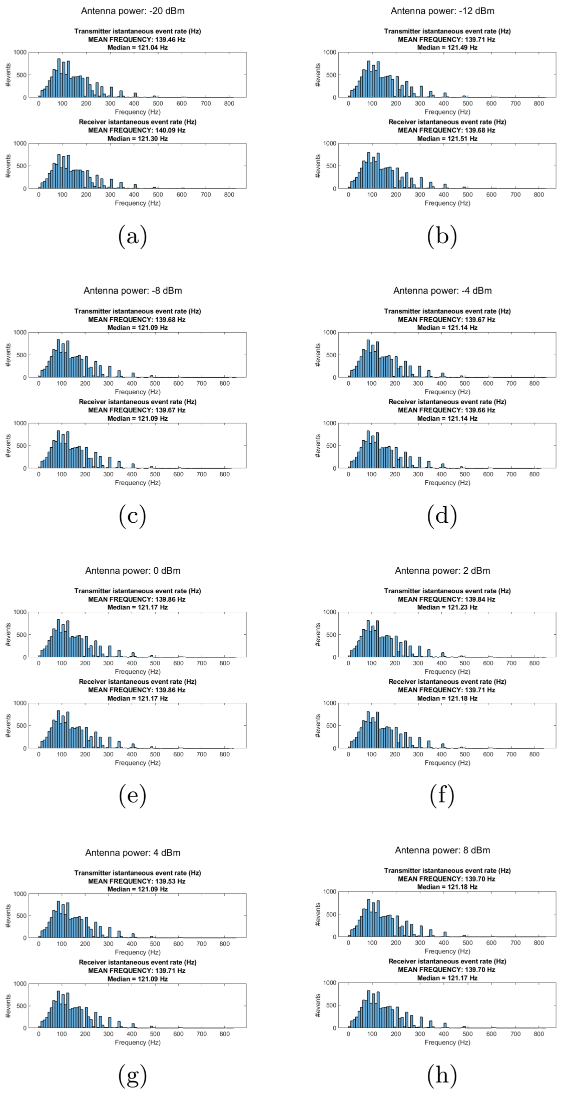


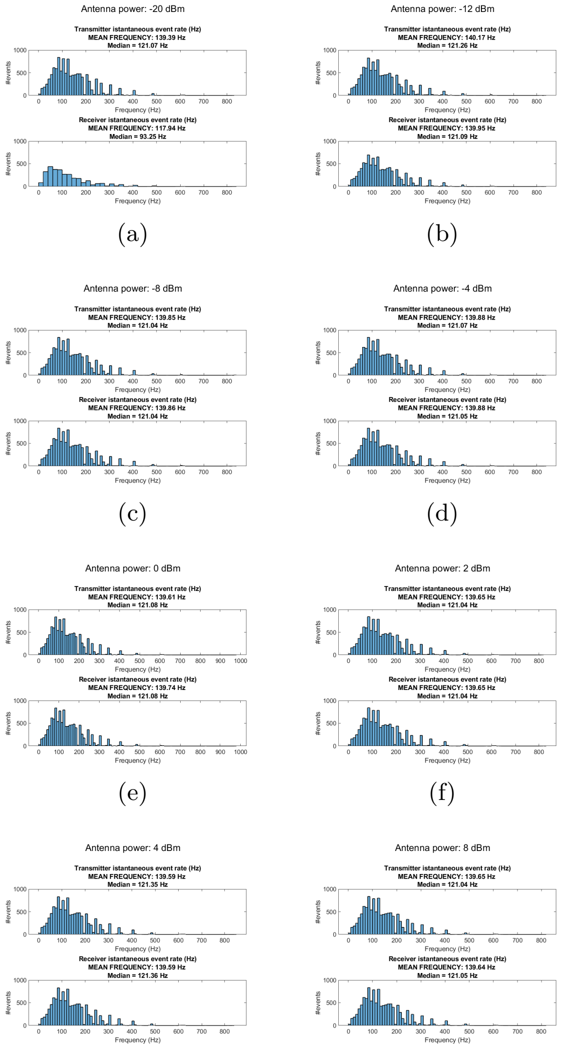
Figure 1.117

Similarity between ATC events has been measured using data plotted in Figure 1.118 (for 1 Mbit/s) and Figure 1.119 (for 2 Mbit/s), which shows events distribution at different antenna power levels.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1.118: Events distribution for 1Mbit/s



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 1.119: Events distribution for 2Mbit/s

Mean frequency and median frequency deviation between tx and rx of transmitted ATC signal, over all trials at fixed power level, is plotted in Figure 1.120, while **cosine and ruzicka similarity and error** of the 2 metrics explained in Chapter 1 are shown in Figure 1.121 and Figure 1.129

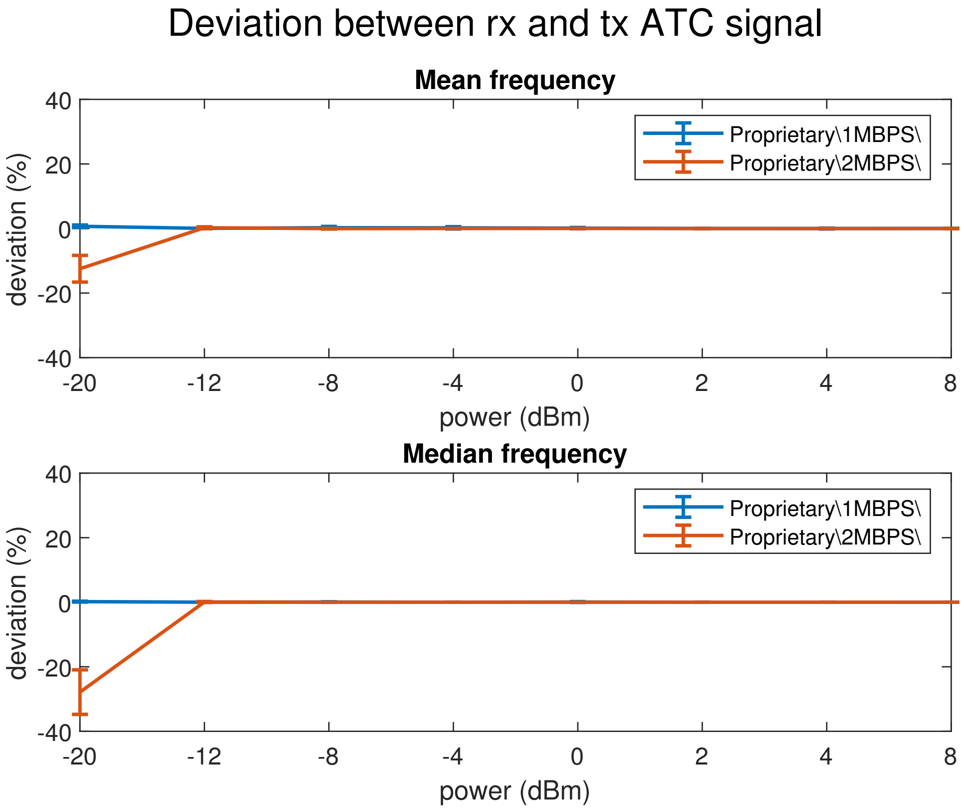
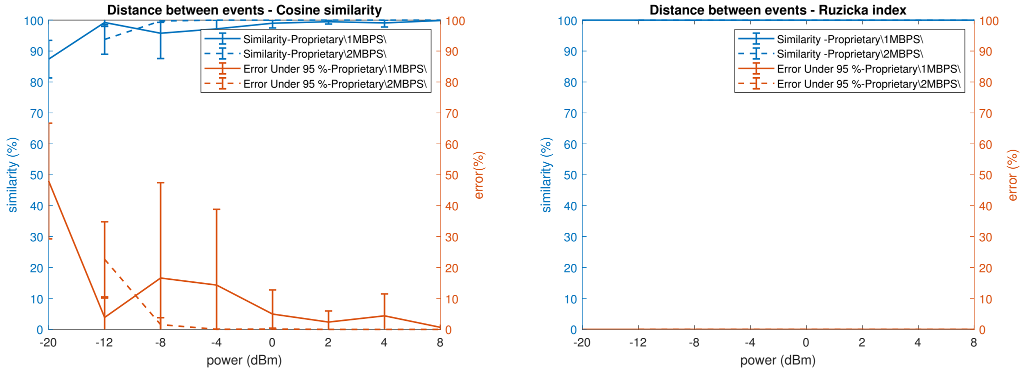
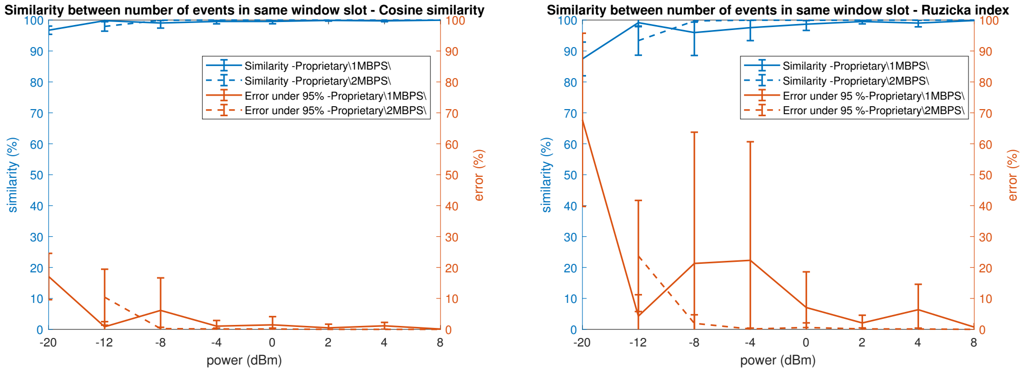


Figure 1.120: Results are closed to zero except for -20 dBm, in which the high volume of events lost reflects in higher deviation



(a) (b)

Figure 1.121: Similarity between distance of consequent events



(a) (b)

Figure 1.122: Similarity between number of events inside corresponding time windows

Reliability is shown in Figure 1.123. For low antenna power levels, a lot of packets are lost: in fact, the implementation of *Proprietary protocol* (for both modulation schemes) does not provide any spread spectrum techniques (as in *IEEE 802.15.4*). In a real implementation, the RF output power should be set a little bit higher, to provide good reliability.

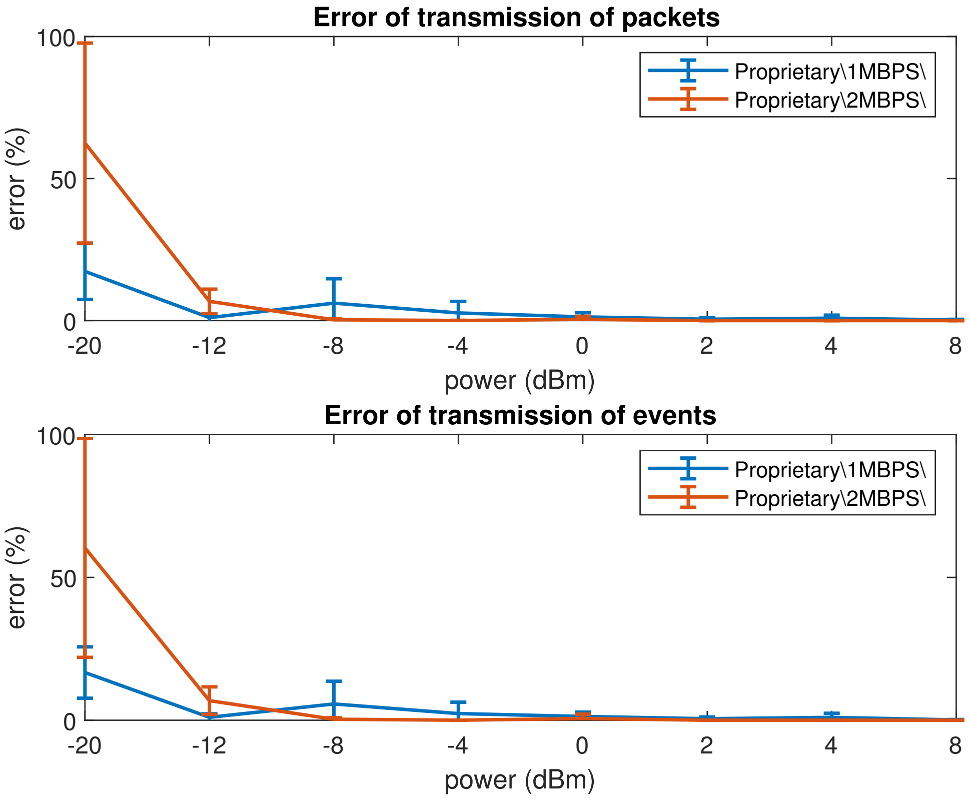
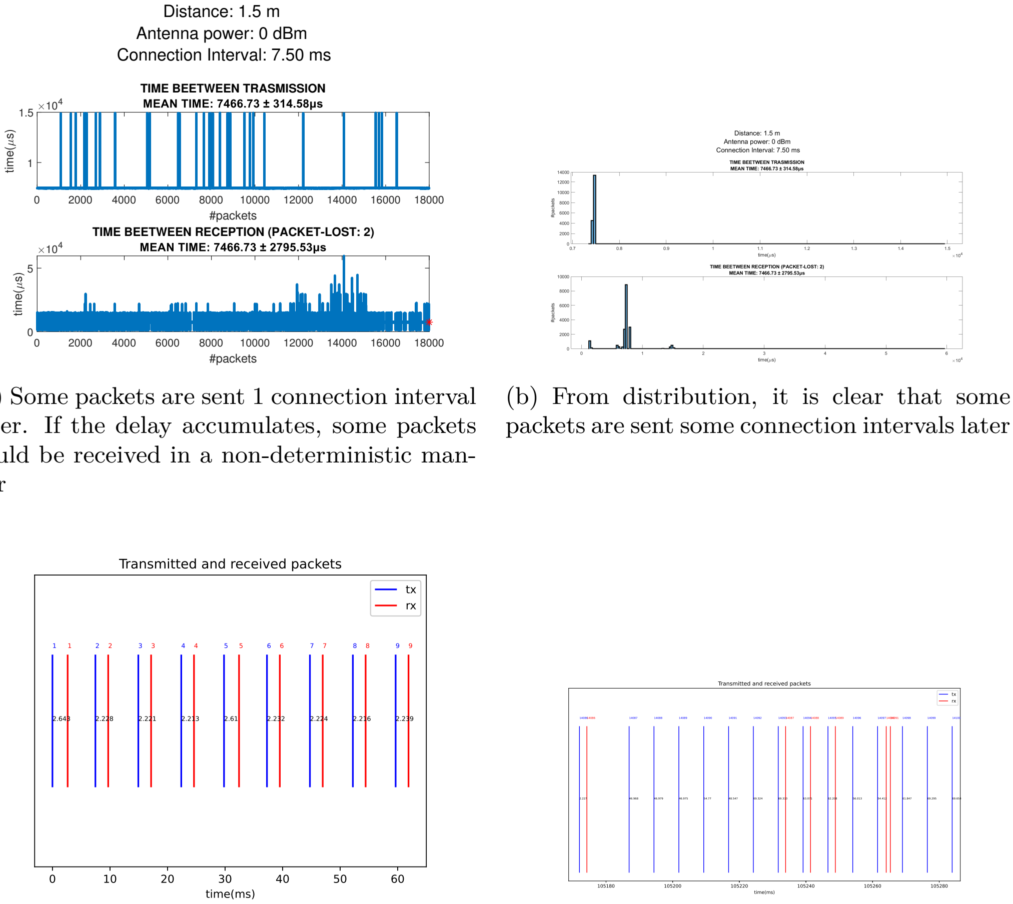


Figure 1.123: Reliability for Proprietary protocol

Bluetooth© Low Energy

The stack used for *BLE* implementation does not allow a total control over the behavior of transmission (as explained in Section 1: data were not always consistent and some results (in particular with low antenna power levels) will miss. In the following, some charts are different than ones presented for other protocols, to try to explain what really happens.
Latency and Jitter are difficult to compute, because the behavior of the transmission is in certain case strange: the following charts will explain what happens. Not all antenna power levels are shown, because in certain situation the data were not consistent and post-processing not possible to be performed. The chats below are related to payload size equal to 200B and physical layer equal to 2Mbit/s



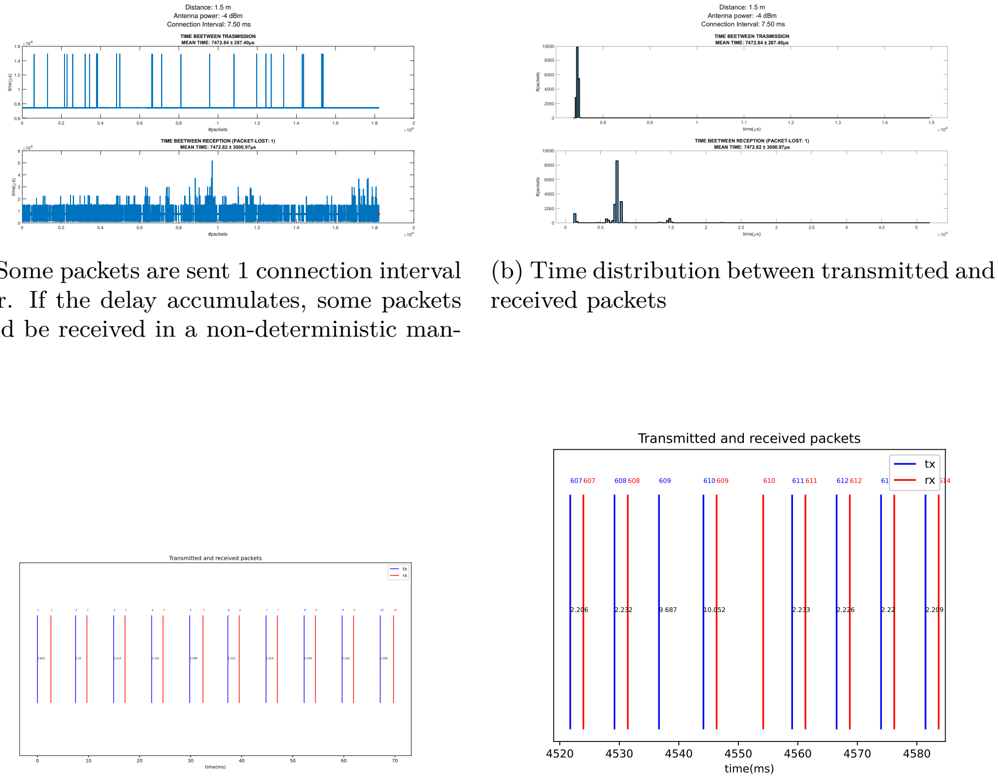
(a) Some packets are sent 1 connection interval later. If the delay accumulates, some packets could be received in a non-deterministic manner

(b) From distribution, it is clear that some packets are sent some connection intervals later

(c) Right transmission: number between line represents time of flight of packet

(d) In some case, one connection interval is lost and real transmission happens 1 connection interval later

Figure 1.124: Latency and jitter for 2Mbit/s



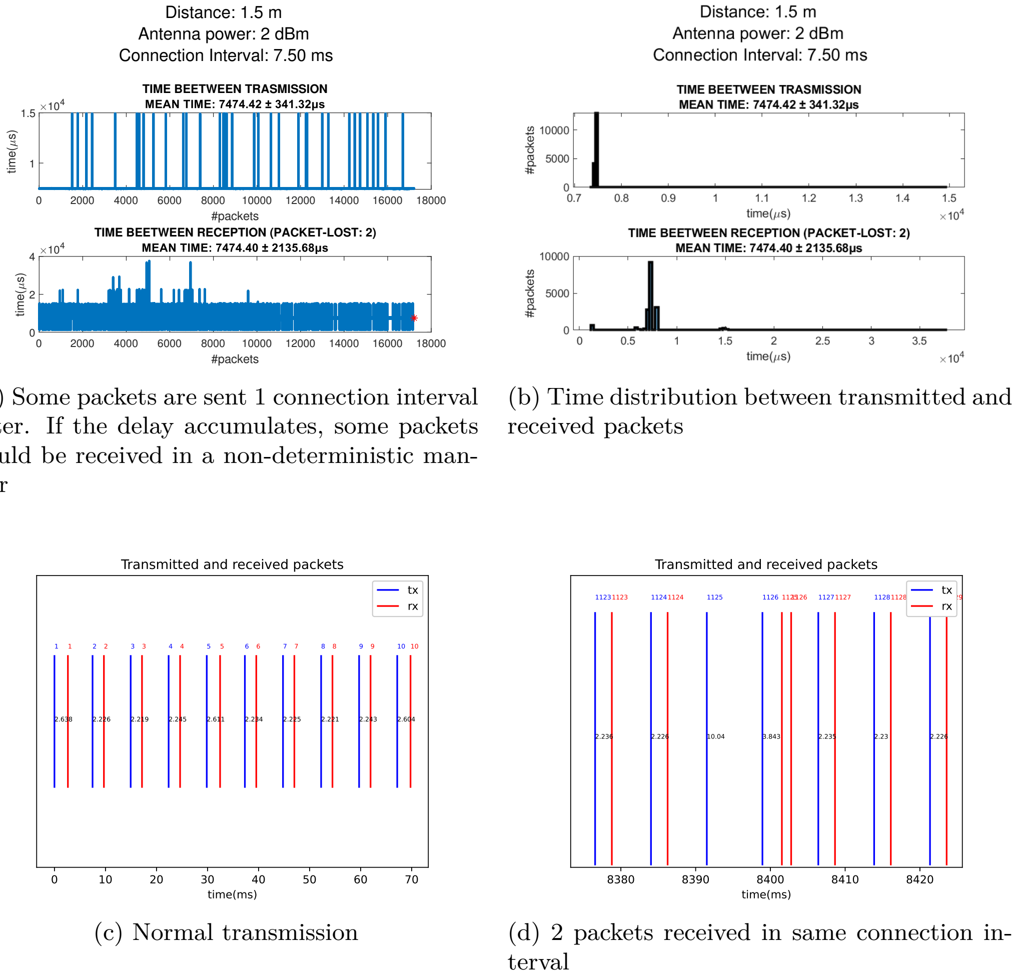
(a) Some packets are sent 1 connection interval later. If the delay accumulates, some packets could be received in a non-deterministic manner

(b) Time distribution between transmitted and received packets

(c) Normal transmission with time of flight plotted

(d) In this case, one packet is lost and it is retransmitted later

Figure 1.125: Latency and jitter for 2Mbit/s



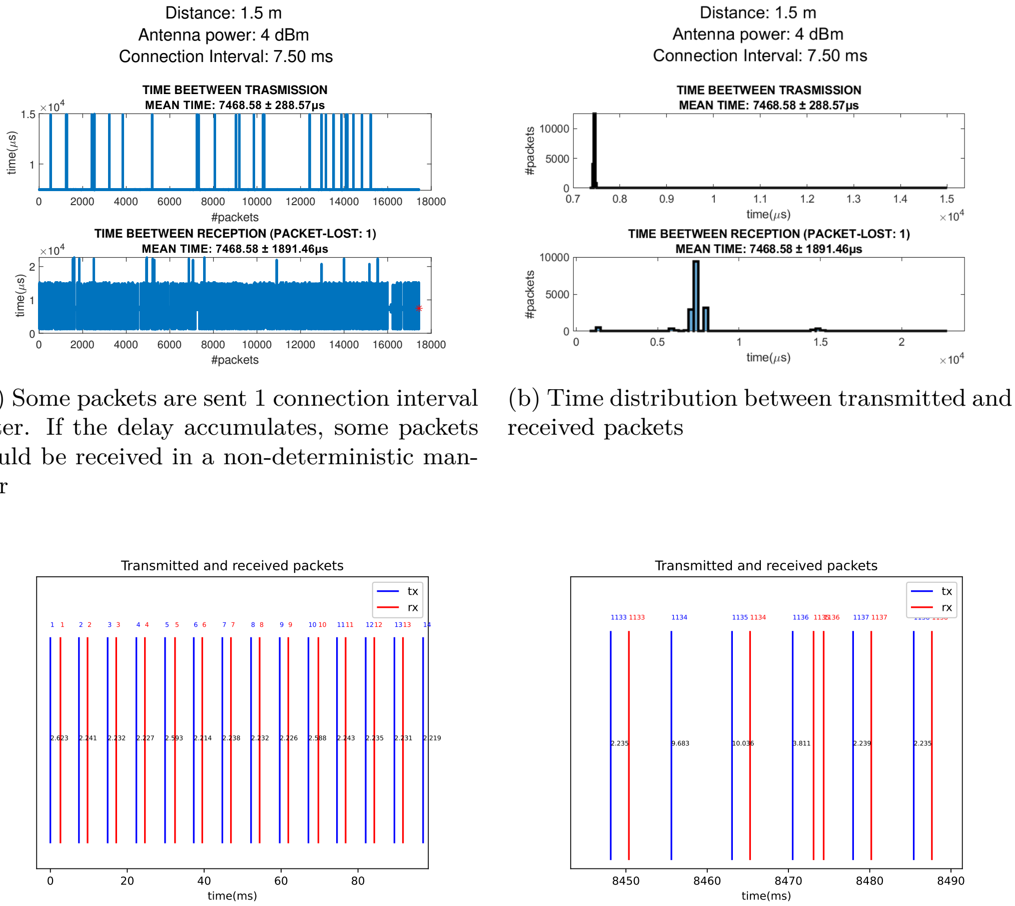
(a) Some packets are sent 1 connection interval later. If the delay accumulates, some packets could be received in a non-deterministic manner

(b) Time distribution between transmitted and received packets

(c) Normal transmission

(d) 2 packets received in same connection interval

Figure 1.126: Latency and jitter for 2Mbit/s



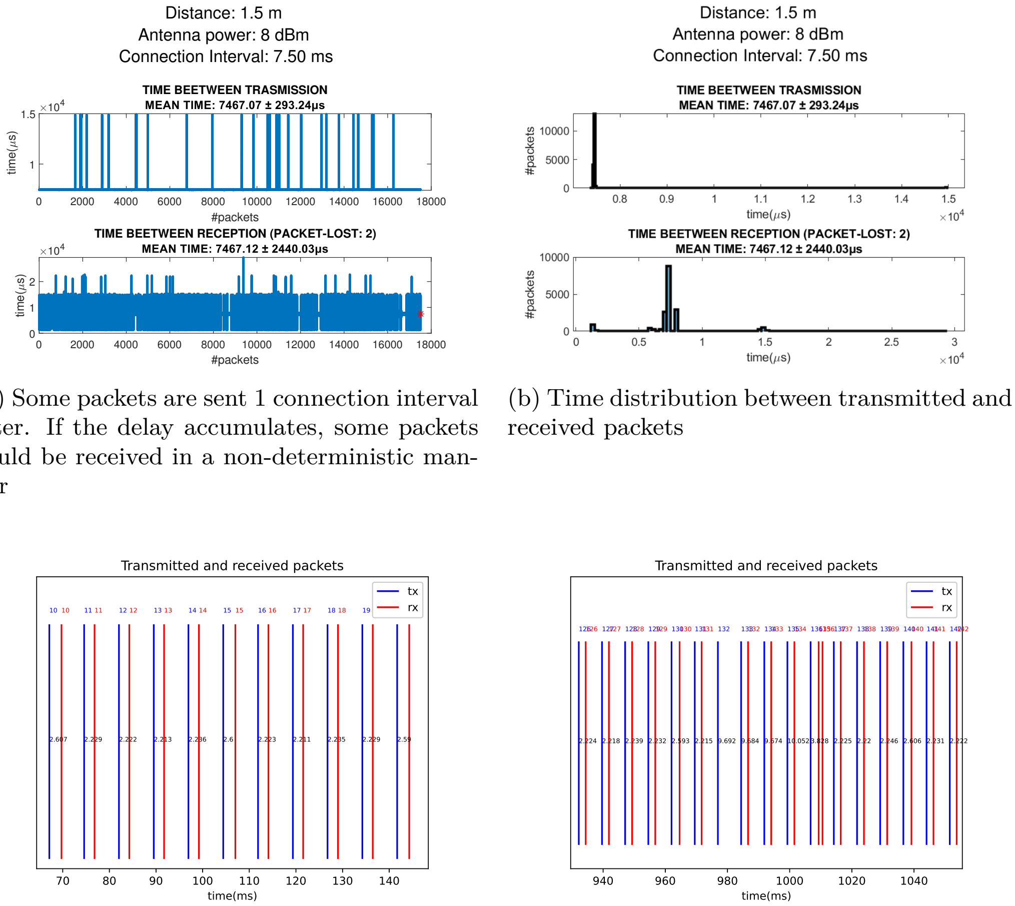
(a) Some packets are sent 1 connection interval later. If the delay accumulates, some packets could be received in a non-deterministic manner

(b) Time distribution between transmitted and received packets

(c) Normal transmission

(d) 2 packets received in same connection interval

Figure 1.127: Latency and jitter for 2Mbit/s



(a) Some packets are sent 1 connection interval later. If the delay accumulates, some packets could be received in a non-deterministic manner

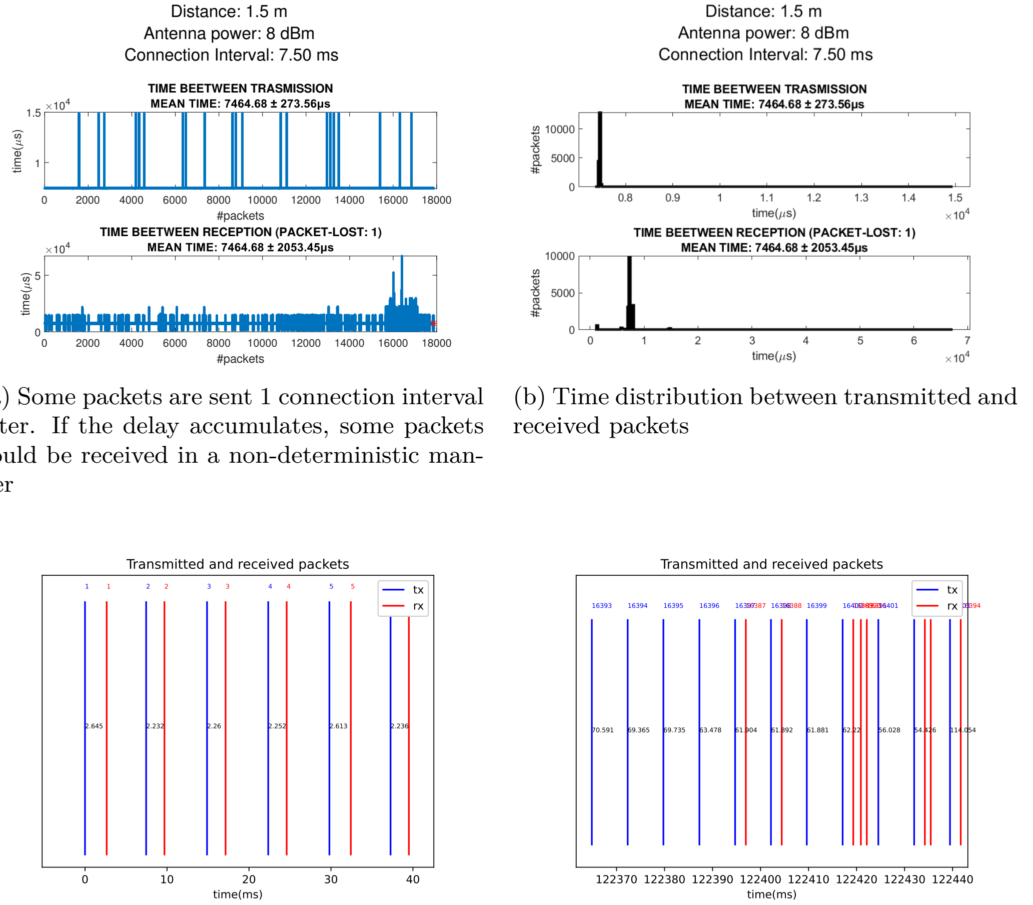
(b) Time distribution between transmitted and received packets

(c) Normal transmission

(d) 2 packets received in same connection interval

Figure 1.128: Latency and jitter for 2Mbit/s

For the sake of completeness some chart for 1Mbit/s are also shown: the results are equal to the one above: thus, only one antenna power level is plotted.



(a) Some packets are sent 1 connection interval later. If the delay accumulates, some packets could be received in a non-deterministic manner

(b) Time distribution between transmitted and received packets

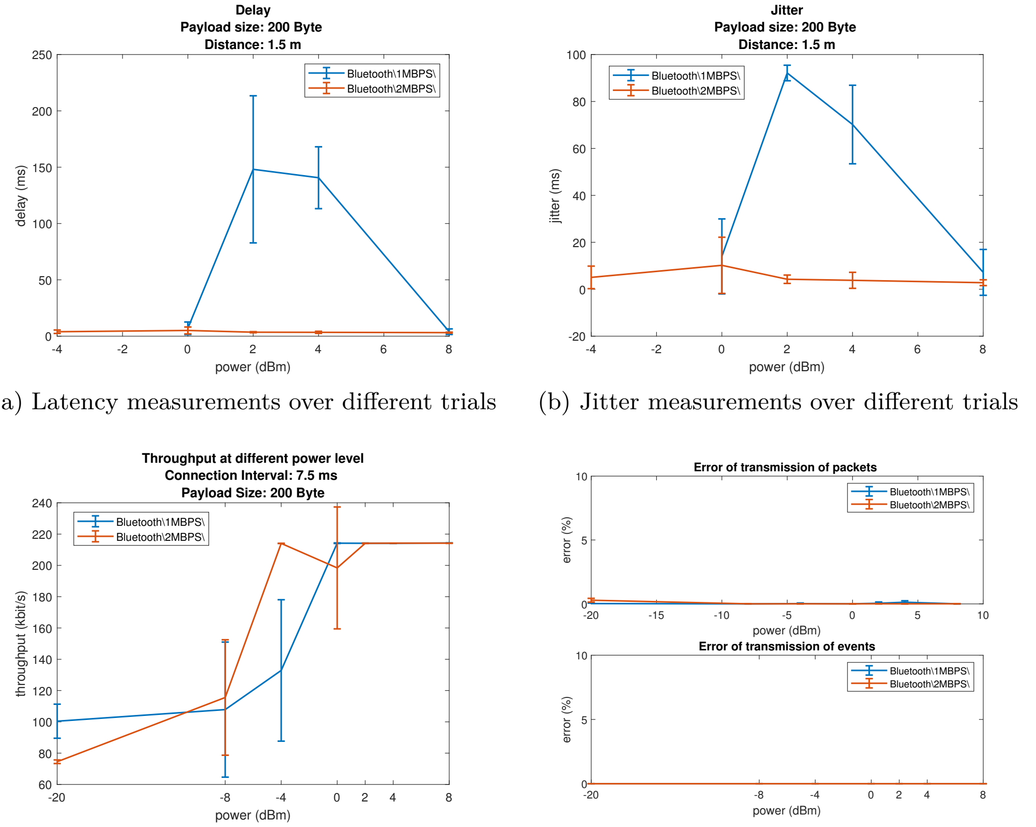
(c) Normal transmission. Everything works fine

(d) Some packets lost and retransmitted later

Figure 1.129: Latency and jitter for 1Mbit/s

Estimate of latency and jitter is difficult to perform, for the strangeness of transmission: with this protocol, reliability stays always high (even with low antenna power levels, Figure 1.130) at the expense of throughput and latency. With *Bluetooth Low Energy*, the packet sent is correctly received, but:

- Not all the packets are sent, as one can see from throughput (Figure 1.130) behavior that should be constant over different antenna power levels.
- Latency between transmission and reception is not deterministic, because transmitter keep to send data until receiver acknowledged it. Thus, one packet could be delivered some connection intervals later.



(a) Latency measurements over different trials

(b) Jitter measurements over different trials

(c) Throughput is sacrificed in favor of reliability for low antenna power levels

(d) Perfect reliability even with low antenna power levels: caution must be taken, because not all the packets are sent (mechanism that allows for re-transmission of packets until they are not correctly received)

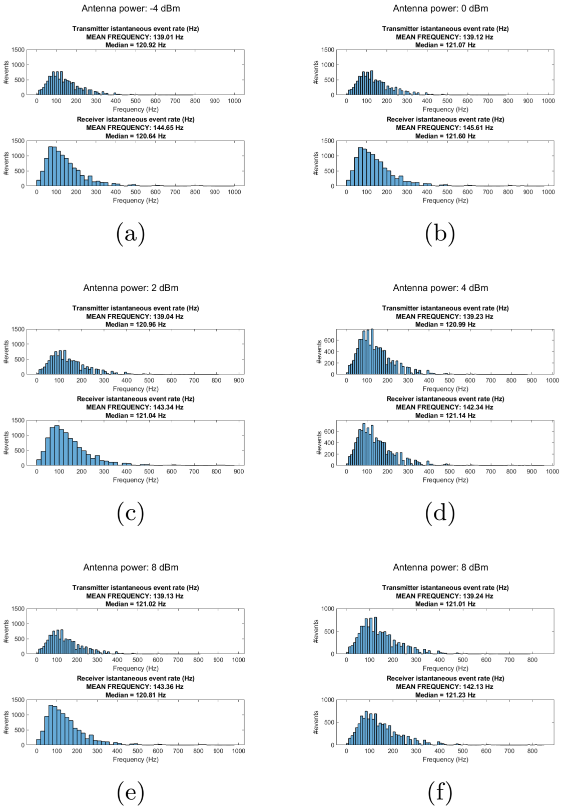
Figure 1.130: Bluetooth© Low Energy results

To recap, latency and jitter for BLE 1Mbit/$sand$2Mbit/s is listed in Table 1.14

Bit rate	Latency [ms]	Jitter [ms]
1 MBPS	74.93	45.87
2 MBPS	3.83	5.22

Table 1.14: Mean value reflects the BLE behavior over entire transmission and power level. As one can see in charts above, when the packet is correctly sent at first shot, the latency is around 2ms for both modulation schemes

The non-total control of transmission (as explained in Section 1), together with the fact that some packets are sent some connection interval later, brings ATC reconstruction performance at low levels, with respect to other protocol. Events distribution at transmitter and receiver side is plotted in Figure 1.131



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1.131: Events distribution for 2Mbit/s. **f)** chart refers to 1Mbit/s modulation

Mean frequency and median frequency deviation between tx and rx of transmitted ATC signal, over all trials at fixed power level is plotted in Figure 1.132, while **cosine and ruzicka similarity and error** of the 2 metrics explained in Chapter 1 are shown in Figure 1.133 and 1.134

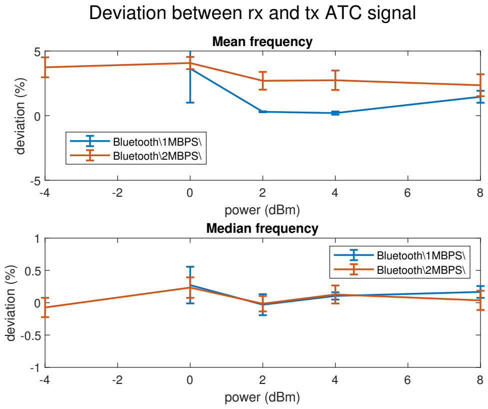


Figure 1.132: Frequency deviation for both modulation schemes. The fact that some packets are sent some connection intervals later, slightly distorts frequency content of ATC signal

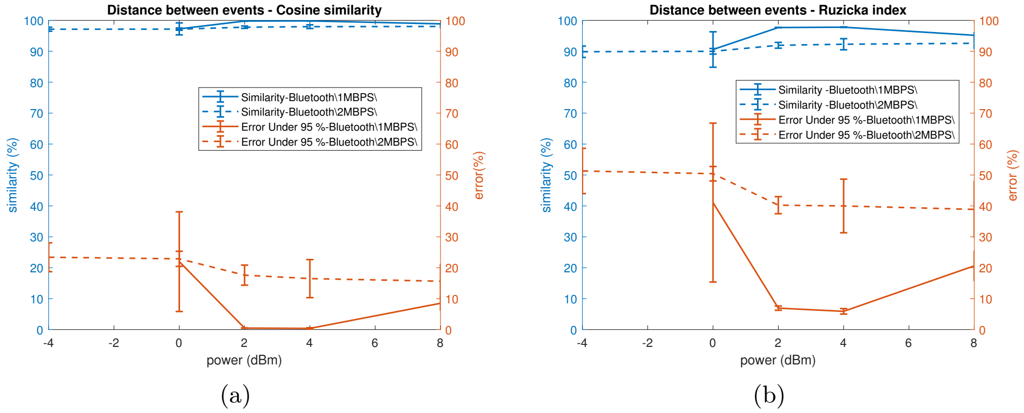
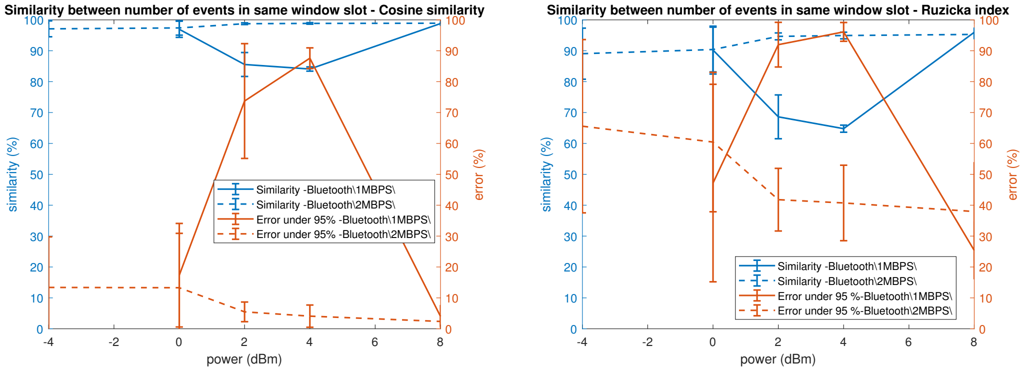


Figure 1.133: Similarity between distance of consequent events over all trials. For both indexes, high values of error are reached



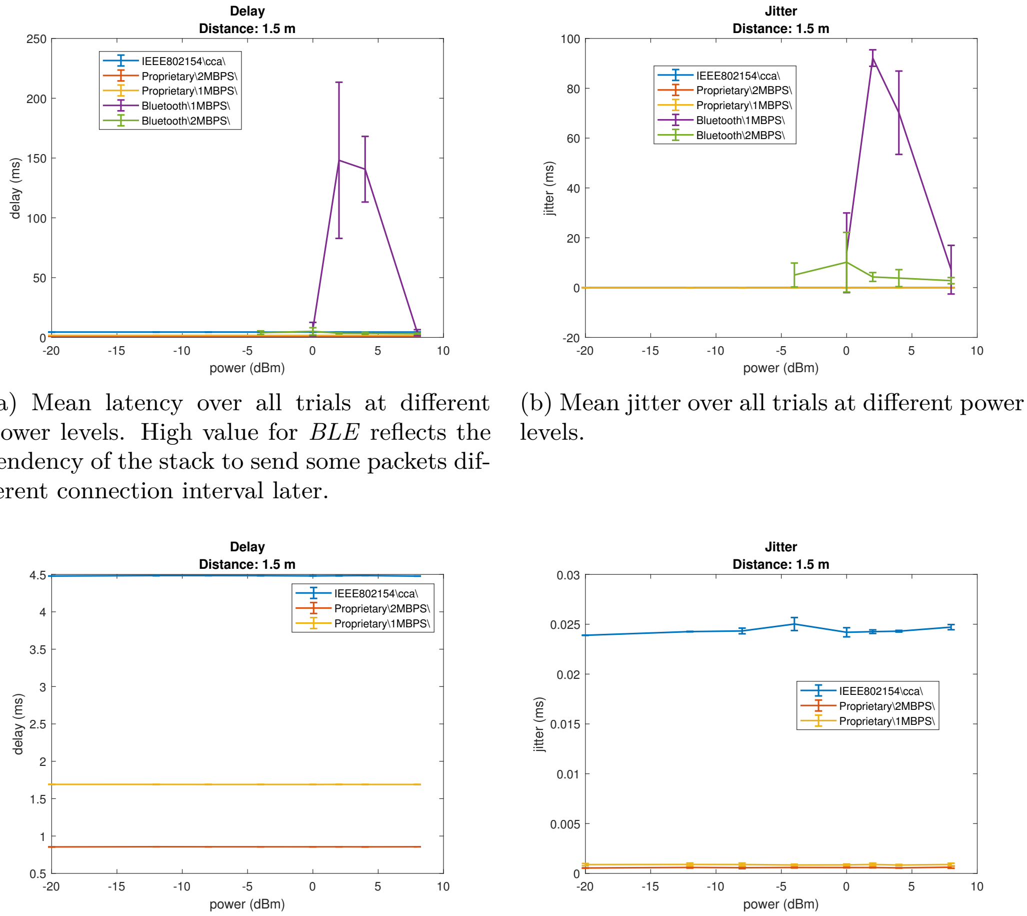
(a)

(b)

Figure 1.134: Similarity between number of events in corresponding time windows over all trials

Comparison

In this section, different tested protocols are compared with respect to different performance metrics. For *Bluetooth Low Energy*, some results are missing at low antenna power levels, for reasons explained in Section 1. **Latency, jitter, throughput and transmission error** are shown in Figure 1.135 and 1.136. All protocols behave well in term of reliability above $power_{antenna} = 0\text{dB}m$, reaching a transmission error around 0%. The throughput is suitable for the application up to 10 channels for *Proprietary and Bluetooth Low Energy* and 5 channels for *IEEE 802.15.4*, due to difference in maximum payload size (constraint imposed by the standards). For *BLE*, at low antenna power levels the reliability of transmission (Figure 1.135-d) is preferred to total throughput: in this way all packets sent are received but not all the packets are sent, losing throughput at transmitter side.
Latency and jitter are far better for *Proprietary protocol*, allowing for a maximum flexibility and customizability, reaching a minimum value of 850 ± 1µs: for *BLE* the stack used does not allow the perfect synchronization between the connection interval and transmission of packets, resulting in some packet sent different connection intervals later; *IEEE 802.15.4* shows stable latency and jitter (4480 ± 24µs), the former higher than *Proprietary protocol* for the modulation schemes used, but still suitable for real-time applications.



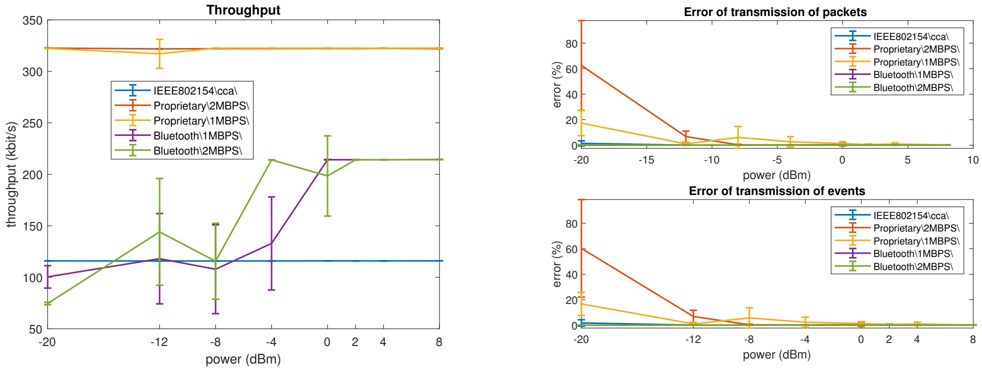
(a) Mean latency over all trials at different power levels. High value for *BLE* reflects the tendency of the stack to send some packets different connection interval later.

(b) Mean jitter over all trials at different power levels.

(c) Zoom on **a**) chart, in order to appreciate the low latency of *Proprietary* and *IEEE 802.15.4* protocols

(d) Zoom on **d**) chart,in order to appreciate the high stability of latency (low jitter) of *Proprietary* and *IEEE 802.15.4* protocols

Figure 1.135: Latency and jitter comparison between protocols



(a) Connection interval for *Proprietary protocol* is set to 5ms, for *IEEE 802.15.4* to 7ms and for *BLE* 7.5ms. The tendency of *BLE* is to "sacrifice" throughput in favor of reliability for low antenna power levels.

(b) Error of transmission of packets and events, expressed as percentage of the total number of packets and events. Two different plots have been performed, because not all the packets lost contain some events: however, the tendency of the two graphs is similar. Direct sequence spread spectrum (*IEEE 802.15.4*) and internal mechanism of *BLE* reflects in lower transmission error also at low antenna power levels.

Figure 1.136: Throughput and transmission error comparison between protocols

ATC reconstruction is evaluated considering both frequency deviation (Figure 1.137) and similarity(Figure 1.138). The frequency content remains stable for higher antenna power levels and all protocols shows good performance, around 0% of deviation between ground truth. The overall similarity, calculated as explained in Chapter 1, shows an increasing trend, for all protocols, with antenna power level: for *BLE*, similarity remains under 80%, while it reaches 99% for other protocols. A total mean is performed on similarity in order to characterize the protocol's similarity with only 1 number (Figure 1.138-d): best protocol appears to be the ***Proprietary protocol***, using 2Mbit/s physical layer.

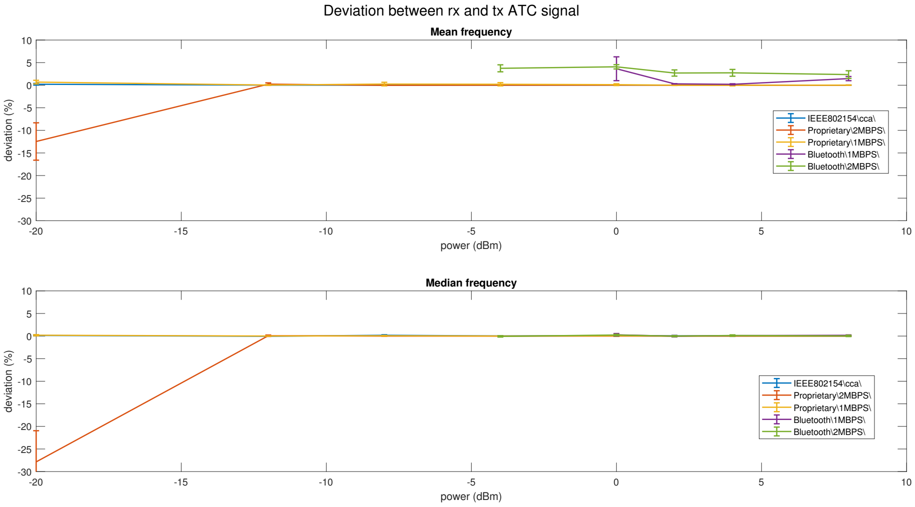
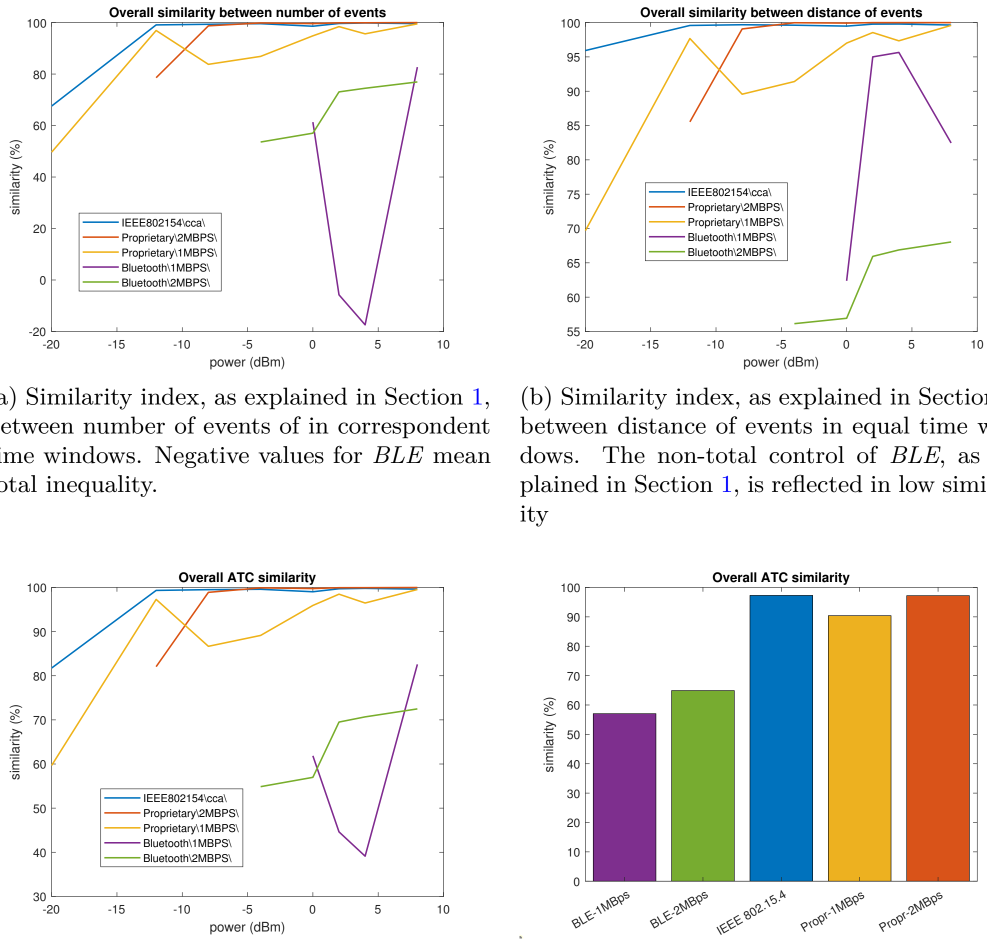


Figure 1.137: Frequency Deviation between transmitted and received ATC signal. All protocols behave well in terms of median frequency, except for *Proprietary-2MBps*, where the high number of packets lost reflects in higher deviation. For mean frequency, *BLE* presents the worst performance for the non-total control over transmission, as explained in Section 1. For all protocols, however, the frequency content remains more or less stable, except in that case where lot of packets are lost.



(a) Similarity index, as explained in Section 1, between number of events of in correspondent time windows. Negative values for *BLE* mean total inequality.

(b) Similarity index, as explained in Section 1, between distance of events in equal time windows. The non-total control of *BLE*, as explained in Section 1, is reflected in low similarity

(c) Mean value between **a and b** chart. Final index used to assess ATC similarity of different protocols at fixed antenna power level.

(d) Mean value of **c** chart: useful to characterize ATC reconstruction with only 1 number

Figure 1.138: ATC similarity comparison between protocols

To compare different protocols in a possible real implementation, value of **8 dBm** is chosen for *antenna power level*, who gives best performance in terms of packets/events lost as well as ATC similarity. Throughput is not shown in Table 1.15 because it is not a key parameter in the choice of best protocol, considering that all protocols' throughput is suitable.

Considering both Table 1.15 and Table 1.11, at maximum antenna power level, best protocol in term both of energy consumption, latency and jitter is **Proprietary protocol 2 MBPS**

Protocol	Payload [B]	Latency [µs]	Jitter [µs]	Conn Interval [ms]	Total latency [ms]	Packets lost [%]	ATC similarity[7] [%]
Proprietary 2MBPS	200	856.10	0.62	7	7.86	0	99.99
Proprietary 1MBPS	200	1690	0.89	7	8.69	0	99.54
BLE 2MBPS	200	3157	2789	7.5	10.6	0	72.49
BLE 1MBPS	200	3981	7186	7.5	11.4	0	82.58
IEEE 802.15.4 CCA	100	4478	25	7	11.45	0.06	99.63

Table 1.15: Total latency is calculated as $connection\_interval + latency$, referred to total latency of one event between transmitter and receiver. High value of jitter for both BLE implementations reflects the non-total control over the transmission of data: in real world the total latency for this technology is a multiple of connection interval

Conclusion and future works

In this thesis work, a simple wireless body area network, composed only of one transmitting and one receiving node, has been investigated.
Acquisition of *sEMG* is made using custom PCB, with standard analog front-end comprising also a voltage comparator: in this way, every time the biological signal overcomes a fixed threshold, a digital event is made available, thus bringing data size, PCB complexity and power consumption to the lowest. A time window (equal to connection interval) is used to queue different events inside a packet, transmitted at the end of it: in this way there isn't the loss of any event, allowing transmission of all *sEMG-ATC* bandwidth.
Different wireless protocols have been studied, implemented, and tested in order to choose the best one considering different parameters:

- Power consumption: it's important, in wearable electronics, to have low power consumption to extend battery's lifetime
- Total latency: it is intended as the latency between transmission and reception, considering both transmission latency, time of flight (negligible) and connection interval set. To have a real-time system, it must be low, in the order of ms
- Jitter: variability of latency. To assure a good transmission, latency should be constant over time, assuring low jitter. A high value of jitter brings to a non-total control over the transmission
- Throughput: to assure high exchange of data and to allow system for more channels
- Reliability: intended both as packets lost ratio and goodness of ATC reconstruction (to not lose EMG information)

Bluetooth© Low Energy shows good performance in terms of energy consumption, both for transmitter and receiver side, consuming (maximum) 62.14µC for one single transmission. The latency (and therefore jitter) is not constant over time: some packets are sent some connection intervals later, bringing to a non-total control on transmitted data. For low antenna power levels, throughput is reduced in favor of packets lost ratio: the latter remains around 0%, meaning the all packets sent are received, but not all packets are sent, reducing throughput at the transmitter side. In a real implementation, the antenna power level must be chosen carefully, considering distance between transmitter and receiver. Using a payload size of 200B a 10-channels system could be created.
IEEE 802.15.4, implemented both with and without Clear Channel Assessment (CCA), performs poorly in terms of energy consumption (124.85µC for one transmission) due to a

lower bit rate used in physical layer. Latency and jitter stay constant over time, allowing for a perfect control of transmitted data: even if the total latency is around 11.45ms, it meets the requirements for a real-time application . Throughput is limited by maximum payload size (117B allowing for a 5-channels system), while transmission reliability is high even with low antenna power levels, thanks to spread spectrum technique in physical layer.
Proprietary protocol performs good in energy consumption, with maximum charge absorbed in one transmission event equal to 58.77µC. Latency and jitter are perfectly constant over time, with few time needed (in the order of µs) for pre-processing the packet: in this way, total latency is strictly dependent on connection interval. Reliability is poor for low antenna power levels, for the huge number of packets lost (no retransmission or spread spectrum tecniques are used), but it becomes excellent as soon as the antenna power is tuned on tx/rx distance. Throughput can be finely chosen, tuning both connection interval and payload size: a 10-channels system could be implemented.
Considering all parameters listed above, the best protocol for this application turns out to be ***Proprietary protocol 2MBPS***. If the transmission must be standardized, a trade-off between latency/jitter and power consumption must be applied: if the battery's lifetime is preferred over latency **Bluetooth Low Energy** is the right choice, while if total control over transmission is preferred, one should choose **IEEE 802.15.4**.

In future works, the receiver for *Proprietary protocol* could be optimized, considering some sort of duty-cycling and synchronization with the transmitter. In this way, both side of system could be optimized towards a low power device. Moreover, a more precise study correlating the antenna power level with distance could be conducted, optimizing in this way the power consumption considering the coverage needed. In the end, the entire *Proprietary protocol* could be changed towards an event-driven transmission, transmitting the event as soon as happens.
Furthermore, the entire system could be replicated using new technology, IR-UWB, which could offer great improvements both in throughput, power consumption and jitter.
Finally, the entire system could be used in hand gesture recognition system, for rehabilitation or robotics purpose.

Bibliography

[1] Bluetooth Special Interest Group (SIG). *Bluetooth Core Specification.* v5.2. URL: https://www.bluetooth.com/specifications/bluetooth-core-specification/.

[2] D'Amico S. et al. "Ultra Wide Band in Medical Applications". In: *Advances in Biomedical Sensing, Measurements, Instrumentation and Systems.* 2010.

[3] Purves D et al. "The Motor Unit". In: *Neuroscience. 2nd edition.* 2001. URL: https://www.ncbi.nlm.nih.gov/books/NBK10874/.

[4] A. Baba and M. J. Burke. "Measurement of the electrical properties of ungelled ECG electrodes". In: 2008.

[5] G. Biagetti et al. "Wireless surface electromyograph and electrocardiograph system on 802.15.4". In: *IEEE Transactions on Consumer Electronics* 62.3 (2016), pp. 258–266. DOI: 10.1109/TCE.2016.7613192.

[6] Falascehtti L. et al. Biagetti G. Crippa P. "A Multi-Channel Electromyography, Electrocardiography and Inertial Wireless Sensor Module Using Bluetooth Low-Energy". In: *Electronics* 9 (2020), p. 934. DOI: https://doi.org/10.3390/electronics9060934. URL: https://www.mdpi.com/2079-9292/9/6/934.

[7] D. Brunelli et al. "Design Considerations for Wireless Acquisition of Multichannel sEMG Signals in Prosthetic Hand Control". In: *IEEE Sensors Journal* 16.23 (2016), pp. 8338–8347. DOI: 10.1109/JSEN.2016.2596712.

[8] Benjamin Caballero. "Skeletal Muscle". In: *Encyclopedia of Human Nutrition (Third Edition).* Third Edition. Waltham: Academic Press, 2013, pp. 193–199. ISBN: 978-0-12-384885-7. DOI: https://doi.org/10.1016/B978-0-12-375083-9.00188-4.

[9] G. L. Cerone, A. Botter, and M. Gazzoni. "A Modular, Smart, and Wearable System for High Density sEMG Detection". In: *IEEE Transactions on Biomedical Engineering* 66.12 (2019), pp. 3371–3380. DOI: 10.1109/TBME.2019.2904398.

[10] G. L. Cerone and M. Gazzoni. "A wireless, minaturized multi-channel sEMG acquisition system for use in dynamic tasks". In: *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS).* 2017, pp. 1–4. DOI: 10.1109/BIOCAS.2017.8325129.

[11] Ka Xiong Charand. *Action Potential explained.* URL: http://hyperphysics.phy-astr.gsu.edu/hbase/Biology/actpot.html.

[12] Austin Community Colleger. *Myofibril structure.* [Section 7, Sarcomere]. 2020. URL: https://www.austincc.edu/apreview/PhysText/Muscle.html.

[13] M. Crepaldi, P. M. Ros, and D. Demarchi. "A 130 nm CMOS IR-UWB receiver based on baseband cross-phase detection". In: *2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2014, pp. 814–817. DOI: 10.1109/ICECS.2014.7050110.

[14] M. Crepaldi et al. "A non-coherent IR-UWB receiver for high sensitivity short distance estimation". In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2014, pp. 1905–1908. DOI: 10.1109/ISCAS.2014.6865532.

[15] M. Crepaldi et al. "A Very Low-Complexity 0.3–4.4 GHz 0.004 mm^2 All-Digital Ultra-Wide-Band Pulsed Transmitter for Energy Detection Receivers". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 59.10 (2012), pp. 2443–2455. DOI: 10.1109/TCSI.2012.2188954.

[16] I. et al. Čuljak. "Wireless Body Sensor Communication Systems Based on UWB and IBC Technologies: State-of-the-Art and Open Challenges". In: *Sensors* (2020). DOI: https://doi.org/10.3390/s20123587.

[17] M. Daoud et al. "Design of an OOK IR-UWB front-end for biomedical devices". In: *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2016, pp. 1–4. DOI: 10.1109/MWSCAS.2016.7870080.

[18] Institute of Electrical and Electronics Engineers(IEEE). *IEEE Std 802.15.4-2006*. v2006. 2006. URL: https://standards.ieee.org/standard/802_15_4-2020.html.

[19] Merletti R. Farina D. "Biophysics of the Generation of EMG Signals". In: *Surface Electromyography : Physiology, Engineering, and Applications*. DOI: https://doi.org/10.1002/9781119082934.ch02.

[20] Mohammad Ghamari et al. "A Survey on Wireless Body Area Networks for eHealthcare Systems in Residential Environments". In: *Sensors* 16.6 (2016). ISSN: 1424-8220. DOI: 10.3390/s16060831. URL: https://www.mdpi.com/1424-8220/16/6/831.

[21] D. A. F. Guzman et al. "Very low power event-based surface EMG acquisition system with off-the-shelf components". In: *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. 2017, pp. 1–4. DOI: 10.1109/BIOCAS.2017.8325152.

[22] Khalid Hasan et al. "A comprehensive review of wireless body area network". In: *Journal of Network and Computer Applications* 143 (2019), pp. 178–198. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2019.06.016. URL: http://www.sciencedirect.com/science/article/pii/S1084804519302218.

[23] Biedermann Heinz-J. Hemingway Monica A. "Electromyographic Recordings of Paraspinal Muscles: Variations Related to Subcutaneous Tissue Thickness". In: *Biofeedback and Self-Regulation, Vol. 20, No. 1* (1995).

[24] Stephanie Hunter. *Motor Neuron Disease*. 2019. URL: https://www.news-medical.net/health/Motor-Neuron-Disease-Can-Dietary-Supplements-Help.aspx.

[25] et al. Koubaa A. Alves M. *IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview*. 2005. URL: https://www.researchgate.net/publication/253418115_IEEE_802154_for_Wireless_Sensor_Networks_A_Technical_Overview.

[26] LumenWayMaker. *Sarcomere*. 2019. URL: https://courses.lumenlearning.com/wm-biology2/chapter/sliding-filament-model-of-contraction/.

[27] T.M. M. Vieira M.A. Cavalcanti Garcia. “Surface electromyography: Why, when and how to use it”. In: *Revista Andaluza de Medicina del Deporte* 4 (2011), pp. 17–28. ISSN: 1084-8045. URL: https://www.elsevier.es/es-revista-revista-andaluza-medicina-del-deporte-284-articulo-surface-electromyography-why-when-how-X1888754611201253.

[28] T. Matić, L. Šneler, and M. Herceg. “An Energy Efficient Multi-User Asynchronous Wireless Transmitter for Biomedical Signal Acquisition”. In: *IEEE Transactions on Biomedical Circuits and Systems* 13.4 (2019), pp. 619–630. DOI: 10.1109/TBCAS.2019.2917690.

[29] Betty McGuire. *Organization of Skeletal Muscles*. URL: https://www.coursehero.com/sg/anatomy-and-physiology/organization-of-skeletal-muscles/.

[30] Farina D. Merletti R. “Analysis of intramuscular electromyogram signals”. In: *The Royal Society* (2009).

[31] Muceli S. Merletti R. “Tutorial. Surface EMG detection in space and time: Best practices”. In: *Journal of Electromyography and Kinesiology* (2019).

[32] Luca Mesin. *Introduction to biomedical signal processing*. 2017.

[33] Zach Michel. *Maximizing BLE Throughput Part 3: Data Length Extension (DLE)*. 2020. URL: https://punchthrough.com/maximizing-ble-throughput-part-3-data-length-extension-dle-2/.

[34] A. Mongardi et al. “A Low-Power Embedded System for Real-Time sEMG based Event-Driven Gesture Recognition”. In: *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2019, pp. 65–68. DOI: 10.1109/ICECS46596.2019.8964944.

[35] S. Movassaghi et al. “Wireless Body Area Networks: A Survey”. In: *IEEE Communications Surveys Tutorials* 16.3 (2014), pp. 1658–1686. DOI: 10.1109/SURV.2013.121313.00064.

[36] Nordic. *Bluetooth Low Energy protocol stack*. [s140]. 2021. URL: https://www.nordicsemi.com/Software-and-tools/Software/S140.

[37] *Nordic Semiconductor- Development Kit*. 2020. URL: https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK.

[38] *Nordic Semiconductor-Dongle*. 2020. URL: https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF52840-Dongle.

[39] *Nordic Semiconductor-nRF52*. 2020. URL: https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52-DK.

[40] *Nordic Semiconductor-nRF52840-DK product brief*. 2020. URL: https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK.

[41] P. M. Ros et al. “A wireless address-event representation system for ATC-based multi-channel force wireless transmission”. In: *5th IEEE International Workshop on Advances in Sensors and Interfaces IWASI*. 2013, pp. 51–56. DOI: 10.1109/IWASI.2013.6576061.

[42] Fabio Rossi. “Low Power System for Event-Driven Control of Functional Electrical Stimulation”. Master’s Degree. Politecnico di Torino, 2017.

[43] Fabio Rossi et al. “Embedded Bio-Mimetic System for Functional Electrical Stimulation Controlled by Event-Driven sEMG”. In: *Sensors* 20.5 (2020), p. 1535. ISSN: 1424-8220. DOI: 10.3390/s20051535. URL: http://dx.doi.org/10.3390/s20051535.

[44] Motto Ros P. et al. Rossi F. “Embedded Bio-Mimetic System for Functional Electrical Stimulation Controlled by Event-Driven sEMG”. In: *Sensors(Basel)* (2020). DOI: https://dx.doi.org/10.3390%2Fs20051535.

[45] S. Sapienza et al. “On-Line Event-Driven Hand Gesture Recognition Based on Surface Electromyographic Signals”. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2018, pp. 1–5. DOI: 10.1109/ISCAS.2018.8351065.

[46] Nordic© Semiconductor. *RF52840 Product Specification v1.1*. 2018. URL: https://infocenter.nordicsemi.com/index.jsp?topic=%2Fps_nrf52840%2Fkeyfeatures_html5.html.

[47] Yu Song et al. “Wireless battery-free wearable sweat sensor powered by human motion”. In: *Science Advances* 6.40 (2020). DOI: 10.1126/sciadv.aay9842. eprint: https://advances.sciencemag.org/content/6/40/eaay9842.full.pdf. URL: https://advances.sciencemag.org/content/6/40/eaay9842.

[48] *Textbook OpenStax Anatomy and Physiology*. 2016. URL: https://cnx.org/contents/FPtK1zmh@8.25:fEI3C8Ot@10/Preface.

[49] *Textbook OpenStax Anatomy and Physiology*. 2016. URL: https://cnx.org/contents/FPtK1zmh@8.25:fEI3C8Ot@10/Preface.

[50] P. Thippun et al. “An Experimental Study of Dynamic Capabilities in a Wireless Body Area Network”. In: *2020 12th International Conference on Knowledge and Smart Technology (KST)*. 2020, pp. 164–167. DOI: 10.1109/KST48564.2020.9059346.

[51] Matteo Tolomei. “Towards an electromyographic armband: an embedded machine learning algorithms comparison.” Master’s Degree. Politecnico di Torino, 2020.

[52] Akiba Townsed K. Cufì C. *Getting started with Bluetooth Low Energy. Tools and techniques for low-power networking*. 2014.

[53] *UUID Generator*. 2020. URL: https://www.uuidgenerator.net/.

[54] Wikipedia. *Cosine similarity*. [Online; last edited on 18 February 2021]. 2021. URL: https://en.wikipedia.org/wiki/Cosine_similarity.

[55] Wikipedia. *Jaccard Index*. [Online; last edited on 2 March 2021]. 2021. URL: https://en.wikipedia.org/wiki/Jaccard_index.