UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Energy aware optimization for low power radio technologies

Ioana Suciu

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

PhD Program in Network Engineering

Energy Aware Optimization for Low Power Radio Technologies

Doctoral thesis by:

Ioana Suciu

Thesis advisor:

Prof. Dr. Xavier Vilajosana

Department of Telematics Engineering

Barcelona, January 2020

ERRATA PAGE

PROLOGUE

The evolution of machine to machine communication, resulted by equipping everyday objects with connectivity and intelligence, has enabled applications that were never before thought possible, such as: smart homes, wearables, remote patient monitoring, smart cities, smart agriculture and industrial automation. Moreover, these applications built around the smart objects target everyone and are created so as to be plug and play. These technologies are generically called the Internet of Things (IoT). What is characteristic to IoT is that it is continuously evolving, with new use cases and new technologies appearing, not leaving enough time for the old technologies to mature. The new use cases are thought so as to consider the interests of more and more individuals, ensuring that in a short time, everybody will directly start using these technologies. While the IoT is clearly improving our day to day life, with sensors monitoring harsh environments, making weather predictions, announcing available parking spots, or even monitoring our health, the enormous number of involved devices already reached an energy consumption equivalent to 10% of the world electricity generation.

This work is part of the SCAVENGE European Training Network, that targets the design and implementation of sustainable 5G networks in Europe. In particular, this thesis is focused on optimizing the machine to machine communication sector of 5G, that will be used to fulfill the vision of building an intelligent and fully connected world. We look for optimization techniques that allow an improvement in the user experience - such as increasing the number of devices accommodated by a network, boosting the network performance, or allowing for smaller and cheaper devices that enable new use cases and more availability to users - while always aiming at lowering the network energy consumption. In order to achieve this, we review the state of the art in energy efficiency techniques that can be applied to this sector, we identify the possible gaps and the cases where we can leverage them to bring both the improved user experience and the energy savings.

This work has received funding from the European Union Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675891 (SCAVENGE project).

ABSTRACT

This work is focused on exploring new techniques for enhancing the user experience and energy efficiency of IoT networks. We divide the proposed techniques in frame and chip level optimization techniques, respectively. While the frame level techniques are meant to improve the performance of existing radio technologies, the chip level techniques aim at replacing them with crystal-free architectures.

The identified frame level techniques are the use of preamble authentication and packet fragmentation, advisable for Low Power Wide Area Networks (LPWANs), a technology that offers the lowest energy consumption per provided service, but is vulnerable in front of energy exhaustion attacks and does not perform well in dense networks. The use of authenticated preambles between the sensors and gateways becomes a defence mechanism against the battery draining intended by attackers. We show experimentally that this approach is able to reduce with 91% the effect of an exhaustion attack, increasing the device's lifetime from less than 0.24 years to 2.6 years. The use of packet fragmentation despite the packet fits the frame, is shown to reduce the probability of collisions while the number of users in the duty-cycle restricted network increases. Combining packet fragmentation with group NACK can increase the network reliability, while reducing the energy consumed for retransmissions, at the cost of adding small headers to each fragment. It proves to be effective in dense duty-cycle restricted networks only, where the headers overhead is negligible compared to the network traffic.

As a chip level technique, we consider using radios for communication that do not use external frequency references such as crystal oscillators. This would enable having all sensor‘s elements on a single piece of silicon, rendering it even ten times more energy efficient due to the compactness of the chip. The immediate consequence is the loss of communication accuracy and ability to easily switch communication channels. In this sense, we propose a sequence of frequency synchronization algorithms and phases that have to be respected by a crystal-free device so that it can be able to join a network by finding the beacon channel, synthesize all communication channels and then maintain their accuracy against temperature change. The proposed algorithms need no additional network overhead, as they are using the existing network signaling. The evaluation is made in simulations and experimentally on a prototype implementation of an IEEE802.15.4 crystal-free radio. While in simulations we are able to change to another communication channel with very good frequency accuracy, the results obtained experimentally show an initial accuracy slightly above 40 ppm, which will be later corrected by the chip to be below 40 ppm.

Keywords: energy efficiency, low power radios, LPWAN, LoRaWAN, preamble authentication, packet fragmentation, IEEE802.15.4, crystal-free radios, frequency synchronization, standard-compliance

ACKNOWLEDGEMENTS

Throughout the development of this work, I have received a great deal of support and assistance. I would like to use this section to thank everyone that has guided me throughout these years and I hope I have managed to include all of you.

First of all, I would like to thank my supervisor, Dr. Xavier Vilajosana, for all his support and guidance, for believing in me, and for never running out of great ideas for the development of this work. I guess I can say now that he is the best advisor a student can ask for.

I would like to thank the Worldsensing team, for including me in their projects, for allowing me to experience the 'corporate life' and for creating an amazing work environment. Here, very special thanks go to Alejandro Lampropulos and Jose Carlo Pacho, for their invested time and patience for teaching me the very basics of writing firmware, developing software and testing. I will forever be grateful.

While developing the second part of this thesis at University of California Berkeley, I had the pleasure to meet and work with an amazing team of people. I want to thank Prof. Kristofer Pister for welcoming me to his team and for trusting me with the responsibility that is developing algorithms for the first prototype implementation of the single chip mote (SCuM). I would like to acknowledge Thomas Watteyne, for being such an inspiring technology enthusiast and for his incredible writing skills. I want to thank David Burnett, for our productive debates regarding the possible ways of improving SCuM, for his time, his trust and continuous encouragements. I also want to thank Filip Maksimovic, Brad Wheeler and Osama Khan for their guidance and support with learning the hardware particularities of SCuM.

I am thankful to David Lopez, of Centre Tecnolgic Telecomunicacions Catalunya, for his patience and support for performing frequency accuracy measurements. Here, a special mention goes to Paolo Dini, for facilitating this collaboration and for always trying to keep this project on track.

I am grateful to Dr. Georgios Papadopoulos, of IMT Atlantique, and to Prof. Diego Dujovne, of Universidad Diego Portales, for taking the time to review this manuscript and for their very useful comments.

Very special thanks go to Borja Martinez, for his advice, his encouragements, and for teaching me that doing a PhD is a marathon, and not a sprint.

Last but not least, I want to thank my parents, Ion and Violeta, and my sister, Larisa, for always being there for me, for always supporting me through all my decisions and for trying to make me see the good in all things. Words of gratitude also go to the rest of my family and to all my friends, as they never gave up on me even if many times I was too busy to keep in touch.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TERMS AND ABBREVIATIONS

5G PPP 5G Infrastructure Public Private Partnership

ACK Acknowledgement

ADC Analog to Digital Converter

AES Advanced Encryption Standard

AP Authentication Preamble

AppSKey Application Session Key

ASA Adaptive Sampling Algorithm

ASIC Application-Specific Integrated Circuit

ASK Amplitude Shift Keying

BER Bit Error Rate

CCA Clear Channel Assessment

CDR Clock and Data Recovery

CMOS Complementary Metal-Oxide-Semiconductor

CRC Cyclic Redundancy Check

DCO Digitally-Controlled Oscillator

DL Downlink

DoS Denial of Service

EE Energy Efficiency

ETSI European Telecommunications Standards Institute

FEC Forward Error Correction

FPGA Field-Programmable Gate Array

GW Gateway

ICT Information and Communication Technologies

IF Intermediate Frequency

IoT Internet of Things

LEACH Low Energy Adaptive Cluster Hierarchical Routing

LO Local Oscillator

LPWAN Low Power Wide Area Network

LQI Link Quality Indicator

MA Moving Average

MAC Medium Access Control

MAuC Message Authentication Code

MEMS Microelectromechanical Systems

MIC Message Integrity Check

MTU Maximum Transmission Unit

NACK Negative Acknowledgement

NwkSKey Network Session Key

OOK On-Off Keying

PDU Physical layer Protocol Data Unit

PLL Phase Locked Loop

PWM Pulse Width Modulation

QoS Quality of Service

RF Radio Frequency

RLS Recursive Least Squares

RPL Routing Protocol for Low-Power and Lossy Networks

RSSI Received Signal Strength Indicator

RX Receive

SCADA Supervisory Control and Data Acquisition

SDU Service Data Units

SF Spreading Factor

SNR Signal to Noise Ratio

ToA Time on Air

TSCH Time Slotted Channel Hopping

TX Transmit

UL Uplink

WBAN Wireless Body Area Networks

WSN Wireless Sensor Network

XTAL Crystal Oscillator

CHAPTER 1

Introduction

1.1 Motivation and objective

It is foreseen that by 2020 there will be more than 50 billion mobile and wireless devices connected to the cloud with requirements of high data rates and low latency [1]. Traditional mobile communications used to connect people will be complemented by an enormous increase in wireless links connecting machines, the so called Internet of Things (IoT). This enormous growth in the number of connected devices will have as an immediate consequence an equally large growth in the carbon footprint of the Information and Communication Technologies (ICT), which is of high importance especially because already by 2013 the ICT ecosystem consumed about 10% of the world electricity generation [2]. The key to control and bring to a stop the negative impact of the ICT sector on the environment, is to build energy efficient, sustainable wireless networks that would enable as much as possible the use of harvested ambient energy.

In this context, the 5G Infrastructure Public Private Partnership (5G PPP) [3], created by the European Commission in collaboration with industry manufacturers, service providers, network operators and researchers, is working on providing technologies and standards for the next generation communication infrastructures and markets. With the emergence of concepts and technologies related to the IoT, smart cities, e-health and intelligent transport, there are new challenges for the ICT sector [3]:

1. Saving up to 90% of energy per service provided, with the main focus on mobile communication networks
2. Facilitating very dense deployments of wireless communication links to connect over 7 trillion wireless devices serving over 7 billion people
3. Ensuring for everyone and everywhere the access to a wider panel of services and applications at lower cost
4. Creating a secure, reliable and dependable Internet with a zero perceived downtime for services provision

Up until now, each new generation of mobile communication has brought an important change in the way we experience the day to day life: the second generation (2G) allowed for voice transmission, the third generation (3G) enabled mobile data transfer and the fourth generation (4G)-

with its increased data transmission speeds - unleashed the mobile streaming revolution. The fifth generation (5G) of mobile communication aims at something much bigger: building an intelligent and fully connected world [4], and it will do so by accommodating massive machine to machine communication. It is this vision for 5G that makes the wireless sensor networks become a very important part of the 5G ecosystem. It is also for this vision that the future wireless networks are challenged to meet extreme performance goals: improve efficiency in both energy consumption and costs, while reliably supporting higher connectivity density and co-existence between various access technologies. In this context, the objective of this thesis is boosting the performance of wireless sensor networks with a main focus on energy efficiency. We identified gaps in the existing energy efficiency techniques for wireless sensor networks, gaps that can be explored for specific use case scenarios. Moreover, we analyze the impact of the identified energy efficiency techniques on communication reliability, network densification and costs, as per the aforementioned 5G PPP challenges.

1.2 Problem statement

The IoT principle is simple: objects that surround us are connected to the Internet for remote sensing and control. The key element in IoT is represented by the Wireless Sensor Network (WSN), with a certain Internet connectivity model for the devices: a WSN consists of battery powered devices ("sensor nodes" or simply, "nodes") that collect environmental information using various sensors and a device ("sink" or "gateway") that receives the information sent by the nodes. The usual internet connectivity model is to connect only the gateway to the Internet (see Fig. 1.1).

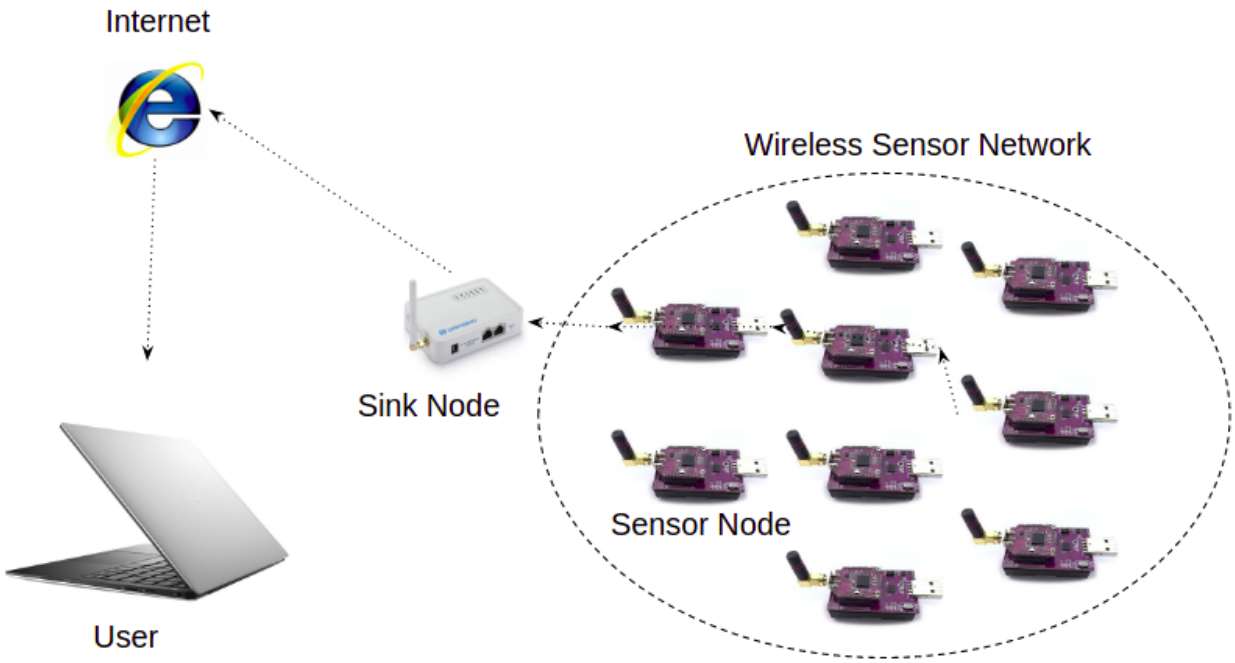


Fig. 1.1 The principle of IoT

There are short-range and long-range IoT technologies. Fig. 1.2 [5] provides a comparison of the wireless solutions for IoT in terms of range and data rate. The short range technologies, such as ZigBee [6], BlueTooth [7], 6TiSCH [8], Wi-Fi [9], are used mostly for indoor applications like home and building automation, as they provide relative high data rates (250kbps for ZigBee to over 100Mbps for WiFi) but short communication range. These networks operate mostly in the 2.4 GHz band.

The long range technologies, also called Low Power Wide Area Network (LPWAN), are used for outdoor applications, providing low power operation and long range connectivity at the cost of reduced data rates and strict duty cycle regulations. Nowadays, there is an increasing demand for LPWANs, which are being deployed and used in many applications as: infrastructure monitoring, smart metering, transportation of oil and gas, traffic flow optimization, maintenance cycle optimization in industry, etc.

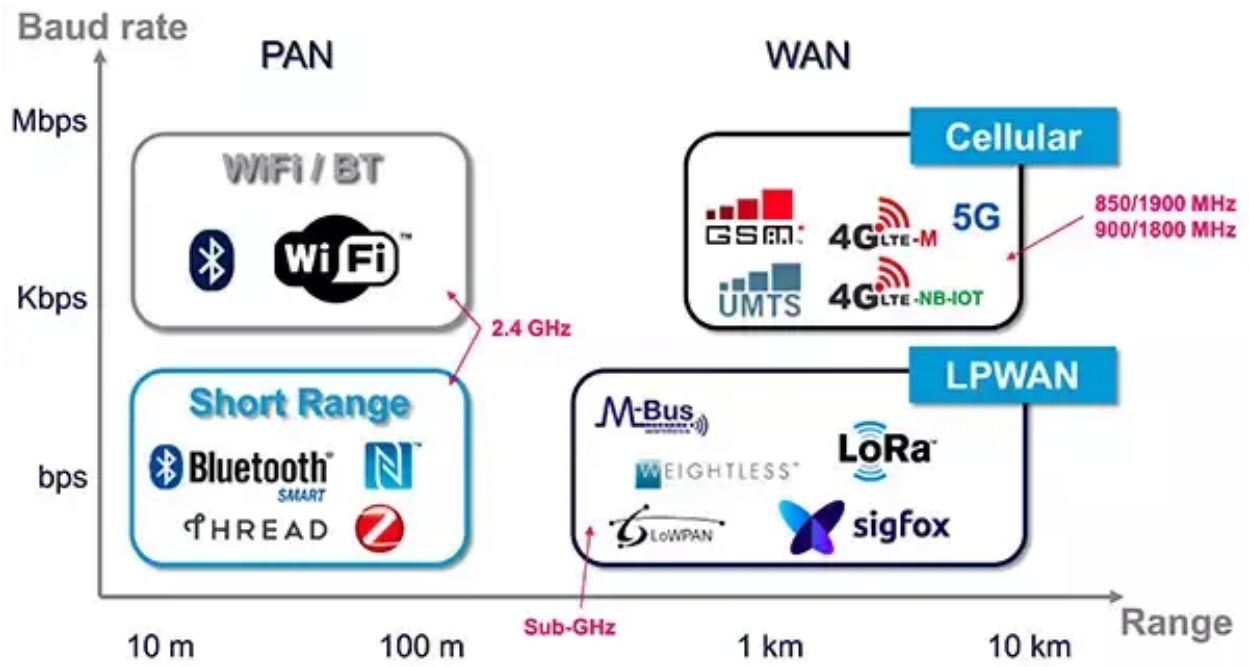


Fig. 1.2 Wireless options for IoT

Most of the current LPWAN technologies operate in the sub-GHz bandwidth in order to cover a communication range in the order of km, have a star network topology, an Aloha-based MAC access in order to keep them as simple and as energy efficient as possible, most of their traffic is uplink and the acknowledgements are limited. Examples of such technologies are LoRaWAN [10], Sigfox [11], Weightless P [12] or RPMA [13].

We can already see (Fig. 1.2 [5]) that there are plenty of IoT communication technologies for each short range or long range IoT application we can foresee: the IoT market is fragmented and is continuously evolving and growing, with technologies that are not mature nor stable. While there are plenty of energy efficiency techniques that can be theoretically applied to IoT (see Chapter 2), when having to apply them to real deployments and also enable the features envisioned by 5G PPP[3], there are a few details that have to be taken into account:

1. The IoT sensors have very low computational power (compared to a computer or simulation environment) and improving it by using better microprocessors would impact their low cost. This means that we need simple and effective approaches for a successful implementation and operation of the devices.

2. The European Telecommunications Standards Institute (ETSI) has strict regulations for the sub-GHz bandwidths in terms of number of available channels, maximum allowed duty cycle and transmission power. Detailed information about these regulations for each type of application and country of operation, along with the permitted license-free operating frequency bands can be found in [14]. Specifically, for the Europe ISM 868MHz band, there are 23 channels with a bandwidth of 125kHz, out of which only 2 channels have a duty cycle requirement of 10% and no requirement respectively, while the remaining channels have to

satisfy duty-cycle restrictions of 0.1% or 1%, depending on the case. The direct consequence of these restrictions is the limited amount of communication these devices can exchange. For this reason, the communication overhead for routing, synchronization or other information exchange other than data transmission would have to be as low as possible.

3. For industrial applications especially, the typical network lifetime requirement is of 10 years. For this reason, the most energy efficient communication protocol that can be used is Aloha-based, as no synchronization or coordination between sensor nodes is required. This impacts the reliability of the network when the number of sensor nodes increase, as the packet collisions also increase.

We can see that as the network is growing, the most energy efficient communication protocols (Aloha-based) and the cheapest microprocessors have a direct impact on the network reliability and device price. Also, in order to operate in the license-exempt bands (free of costs), the sensor nodes have to limit the number of transmitted packets, which impacts the communication quality in terms of latency. Also, as they are free, these bands are very populated, which makes that collision-avoidance schemes have to be implemented in order to be able to successfully communicate, which, in turn, impacts the energy consumption, communication latency and overhead. These wireless sensor networks that were built specifically to be cheap and simple so as to provide a basic amount of communication and keep a low energy consumption, are now challenged to provide massive amounts of quality communication while becoming even cheaper and more energy efficient. This being the case, there are trade-offs to be made between the cost of the devices, the energy consumption and the quality of the other network parameters. In this manuscript, we identified unexplored energy efficiency techniques that we present along with the proper use case scenarios, so as to understand the trade-offs (as per 5G PPP challenges [3]) when trying to maximize the energy efficiency of the network.

1.3 Research methodology and overview

Energy efficiency in WSN, either if part of a short range network or a LPWAN, can be achieved by various techniques, from enhancing the performance of the hardware components, to turning various sensor nodes off, or using natural energy resources. Having more energy efficient networks increases the network lifetime which at the same time postpones the need to replace batteries and increases the possibility to completely function using only harvested energy.

In this manuscript, we identified energy efficiency techniques that have not been explored much in the literature and that could be applied at the frame and chip level of an IoT sensor node. More specifically, we identified that at frame level, we could use authentication preambles and aggressive packet fragmentation. We found these techniques to be more suitable for LPWAN, that have very small data rates and important energy consumption restrictions in particular, but they can be extended to any technology. While using authentication preambles brings some communication overhead, it also decreases 91% the effect of an energy exhaustion attack that aims at draining a sensor node's battery and so increases the network lifetime in the long run. We explored packet fragmentation despite a packet fits the frame and we observed that in dense and slow networks it improves the energy efficiency, scalability and goodput of a wireless sensor network, even when using Aloha protocol. Aggressive packet fragmentation can be combined with a retransmission policy that provides user guarantees in terms of packet delivery ratio.

At the chip level, we consider the use of crystal-free radios, that could bring a 10 times more energy efficient communication, together with the decrease in size and price of the devices. This technique can be considered by any IoT manufacturer, but its advantages in terms of size, price and consumption are of high importance for short range technologies used for applications such as, but not limited to, in-body sensors or microrobots. Still, using crystal-free radios for communication has an important impact on the frequency accuracy, frequency stability with time and temperature, and the ability to synthesize the communication frequency.

In this manuscript, together with numerical computation or NS3/MATLAB simulations, we also present experimental evaluations for the energy efficiency of using authentication preambles and the communication accuracy of using crystal-free radios. For this purpose, we used dedicated platforms: Loadsensing [15] sensor nodes for implementing authentication preambles and SCuM [16, 17] single chip mote for implementing frequency synchronization algorithms that allow the use of crystal-free radios in WSN.

The Loadsensing devices are produced by Worldsensing [18]. This product (see Fig. 1.3 [15]) is used for civil engineering applications (infrastructure monitoring: buildings, tunnels, roads, dams, etc) and for mining applications, including underground mines. Loadsensing uses the LoRa technology developed by Semtech [19], with an Aloha MAC and very low power consumption, guaranteeing 10 years of lifetime.

Fig. 1.3 Loadsensing components (left to right): sensor node, gateway and user interface.

SCuM ("Single-Chip μMote") is a crystal-free platform that features a 2.4 GHz transceiver (IEEE802.15.4 TSCH), a Cortex-M0 microprocessor, 128 kB of RAM memory, and all the support hardware in a single chip with an area of $2.5mm \times 3mm$ (Fig. 1.4 [20]). It is a prototype implementation of the "Single-Chip Mote" [16, 17]. By removing the off-chip crystal reference and integrating all the elements of a sensor node on a single piece of silicon, SCuM achieves a 10 m communication range, and an energy consumption of 670 uA in reception and 1 mA in transmission mode at a transmission power of -10 dBm. As a point of comparison, the crystal-based industry leading low-power technology (2.4 GHz, IEEE802.15.4 TSCH) is the $10\ mm\ \times\ 10\ mm$ LTC5800-IPM [21], consuming 4.5 mA when receiving and 5.4 mA when transmitting at 0 dBm.

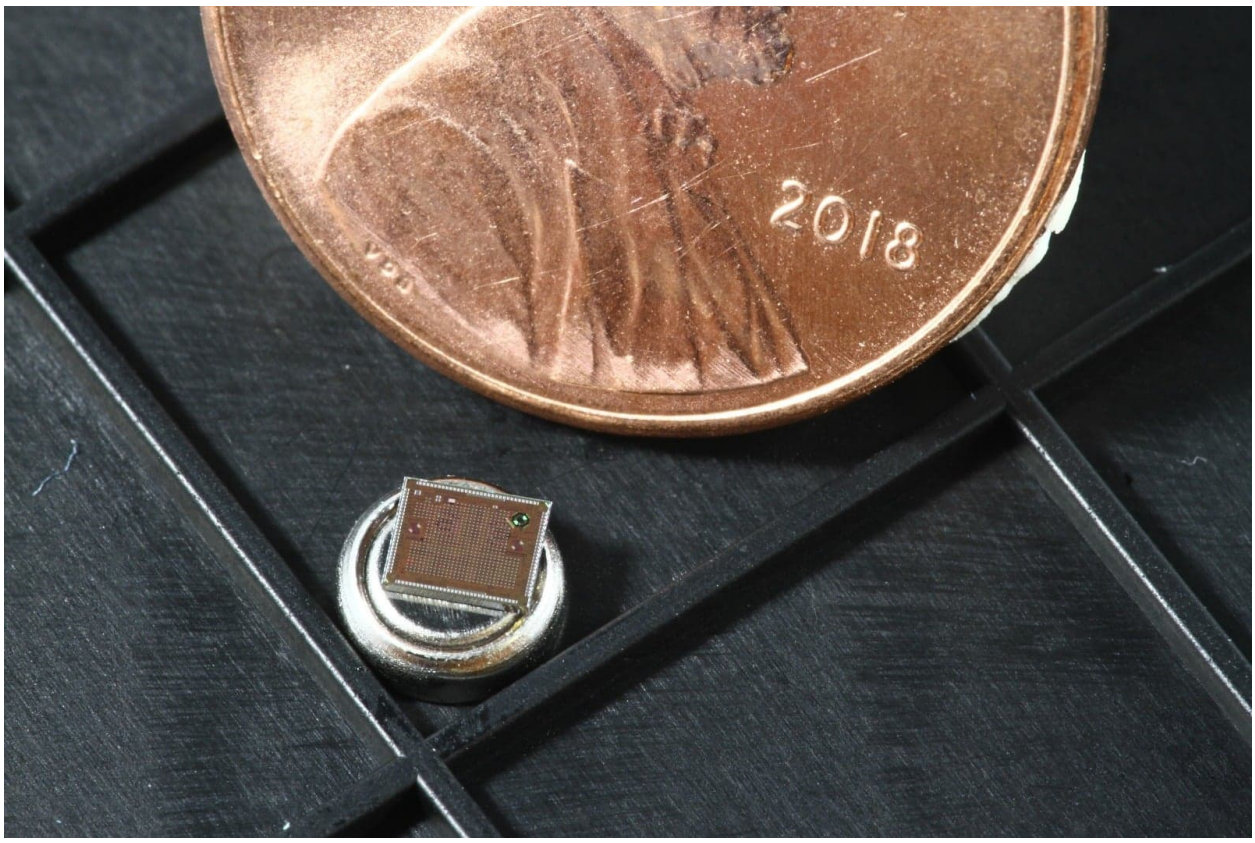


Fig. 1.4 The Single-Chip Mote next to a 1 U.S. cent coin.

1.4 Thesis organization

The structure of this manuscript is as follows. Chapter 2 presents the state of the art in energy efficiency techniques for WSN. We also add to this chapter our proposed techniques of preamble authentication and packet fragmentation as energy efficiency techniques at frame level and the use of crystal-free radios at the chip level. The thesis is then split in two parts: PART 1 (Chapter 3, Chapter 4 and Chapter 5) explores the energy efficiency techniques at the frame level, while PART 2 (Chapter 6 and Chapter 7) deals with the chip level technique of using crystal-free radios. Even if it seems counter-intuitive, we chose to organize this thesis by starting with frame level and then going to chip level as we are going through the most simple options for achieving energy efficiency and good network performance to the most complicated options that we have in order to achieve that. In this sense, it is much easier changing the packet format of an existing technology and protocol (as we do for preamble authentication) than bringing changes to the protocol itself (as we do for packet fragmentation). We then realize that for bringing a significant change in terms of energy efficiency, costs and performance, we have to go beyond the protocol and consider new possible hardware designs for the future low power radio technologies, that would eventually replace the existent ones.

For PART 1, Chapter 3 introduces the use cases for authentication preambles, the usage protocol and computes the energy efficiency of this technique in terms of battery lifetime. Experimental evaluation is provided using Loadsensing nodes. Chapter 4 analyzes using MATLAB simulations

the advantages of using packet fragmentation despite the packet fits the frame. The usage scenarios where this strategy proved to be effective are then reconsidered in Chapter 5, where a retransmission protocol is also evaluated in NS3 as a way to increase the communication reliability as the network grows.

For PART 2, Chapter 6 presents the challenges in terms of communication accuracy of using crystal-free radios when wanting to improve the energy efficiency and costs of wireless sensor networks. Algorithms to control the communication frequency accuracy are presented and experimental evaluation is provided using SCuM. Chapter 7 continues this work by providing a full solution for a crystal-free radio to be able to synchronize to a network, initiate communication, synthesize the communication channels needed for transmission and reception and be able to maintain this communication as the temperature in the environment changes. Chapter 8 concludes the manuscript.

CHAPTER 2

State of the art

2.1 Introduction

The design of sustainable WSN is a very challenging issue, as they have been designed for very specific applications and the requirements they have to satisfy differ from one application to another. As WSN devices are battery-operated, replacing the batteries could be impossible either because of deployment in a hostile environment, or because of the costs and the mechanical problems that arise. The expected battery lifetime is in terms of years, for the long range networks deployed in harsh environments. WSN applications range from small-size healthcare surveillance systems to large scale environmental monitoring.

Proposing procedures for Energy Efficiency (EE) in IoT is a difficult task, because of the following characteristics of the IoT market and technologies [22]:

- the existing technologies are not mature and stable, evolving at a high pace
- IoT is not a homogeneous application area, being comprised of many and very diverse product categories

A classification of WSN applications is shown in Fig. 2.1 [23]. As the WSN applications cover very different areas, the technological solutions are built so as to satisfy the respective area's specific demands in terms of robustness, mobility, security, scalability, Quality of Service (QoS), latency or coverage. For example, for the healthcare domain, the most important factor would be the latency, as for detecting life-critical emergencies real-time communication is a must. The next factor in the order of importance could be the mobility, for ensuring connectivity even when the patients wearing the sensors are moving. In industry though, for Supervisory Control and Data Acquisition (SCADA) [23], the real time requirement is replaced by the need of bounded latency, together with robustness and security. Another domain that changes the importance order of the requirements is agriculture, that adds more weight to factors such as coverage, lifetime and scalability. Rault *et al.*[23] give a nice summary of the requirements of each application area (see Fig. 2.2 [23]).

In order not to have a large scale of poor efficiency cheap devices, the EE procedures should be independent of specific technological solutions. They should be generic and adaptable to technology changes. The research that has been done in order to propose solutions for the energy-efficiency problem, covers areas going from network layer solutions to physical layer optimization,

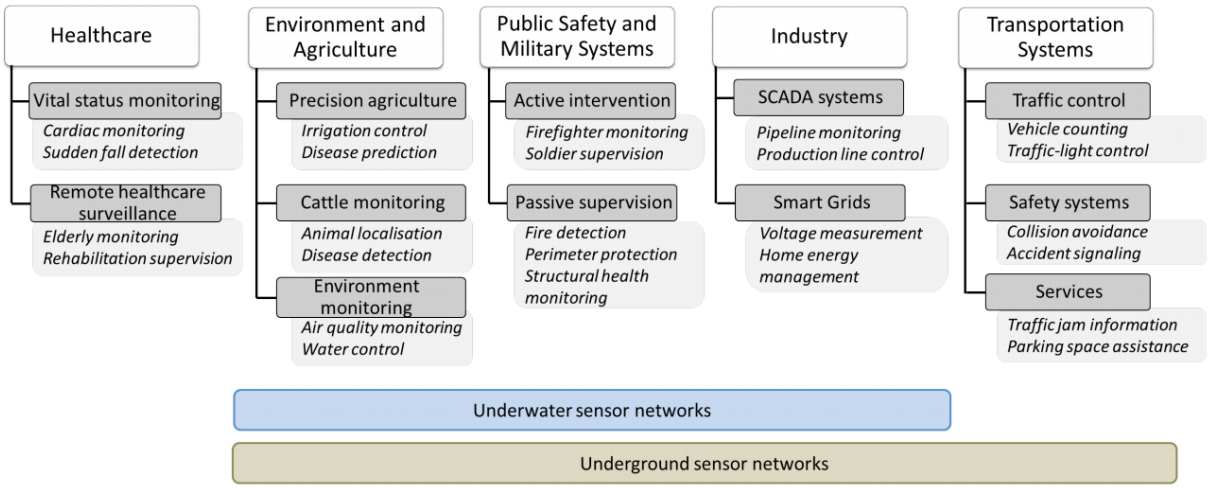


Fig. 2.1 Taxonomy of WSN applications

and the WSN designer has to select the techniques that are best for its application-specific WSN architecture and the specific radio technology used [22, 24].

In the following, we describe techniques for energy efficiency in IoT. We choose to go through these techniques in the order requiring less changes to the existing protocols and technologies down to the more invasive techniques. The first category of energy efficiency techniques that we describe deals with managing the activity of sensor nodes so as to stay as much as possible in sleep state and save energy. Then we review the techniques related to the frame, that aim at reducing (where possible) or better organizing the number of transmitted bits. The last category treated consists in creating new radio chip designs and/or explore PHY layer parameters.

2.2 Node activity management

The radio of a sensor node, when not turned off, can have the following states: transmitting, receiving, idle listening and sleep. When not sending or receiving data, keeping the radio in idle mode is energy consuming. In order to increase the energy efficiency of the sensor nodes there is the need for techniques that keep the radio in sleep mode as much as possible without altering the functionality of the network. Such techniques are: functioning according to a duty-cycle, using passive wake-up radios or controlling the topology of the network using redundant nodes [23].

2.2.1 Duty cycle schemes

When duty cycle schemes are used, the idle listening time of a sensor node is minimized. The sensor node will wake up on-demand, according to a schedule or in an asynchronous way [23, 25].

On demand wake-up is achieved by using a broadcasted wake-up signal that does not need to be decoded and it is heard by all the sensor nodes in the area. This approach is common in event driven applications and while it manages to wake up a sensor node only when communication is needed, it also wakes up all the surrounding sensor nodes that may be not needed [25].

Unlike on demand wake-up, the scheduled wake-up requires synchronization between the sensor nodes in a network so that these share a common notion of time. In this way, each node can wake up in order to receive data that its being send by its parent node, forward it and then go back to sleep. For example, in BLE, the slave device wakes up each 100ms in order to listen to the master device for data to be sent/received [24]. This is similar to LoRaWAN class B, in which a device has to listen for downlink beacons from the gateway each 128s and has additional configurable ping slots to listen for data [10].

In asynchronous wake-up, no synchronization is needed between the sensor nodes and a node wakes up according to its own schedule. In order to enable communication, nodes need to wake up more frequently to check if messages are being sent: either the sender uses a long preamble so as to ensure that the receiver will hear it during one of its wake-ups, or the receiver listens for a longer period of time [23]. In LoRaWAN class A [10], for example, the networks are organized in star topologies and the sensor nodes are not synchronized. Each sensor wakes up with a given periodicity, sends data to the gateway and immediately opens two receive windows in order to receive any downlink message from the gateway. This ensures the lowest power consumption, as the nodes do not have to keep tight synchronization and are asleep more than 99% of the time (in Europe, LoRaWAN operates in 868MHz ISM band, duty cycle restricted [14]).

While the above techniques make the wireless sensor networks very energy efficient, they have as a drawback a significant increase in communication delays.

2.2.2 Wake-up radios

Mostly in multi-hop networks, the sensor nodes have to spend time in idle listening for an incoming packet in order to minimize latency and packet loss. In idle listening, the node is not actively

receiving or transmitting any data, but it is consuming energy. A solution to the problem of idle listening, without increasing the overhead or complexity of time synchronization algorithms, is the use of wake-up radios. The wake-up radios can be active or passive. The active wake-up radios need to be powered, but consume very little compared to the main radio, whereas the passive wake-up radios are harvesting the energy from the incoming wake-up signal [26]. The wake-up range of a passive radio is much shorter than that of an active wake-up radio.

Using very low power wake-up (active) radios additional to the main power-hungry radio aims at keeping the main radio in sleep state until it is needed for communication, while not compromising the communication range. In order to do so, the very low power wake-up radio is always on until it receives a wake-up signal that is used to trigger the main radio. In order to achieve very low power consumption in continuous listening (as low as $1.8\mu W$ for an active radio [27], or $6.1nW$ for a passive radio [28]), these radios usually use basic modulation schemes, such as On-Off Keying (OOK), Pulse Width Modulation (PWM) or Amplitude Shift Keying (ASK) to simplify the receiver circuitry, and operate in the sub-GHz band to achieve higher range (tens of meters) [29]. By integrating address decoding capabilities, the unneeded wake-ups can be avoided. Still, in [30], they conclude that the selective wake-up signal requires more energy consumption (to be transmitted and decoded upon reception) than the energy loss during false-wake up. This makes the address-based wake-up signal advisable only for very dense networks, where much energy would be consumed in false wake-ups.

The wake-up signal can be initiated either by the sender before sending data, or by the receiver, when needing data. If initiated by the sender (broadcast), multiple senders can transmit data concurrently, causing collisions and unneeded wake-ups. If initiated by the receiver (usually, a gateway), no synchronization between nodes is required and each node can be woken up only when its data is needed, achieving 100% reliability in asynchronous networks [27].

2.2.3 Redundancy in topology

Some level of redundancy should exist in a wireless sensor network in order to improve the data reliability and safety. There can be two types of redundancy: spatial redundancy, when multiple nodes are placed in the same geographic location so as to obtain similar data from multiple sources; or temporal redundancy, when the same node is sending data from multiple successive measurements. If not used in an intelligent way, redundancy wastes energy [31], while if used, it can save more energy than if there was no redundancy at all in the network [32].

When sensor networks are dense and redundant, some of the nodes could be deactivated in order to save energy. The nodes should be organized in subsets or clusters, so as to ensure connectivity and at each time only the minimum number of nodes in the network to be active [33, 31]. However, this technique requires complex algorithms to create the subsets: the topology of the network, the physical location of the sensor nodes and their available energy has to be known at each time so as to optimize the energy consumption. Keeping this information updated may require extra communication between sensor nodes (as for exchanging routing tables), adding to

the communication overhead and to the energy consumption. In [32], a probabilistic model that combines the geographical position of a sensor with its probability to detect an event is used for network partitioning and scheduling; this probability is also used for a fault-tolerance mechanism, in order to ensure that the transmitting node at a time will reliably send its data. Combining this fault mechanism with the network partitioning proves to be more energy effective as the network becomes denser, as a certain node sends data less often, without affecting the reliability of the network.

In order to leverage the redundancy in a network, it is important to determine how to actually use it given the network topology. In [34], they show that using a geometrically increasing number of redundant nodes from the source to the sink cannot be chosen as a universal solution. They try to find the optimal distribution model of sensor nodes given a pre-determined deployment and the number of available redundant nodes. Their model assumes that all nodes send a packet of the same size, that the distance between two sensors is not variable and that a node can act both as a source and as a relay. They propose the number of redundant sensor nodes around a node to be proportional to the energy consumed by the sensor.

The fact that the wireless sensor nodes are built to be cheap and have reduced computing capabilities renders the decision making process at the node's side challenging, e.g: staying active/switch to sleep, based on parameters like network connectivity, collision rate, etc. In [33], the authors try to eliminate the aforementioned communication overhead by leveraging the neighbour discovery process. In the same context of limited capabilities, in [35], they choose the cluster heads of the redundancy zones of a network and assume that the cluster heads can perform data aggregation and compression, which would hardly be the case in a real deployment.

2.2.4 Routing

In multi-hop wireless sensor networks, routing protocols are needed in order to gain knowledge of the network topology and to establish the communication path from each sensor node to the sink [36]. The most common routing option is for each node to choose a parent that is one hop away (closer to the sink) and to send its data to this node. Then the parent node forwards the data to its own parent node and so on, until the data arrives at the sink/gateway. A routing protocol that achieves such a tree topology is Routing Protocol for Low-Power and Lossy Networks (RPL) [37]. The immediate consequence is that the nodes closer to the sink spend more time in active state in order to forward all the incoming packets and so, their battery drains quicker.

Organizing the sensor network in clusters coordinated by cluster heads is one way to avoid this burden, as it reduces the distance between communicating nodes and allows the rotation of nodes for the role of cluster heads, homogenizing the energy consumption inside the cluster and implicitly, inside the network. A well known routing protocol of such type is Low Energy Adaptive Cluster Hierarchical Routing (LEACH) [38]. Only the cluster heads are in charge of aggregating the data and for sending it to the sink [35].

Another way to balance the energy consumption in a multi-hop network is to enable routing

by choosing the available energy of the sensor node as metric, and not the distance to the sink [23, 39]. Still, in this case, knowledge of the available energy of all nodes in the network is needed in order to continuously update routing tables, even in static networks. In [39], they implement RPL with energy as a metric, and each sensor node executes an algorithm in order to estimate its available energy, value that will be used as path cost for when the nodes choose their parents. Their scheme proves to distribute the energy consumption evenly among nodes without an impact on the transmission accuracy. In [40], the estimation of the expected lifetime of a sensor node is used as a metric with RPL. This estimation is based on the remaining energy, the link reliability to its parent and the amount of forwarding the a certain node has to perform. The obtained performance is comparable to using RPL with link reliability as a metric, but with a more balanced energy consumption.

Allowing multi-path routing is more complex than single path routing, but it can be a way to balance the energy consumption within the network by alternating between multiple available paths for transmitting data packets [41, 42, 43].

2.3 Frame and medium access layer level

The frame and medium access layer techniques for energy efficiency in IoT are directly related to reducing the amount of data sent in the network, or changing the way data is sent in the network. Data reduction is a solution to improving the energy efficiency of wireless sensor nodes and can be achieved directly by reducing the sensing and sampling tasks performed by each sensor node. Besides reducing the energy consumption, data reduction is important for reducing the network traffic and saturation, and it is suitable for applications that do not require real time data [44]. Traditional approaches for data reduction are: data aggregation, adaptive sampling, network coding and data compression.

For the cases where the amount of data cannot further be reduced, as it is the case of duty-cycle restricted networks that send a very small amount of data, we identified that using packet fragmentation despite a packet fits the frame could be a way of improving the energy efficiency of the wireless sensor networks under congested situations. Moreover, to protect the network against energy exhaustion attacks, using authenticated preambles enables a node to discard a malicious packet before completing its reception, saving energy and contributing to the overall energy efficiency of the network.

2.3.1 Data aggregation

Data aggregation schemes require a node to act as "aggregator", in charge of collecting and aggregating the data from its cluster in order to reduce the amount of data forwarded towards the sink. While data aggregation reduces traffic and latency, it also reduces the accuracy of data, as only the minimum necessary data is sent.

An important step for implementing data aggregation is managing the clustering of the network. In [45], the authors propose data aggregation for improving the energy efficiency of the data collection schemes used for smart cities under LTE-A. The aggregator is selected by the base station and all the neighbouring nodes send data to the aggregator via low power links, while an optimal modulation and coding scheme is chosen for the aggregator to send its data to the base station. Their proposal is shown to be 11 times more energy efficient than LTE-A.

There are also tree models for data aggregation, where each node sends its data to its parent, which will aggregate it to its own data and send it further, but this scheme assumes that all devices have the computational capability of aggregating data [46, 47].

2.3.2 Adaptive sampling

Adaptive sampling is a technique that adjusts the sampling rate of the sensors while leaving the information precision unaffected. In [48], an online Adaptive Sampling Algorithm (ASA) is proposed, that adjusts the sampling rate of the sensor to the physical phenomenon under monitoring. They evaluated it experimentally as part of a snow composition monitoring application. The algorithm starts with a low sampling rate and after detecting a change in value of the measured

parameters, the sampling rate is increased.

The sensing task can be energy-consuming and may generate unneeded samples which affects communication resources and processing costs. For example, in a supervision application, low-power acoustic detectors can be used to detect an intrusion. Then, when an event is reported, power-hungry cameras can be switched on to obtain finer grained information [23]. Spatial correlation can be used to decrease the sampling rate in regions where the variations in the sensed data is low.

In [49], the authors propose adapting the sensing activity to the behavioural patterns of the application. This proposal is done for sensor networks used in smart parking applications. Each sensor node has customized sampling rates based on the data evolution characteristic to its specific location, but the target is optimizing the network energy consumption. Their approach constrains the operation of the most active sensors, while relaxing that of the sensor nodes with less activity, such that all sensor nodes meet the expected lifetime required by the application.

2.3.3 Network coding

Network coding is a technique suitable for broadcast applications. In order to reduce data traffic and implicitly energy consumption of the nodes forwarding the data, network coding implies a node sending a combination of the received packets, and not all the received packets. This means sending an XOR of the packets, or a random linear combination of the packets [50]. Then, at the receiver's side, the decoding is done either with an XOR operation with the overheard packets, or by solving a system of linear equations, respectively.

IoT devices have limited computational resources, such that using complex network coding schemes may result in important delays. This is why, in [51], network coding is used at the fog layer. This layer is an intermediate layer between the WSN and the cloud and it is composed of devices that have storage, computing power and network connectivity in order to collect data from the sensor nodes and to reduce the amount of data sent to and stored in to the cloud. Network coding is also shown to reduce additional data download in highly loaded P2P or multicloud environments.

Network coding is also a way to increase the wireless sensor network reliability and to reduce latency. In this sense, based on the link reliability, sending multiple times a linear combination of packets proves to be more energy efficient than sending individual copies of packets, waiting for feedback from the receiver and eventually retransmitting [50]. Still, the linear coding schemes make all packets have the same importance, which may not always be the case, as there could be some packets that have high priority for real-time applications. In this context, in [52], they propose a weighted coding scheme that is able to differentiate between the priorities of different packets.

2.3.4 Data compression

Data compression is a technique suitable for increasing the energy efficiency of networks used for applications that do not require real-time data. It consists in reducing the number of bits used to represent the information. The limited computational resources of the sensor nodes used in IoT make it difficult to apply compression techniques to the data they sent.

In [44], the authors apply statistical processing to the temperature measurements collected by sensor nodes and take advantage of the fact that temperature is a slowly changing process resulting in statistically dependent symbols from the Analog to Digital Converter (ADC): the most probable values of the change in temperature are -1, 0 and 1 and will be encoded using less bits. The data is sent in frames containing the initial temperature value followed by the measured changes at each sampling (if these changes exceeded a given threshold). Their approach can bring as much as 85% of energy savings compared with sending raw data, but it is only recommended for applications that do not require real-time data.

A similar idea is presented in [53], where they are aware of the limited computational capabilities of sensor nodes and explore the high correlation between consecutive samples taken by the sensor in order to compute a compressed version of each sample. In order to do so, they map the sample values to a dictionary of size determined by the resolution of the ADC and then use Huffman coding to produce variable length codes for each sample evolution. They obtained lossless compression ratios of $\approx 67\%$ for temperature and humidity data.

2.3.5 Packet fragmentation

We propose the use of an aggressive packet fragmentation strategy, which consists in using packet fragmentation, despite a frame fits into the Physical layer Protocol Data Unit (PDU). This strategy could be a way to take better advantage of the available channels in the network, as the smaller the fragments, the shorter the time on air and the higher the opportunity to transmit without collisions. For multi-channel networks, using packet fragmentation spreads the transmission of a packet over a set of channels in a more homogeneous way, thereby allowing channel hopping by fragment. Also, in case of fragment loss, there is no need to retransmit the entire packet, but only the lost fragment, leading to energy savings. This strategy and the challenges that come with it will be the focus of Part I of this thesis.

2.3.6 Authentication preambles

A subtle type of attack directly related to the energy efficiency of wireless sensor networks, that can be classified as denial of service, is represented by the energy exhaustion attacks: it exploits the communication protocol in order to drain the battery of the device. The lack of authentication at link or network layer can be exploited by causing a sensor node to continuously transmit and receive messages. An attacker sending useless messages can exhaust the node's resources, as these messages are completely received before being discarded. We propose a verification method at

the PHY-layer that is extensible to any wireless protocol. By allowing a gateway to insert an Authentication Preamble (AP) in the packet format, the node is able to reject malicious packets sooner, saving energy and guaranteeing the network's long-term availability. The end-device does not have to authenticate its messages to the gateway at the physical layer. This is because the gateway does not have energy constraints. The details of the proposed method and the efficiency of using authentication preambles are treated in Part I of the thesis.

2.4 Radio and physical layer level

While most of the radio optimization techniques for energy efficiency in IoT focus on optimizing parameters such as the modulation scheme, the transmission power or antenna directivity ([23, 25]), a factor that is sometimes left out is the importance of choosing the most energy efficient radio chip around which the WSN solution is built.

2.4.1 Radio chip manufacturers

The radio module is the element that impacts the most the energy consumption of the WSN devices and it has to be chosen/designed so as to minimize the energy dissipation. For short-range communication, the radio chip consumption is dominating the consumption of the transmitted signal, while for long range communication, the opposite happens.

The importance of the processor used for a WSN application can be seen in Fig. 2.3a and Fig. 2.3b [54], where, for the same protocol, IEEE802.15.4 and LoRaWAN respectively, same voltage and the same bit rate, the consumption during TX/RX/sleep varies much with the radio module used.

Company	**Module**	**IEEE Protocol**	**Designed for network protocols**	**V_{DD} (Volt)**	**I_{TX} (mA)**	**I_{Rx} (mA)**	**I_{sleep} (µA)**	**Bit Rate (Kb/S)**
ANS [19]	ANY900	802.15.4	ZigBee	3.3	33	17	<6	250
Microchip [20]	MRF24J40MA	802.15.4	ZigBee	3.3	23	19	2	250
Radiocrafts [21]	RC2400	802.15.4	ZigBee + 6lowpan	3.3	34	24	1	250
Texas Inst. [22]	CC2430	802.15.4	ZigBee	3.3	25	27	0.9	250
Dresden Elektronik [23]	deRFmega128-22M00	802.15.4	Zigbee + 6lowpan	3.3	12.7	17.6	<1	250
Dresden Elektronik [24]	deRFsam3 23M10-2	802.15.4	ZigBee + 6lowpan	3.3	42	40	<2	250

(a) IEEE 802.15.4

Company	Module	IEEE Protocol	Designed for Network Protocols	V_{DD} (Volt)	I_{TX} (mA)	I_{Rx} (mA)	I_{sleep} (µA)	Bit Rate (Kb/S)	Operation Range Km
Microchip [2]	RN2483	Close Alignment with IEEE 802.15.4	LoRaWAN™ Protocol Stack	3.3	38.9* 32.9**	14.2	9.9	5.468 - 300	5 - 15 Km
Multiteh [26]	MTDOT-868-X1-SMA	Close Alignment with IEEE 802.15.4	LoRaWAN™ Protocol Stack	3.3	26 - 41	12	30.9	5.47 - 21.9	Up to 8 km
Nemeus [27]	Nemeus-MM002	Close Alignment with IEEE 802.15.4	LoRaWAN™ Protocol Stack	3.3	20 - 39.5	11.7	<2	0.3 - 40	12 Km line of sight

*Maximum transmitted power and 868MHz band. **This value for Maximum transmitted power and 433MHz band.

(b) LoRaWAN

Fig. 2.3 Comparison of the energy consumption of various radio chips: a) IEEE 802.15.4; b) LoRaWAN.

2.4.2 Removing the crystal oscillators

The elimination of the off-chip frequency reference, typically a Crystal Oscillator (XTAL), would bring important benefits in terms of size, price and energy efficiency of radios and systems-on-chip.

Crystals are typically used as the reference oscillators for synthesizing the radio frequency, and for keeping time. Crystals are popular because they provide the necessary stability while remaining low-power. It is common to use two crystals: a "slow" ultra low-power crystal oscillator for time-keeping (typ. 32,768 Hz) and a "fast" and more power-hungry crystal oscillator for radio frequency synthesis (typ. 12-48 MHz) [55]. Crystal oscillators have undoubtedly become cheaper and smaller and offered better performance [56]. Their main drawback is that they are inherently external (off-chip) components: using them requires special fabrication and packaging steps, and process integration beyond Complementary Metal-Oxide-Semiconductor (CMOS) [57]. They contribute to the energy consumption, size and cost of the final product, which becomes significant at high volumes (millions of chips) [58].

2.4.2.1 MEMS

There is important trend towards adopting Microelectromechanical Systems (MEMS) oscillators as an alternative to crystals [59]. MEMS oscillators are complex, silicon-based structures, consisting of a resonator, a fractional-N Phase Locked Loop (PLL) with a temperature sensor and compensation circuitry [59]. MEMS oscillators allow for a more customizable package size, a much higher accuracy than equivalent crystals over high temperature variation, and increased resistance to shock and vibration [60, 61]. While crystals have to be cut into a fixed size, MEMS oscillators can have a $60 - 80\%$ smaller footprint, placed on top of the Application-Specific Integrated Circuit (ASIC) [62]. They typically feature higher phase noise and energy consumption than equivalent crystals [60]. Texas Instruments [63], a global semiconductor design and manufacturing company, has released early in 2019 the industry's first crystal-free microcontroller unit, the SimpleLink CC2652 [64], that uses in its structure a MEMS-based acoustic resonator [65]. MEMS oscillators are, as crystal oscillators, off-chip components. And while the CC2652 is a good example of a MEMS+CMOS integration in a single package, it is still a packaging exercise, rather than a full single-chip solution.

2.4.2.2 Single chip radio design

To further integrate communication chips, research is being done on designing on-chip oscillators that could replace the off-chip (crystal) oscillators [66, 67, 68], aiming to obtain crystal-free architectures. Not needing to add crystals to a radio has several key advantages. All elements – including the oscillators – are part of a single piece of silicon. This significantly reduces the price of the device. It also reduces its footprint (size) in a design, to the point that it can be used completely standalone, provided it embeds a power source. In the extreme case, such a mote-on-a-chip can even be considered disposable. There are some further benefits, such as the fact that

the start-up time of an on-chip resonator is much shorter than that of a crystal, making the system faster to switch on and off. The resulting more efficient duty-cycling further reduces the average power consumption of the crystal-free modules [58]. Besides eliminating the off-chip elements, to further enhance the energy efficiency of a radio, the power-hungry circuitry such as a PLL [69] could be removed. This will result in an energy consumption even 10 times lower than that of a conventional crystal based chip, but will have important consequences to the obtained accuracy of the frequency and time reference [70].

True single-chip radio design requires neither a crystal nor a MEMS oscillator. The space savings of removing off-chip elements is significant. Wireless communication systems ("radio chips") have been demonstrated that are as small as $0.46\ mm^2$ [71], $2.025\ mm^2$ [72], $\approx 4.4\ mm^2$ [73], $\approx\ 9\ mm^2$ [74]. As a point of comparison, the area occupied by a small crystal oscillator is $\approx 2mm^2$ [56, 75]. Part II of this thesis describes the challenges of removing the off-chip crystal reference and PLL, analysing the obtained frequency accuracy and proposing techniques for frequency calibration and channel discovery. This is, from the best of our knowledge, the first time such analysis is done experimentally on a crystal free platform (we use SCuM), that we also use to apply the proposed calibration techniques.

2.4.3 Modulation and coding scheme

The optimization of the modulation and coding scheme is more important for the networks that have high communication range, as in those cases the energy required for packet transmission is significantly higher and the gain brought by this optimization would be more significant.

Deepak and Babu in [76] analyze the impact of modulation and coding scheme on the energy consumption of Wireless Body Area Networks (WBAN)s using IEEE802.15.6 standard. Their results showed that using coding schemes helps reducing the Bit Error Rate (BER) and that when using higher order modulation schemes in WBANs, the total energy consumption of correctly transmitting a packet is reduced given a specific BER performance value. This is the main consequence of the fact that increasing the modulation order, the transmission time of the packet is reduced. Still, for the same BER performance, the higher order modulation schemes require higher Signal to Noise Ratio (SNR)s, but their results account for this.

A similar analysis is done by Bouguera *et al.*[77] for LoRaWAN and specifically, LoRa modulation [78]. Higher coding rates ensure lower time on air. Higher data rates (low Spreading Factor (SF)) decrease the time on air as well (and so the consumed energy per transmitted bit), while needing more transmission power for reaching the same range as a low data rate modulation would. The maximum communication range at a given transmission power is achieved with the lowest communication data rate (SF12). An important thing to mention is that with high data rates, the payload size does not impact the energy consumption per useful bit. For low data rates though, a longer payload significantly decreases the energy per useful bit.

We can see the impact of the chosen modulation scheme in terms of energy consumption for data transmission. In order to determine the optimal modulation scheme that minimizes the energy

consumption for a given application, other application-specific factors need to be considered, such as the maximum allowed transmission power, the targeted BER and QoS, the network lifetime or the required transmission range.

2.4.4 Transmission power control

Adjusting the transmission power at the physical layer is a technique that can be used to enhance the energy efficiency of the wireless sensor networks. It leads to longer network lifetime, by either:

- increasing each device's lifetime
- using cooperation between nodes

For the case of a flat network topology in which every node has the same responsibility and same assigned traffic, the lifetime of the network can be increased by individually increasing the lifetime of the sensor nodes. By considering the distance to the next-hop or the gateway, the maximum transmission power of a device can be adjusted as often as possible. Dense networks can have lower energy consumption than the sparse ones, as for the latter, the distance between nodes may be higher. In this sense, in [79], they manage to extend ten times the lifetime of a sensor node by continuously adjusting its transmission power based on the Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI). In [80], they notice that transmission power control proves to be more effective when the link quality is poor.

Increasing the network lifetime in a cooperative way can be done when some nodes have higher requirements than others, or have more traffic assigned, by regularly adjusting the transmission power of every node in order to take into consideration the uneven energy consumption profile of the sensors [81]. Therefore, a node with higher remaining energy may increase its transmission power, which will potentially enable other nodes to decrease their transmission power, thus saving energy. This strategy has an effect not only on energy but also on delays, link quality, interference and connectivity.

Decreasing the transmission power has the advantage of decreasing the risk of interference, favouring the spatial reuse of bandwidth. It also decreases the number of nodes that can overhear the transmission. On the other hand, decreasing the transmission power reduces the transmission range and then more hops are needed for forwarding a packet. This translates into higher delays and a varying connectivity between sensors that would influence the network topology.

2.4.5 Directional antennas

Traditional sensors use omnidirectional antennas for communication, which means that they radiate energy in every direction when transmitting data. In these conditions the sensors waste a lot of energy when communication is supposed to happen only between two sensor nodes, and not as a broadcast. Using directional antennas would allow packets to be sent and received by focusing the energy on one direction at a time, improving the transmission range.

In [82], reconfigurable directional antennas (RDA) are shown to present even more advantages than conventional antennas, such as: radiation pattern, frequency, polarization and compound reconfiguration, further improving the energy efficiency of the network. Their study shows that using RDAs benefit the network by presenting an almost 50% reduced energy consumption, collision rate and latency when compared to networks using omnidirectional antennas. This happens because directional antennas allow the spatial reuse of bandwidth, limiting overhearing and by having an increased antenna gain require less power for reaching the same range as an omnidirectional antenna.

In spite of the presented advantages regarding the energy efficiency, using directional antennas require more processing and complexity at the sensor's side, as localisation techniques are needed for orienting the antenna and for maintaining information about the location of other nodes in the network [83]. Other problems that could rise are related to deafness or hidden terminal. Deafness happens at the receiver's side, when its antenna is oriented in a different direction than the direction of arrival of the packet [84]. The hidden terminal problem occurs at the transmitter's side, when it causes a collision by initiating a transmission to the receiver of an ongoing directional communication [85].

2.4.6 Battery charging

An important thing to mention is that the wireless sensor networks were introduced in order to replace the need for visual inspection and time-based maintenance done by people, with autonomous processes that by definition, would need as little as possible human intervention [86]. This is why, changing the batteries of the sensor nodes should be out of question. Anyway, there are more reasons for not considering the possibility of changing the batteries of the sensor nodes. One reason could be that the nodes are located in hard to reach environments or that opening the case of the sensors would cause mechanical problems. Another reason could be related to a very high number of nodes in the network which would make it a tedious and cost ineffective work for humans. The longer the lifetime of the wireless sensor network, the better the network is working towards its purpose. The usual requirement for the sensors lifetime is 10 years. Besides using techniques as those enumerated throughout this chapter, research in energy efficient IoT has focused on using energy harvesting or wireless charging, techniques that could help meeting the 10 year lifetime requirement or could even allow unlimited operation.

2.4.6.1 Energy harvesting

The energy can be harvested from sources like solar, thermal, vibration, RF signals, or even the human body. Even if energy harvesting is a promising solution to enhancing network lifetime, it requires careful considerations in many aspects, such as the availability of the harvesting source, the harvesting transducer type, its implementation and siting, the required power levels, the energy storage component, the associated power-management electronics, etc [86]. The most important factor to know is the availability and magnitude of the harvesting source, in order to be able to

predict the next charging cycle and to optimize the behaviour of the sensors between two charging cycles. In [87] a solar energy harvester is dimensioned so as to fit the IoT application requirements. It is used with a supercapacitor that serves as energy storage. The idea the authors use for the dimensioning is straightforward: the averaged scavenged power should be greater or equal to the average power consumption of the application. In their case, as solar energy is used, the sensor nodes have to adapt their sampling activity to two periodic behaviours: the day-night cycle and the season change. Even if during daytime in summer the nodes have enough energy to allow high sampling rates, during nighttime or winter days, they have to implement other energy efficiency techniques in order to be able to minimize the use of the stored energy. In [88], a case of vibrational energy scavenging is presented, with applications to Time Slotted Channel Hopping (TSCH) networks. Knowing the energy demands of such a deterministic and slotted network, a proper scavenging source is chosen so as to have a self-sustainable network.

2.4.6.2 Wireless charging

Wireless charging allows to transmit power between two devices without the need of physical contact. Besides the direct consequence of being able to charge sensor nodes using a drone, for example, wireless charging would also enable cooperation between sensor nodes that would trade energy. There are several identified techniques of wireless charging, but the more promising are: inductive coupling, magnetic resonance coupling, electromagnetic radiation and distributed laser charging [89].

While wireless charging can be done using electromagnetic radiation and reaching tens of meters, it could raise safety concerns for humans. This is the reason why this wireless charging option is regulated by the Federal Communications Commission to the order of milliwatts over several meters [90]. Magnetic resonant coupling seems to be more effective and can be used for near field power transmission, for example, for medical or in-body sensors, even electric car charging, without requiring line of sight [89]. Inductive coupling requires a very short distance and tight alignment for charging, with a range of few milimeters to few centimeters and is more suitable for home appliances (e.g.toothbrush). It also has a heating effect [89].

In [89, 90], a distributed laser charging technology is introduced. This technology can transfer approximately 2 Watts of power over a range of 10 meters. The efficiency of the transfer is affected by factors such as temperature, wavelength, distance, material and air quality. While suitable for mobile devices and home appliances, it has a low charging efficiency and requires line of sight. In [91], they consider a mobile charger that could charge multiple sensor nodes at the same time. They propose charging plans so as to minimize the number of stops a charger has to make and the total required charging time.

		Scalability	Coverage	RT Delay	QoS	Security	Mobility	Robustness
Healthcare	Vital status monitoring	- -	- -	+ +	+	+ +	+ +	+
	Remote surveillance	- -	- -	+	+	+ +	+ +	-
Agriculture and Environment	Precision agriculture	+ +	+ +	- -	- -	- -	- -	+
	Cattle monitoring	+ +	-	-	- -	- -	+	-
	Environment monitoring	- -	- -	+	+	+ +	+ +	-
Public Safety & Military systems	Active intervention	- -	- -	+ +	+	+ +	+ +	+ +
	Passive supervision	- -	+	+ +	+	+ +	- -	-
Transportation systems	Traffic control	- -	-	+ +	+ +	+ +	+ +	-
	Safety system	- -	-	+ +	+ +	+ +	+	+
	Services	- -	- -	-	+	+	+	-
Industry	SCADA systems	- -	-	+ +	+	+ +	- -	+ +
	Smart grids	+	-	+ +	+	+ +	- -	+ +

Requirement importance	
- -	very low
-	low
+	high
++	very high

Fig. 2.2 Application-specific requirements for wireless sensor networks

Part I

Frame level optimization

CHAPTER 3

Authenticated preambles in energy restricted LPWANs

3.1 Introduction

In the context of new market demands, such as smart cities, e-health, intelligent transportation, infrastructure monitoring and many other industrial applications, wireless sensor networks allow users to remotely access data and take decisions based on it. For example, an infrastructure monitoring application could trigger alarms when a maintenance cycle is needed, allowing for taking timely and appropriate actions.

LPWANs are wireless sensor networks that provide low power operation and long range connectivity at the cost of reduced data rates and strict duty cycle regulations. Current technologies [78, 11, 92], operate in the sub-GHz bands in order to cover a communication range in the order of kilometers, have star network topologies and Aloha-based Medium Access Control (MAC) access. Moreover, a gateway is able to accommodate thousands of sensor nodes, allowing for low cost deployments and customizable applications and services. The sensor nodes hardware and software is built to be simple and minimalistic with the goal to ensure years of battery lifetime. The low energy consumption and the possibility of powering the nodes using energy harvesting, reduces even more the costs of batteries and maintenance. All these factors make LPWAN the technology with the lowest energy consumption per provided service, but also makes it be vulnerable in front of attackers.

In [93] the main cyberattacks faced by the critical infrastructure owners and operators are introduced. Amongst others, those wireless devices are exposed to phishing, unpatched vulnerabilities and Denial of Service (DoS) attacks. Phishing opens the door to a wide range of possibilities: traffic capture, network flood, controlling of network parameters and other further possible exploitations. In mesh networks, wormhole attacks are used to create false route information and routing loops that increase the energy consumption of these networks [94].

The eventual vulnerabilities existent in the application or operating system allow an attacker to perform actions for which it's not authorized and it mainly leads to collection of information. For example, in the case of an energy management system, an attacker could get information about when and where power is used, that could further lead to knowing if and when anyone is on that property [95].

Both phishing and the exploit of vulnerabilities can lead to DoS. The DoS prevents a system

from carrying its designated tasks. Jamming can be a way towards service disrupt [95]. Moreover, in LPWANs, because of the low data rate that leads to a long time on air of the messages, jamming is possible and effective [94].

In this chapter, our attention is focused on the LoRa technology, one of the most used industrial LPWAN technologies. Security issues have been analyzed for LoRa networks [94], [96], [97]; Still, the analysis made in the literature explores how these issues impact the traffic performance and the actual data privacy, while in this chapter we focus on the impact on energy consumption and network lifetime. We analyze the impact of a DoS attack on the network lifetime through real data collection using the Loadsensing sensor nodes developed by Worldsensing [18]. The novelty of this work consists in considering LoRaWAN class B in a scenario in which an attacker aims at draining the batteries of the sensor nodes in order to kill the network. Then, we evaluate the efficiency of a possible solution based on authenticated preambles against this type of attack.

3.2 Security mechanisms in LoRaWAN

LoRaWAN [78] is a promising technology for IoT. It's proprietary physical layer uses CSS modulation [19]. Orthogonal SFs allow for variable data rates ranging from 0.3 kbps to 27 kbps. SF can vary from 7 to 12, the least corresponding to the smallest datarate and highest communication range. LoRa enables the trade-off between throughput for coverage range and robustness while keeping a constant bandwidth. In Europe, the sensor nodes can send data on randomly chosen channels in the 868MHz ISM band, subject to the allowed duty cycle [14, 98]. A typical gateway can listen on 8 channels at once.

LoRaWAN is organized as a star topology: the end-devices or sensor nodes communicate directly with a LoRa gateway. There are three categories of end-devices [10]: Class A, Class B and Class C, but it is mandatory that all devices support class A by default. A class A end-device supports bi-directional communication, in the sense that a Downlink (DL) transmission can be received only in the pre-defined reception windows that the device opens following it's Uplink (UL) data. Class A devices provide the lowest possible energy consumption: the end-device transmits messages using Aloha protocol restricted by a mandatory 1% duty cycle [14]. Normally, no acknowledgements are provided by the gateway, as these are expensive in terms of energy. The DL traffic is mainly dedicated for MAC commands that control the end-device datarate, channel or transmission power [10].

Class B end-devices have additional receive windows determined by the gateway's beaconing interval (1s to 128s). Class C devices allow continuous reception of data. Industrial solutions based on LoRaWAN use class A devices. Class B devices would allow for more feedback from the gateway.

Because of the Aloha-based protocol, collisions of signals can happen at the gateway. Collisions happen if two or more packets are sent on the same channel, with the same SF and they overlap in time. In case of collision, all of the collided packets are dropped. For two packets arriving at the gateway with the same SF at the same time, the gateway can decode one if it has a power greater than 6dB above the other peak [99]. As for different SFs the rejection gain ranges from 16 to 36 dB, we can consider there is no inter-spreading factor interference.

In what concerns non-LoRa interferers, due to the redundancy associated with wideband spread-spectrum modulation, LoRa is resilient to the interference mechanism that appears as bursty short duration pulses [100]. According to [101] , LoRa can tolerate a non LoRa interferer if this is less than 5dB (19.5dB) above desired signal for SF=7 (12) for the case of an error coding scheme of 4/6. Being wide-band, a narrow band jamming signal would only add noise on a very small portion of this band and the LoRa signal would still be recoverable. Also, a jammer that floods the channel can easily be detected and dealt with. Moreover, it would need to transmit with high power on a very wide band of the radio spectrum, which poses an important problem for a potential attacker.

LoRaWAN offers a good degree of protection against impersonation, as the end-device needs to be authenticated: a Message Authentication Code (MAuC) confirms that the message comes from an authorized sender. The LoRaWAN network and application layer use EUI64, while the device

specific key, EUI128. AES CCM (128-bit) is used for encryption and authentication [10]. The Network Session Key (NwkSKey) is used for checking the validity of messages, Message Integrity Check (MIC). The Application Session Key (AppSKey) is used for encryption and decryption of the payload.

Regarding replay attacks, LoRaWAN offers a mechanism to prevent them [94]: the MIC of a message, once validated by a gateway, prevents any further occurrences of the same sequence number. The lack of timestamp in LoRaWAN headers, makes it possible for a packet to be replayed at a later time as legitimate, only if the original message was jammed and no message with a higher sequence number has been received by the gateway. This attack could be used to hide the changes detected by the sensor nodes.

A more subtle type of attack that can be classified as a denial of service is represented by the exhaustion attacks: it exploits the communication protocol in order to drain the battery of the device. The lack of authentication at link or network layer can be exploited by making a given sensor node continuously transmit and receive messages. In this case, an attacker sending useless messages can exhaust the node's resources, as these messages are completely received before being discarded. This type of attack is unpractical for the case of class A devices, as the only opportunity it has is after the uplink transmission, which happens normally every few hours. Class B and Class C devices have a higher listening rate and so are much more exposed to this type of attack. Class C devices are in continuous reception mode, having high energy consumption and are more likely to be used for smart home applications, where they can be plugged in, e.g for smart plugs. This is why in this chapter we consider the case of a class B device, as it is more likely to be used in industry due to its good trade-off between energy consumption and reactivity.

3.3 Preamble authentication in LoRaWAN

3.3.1 Exhaustion attacks in LoRaWAN

In the industrial context, sensor networks must be resilient and robust against any external disruption. We address the problem of exhaustion attacks, which is a type of DoS attack for battery powered devices. This type of attack exploits the communication protocol in order to drain the device battery, rendering it inoperative. We consider LoRaWAN class B end-devices, as class A devices have a reactivity limited to the transmission rate of uplink messages, although the attack still applies. Class B allows for an efficient downlink communication at the expense of increased power draw due to periodic listening for beacons.

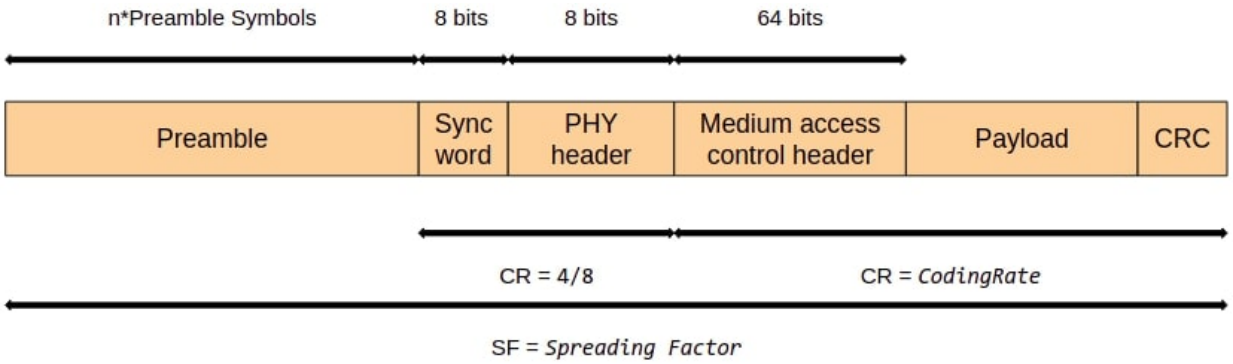
To carry out this attack, an attacker sniffs the medium for any downlink message addressed to the target end-device (for class A a node may listen for the uplink message to be able to attack the downlink windows afterwards). Then the attacker would synchronize with the listening window of the LoRaWAN device and send a single but very long packet on each listening window. Since the device needs to receive the whole packet in order to calculate the network level MAuC before discarding it, it would be forced to receive up to ≈ 250 bytes of payload message which could take up to 14 seconds depending on the SF in use.

3.3.2 Early message authentication

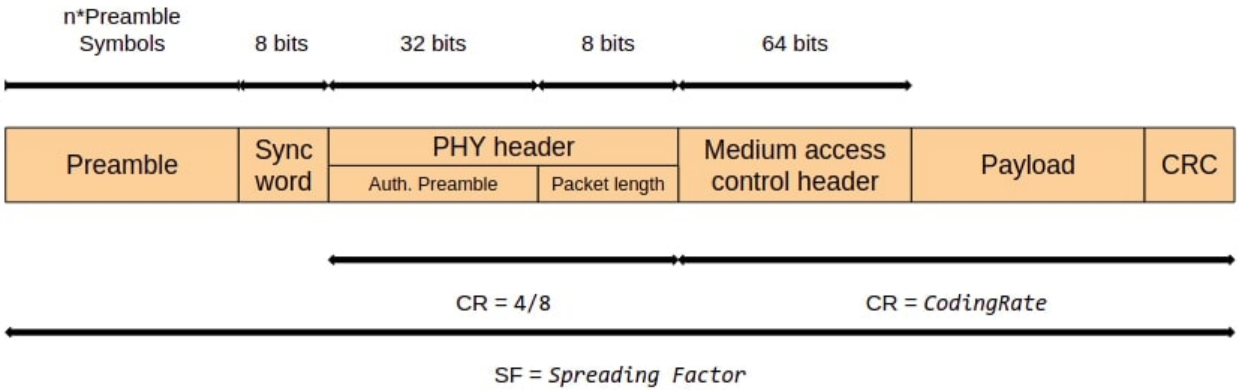
We propose a verification method at the PHY-layer that is extensible to any wireless protocol. The use of an AP is able to reject malicious packets sooner, saving energy and so guaranteeing network's long-term availability. Fig. 3.1a shows the packet structure as used in LoRaWAN, while Fig. 3.1b shows the proposed packet structure that would lead to a sooner verification of a message authenticity. This allows discarding the malicious packet after receiving the first 4 bytes after the synchronization word.

Normally, the MAuC is generated using the payload of the message it accompanies. As we want to be able to reject the packet sooner, we would not have the payload in order to compute the MAuC. Therefore we must generate a MAuC that is known at the start of the reception frame and that is different at each frame. We propose a token exchange scheme: the end-device uses a token that the gateway will use to authenticate at the physical layer all subsequent communication.

The end-device does not have to authenticate its messages to the gateway at the physical layer. This is because the gateway does not have energy constraints. In Fig. 3.2 we propose to use a frame counter as the token upon which the MAuC will be generated. Given that each reception frame has a fixed duration, the gateway, once it has obtained the frame counter from the node, can easily predict the frame counter no matter how much time has passed. If the gateway knows the counter value ϕ for the frame f_i, to know the counter for a frame in the future f_m it simply needs to divide the elapsed time between f_i and f_m by the frame duration. The resulting value is then added to ϕ to obtain the frame counter for f_m.



(a) Packet structure as defined by LoRaWAN



(b) Packet structure including authentication preamble

Fig. 3.1 Packet structure: a) as defined by LoRaWAN; b) new structure allowing early authentication of the packets received by an end-device.

3.3.3 Securing the token exchange

The token is sent to the gateway once the end-device boots. This ensures that reception is available as soon as possible. To deal with possible de-synchronization, the token is retransmitted periodically at a predefined interval. The token exchange must be coupled with a strong cipher algorithm such as Advanced Encryption Standard (AES). The application key is distributed during the device manufacturing and it is only known by the device itself and by the network server. A possible vulnerability arises when an adversary can capture the moment a device reboots because it will then know the exact token value. The token value can be initialized during manufacturing to a random value for each device, which would remove this vulnerability.

The MAuC is then calculated as follows:

1. The frame counter value k is computed
2. The value k is padded to up to 16 bytes. The 16 byte block is then encrypted using an AES algorithm with a shared key
3. The last 4 bytes of the resulting cipher text represents the MAuC value.

The end-device follows the same procedure in order to determine the accepted MAuC value for the current reception frame.

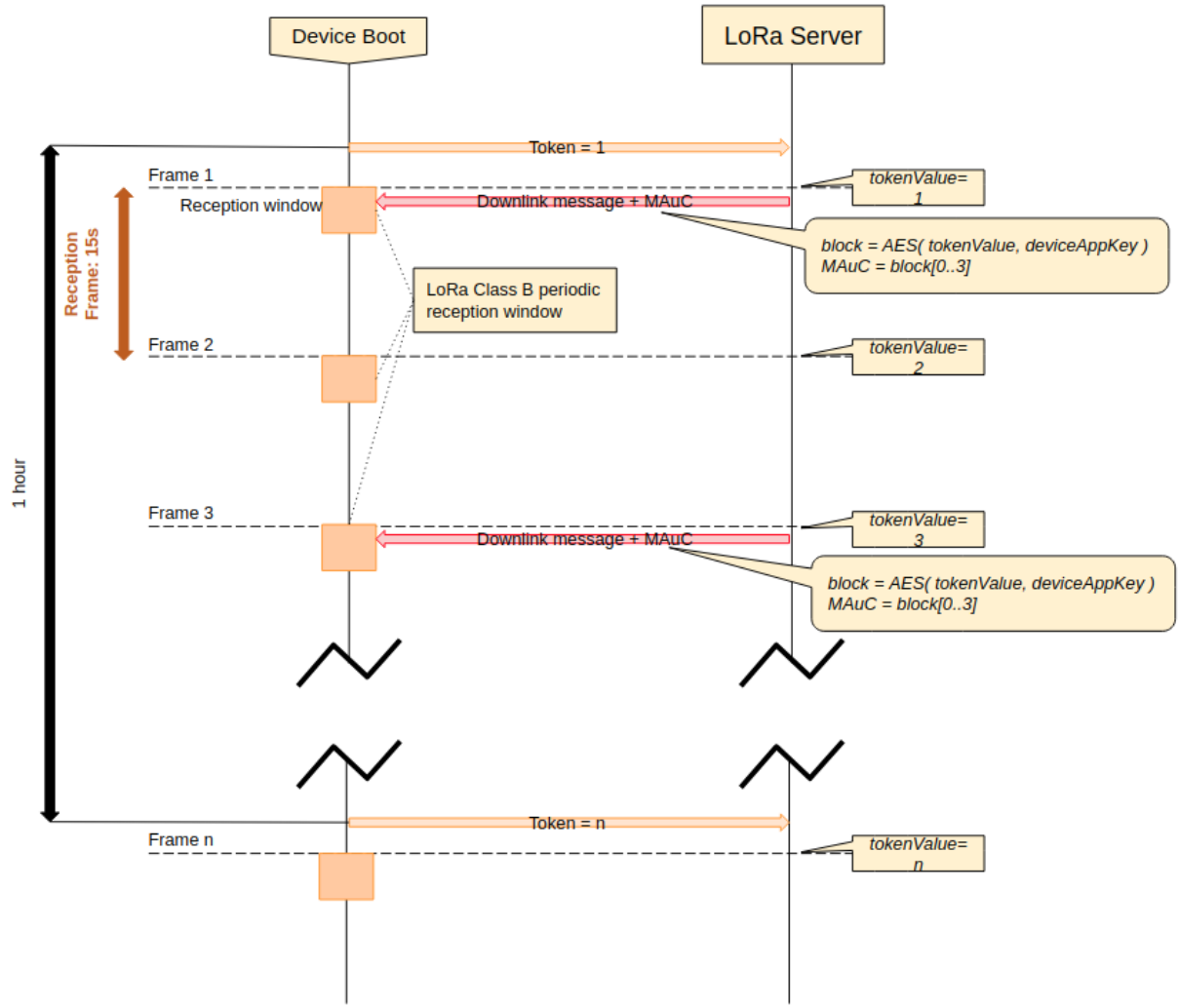


Fig. 3.2 Token exchange as a form of message authentication: the first token value is generated when the end-device is booted. The token value is then used to authenticate any incoming packet.

Regarding the suitability of our proposed approach in face of a brute force attack, which is an attacker trying to sniff and guess the next MAuC value, the 64-bit token makes the search space close to intractable, even if using rainbow tables [102]. Also, a cipher-text attack would result difficult because when using either a cipher or a hash on the token, the entire result of the computation, usually a 16 bytes block, is not sent over the air, only 4 bytes of it, in the MAuC field. The adversary does not then have a complete set of cipher-text to attack the cipher algorithm.

3.4 Evaluation

In this section, we evaluate the efficiency of using authentication preambles in a realistic LoRaWAN setting. Fig. 3.3 shows the behavior of an end device in both cases of using and not using authentication preamble. On the left side of the figure we can see that if the end device does not use AP it will stay in reception mode for all the packets sent by an attacker, without being able to discard them before their transmission is completed. On the right side of Fig. 3.3, an end device implementing AP is able to discard the attacker's packet, 4B after the synchronization word composing this message.

The experimental setup we used was composed of a Loadsensing sensor node [15], a LoRaWAN gateway and a sniffer that would synchronize with the node and would send packets, acting as if it was a legitimate gateway. The end device is implementing LoRaWAN class B and is set to wake up every 15s to listen for DL messages coming from the LoRaWAN gateway. The predefined listening period for downlink packets from the gateway is 300ms. The sensor node is configured to take samples at a very low rate (12h - 24h), and thus most of the time is sleeping.

In a typical scenario, the DL messages from the gateway are very limited, so the end device would normally wake up, listen for 300ms and go back to sleep, as there would be no packets to hear. This is why, in our setting, the attacker is the only one sending DL messages to the end device. The attacker would send every 15s (the wake-up period of the end device) a packet with the payload set to the maximum allowed value for SF12, (242B)[10], transmission that takes 14s. The goal of the attacker is to keep the node awake as much time as possible.

3.4.1 Energy exhaustion attack: end device does not implement AP

Fig. 3.4 shows the end device current consumption versus time, for the case when AP is not implemented and the attacker sends packets with the maximum payload size.

We can observe the periodicity in node's activity: the node wakes up, starts receiving the packet and cannot discard it, it has to receive it completely before being able to see that it is not a legitimate packet. For obtaining this plot, we used the PowerScale tool [103], taking measurements for 1 minute and repeating the tests 1000 times. There are $10,000$ samples/s, so the whole test shows $600,000$ samples. We can see in Fig. 3.4 that the packet reception lasts for approximately 14s out of the 15s configured as beaconing interval.

3.4.2 Energy exhaustion attack: end device implements AP

This scenario is similar to the previous one, with the only difference that the end device implements the AP and it is expecting the gateway to send the correct message authentication code described in Section 3.3.3.

In Fig. 3.5 we can see that when using AP, the node wakes up every 15s, but it is able to discard attacker's packet after checking for the existence of AP. As the attacker does not have any AP, or it is not able to generate the correct MAuC, the end device is able to go back to sleep state. Every

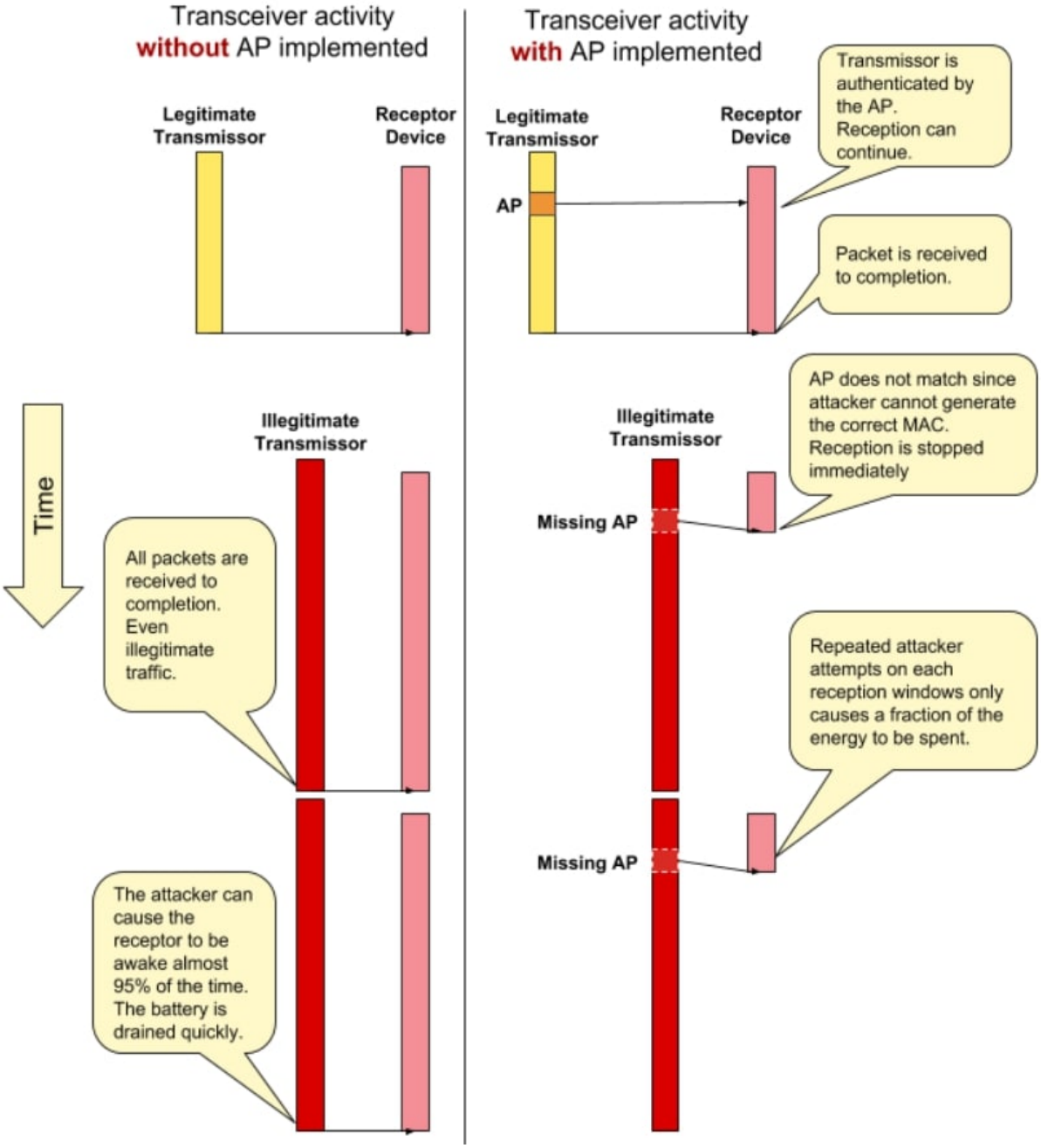


Fig. 3.3 The behavior of an end device subject to an energy exhaustion attack: (left) the end device does not implement AP; (right) end device implements AP.

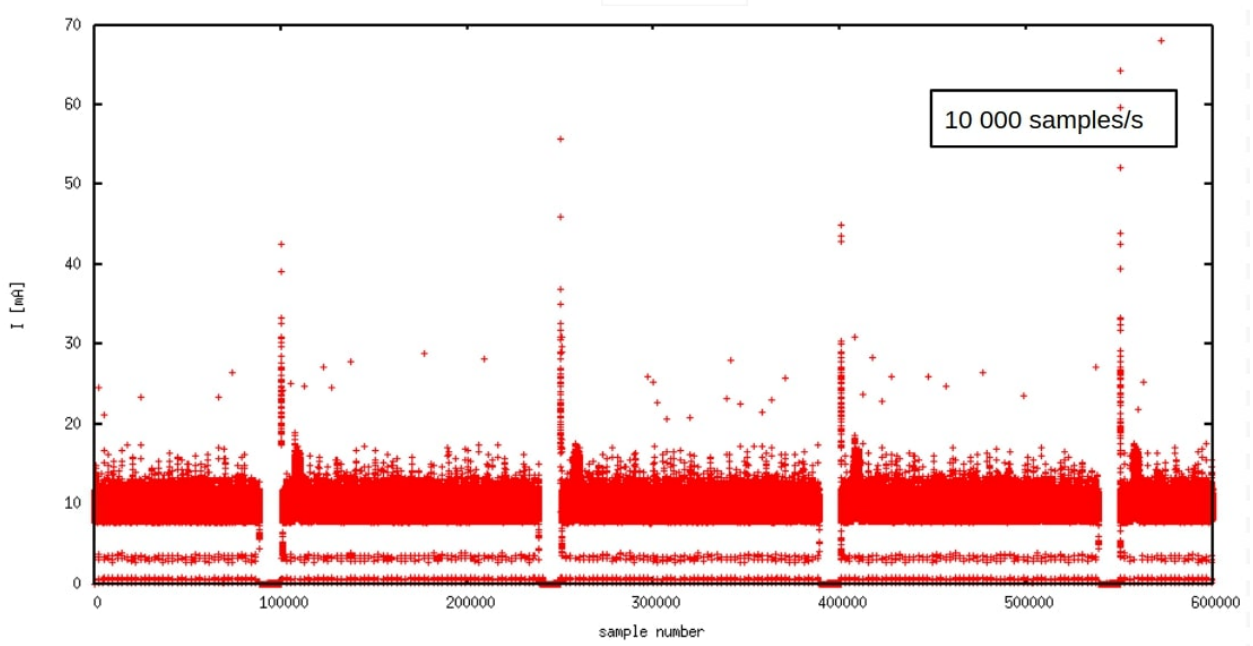


Fig. 3.4 End device current consumption versus time (sample number), for the case when AP is not implemented and the attacker sends packets with the maximum allowed payload size. Listening periodicity: 15s; Packet duration: 14s; Awake time: 14s; Test duration: 1min.

15s the node will wake up for approximately 1s.

3.4.3 Analysis of energy consumption

In this section we analyze the energy consumption of a node running in a typical Class B LoRaWAN network. The results have been extrapolated from real measurements conducted on a LoadSensing device [15]. The performed tests aimed at understanding what is the energy drained from the battery of a node in normal operation conditions and under a DoS attack, when (not) implementing the preamble authentication. The parameters we used for the analysis are presented in Table 3.1.

Table 3.1 Technical parameters

Parameter	Value	Unit
Battery Capacity	23.2	Ah
Sensor power draw	2.2	Ah/year
Sensors UL data	0.025	Ah/year
DL listening	1.983	Ah/year
AP Beacon	0.17	Ah/year
Attack power drain (no AP)	94.024	Ah/year
Attack power drain (with AP)	6.354	Ah/year

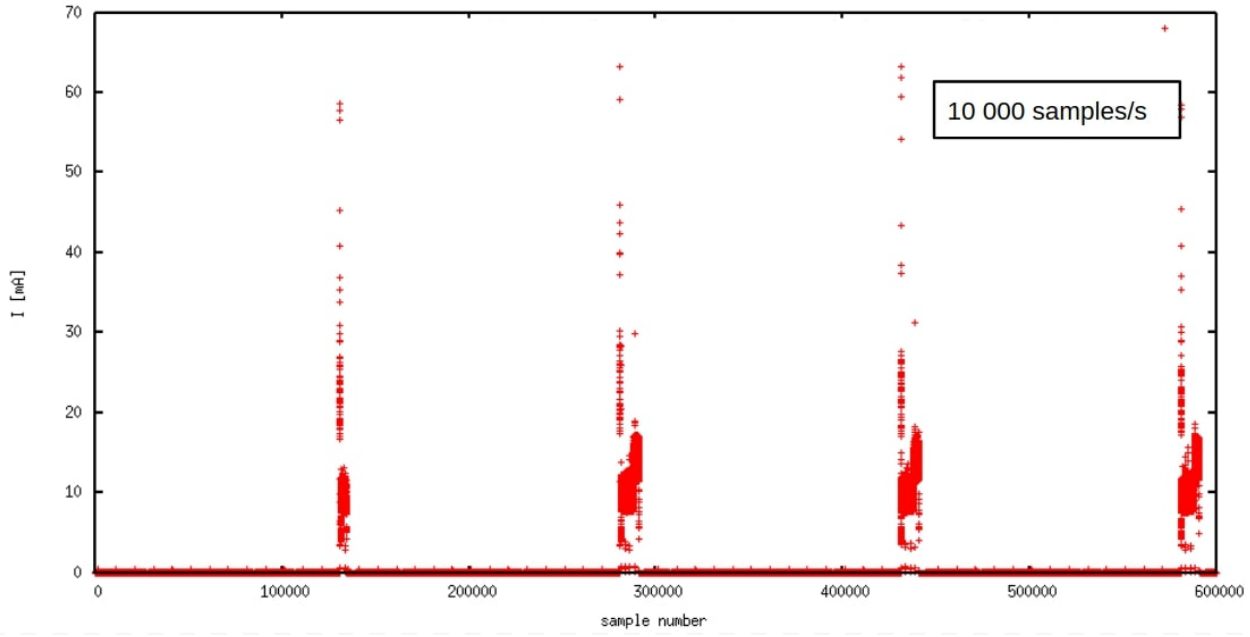


Fig. 3.5 End device current consumption versus time (sample number), for the case when AP is implemented and the attacker sends packets with the maximum allowed payload size. Listening periodicity: 15s; Packet duration: 14s; Awake time: 1s; Test duration: 1min.

Table 3.2 Expected battery lifetime of a sensor node

Operation Mode	Authentication Preamble	Battery Duration
Normal Operation	No	5.51 years
	Yes	5.3 years
DoS Attack	No	2.9 months
	Yes	2.65 years

We considered the real case of a Loadsensing end device using 4x Li-SOCl2 primary batteries of 5.8 Ah each. It sends a 20 B data message per day, transmission which takes about 5s in the air. The used radio has a transmission energy consumption (at 7dB) of 18mA. Reception energy consumption is 11.5mA.

The sensor node wakes up every 15s to listen for DL data. In normal operation mode the listening duration is 300ms, while in the case of a DoS attack, if the node implements the AP, it can discard the packet after 0.9s. If no AP is implemented, the node stays in reception mode for 14s.

In Table 3.2 the expected battery duration of the device is presented, considering a normal network operation and when a DoS attack is performed. Preamble authentication is considered incurring an extra overhead of 4B which causes extra energy consumption.

As observed, the impact of the preamble authentication technique on the energy consumption under normal operation conditions (no attack) is small, reducing the battery less than 4%.

These results confirm the fact that using the authentication preamble can reduce 91% the effect of an exhaustion attack, this means increasing the battery lifetime from 0.24 years to 2.65 years.

The battery lifetime reduction due to using the authentication preamble under an exhaustion attack is 53.9% (this is the worst case scenario, when in each 15s the node has to stay awake to check the preamble of the malicious message), compared with 95.6% when not using the preamble.

3.5 Conclusions

In this chapter we studied the suitability of authenticated preambles to cope with exhaustion attacks in LoRaWAN networks. We show that a short 4B preamble can incur a small energy consumption overhead of less than 4% in operational networks while it prevents malicious attackers to significantly impact the operation of a network. This chapter presented results based on an industrial data logger platform used commercially for critical infrastructure control and monitoring. Even if exemplified on LoRaWAN devices, this technique is extensible to any wireless protocol.

CHAPTER 4

Packet fragmentation in LPWANs

4.1 Introduction

LPWANs are being challenged to offering higher quality of service connecting an increasing number of industrial assets at low costs. The demands for the industrial IoT applications incremented from simple monitoring applications with low traffic needs and low power consumption, to applications where the guarantee of a timely packet delivery is required while maintaining the low power consumption [3]. LPWANs are star-topology networks composed of battery-operated devices mostly deployed in harsh environments, where the battery replacement is costly. Those devices are required to operate with very low power consumption in order to provide 10 years of network lifetime. The data being sent by sensor nodes to the gateway consists in a few packets/day, most of the times without being acknowledged, as a way to maintain the sensor nodes more time off and to satisfy the 1% duty-cycle restrictions imposed by ETSI in the license-free bands [14].

In this context, with duty-cycle restrictions and limited transmission opportunities, packet fragmentation opens up new challenges and opportunities to be explored. In order to analyse this, we consider the approach taken by the IETF LPWAN WG where packet fragmentation headers are being defined with an overhead of one extra byte [104]. This limitation is mandated by the technology characteristic that impose strict limitations in the amount of data to be sent. In this chapter, we study the impact of fragmenting the data packets considering 1B fragmentation header for the LoRa particular use case. The LoRa technology developed by Semtech [78] was chosen as a case study for this work, as it is one of the most adopted technologies for the industrial IoT applications [105]. These networks have very low data rates, ranging from 0.3 kbps to 27 kbps depending on the Spreading Factor (SF) in use. More details about this technology were provided in Chapter 3.2.

4.2 Related work

In the current literature, we found very few references related to packet fragmentation in LPWANs. Most of them are related to allowing the transmission of IPv6 packets or of large volumes of data over LPWANs. The rest of the available information related to fragmentation refers to the condition only when transmission cannot be completed or there is the need to adjust the packet size to the channel conditions.

The 6LoWPAN [106] (IPv6 over Low-Power Wireless Personal Area Networks) working group of IETF is working on encapsulation and header compression mechanisms as an adaptation layer that allows IPv6 packets with a Maximum Transmission Unit (MTU) requirement of 1280B to be sent and received over IEEE 802.15.4 based networks [107, 108]. While the frame payload of 802.15.4-2006 has a size of 81B to 102B (depending on IPv6 Headers), supporting 4-5B fragmentation headers, many LPWAN technologies have a maximum payload size that is one order of magnitude below it, so this header size causes a high overhead.

The IETF LPWAN [109] WG is developing a set of mechanisms to compress IPv6 on top of LPWAN networks such as LoRa, Sigfox, NB-IoT or WiSUN. The approach is based on defining static IPv6 contexts and compressing all the header information that can be mapped to a context. Packet fragmentation is also considered with a header overhead of one extra byte.

The work in [110] presents a data fragmentation scheme for the IEEE 802.15.4 wireless sensor networks that has as a purpose the avoidance of packet collisions in densely deployed networks. Fragmentation is introduced only for the case when the remaining number of back-off periods in the current superframe are not enough to complete the data transmission procedure and the sensor nodes would normally hold the complete transmission until the next superframe. If two or more nodes have these pending transmissions for the next superframe, they collide at the beginning of it and cause the waste of the channel utilization. Their proposed fragmentation scheme is to adapt the packet size to the number of remaining back-off periods in the current frame, send it, and send the remaining fragment at the beginning of the next super-frame, after performing Clear Channel Assessment (CCA), showing good improvements in aggregate throughput and collision probability.

In [111] packet fragmentation is used for allowing the transmission of large volume of data in WSNs, such as images and videos, and block acknowledgements are introduced for reducing the costs of exchanging control packets. As they consider S-MAC in their work, errors are caused by BER rather than by collisions, as S-MAC has frames with listen and sleep periods. Their results show that there is no single best fragment size, but that it is dependent on the characteristics of the deployment environment.

In [112] an optimization work of the optimal packet size in wireless sensor networks is presented, with the varying of packet size based on the channel conditions for throughput enhancements. There are no collisions assumed in the network, and again the only source of error is the BER of the channel. It was shown that applying Forward Error Correction (FEC) can significantly improve the energy efficiency of the communication links.

Other works like [113, 114] showed that the optimum packet length is dependent on the application and communication protocol, but this is not necessarily studied in a fragmentation scenario.

Our work focuses on the analysis of the impact of using packet fragmentation in industrial LPWANs operating in 1% duty-cycle restricted channels, where the data does fit in the frame, but the advantages of using smaller fragments is studied. We take into consideration only the errors caused by colliding packets, while considering perfect channel conditions. We consider that in these Aloha-type networks, the impact of the channel becomes of low importance with the network densification, as collisions drastically increase and limit their performance. This is contrary to works like [111, 112], where as there are very few or no collisions in the network, the impact of the channel BER must be taken into account. We drive our work towards the impact of fragmentation on reliability of communication, network densification and energy consumption.

4.3 Analysis scenarios

In order to match the characteristics of real LPWAN deployments, the analyzed networks consist of sensor nodes accessing the channel using the Aloha protocol (as in LoRaWAN). In average, each sensor node has data to send every 10 seconds, in order to ensure a full buffer. The data packets generated by the nodes have a payload of 250B, with 1B of header. This means that every time fragmentation is used, the payload is divided and one extra Byte of header is added to each fragment.

As the LPWAN networks are very restricted from the energy and duty-cycle point of view, no packet acknowledgements are considered in the downlink, so the collided packets are lost. Note that an Acknowledgement (ACK) has a two-fold impact on LPWAN performance: it increases the channel utilization and reduces the time in sleep state of the nodes (thereby increasing energy consumption). However, the lack of ACKs impedes quality of service guarantees.

The considered scenarios allow us to compare the implications of using packet fragmentation in ideal networks (no duty cycle restrictions) with those in real, 1% duty cycle restricted network deployments. For both cases, the impact of packet fragmentation on energy consumption, throughput, goodput and end-to-end latency is studied when using two data rates, corresponding to LoRa SF7 (5470 bps) and SF12 (250 bps) in 125 kHz bandwidth channels. We consider that these performance metrics are the most relevant in the case of LPWANs, and LoRaWAN in particular.

4.4 Performance evaluation metrics

For the sake of clarity, in this section we present basic terms characteristic to LoRa networks, followed by definitions of the performance metrics we used in our simulations. In LoRa networks, the notion of Time on Air (ToA) is used for defining the packet or fragment transmission duration at a given spreading factor and channel bandwidth, and it is expressed as the sum of the preamble duration and payload duration after converting their respective lengths from bytes to symbols [19].

The preamble is a sequence of a programmable number of symbols used for synchronizing the receiver and enabling it for the detection of the following LoRa chirps that make the user payload. The symbol period is dependent on the channel bandwidth (BW) and spreading factor (SF) used:

$$T_{sym} = \frac{2^{SF}}{BW} \tag{4.1}$$

A LoRa symbol is composed of 2^{SF} chirps, which cover the entire frequency band, so the SF is defined as the logarithm in base 2, of the number of chirps per symbol used for modulation [115].

As the LPWAN networks operating in the 868MHz ISM band in Europe have to restrict their duty-cycle to 1% per channel, each node will send one packet or fragment of data and then switch to sleep for the amount of time that it is obliged to stay off because of this restriction. The off period (T_{off}) is defined as:

$$T_{off} = ToA \times \frac{100 - DC}{DC} \tag{4.2}$$

where ToA is the time on air and DC is the duty-cycle in %. As we can see, T_{off} is dependent on the packet length by its corresponding time on air (ToA). When using packet fragmentation, the node has to stay off until the T_{off} corresponding to the fragment ToA expires and then wake up to send the following fragment.

The impact of fragmentation on energy consumption is defined as an overhead, so as to show how much extra energy is consumed when fragmenting with respect to the case when the same data is sent unfragmented. This is expressed as

$$Energy\ Consumption\ Overhead\ [\%] = \frac{E_f}{E} \times 100 \tag{4.3}$$

where E and E_f represent the energy consumption for sending the data unfragmented and fragmented, respectively. The energy consumption is proportional with the number of packets being sent and with the packet duration.

The goodput is defined as the ratio between the number of packets that arrive at the destination undamaged by collisions (N_u) and the total number of packets sent in the network (N_{pkt}). For the case when fragmentation is used, the damage of at least one fragment causes the loss of the complete packet, since no retransmissions are considered. This is the case in most of the industrial LPWANs.

$$Goodput[\%] = \frac{N_u}{N_{pkt}} \times 100 \tag{4.4}$$

In our analysis, the throughput is defined as the data successfully received by the gateway for a given period of time. That is, $N_{data_u} \times S_{data}$, where N_{data_u} is the number of received packets or fragments and S_{data} is the size of the data packets or fragments.

$$Throughput[bps] = \frac{N_{data_u} \times S_{data}}{Time} \tag{4.5}$$

The average delay in the network is defined as the mean time required for a packet to be received in the gateway. This delay is computed for various scenarios, consisting of variable number of sensor nodes and variable options for fragmentation.

4.5 Results

The results presented in this section for no-duty cycle restricted and 1% duty cycle restricted networks, were obtained with a custom-made Matlab simulator that gathered all the information presented in the previous sections. For reasons of computational complexity, for each analyzed case, the results are averaged over 300 different arrival patterns that were generated for the given arrival rate.

4.5.1 No duty-cycle restricted network

In the following subsections we analyze the impact of fragmentation on throughput and goodput, for the case of a network that has no restrictions in terms of duty-cycle.

4.5.1.1 Throughput

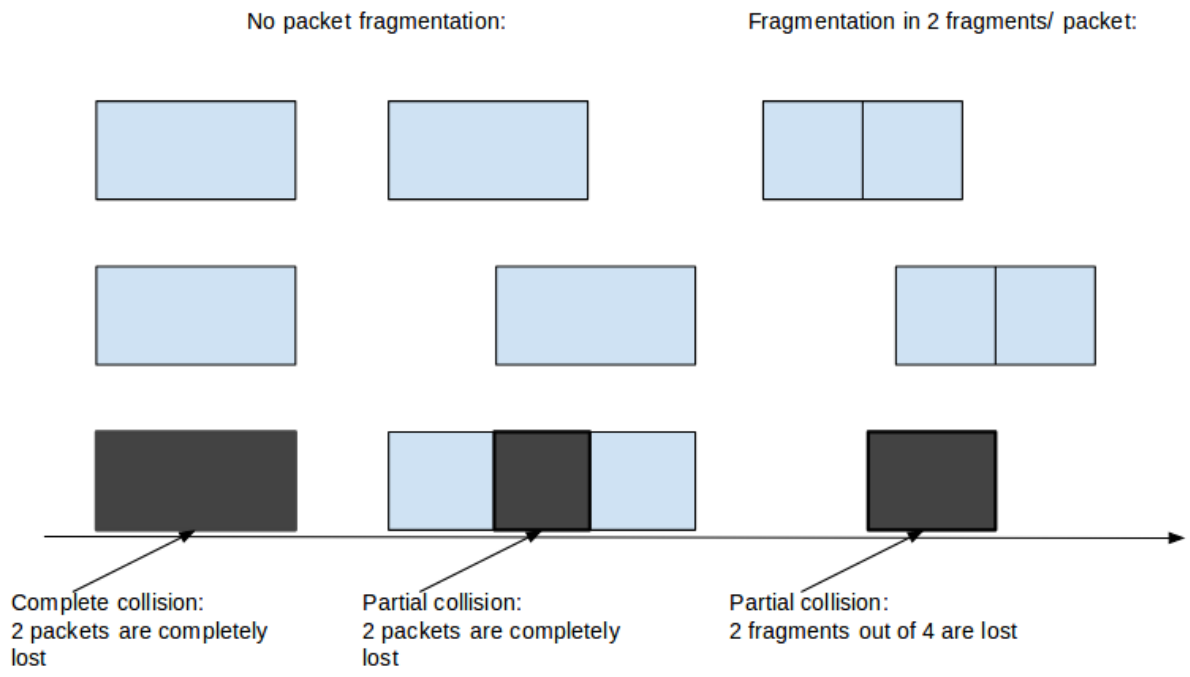
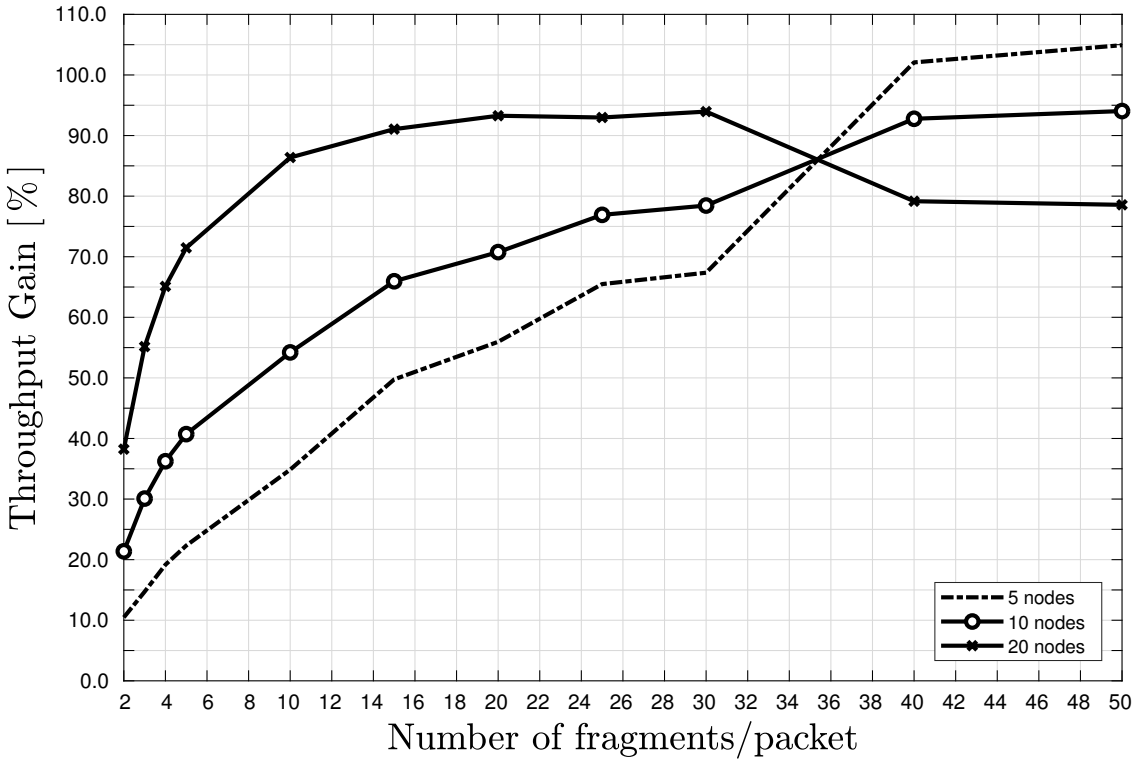


Fig. 4.1 Collisions in Aloha networks: data loss when sending a packet unfragmented and fragmented in 2, respectively.

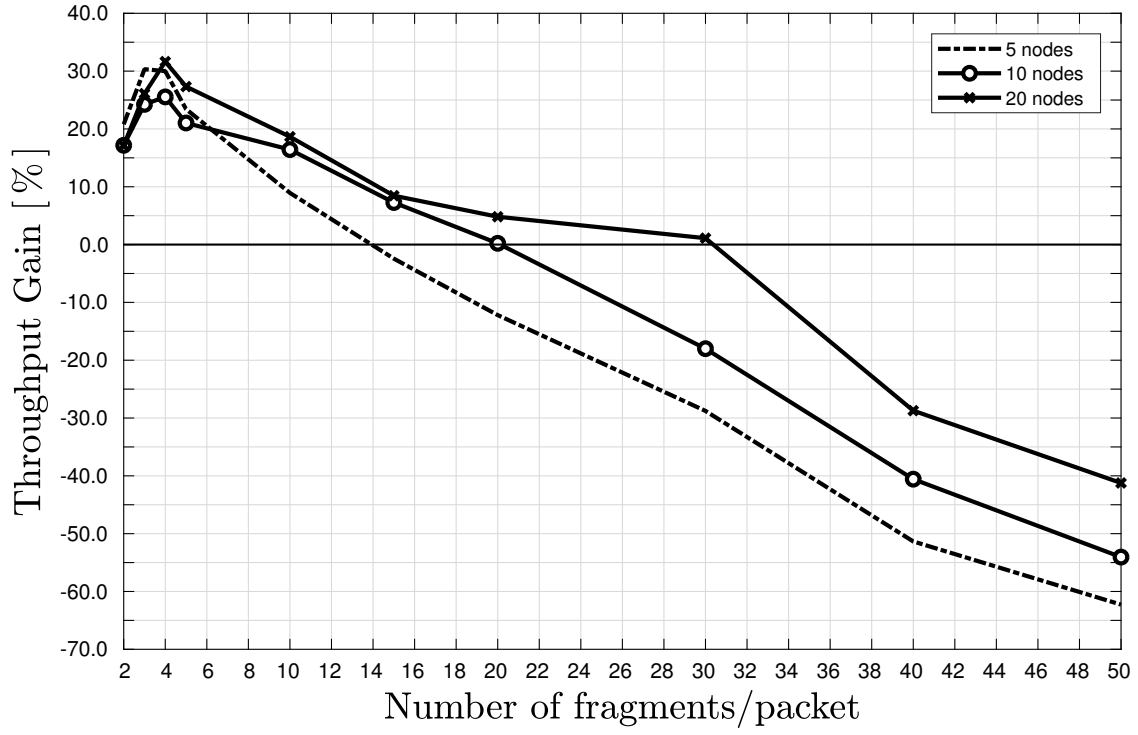
In Aloha networks, collisions can happen anytime during packet transmission and so, for a 250B packet the probability of collision during its transmission is very high, causing the loss of the whole packet. Fragmenting the packet will reduce this loss to the interfering sections of the colliding packets, leaving some of the fragments of the packet unaffected by this collision. This can be seen in Fig. 4.1, where, choosing not to fragment the packet will cause the loss of the complete colliding packets, even if the collision is caused by the complete or only partial superposition of the packets; fragmenting is a way not to lose the complete data when colliding, the damage being restricted to the affected fragments only, and causing an increase in throughput.

Considering various numbers of nodes in the network, Fig. 4.2a plots the relative increase in throughput with respect to non-fragmentation case, for a network operating at SF7. We can see that the impact of fragmentation on throughput increases with the network density. This means that fragmentation could be a valid way towards network densification. Also, increasing the number of fragments/packet too much will cause the decrease of the obtained gain, because of network saturation, which is the case in Fig. 4.2a for a 20 nodes network, after a number of 30 fragments/packet/node. For the other two cases, of 5 and 10 sensor nodes, the network load is too small to be able to witness the same effect as for 20 nodes.

The network load can be ”artificially” increased by lowering the datarate of the network from SF7 to SF12. In Fig. 4.2b we see that the impact of fragmentation is higher in denser networks and that the throughput gain increases up to a certain number of fragments, after which it has a continuously decreasing trend, caused by the increasing amount of collisions in the network.



(a) SF7



(b) SF12

Fig. 4.2 Throughput gain when fragmenting a 250 bytes packet with respect to the case when packet fragmentation is not used, for LPWAN networks composed of 5, 10 and 20 sensor nodes operating at a) SF7 b) SF12.

4.5.1.2 Goodput

The goodput is given by the percentage of uncollided packets for the case where the sensor nodes do not fragment the data and by the percentage of packets that have no damaged fragments for the case when nodes do fragment the data (Fig. 4.3). Even if the throughput increases considerably while fragmenting, it is not used as a measure of the ”reliability” of the transmission, because of the lack of acknowledgements in the network. Adding 1B header to each of the fragments means that the final ToA occupied by a packet is slightly longer than when not fragmenting. This can cause new collisions in the network and then the goodput, in this scenario of one channel and no duty-cycle restrictions, is not better than when not fragmenting. There is a relative decrease in goodput with respect to the non-fragmentation case that can be seen in the Fig. 4.3, where the decrease is higher for denser networks, as the network gets saturated. The same trend is valid for networks operating at SF12, but for these, the goodput is already close to 0% because of their very long ToA considering arrivals at each node occur every 10 seconds.

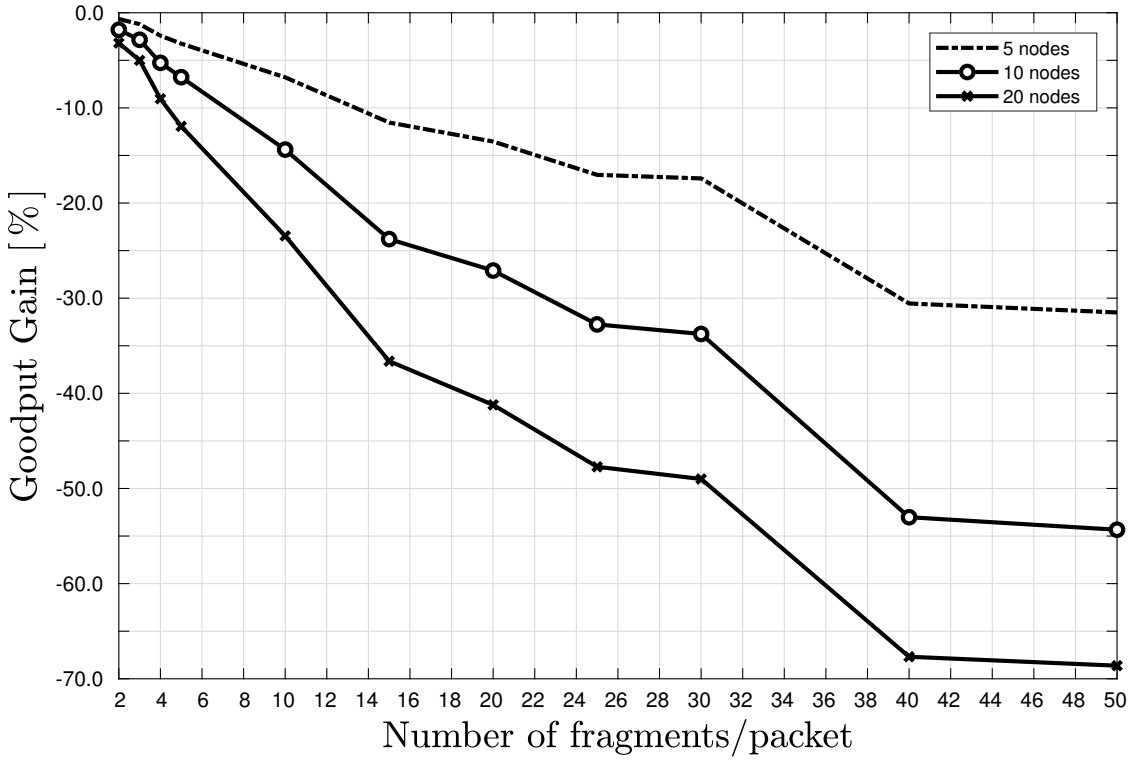


Fig. 4.3 Goodput decreases when fragmenting a 250 bytes packet with respect to the case when packet fragmentation is not used, for LPWAN networks composed of 5, 10 and 20 sensor nodes operating at SF7.

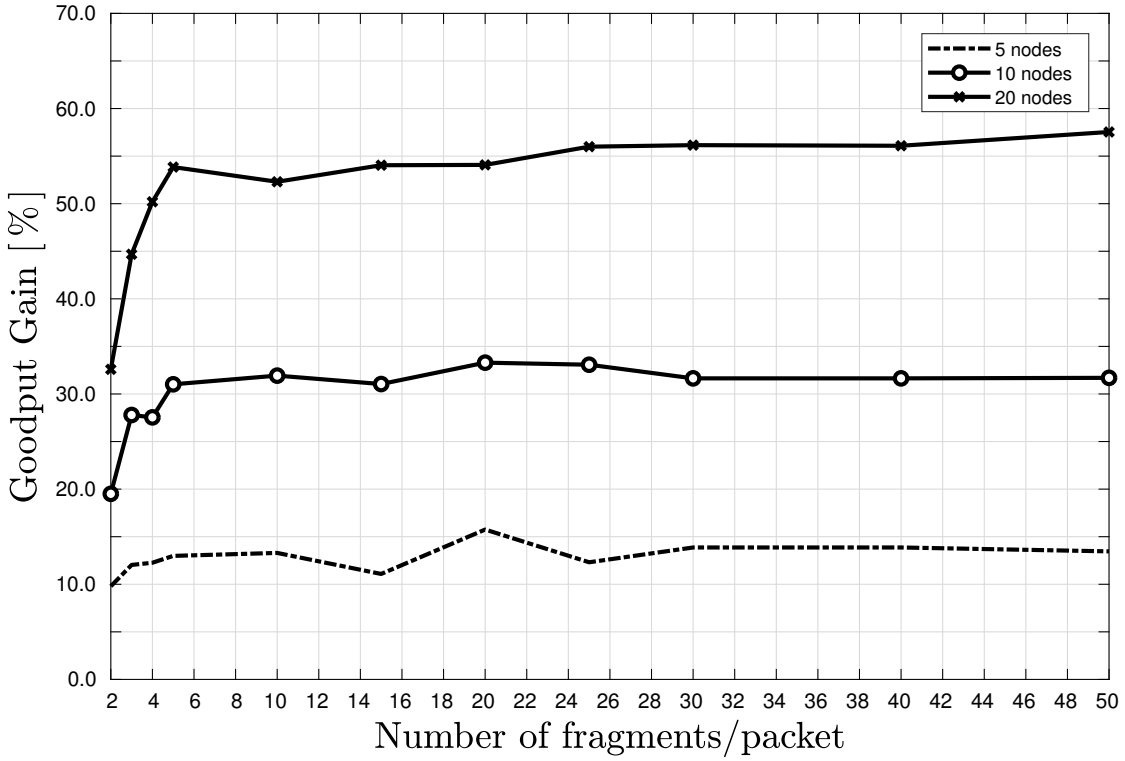
4.5.2 Duty cycle restricted networks

The following subsections describe the effect of packet fragmentation on goodput, energy consumption and end to end delay, considering 1% duty-cycle restricted networks.

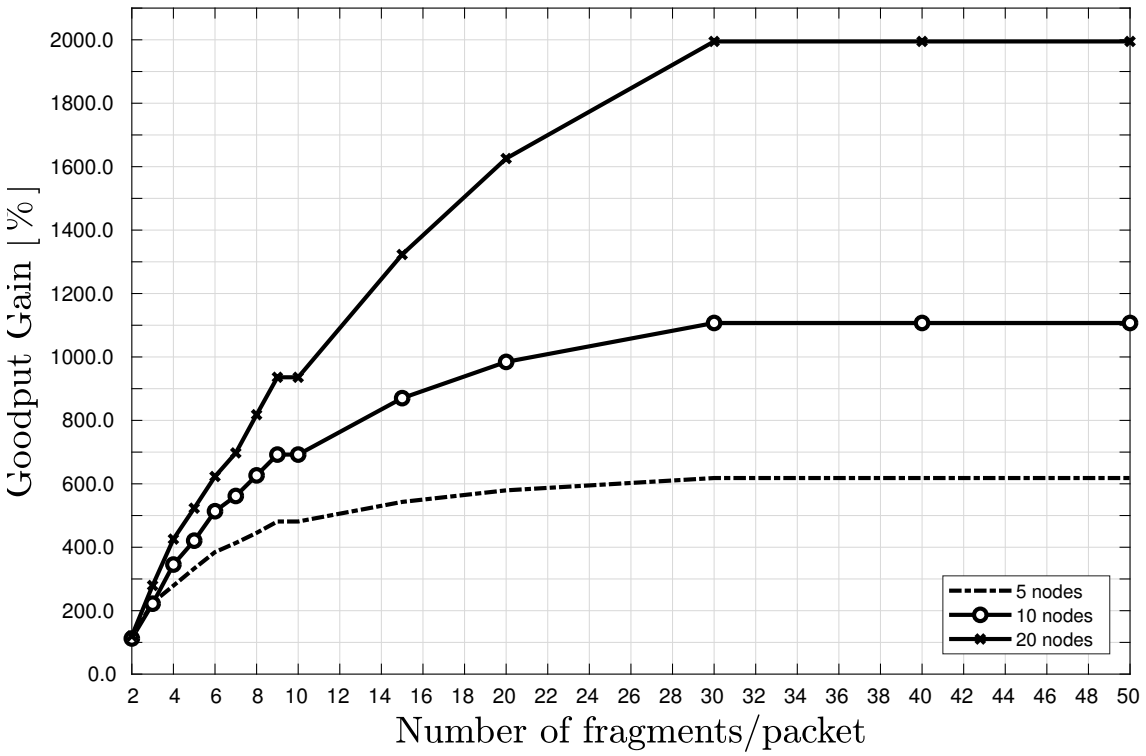
4.5.2.1 Goodput

The results in Fig. 4.4a show the gain in network goodput brought by fragmentation, in 1% duty-cycle restricted networks composed of 5, 10 and 20 sensor nodes operating at SF7. The same can be seen in Fig. 4.4b for networks operating at SF12, but in this case, the gains are much higher, as fragmentation could bring the goodput of such slow networks, for example, from 10% to more than 80% by fragmenting in more than 25 each packet sent in a 5-nodes network. This improvement observed in these slow networks is of high importance, because, in LPWANs/LoRa, using higher spreading factors is a way to reach higher distances, at the costs of higher ToA values. Sending a 250B payload at SF7 is characterized by a ToA value of roughly 0.5s, while sending the same payload at SF12 can take up to 9s.

Again, we notice that the higher the number of nodes in the network, the higher the gain brought by fragmentation. An interesting fact is that in 1% duty-cycle restricted networks, the goodput variation trend is not strictly decreasing with the number of fragments anymore, but increases until it reaches a steady value. This is because of the mandatory T_{off}: the various arrivals in the network will become more separate in time with fragmentation, decreasing the occurrence of collisions, up to a point when fragmentation does not help anymore, but does not decrease the goodput performance of the network either.



(a) SF7



(b) SF12

Fig. 4.4 Network goodput variation when fragmenting a 250 bytes packet with respect to the case when packet fragmentation is not used, for 1% duty-cycle restricted LPWAN networks composed of 5, 10 and 20 sensor nodes operating at a) SF7 b) SF12.

4.5.2.2 Energy consumption overhead

The impact of using packet fragmentation on energy consumption is studied in Fig. 4.5 by comparing the energy consumption of sending the data in one packet, to that of sending it fragmented. As these results are independent on the duty-cycle restrictions of the network, they are valid for both study cases. What has a high impact on the energy consumption of the sensor nodes is the spreading factor used, as the higher the spreading factor, the higher the ToA of the packet and implicitly, the energy consumption for sending it.

Results show that fragmenting brings an energy consumption overhead ranging from 2% when sending the user data in two fragments, to almost 120% when sending each data packet in 50 fragments, for networks operating at SF7. This overhead is caused by the 1B header attached to each fragment that makes the sum of the ToA of the fragments to be higher than the ToA of the original data packet. This energy consumption overhead is much higher for networks operating at SF12, as the slower the datarate, the higher the impact of the fragmentation headers.

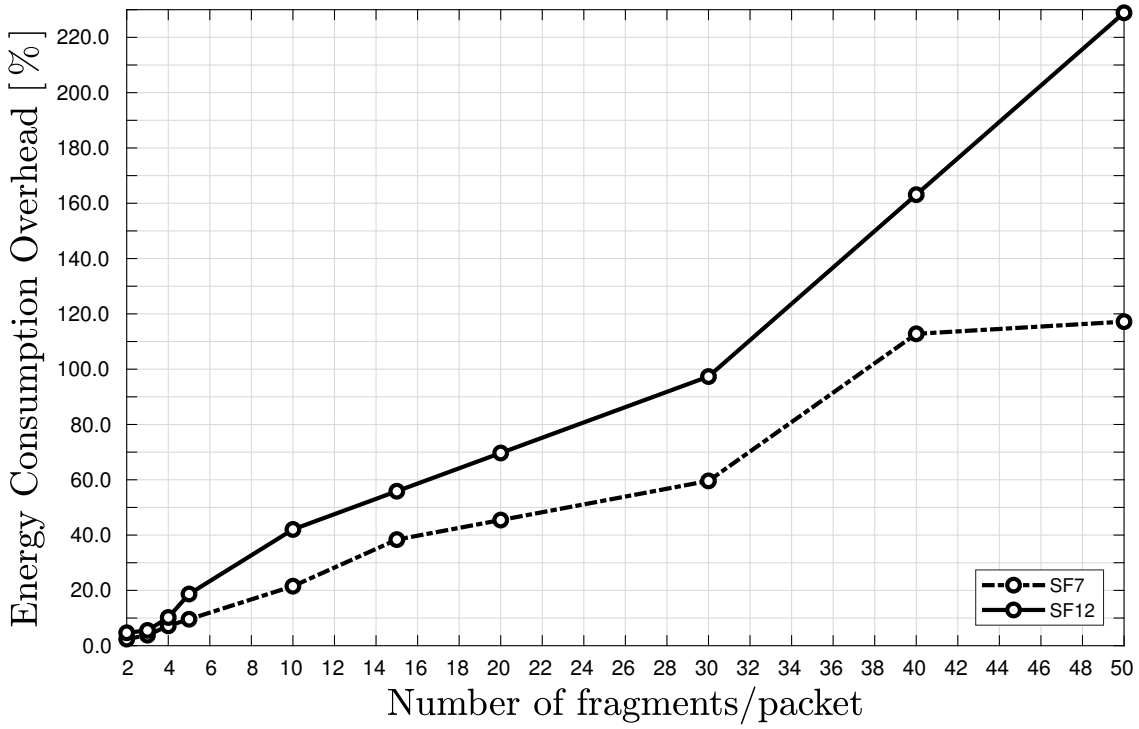
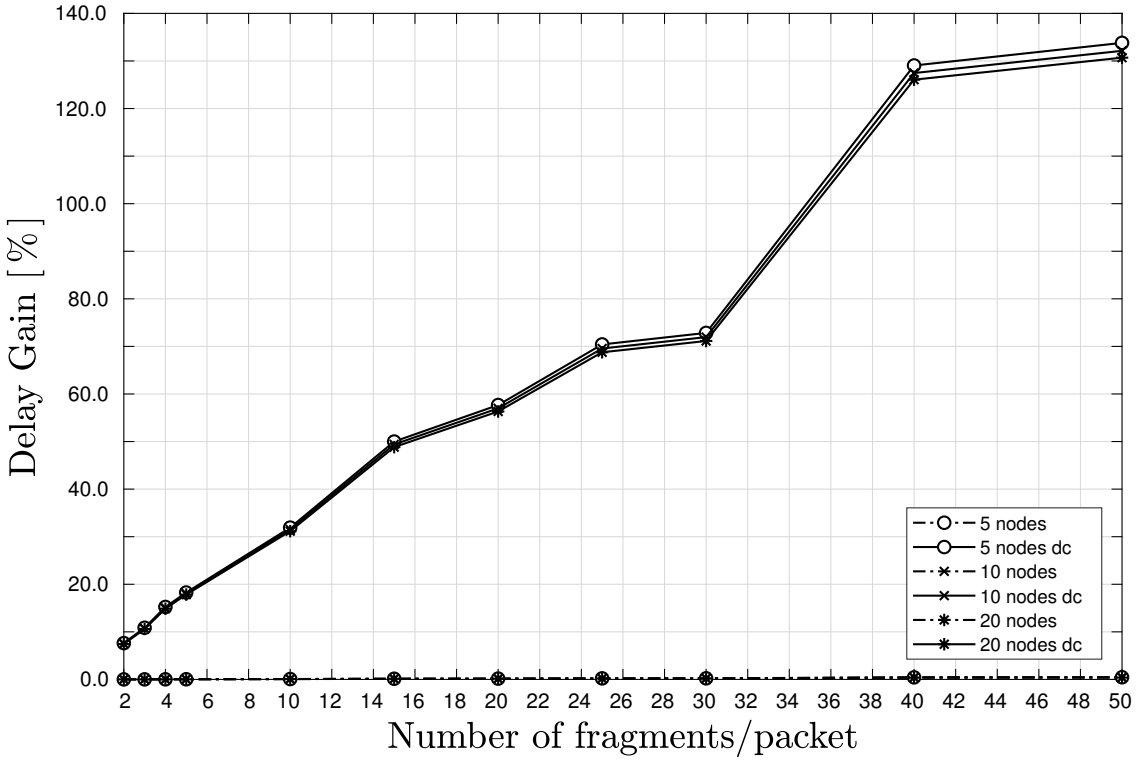


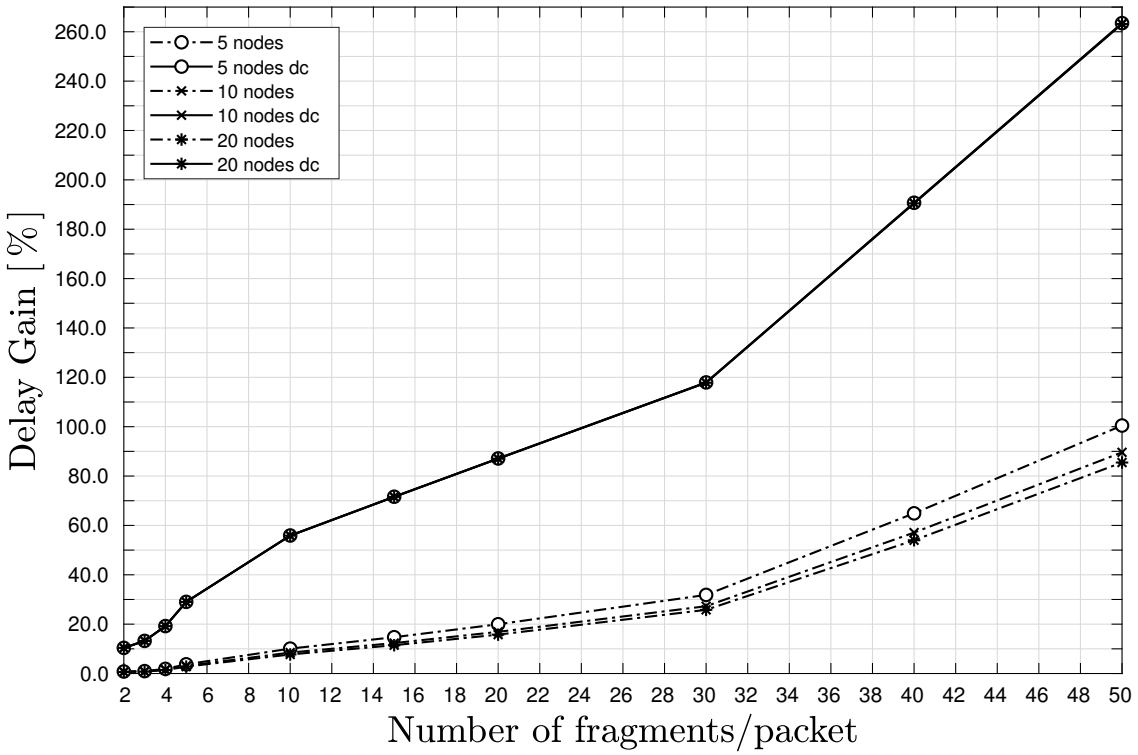
Fig. 4.5 Per-sensor node energy consumption increase when fragmenting a 250 bytes packet with respect to the case when packet fragmentation is not used for networks operating at SF7 and SF12.

4.5.2.3 End to end delay

For the sake of clarity, we choose to describe in this section the impact of fragmentation on end to end delay, for both the cases of an unrestricted and a 1% duty-cycle restricted network, respectively.



(a) SF7



(b) SF12

Fig. 4.6 End to end delay increase when fragmenting a 250 bytes packet fragments with respect to the case when packet fragmentation is not used, for duty-cycle unrestricted and 1% duty-cycle restricted LPWAN networks composed of 5, 10 and 20 sensor nodes operating at a) SF7 b) SF12.

In Fig. 4.6a it can be seen that using packet fragmentation in a network where there is no duty-cycle constraint seems to have no impact on the end-to-end delay in the networks operating at SF7. This is because the network is operating at a relatively high data rate and fragments are sent successively. For the networks operating at SF12, there is an increase of the end to end delay with the number of fragments, as the 1B of header has a higher impact due to their very low data rate, as it can be seen in Fig. 4.6b.

When operating in duty-cycle restricted networks, the end to end delay increases with almost 120% when fragmenting in 50 fragments compared to the case when no fragmentation is used, for networks operating at SF7, as shown in Fig. 4.6a. For networks operating at SF12 (Fig. 4.6b), the end to end delay increase with the number of fragments is much higher, reaching an overhead of 260%, independent of the number of nodes in the network. The delay increase is caused exclusively by the fragmentation headers that cause the nodes to spend extra time off, in order to satisfy the duty-cycle restrictions.

4.6 Conclusions

In this chapter, an analysis of the impact of packet fragmentation in LPWANs has been done following standardization directions defined by the LPWAN working group of IETF. In the presented study we focused on the use of fragmentation despite longer packets do fit in the frame. Our goal was to identify the advantages and disadvantages of sending the data in smaller fragments when deploying duty-cycle restricted LPWAN networks. The analyzed parameters were the energy consumption, throughput, goodput and end to end delay introduced by fragmentation. Two spreading factor values were considered, for seeing the effects of fragmentation in the slowest and fastest cases of LoRa networks. Also, the impact of packet fragmentation was analyzed for the case of an ideal network, where nodes can send data unrestricted, in an Aloha way, as well as for the case of the industrial LPWANs, where data has to be sent while satisfying the 1% duty-cycle restriction of the channel.

The results of our analysis show that packet fragmentation can increase the reliability of the communication for the case of duty-cycle restricted networks: important goodput improvement was obtained with fragmentation, with higher impact in denser and slower networks. In what concerns the optimal number of fragments/packet to be used, there is a trade-off between the goodput performance that can be obtained and the extra costs in terms of energy consumption and latency. Typically, in LPWAN, there are no packet acknowledgements, so in order to increase the probability of successful packet delivery, sensor nodes send a packet multiple times, by default. As with packet fragmentation, the probability of successful delivery increases, it could be a technique to decrease the energy consumption of the network by replacing the technique of multiple packet transmissions.

This chapter unveils that using packet fragmentation despite the packets fit the frame is relevant to scale and densify industrial LPWAN networks. This leads us to identify the need for acknowledgements from the gateway/sink in order to further increase the communication reliability, strategy in which packet fragmentation could also reduce the impact of retransmissions on energy consumption and on network saturation. The following chapter studies the impact of packet fragmentation in LPWANs with a protocol that provides user guarantees of packet delivery while still respecting the duty cycle limitations of the LPWAN common bands.

CHAPTER 5

Packet fragmentation and group acknowledgements (NACK) in duty-cycle restricted LPWANs

5.1 Introduction

In this context of the 5GPPP demands for IoT [3], the technology provider's aim is to evolve the communication technology towards more reliable and scalable long range wireless by adopting new access mechanisms or using dedicated bands [116, 117, 118]. However, we foresee opportunities to leverage the combination of packet fragmentation and group Negative Acknowledgement (NACK) to improve the network scalability, that have not been studied.

The group NACK combined with packet fragmentation will only acknowledge a packet after all its corresponding fragments have been sent and only if there are fragments that need to be resent. This brings reliability while reduces the impact of individual fragment acknowledgements in terms of duty cycle and energy consumption. Yet, packet fragmentation opens up new challenges and opportunities to be explored for improving the efficiency of these very restricted networks under congestion situations [105].

Packet fragmentation has been traditionally seen as an adaptation mechanism to divide MAC layer Service Data Units (SDU) into a set of smaller PDU with a dual purpose: i) better adapt to the channel conditions by reducing the length of the PDU in noisy channels, and ii) fit long SDUs into maximum length PDUs. However, the impact of an aggressive packet fragmentation strategy in strict duty cycle and energy constrained networks such as LPWANs has not been analyzed in depth. An aggressive packet fragmentation consists in using packet fragmentation, despite a frame fits into the PDU. This strategy could be a way to take better advantage of the available channels in the network, as the smaller the fragments, the shorter the time on air and the higher the opportunity to transmit without collisions. For multi-channel networks, using packet fragmentation spreads the transmission of a packet over a set of channels in a more homogeneous way, thereby allowing channel hopping by fragment. Also, in case of fragment/s loss, there is no need to retransmit the entire packet but, only the lost fragment/s, leading to energy savings.

In Chapter 4 we learned that using packet fragmentation has a good impact on the goodput performance of low data rate, duty-cycle restricted networks at the expense of increased end to end delays and more energy consumption per transmitted packet due to the use of 1B fragmentation headers and the increase in the number of access attempts in the network [119].

The aim of this chapter is to shed light on the potential gains of packet fragmentation combined with group NACK in duty cycle restricted LPWANs and show which network conditions this strategy is advisable for. From the best of our knowledge, similar studies have not been done in the existing literature. The analysis carried out in the sequel is based on the LoRaWAN networks, one of the most adopted technologies for the industrial IoT applications [105] that provides very low data rate, ranging from 0.3 kbps to 27 kbps. A more detailed description of LoRaWAN was provided in Chapter 3.2. In the following sections, we describe the aggressive fragmentation protocol we implemented, the used metrics and the simulation setup. We close the chapter with the simulation results and conclusions.

5.2 Aggressive fragmentation strategy

The medium access protocol of LoRaWAN is based on ALOHA random access combined with a duty cycle per channel, which for instance in Europe is set to 1% for the 868MHz ISM band [14]. That is, upon the generation of a packet, the node transmits the packet only if there is a channel available for transmission. Yet, the availability of a channel is defined based on its duty cycle. Specifically, in a channel with a duty cycle DC and for a packet with Time on Air T_{oA}, the channel only becomes available for the node after an off period, namely T_{off}, equal to:

$$T_{off} \text{ [sec]} = T_{oA} \times \frac{100 - DC}{DC} \tag{5.1}$$

When more than one channel is available, the node randomly selects the channel.

The aggressive fragmentation strategy consists in using packet fragmentation even if the frame fits the PDU, in order to make use of the advantages that come out of using smaller data size [119]. For enhanced network performance, we propose a group-NACK scheme, allowing for fragment retransmissions.

According to LoRaWAN specification [10], the payload of a packet needs to be sent together with a frame header and a MAC header. The MAC header (1B) contains 3 bits identifying the message type, 2 bits for the major version of the frame format and 3 bits that are reserved for future use. The frame header (7-21B) uses 4B for the device address, 1B for frame control, 2B as frame counter and up to 15B as frame options.

When the aggressive fragmentation strategy is used, the payload of the generated packet is divided into a set of equal size fragments. To each fragment, a 9B header is added, accounting for the MAC and frame headers.

Throughout this paper, in order to determine in which network conditions the aggressive fragmentation strategy is advisable, the following transmissions strategies will be analyzed:

- Aloha: represents the baseline protocol; the data packets are sent unfragmented and only if the channel is available for transmission, otherwise the packets are discarded.

- Buffered Aloha: the data packets are buffered until a channel becomes available and then sent consecutively, unfragmented and subject to the duty cycle restrictions of the network .

- Buffered Aloha with fragmentation: the data packets are fragmented and buffered until the channel becomes available for transmission; the fragments are sent consecutively and respecting the duty cycle restrictions of the network. If after all the fragments of a packet have been sent, at least one of the fragments is lost, the whole packet is dropped by the gateway.

- Buffered Aloha with fragmentation and retransmissions: in this case, after all the fragments of a packet have been sent, the node waits for a NACK. The NACK indicates which fragments have not been received. In case a NACK is received, it will proceed with resending the missing fragments, following the same protocol and respecting the duty cycle restrictions of

the network. If even after the corresponding retransmission sessions for that packet at least one of the fragments is still lost or corrupted, the whole packet is dropped by the gateway.

The NACK should contain a MAC and frame header in order for the node to identify if the message is meant for it. There can also be a payload attached to it, of variable size. Our choice was to map the fragments status in the NACK on a 0-1 basis, with respect to the sequence number of the fragments: 0, if the fragment was not received and needs retransmission and 1, if it was correctly received at the gateway. This strategy needs the Gateway (GW) to be aware of the number of fragments that the nodes in the network use and that all the nodes use the same number of fragments/packet. Also, the retransmission of a fragment is made using the same sequence number as it had when it was first sent, so that this mapping can be correctly updated.

For the retransmission protocol, we are proposing a scheme in which the last fragment of a packet will be the one triggering the NACK request. The NACK can be received in one of the two reception windows that will be opened by the sensor node after the UL data is sent, as described by LoRaWAN [10]. In case this last fragment is lost, there will be no NACK and the node will continue its activity by sending other packets. If a NACK is received, the node will start sending the fragments that are marked as lost. All these lost fragments that are being resent correspond to one 'retransmission session', as shown in Fig. 5.1 for the case of a network configured to use 3 fragments/packet.

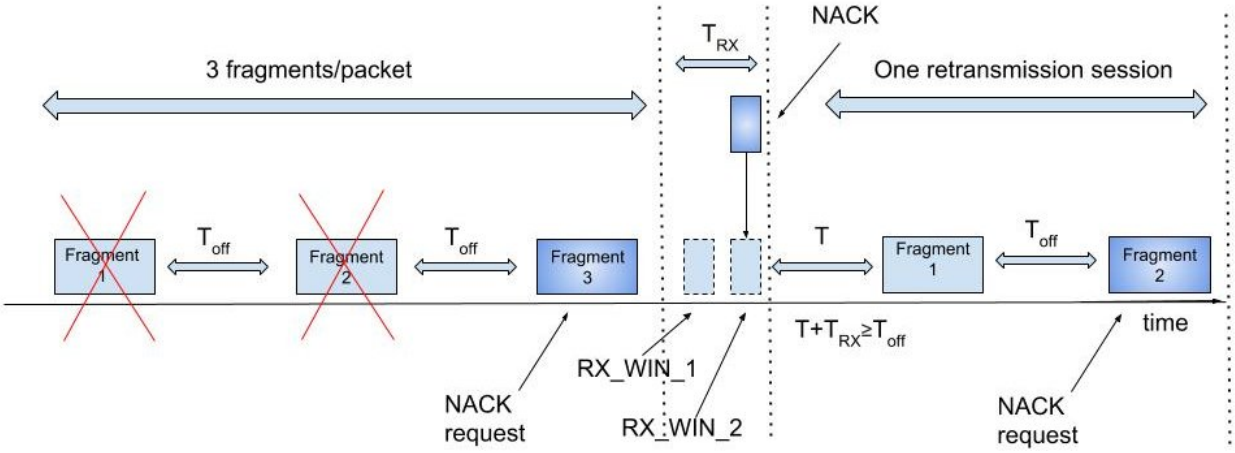


Fig. 5.1 Sending a packet using 3 fragments: the last fragment is the one requesting a NACK. If a NACK is sent by the gateway, it is sent during the first or second receive window opened by the sensor node. The two failed fragments will be sent as soon as possible, after the mandatory T_{off} expires. The last fragment sent can request again for a NACK, if more retransmission sessions per packet are allowed.

We chose to implement the retransmission scheme in this way because these networks are restricted by the duty cycle and by the energy consumption: choosing to ACK each fragment or packet and retransmitting until the ACK is received is too expensive in both duty cycle and energy consumption [120]. This scheme ensures that the nodes will resend only if they are explicitly told so.

5.3 Performance evaluation

In duty cycle restricted networks, after a node sends data, it has to stay silent for the mandatory T_{off} corresponding to that data, as defined in Section 5.2. This means that if the IoT application running on that node asks for more data during T_{off}, the node will drop that data (Aloha) or will buffer it for until it is allowed to send again (Buffered Aloha). This is managed in Fig. 5.2 by the 'DC control' module.

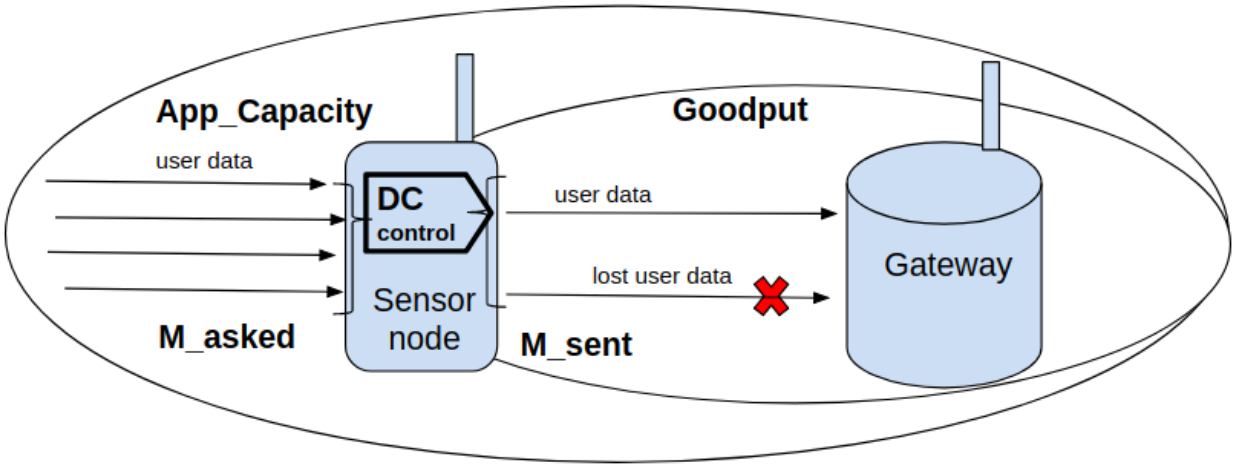


Fig. 5.2 Behavior of a sensor node operating in a duty cycle restricted network: data can only be sent to the GW when the duty cycle allows it.

The first part of this section presents the metrics that we used for the performance evaluation of the transmission strategies presented in Section 5.2, while the second part discusses details of the network simulations we developed using NS3.

5.3.1 Performance metrics

The performance of LPWANs is mainly evaluated in terms of goodput and energy consumption. These, as well as other metrics used in this paper are described in the following.

Goodput: It is defined as the percentage of packets correctly received by the gateway, with respect to the amount of data sent in the network. It is expressed as

$$Goodput[\%] = \frac{M_c}{M_{sent}} \times 100 \tag{5.2}$$

where M_{sent} corresponds to the number of data packets sent in the network and M_c to the number of packets correctly received by the gateway. In case of fragmentation, the packet is only received correctly if all its fragments have been correctly received. M_{sent} does not account for packet retransmissions.

Application Capacity: It is defined as the percentage of packets correctly received by the gateway, with respect to the amount of data asked by the application. This metric allows us to identify the region starting with which packet fragmentation brings a gain to the network performance,

despite the headers overhead. It is expressed as

$$App\ Capacity[\%] = \frac{M_c}{M_{asked}} \times 100 \tag{5.3}$$

where M_{asked} corresponds to the number of data readings asked by the IoT application. This data may not all be sent in the air interface because of the duty cycle restrictions of the network (Fig. 5.2). M_c is the same parameter as defined for Goodput.

Energy Efficiency: it is defined as the total energy consumption of the network divided by the number of successful packets delivered by the sensor nodes to the gateway:

$$Energy\ Efficiency\ \text{[J/packet]} = \frac{E}{M_c} \tag{5.4}$$

where E is the energy consumption of the network and M_c represents the number of correctly received packets at the gateway, as defined for Goodput. The energy consumption of the network accounts for the processes of sending data (packets, fragments, headers) and for processing the NACKs, if it is the case.

Header overhead: This overhead is caused by the need to transmit an additional header for each fragment, as described in Section 5.2. In order to assess this impact, we define the fragmentation header overhead as the percentage of extra energy devoted to transmit a packet in a certain amount of fragments compared to the energy required to transmit the packet in one piece. Therefore,

$$Header\ Overhead\ [\%] = \left(\frac{n_f * E_f}{E_{packet}} - 1 \right) \times 100 \tag{5.5}$$

where n_f is the number of fragments required for sending a packet, E_f is the energy required to transmit one fragment of the respective size and E_{packet} is the energy required to transmit the packet unfragmented. E_f and E_{packet} are proportional to their corresponding transmission duration.

5.3.2 Simulation setup

The simulations have been developed using the NS3 network simulator. We evaluated our approach with network sizes ranging from 1 to 50 sensor nodes for a single gateway and fixed coverage area.

In order to assess the performance of a *dense* network, we chose having all the nodes operating in a single channel and with the same SF: the network operates in a channel of 125 kHz bandwidth in the 868 MHz ISM band and all the nodes transmit with SF=7. The NS3 simulator evaluates the network performance by taking into account not only the packets destroyed by collisions but also the ones destroyed by interference or having a power below the sensitivity threshold of the gateway.

The IoT application running on each node will ask for a fixed amount of data, M_{asked}, independent of the transmission strategy. Because of the duty cycle restrictions of the network, only M_{sent} out of M_{asked} will be delivered to the gateway (as shown in Fig. 5.2).

The data packets have a fixed payload of 200 B, close to the maximum size that LoRaWAN can send using SF7 [10]. If considering other values for the SF, the payload should be modified accordingly so as to be close to the maximum allowed value [10]. In this way, the protocols described in Section 5.2 can be evaluated: Aloha, Buffered Aloha, Buffered Aloha with fragmentation and Buffered Aloha with fragmentation and retransmissions.

Whenever the fragmentation option is used, each data packet will be split into 2 to 5 fragments, but all the sensor nodes in the network will use the same number of fragments/packet. The gateway keeps track of the arrived fragments from the sensor nodes and will be able to provide them with a Group-NACK per packet. After the maximum number of retransmission sessions is completed, the gateway will discard the packets that still have missing fragments. A retransmission session means sending all the fragments that a NACK marked as lost or damaged (see Fig. 5.1).

5.4 Results

In the following, the metrics defined in Section 5.3 are analyzed in order to determine if and when the aggressive fragmentation strategy is advisable for the case of duty cycle restricted LPWANs.

5.4.1 Network goodput

The network goodput (Fig. 5.3) starts with a value of 100% for any transmission strategy when there is only one device in the network. This value decreases as the number of devices (and collisions) in the network increases. Aloha and Buffered Aloha (B.A) will deliver almost the same goodput performance, as they only differ in timing.

When using B.A with fragmentation and retransmissions, the variation of the network goodput with increasing number of devices becomes smoother. Also, the higher the number of fragments/packet, the higher the increase in goodput, as more correct packets are delivered correctly to the gateway. This happens because having smaller data packets reduces the probability of collisions and increases the probability of receiving NACKs (Fig. 5.4).

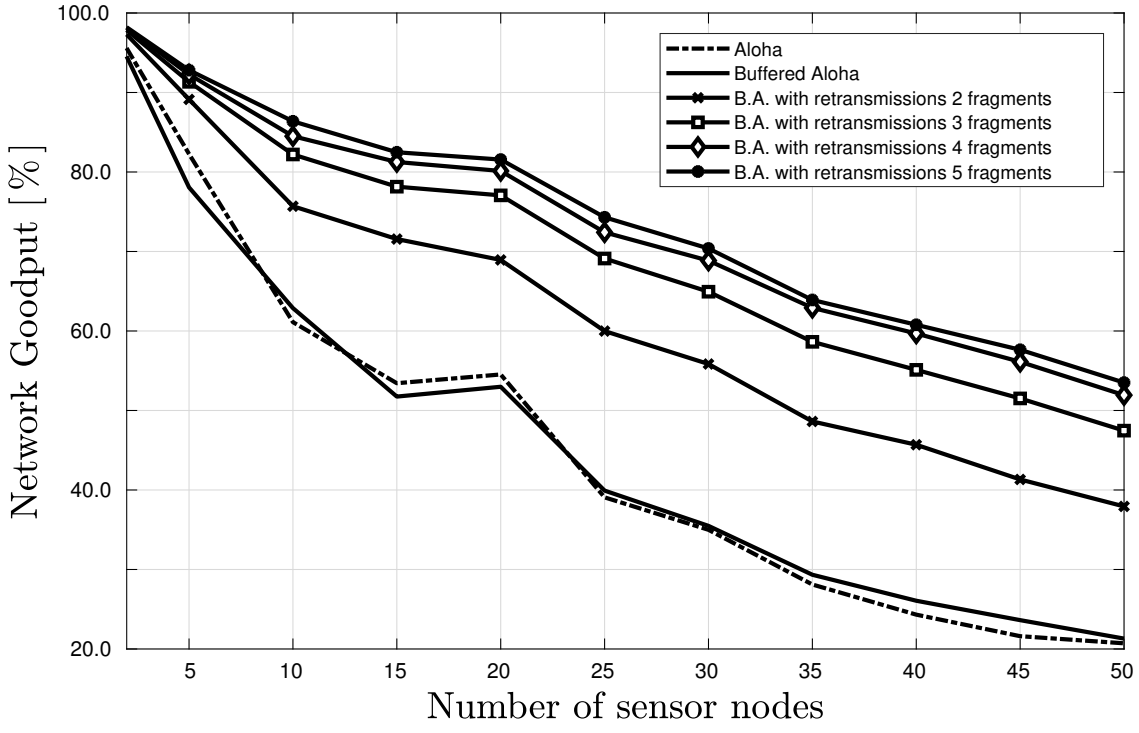


Fig. 5.3 The variation of the network goodput with an increasing number of sensor nodes in the network. Transmission Strategies: Aloha, Buffered Aloha and Buffered Aloha with fragmentation and one retransmission session per packet (2, 3, 4 and 5 fragments/packet).

In Fig. 5.3 we could not show both the case of B.A with fragmentation only and B.A with fragmentation and retransmissions, as the scale didn't allow for it. This is why, Fig. 5.4 shows the gains in goodput that are obtained when using B.A with fragmentation policy, compared to B.A policy. On the same figure, there are plotted the extra gains obtained when upgrading to B.A with

fragmentation and one retransmission session/packet, followed by the gains brought by using 2 retransmission sessions/packet.

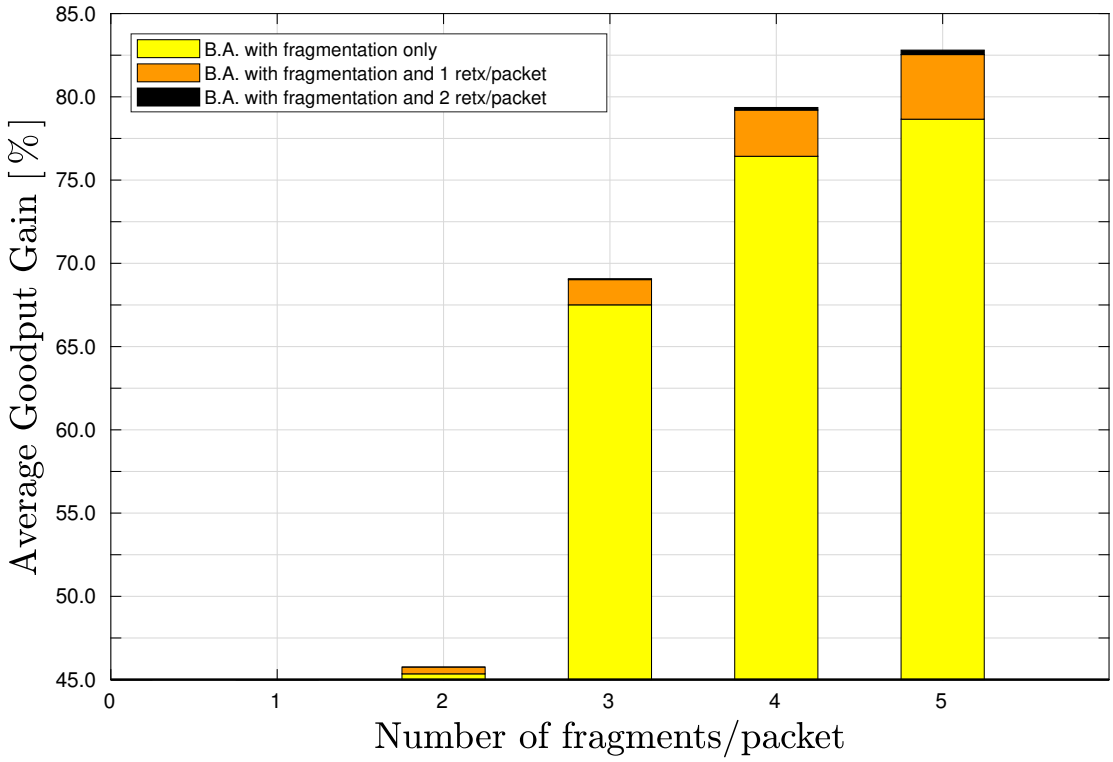


Fig. 5.4 The average gains obtained in network goodput with respect to only using Buffer Aloha transmission. Transmission strategies: fragmentation in 2 to 5 fragments/packet, fragmentation and 1 retransmission session/packet and fragmentation and 2 retransmission sessions/packet.

As we can see, using 5 fragments/packet and 1 retransmission session/packet brings in average an additional 4% gain to using B.A with fragmentation only. Moreover, using 2 retransmission sessions/packet brings additional gains that are smaller than 0.5% and happen only for configurations of more than 3 fragments/packet. This is why the remaining of the chapter will not treat the case of using 2 retransmission sessions/packet.

5.4.2 Application capacity

Fig. 5.5 shows the variation of the application capacity with an increasing number of devices operating in the same channel and using the same SF. This metric helps us identify the network conditions in which the packet fragmentation strategy becomes helpful.

For a small network load (region marked as '1' in Fig. 5.5), data can be sent using full packet size (in our case, 200B). This strategy provides the best results because the probability of collision is low, so using fragmentation would add overheads that are not necessary. Aloha provides worse results than Buffered Aloha, as it is wasting the time resource of the network, directly affecting the application capacity of sending user data.

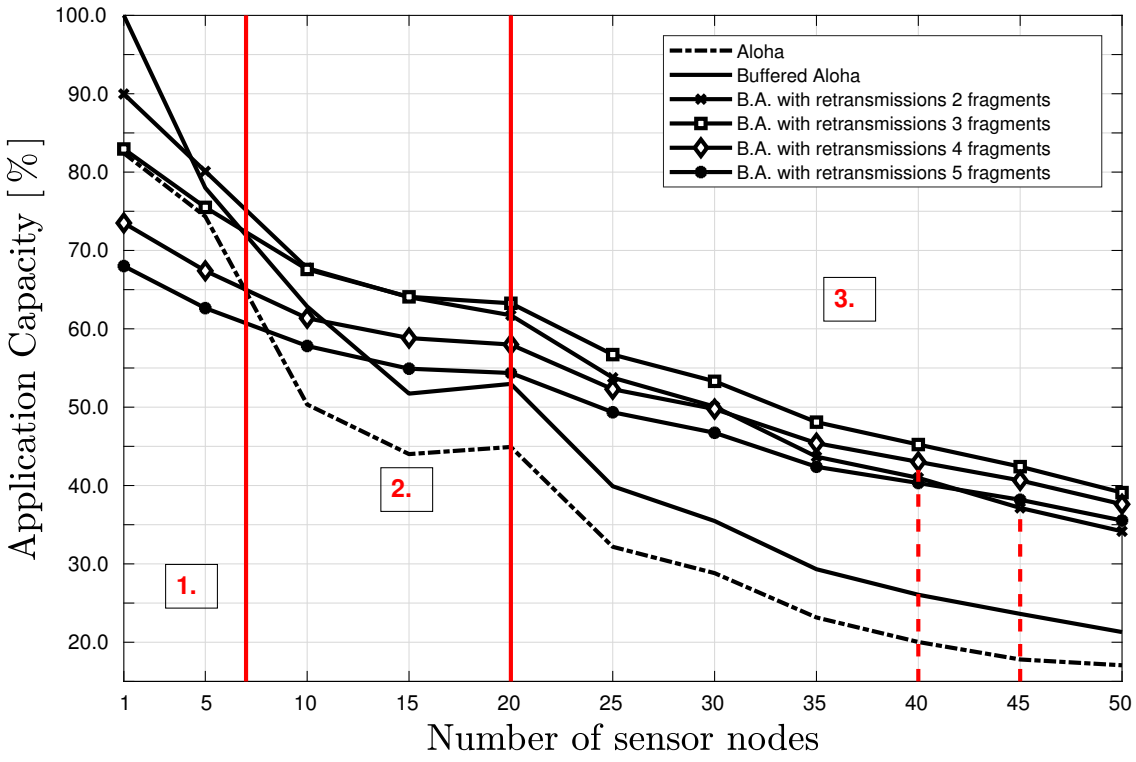


Fig. 5.5 The variation of the application capacity with increasing number of devices in the network. Transmission Strategies: Aloha, Buffered Aloha and Buffered Aloha with fragmentation and one retransmission session per packet (2, 3, 4 and 5 fragments/packet, respectively).

The second region of the plot shows that sending data using 2 fragments/packet is the strategy leading to the best obtainable results. With an increasing number of devices in the network and increased number of collisions, the third region is the one where sending 3 fragments/packet exceeds the other transmission strategies. The two dashed lines in the plot mark the regions where sending data in 4 fragments/packet and 5 fragments/packet, respectively, overtake the performance provided by using 2 fragments/packet. Still, they cannot exceed the application capacity corresponding to 3 fragments/packet. This happens because the lowering in probability of collision that they cause it is not high enough so as to compensate for the fact that their extra T_{oA} directly affects the application capacity.

Going back a step, Fig. 5.3 showed us that the smaller the data size the better the network goodput obtained. Now, Fig. 5.5 shows us that depending on the region in which the network operates, there is a trade-off in the number of fragments/packet to be used, so that fragmentation doesn't have a negative impact on the application capacity.

5.4.3 Energy efficiency

The energy efficiency of the network (Fig. 5.6) follows a similar trend with the application capacity, but it is strongly dependent on the network goodput (amount of data sent, amount of data correctly received by the gateway). The region marked with 'a' corresponds to Aloha as being the

most energy efficient protocol. This happens because Aloha sends less data than Buffered Aloha. Using packet fragmentation and retransmissions in this region is not recommended, as this would imply extra energy consumption for providing a similar network goodput.

The 'b' region shows a number of 2 fragments/packet as being the most energy efficient strategy, very close to the performance that using 3 fragments/packet provides. This happens because the extra energy consumption of using 3 fragments/packet is compensated by the goodput improvement that this strategy brings.

For the networks operating in the 'c' region, using 3 fragments/packet is a good trade-off between the network energy consumption and the obtained goodput performance. The two dashed lines have the same significance as for Fig. 5.5.

We see that Aloha and Buffered Aloha have the worst energy efficiency for dense networks. Using 4 or 5 fragments/packet would provide a better network goodput than using a lower number of fragments/packet, but a price needs to be paid in terms of energy-efficiency.

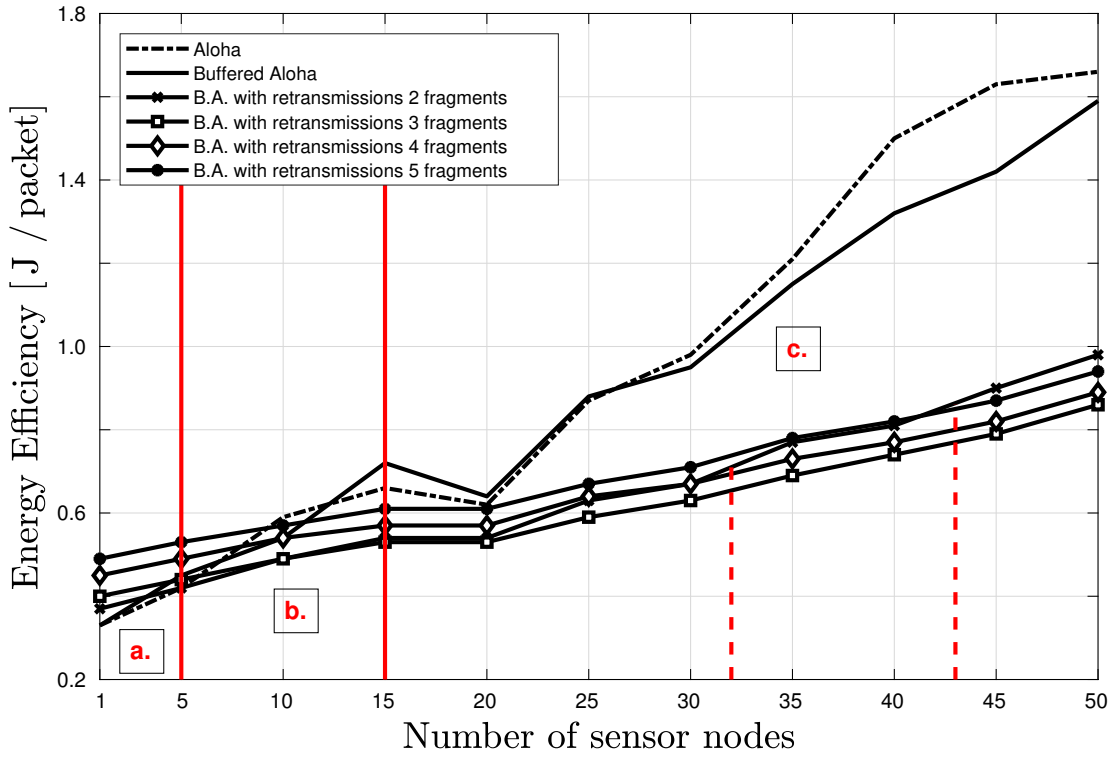


Fig. 5.6 The energy efficiency of Aloha, Buffered Aloha and Buffered Aloha with fragmentation and one retransmission session per packet (2, 3, 4 and 5 fragments/packet)

5.4.4 Header overhead

In Table 5.1, the overhead that packet fragmentation brings in terms of T_{oA} and implicitly, energy consumption, is computed. In the middle column, 9B headers are assumed for each fragment, while in the left column we consider 1B headers. If a way to shrink the 9B MAC and frame header

into a 1B fragmentation header (in the direction of the IETF LPWAN working group) is found, the energy efficiency of the network would be improved.

Table 5.1 Header overhead for multiple fragmentation options

Fragments/packet	Header impact/packet [9B] [%]	Header impact/packet [1B] [%]
2	8.93	5.71
3	19	12.61
4	26.8	17.14
5	35.71	22.86

5.5 Conclusions

In this chapter, we proposed a transmission strategy that combines packet fragmentation with group NACK in duty cycle restricted LPWANs. Packet fragmentation is used despite the packet fits the frame, so as to reduce the probability of collisions while the number of users in the network increases. The group NACK is requested by the last fragment of a packet and accounts for all the fragments of that data packet. This strategy is shown to provide increased network goodput and energy efficiency for dense networks. The retransmission policy is more efficient for smaller fragment sizes, where the probability of successful NACK request is higher.

We provided insights so as to show what transmission strategies are advisable as function of the network size. We showed that for small networks, it is better not to use packet fragmentation, but to use Aloha or Buffered Aloha, which provide similar goodput at increased application capacity and energy efficiency. This is true also for IoT applications that only need to send packets of very small payload, below the size of any fragment considered in this work.

As the network size increases, the aggressive fragmentation strategy provides better network performance. The number of fragments/packet to be used could be dynamically adapted so as to provide the best network performance: goodput, application capacity or energy efficiency. Here, there is a trade-off that needs to be done: smaller fragments provide better goodput but they are less energy efficient and decrease the IoT application capacity. This is mainly because of the fragmentation headers that represent a high overhead in terms of extra time on air and energy consumption. The gateway could control the number of fragments/packet that the nodes use by issuing a MAC command.

The performance of dense industrial duty cycle restricted LPWANs could be further improved if a more collision-resilient acknowledgement scheme is found and if the fragmentation header size is reduced.

Part II

Radio level optimization

CHAPTER 6

Frequency stability analysis with crystal-free radios

6.1 Introduction

IEEE802.15.4 [121] is a popular standard for short-range wireless communication and compliant radios typically operate in the 2.4 GHz frequency band. It is the underlying technology in protocol stacks such as ZigBee or 6TiSCH, which organize a number of IEEE802.15.4-compliant devices in a mesh network topology. These types of networks are widely used in home automation, smart building, smart city and industrial applications.

Manufacturers building IEEE802.15.4-compatible chips must ensure a radio frequency stability not exceeding $\pm 40ppm$ while transmitting a packet [122]. Crystals are typically used as the reference oscillators for synthesizing the radio frequency, and for keeping time. Crystal oscillators are, however, off-chip elements. They contribute to the energy consumption, size and cost of the final product, which becomes significant at high volumes (millions of chips) [58].

For short range communication, the radio module consumption dominates the energy consumption of transmitting data. In Chapter 2.4 we discussed the importance of choosing the manufacturer that built the radio module with the least energy consumption in IDLE/TX/RX states. Moreover, we saw that eliminating the off-chip frequency reference (e.g. XTAL, MEMS) enables having single chip radios of size comparable with that of a XTAL and a ten times more energy efficient communication.

There are significant challenges to making crystal-free radios. The main advantage of a crystal is that the frequency it oscillates at is very stable over time and temperature, typically in the 10-30 ppm range for regular crystals, even down to 2-3 ppm for temperature-compensated versions [123]. On the other hand, on-chip oscillators suffer from high variations over time and temperature, which we characterize in Section 6.3. These variations would need to be detected and compensated by calibration algorithms running continuously during the lifetime of the crystal-free radio. Efforts have been made towards achieving an on-chip frequency reference of higher accuracy [66, 67, 68], but the temperature influence on these oscillators is still too high for compliance with IEEE802.15.4, as that standard mandates a drift below 40 ppm at all times.

The contribution of this chapter is three-fold. First, we characterize the stability in time and with temperature of two on-chip oscillators that drive the wireless capability of the radio. Second, we develop a method by which a crystal-free platform corrects the drifts of its on-chip oscillators

caused by the temperature variation, in order to be able to keep receiving IEEE802.15.4-compliant frames. The resulting algorithm tracks the IEEE802.15.4 frames it receives and continuously fine-tunes the radio to stay within the IEEE802.15.4 oscillator specifications [122]. The algorithm is generic and can be applied to any crystal-free platform. Third, we implement and test the algorithm on the SCuM "Single-Chip μMote", a crystal-free platform that contains a micro-controller and an ultra low-power IEEE802.15.4-compliant radio in a single chip. We evaluate the performance of the solution by having SCuM communicate with an OpenMote [124], a well-known crystal-based IEEE802.15.4 compliant platform. To the best of our knowledge, this work is the first one to show a crystal-free radio successfully communicating with a crystal-based IEEE802.15.4 compliant radio.

As described in Chapter 1.3, SCuM is a prototype implementation of the "Single-Chip Mote" [16, 17]. Including a 2.4 GHz IEEE802.15.4 transceiver, a Cortex-M0 microprocessor, 128 kB of RAM memory, and all the support hardware, it has a die area of $2.5mm \times 3mm$ compared to the crystal-based industry leading low-power technology, LTC5800-IPM [21], that has an area of $10\,mm \times 10\,mm$. The radio offers a 10 m communication range, and draws 670 uA in reception and 1 mA when transmitting at -10 dBm. As a point of comparison, LTC5800-IPM, consumes 4.5 mA when receiving and 5.4 mA when transmitting at 0 dBm.

6.2 Related work

Watteyne *et al.* did some early work with the eZ430-RF2500 platform to replace the "slow" 32 kHz crystal by the internal oscillators of the MSP430 micro-controller on that platform [123]. They implement an adaptive synchronization technique where neighbor nodes re-synchronize to one another at least every 10 s. The resulting drift is approx. 100 ppm. While this work does not attempt to replace the "fast" crystal used by the radio (which is what we target), it does show that getting < 40 ppm using on-chip oscillators is a challenge.

Mehta *et al.* explore whether it is possible to relax the requirement IEEE802.15.4 puts on maximum oscillator drift requirements for the Radio Frequency (RF) accuracy from $\pm$ 40 ppm to $\pm$ 1000 ppm [125]. They show, by simulation, that standards-compliant narrow-band wireless communication is still feasible with $\pm$ 1000 ppm oscillators by compensating their drift using a wide bandwidth channel-select filter, a demodulator and an adaptive feedback loop in the receiver.

Wheeler *et al.* [57] were the first to demonstrate the crystal-free demodulator feedback that we use in this chapter. They use a free-running on-chip LC tank as the local oscillator of an IEEE802.15.4 transceiver that is characterized by a temperature drift of 95 ppm/°C. To deal with temperature variations, the authors use demodulator-based feedback to allow the receiver to track the drift of the transmitter when it is placed in a temperature chamber and subjected to a temperature variation of $2^\circ\ C/min$. The receiver is able to track the transmitter's signal by adjusting the LO at each received frame to keep the value of the Intermediate Frequency (IF) constant. The devices used in the experiments are composed of 65 nm CMOS RF chips with the digital part running on Field-Programmable Gate Array (FPGA) boards.

Khan *et al.* [126] show that an RC oscillator could be disciplined with network feedback to provide a time reference with sufficient accuracy. In order to demonstrate the feasibility of their proposal, the authors use the off-the-shelf OpenMote-CC2538 [124]. One device is programmed to use its crystal and transmit. The other two OpenMote-CC2538 boards simulate a crystal-free device: RX and RC connected to a FPGA that calibrates the RC oscillator. Khan *et al.*obtain an accuracy of 70 ppm for a 1 MHz RC oscillator. In [58], the authors extend this work by testing their approach on a FPGA implementation of the digital system of a crystal-free mote. Using a wired setting, the accuracy obtained using network calibration is 47 ppm for a 25 MHz oscillator.

This chapter takes the state of the art one step further, as it presents a solution for crystal-free radios to be able to maintain their radio frequency accuracy so as to receive IEEE802.15.4-compliant frames sent by an off-the-shelf crystal-based device against temperature change. We provide experimental results showing the efficiency of the proposed methods. To the best of our knowledge, this is the first work to experimentally achieve this by having a crystal-free radio communicating with an off the shelf, standards-compliant device.

6.3 Stability of on-chip oscillators

This section details the challenges in terms of clocking of a crystal-free radio. In order to receive and/or transmit IEEE802.15.4 frames, there are two important frequencies that need to be correctly generated by any compliant-device: the radio channel frequency (in the 2.4 GHz band) and the "chipping" frequency (2 MHz) used to modulate/demodulate the packets. We will refer to the oscillators generating the radio channel frequency and the 2 MHz frequency as the "RF clock" and the "chipping clock", respectively. Link-layer level time synchronization between devices that communicate over IEEE802.15.4 is a well studied topic [123] and it is out of scope of this work.

When in Receive (RX) mode, the RF clock is the only one that needs calibration for setting it on the chosen communication channel. This happens because, for RX mode, the chipping clock can be recovered from the incoming IEEE802.15.4 frames by a Clock and Data Recovery (CDR) module. When the platform is in Transmit (TX) mode, the on-chip oscillator generating the chipping clock has to be calibrated along with the one generating the RF clock. The system is half duplex: at any given time, the RF clock is used either for transmit, or for receive.

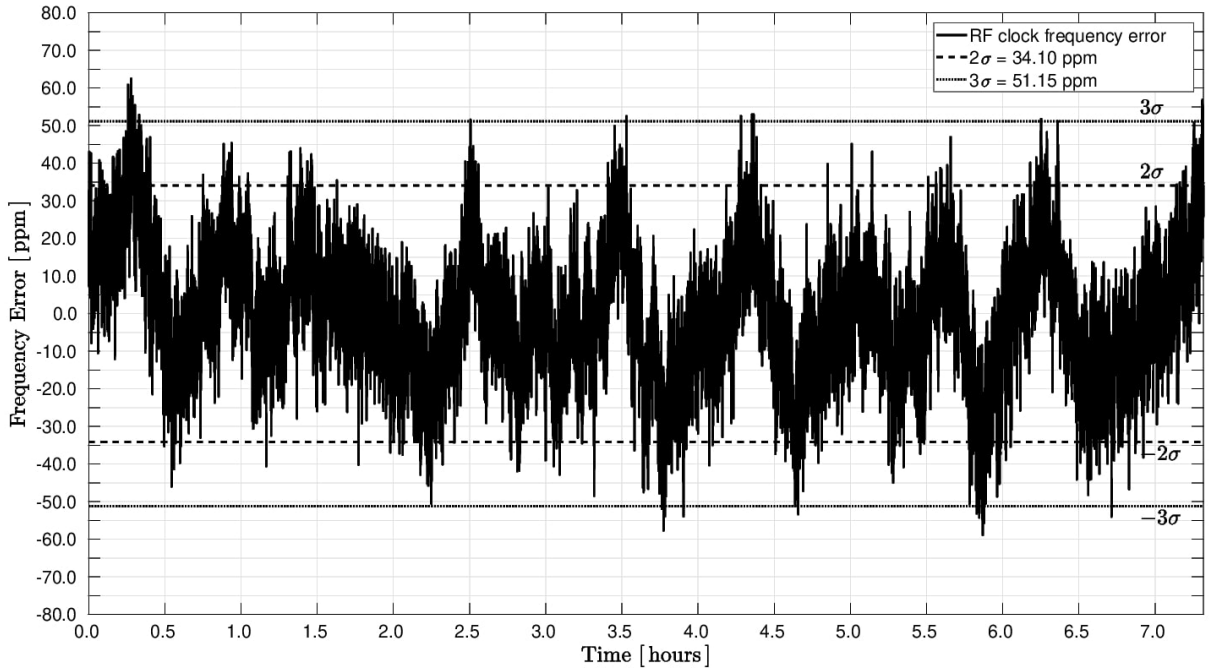
In the remainder of this section, we discuss the stability over time and temperature of the two oscillators that generate the RF clock and TX mode chipping clock, respectively.

6.3.1 Stability over time

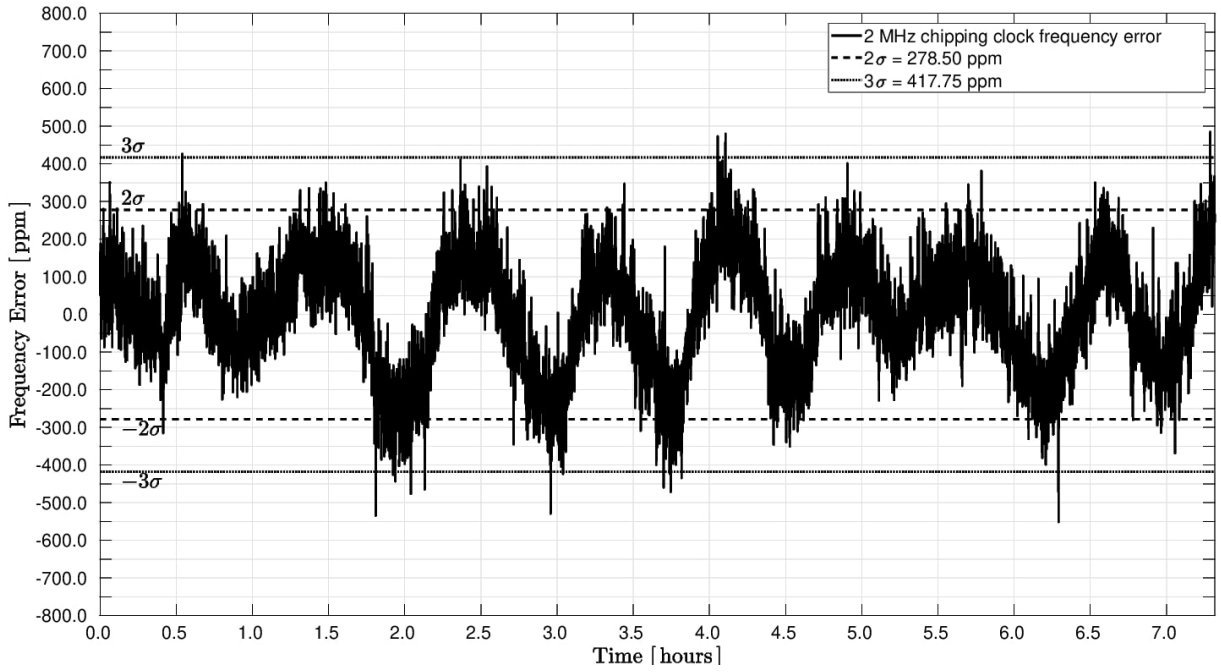
An oscillator drifts over time because of temperature variations and accumulated phase noise. To characterize the stability over time of an oscillator, the external factors that contribute to its drift need to be eliminated (in this case, the temperature variation). Figs. 6.1a and 6.1b show the frequency error over a 7 h period of the RF clock and chipping clock, respectively. For obtaining this data, the crystal-free platform was placed in a temperature chamber at constant temperature ($\pm 0.3^{\circ}C$ temperature stability) for over 7 h. The output frequency of the two oscillators was measured using the frequency counter capability of the Agilent E4440A spectrum analyzer.

For the RF clock, in roughly 95% of the cases (2σ), the frequency error is within $\pm$ 34.1 ppm. This means that, at constant temperature, this oscillator would meet the IEEE802.15.4 requirements of $\pm$ 40 ppm accuracy. Some samples exceed this ($3\sigma = 51.15ppm$), but these variations can also be a consequence of the fact that the temperature chamber used can only keep the temperature constant with a stability of $\pm 0.3^{\circ}C$.

The 2 MHz chipping clock, generated by a different oscillator, drifts mostly within $\pm$ 278.5 ppm (2σ), when at (approximately) constant temperature. This clock has worse stability than the RF clock, but based on experience and measurements, the tolerable accuracy of this clock for communication can be up to $\pm$ 1000 ppm, which is also documented in [127].



(a) RF clock



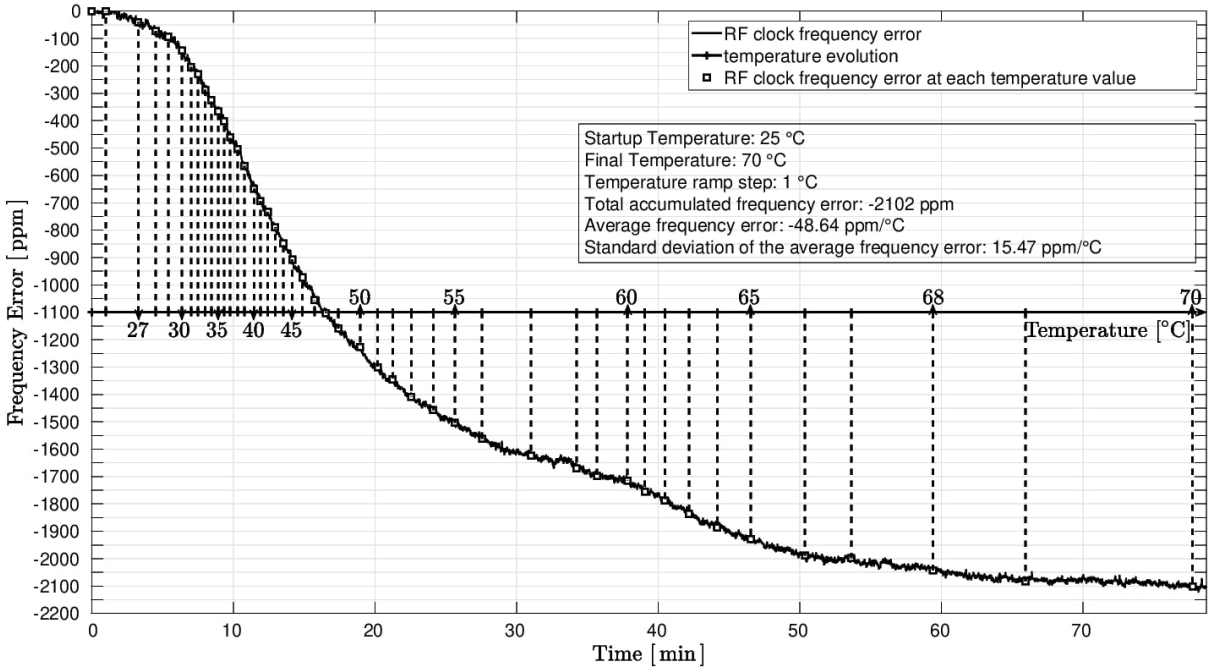
(b) 2MHz clock

Fig. 6.1 RF (a) and chipping (b) clock frequency error during more than 7 hours of run at constant temperature ($\pm 0.3°C$ stability).

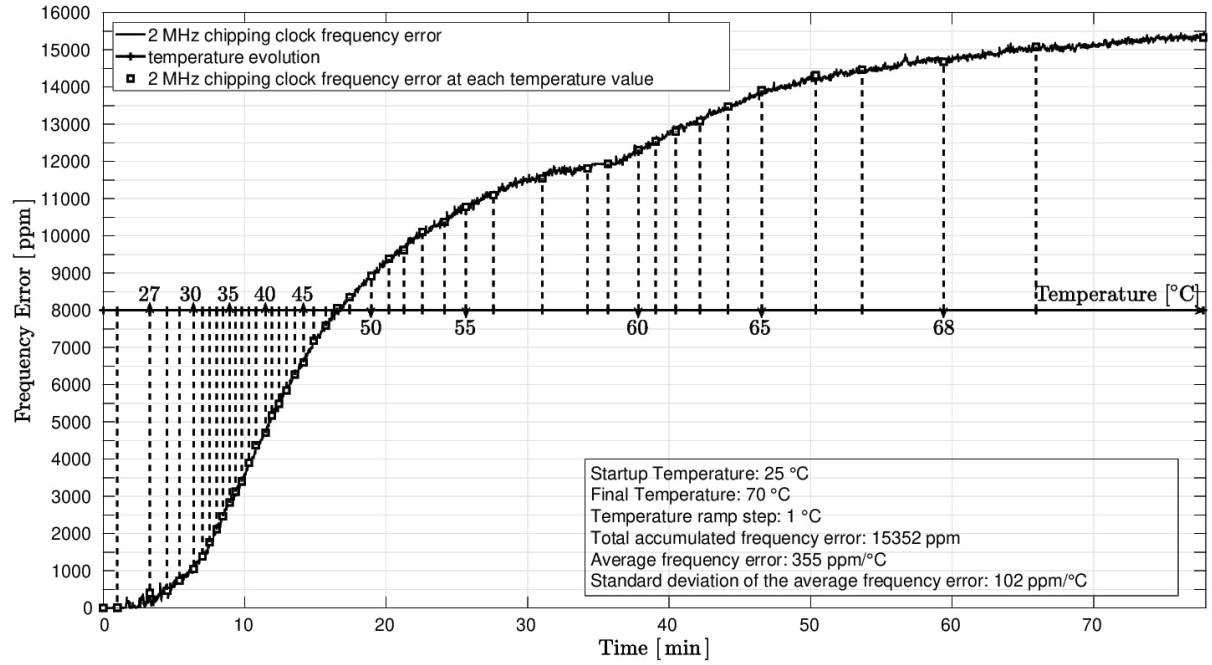
6.3.2 Stability over temperature

When setting the temperature chamber to vary the temperature from $\approx 25°C$ (room temperature) up to $70°C$, the two clocks experience important drifts. Figs 6.2a and 6.2b show the frequency error of the RF clock and chipping clock, respectively, as the temperature increases over time. The two clocks behave differently. The RF clock frequency decreases in average with $48.64 ppm/°C$, while

the chipping clock frequency increases in average with $355ppm/°C$. Deviations from these values could be caused by a slightly different behavior of the oscillators at higher temperatures and/or influenced by the characteristics of the temperature chamber itself ($\pm 0.3°C$ temperature stability, $\pm 3.25°C$ homogeneity, ± 2 % temperature set error). These significant drifts over temperature need to be corrected in order to initiate/maintain communication with an IEEE802.15.4 compliant device.



(a) RF clock



(b) 2MHz clock

Fig. 6.2 RF (a) and chipping (b) clock frequency error when increasing the temperature from $25°C$ to $70°C$.

6.4 Correction of drifts

In order to be able to initiate and maintain communication with an IEEE802.15.4 compliant device, the drifts experienced with temperature variation need to be continuously corrected during the lifetime of the crystal-free device. In order to do so, we will make use of the characteristics of the IEEE802.15.4 Time Synchronized Channel Hopping (TSCH) mode of the IEEE802.15.4 standard, especially of the fact that beacons are periodically sent in the network.

TSCH is a technique which has been designed to provide ultra high reliability and ultra low power operation, and is at the heart of standards such as WirelessHART [128] and 6TiSCH [8, 129]. When joining a TSCH network, a node first listens for a beacon frame. Beacon frames are regularly sent by nodes already part of the network. Once the joining node has received a beacon, it is synchronized to the network, and initiates a secure join handshake. This handshake ensures mutual authentication between the joining node and the network [130].

For our experimental validation, we will have an OpenMote acting as a join proxy node of the IEEE802.15.4 network. It will send periodic beacons on a single communication channel and we will make use of the beacons for maintaining the calibration of the RF clock (using IF calibration) and of the remaining on-chip oscillators (using the periodicity of the beacons). In a future deployment of a network composed only of crystal-free devices, there can always be a crystal-based device acting as a base station, so as the other devices can keep time and frequency synchronization.

6.4.1 RF clock drift correction

As in any standard receive chain, when the RF clock is tuned to the beacon frequency and the crystal-free mote receives a beacon, the received RF signal is down-converted to a specified IF and then demodulated [17]. The IF frequency is a fixed hardware design parameter and can be measured during packet reception. As the temperature in the environment changes, the RF frequency drifts ($\approx 48.64 ppm/^{\circ}C$) and the measured IF frequency shows an offset from the expected value, when receiving a packet. Correcting this offset means correcting the RF frequency drift and maintaining communication against temperature change.

As shown in Fig. 6.3, when the crystal-free device uses an initial tuning code (e.g. C) for receiving packets on channel Y, once a packet is received on that channel, the device can count the IF value [17]. The IF is created by mixing the incoming signal on channel Y with the Local Oscillator (LO) signal, resulting in a signal at a difference frequency. If the IF value is within 40 ppm of the expected value (2.5 MHz for SCuM, a design parameter of the chip hardware), the local oscillator (RF clock) is tuned on channel Y. If there is an offset from the expected value, the LO initial tuning code needs to be adjusted to compensate for that offset, until it becomes null (or a value within $\pm$40 ppm accuracy). The adjusted tuning code is C_{IF}.

This IF-based tuning of a crystal-free radio is one of the key techniques which allow low-accuracy low-frequency clocks to be used to track, and indeed generate, standards-compliant RF

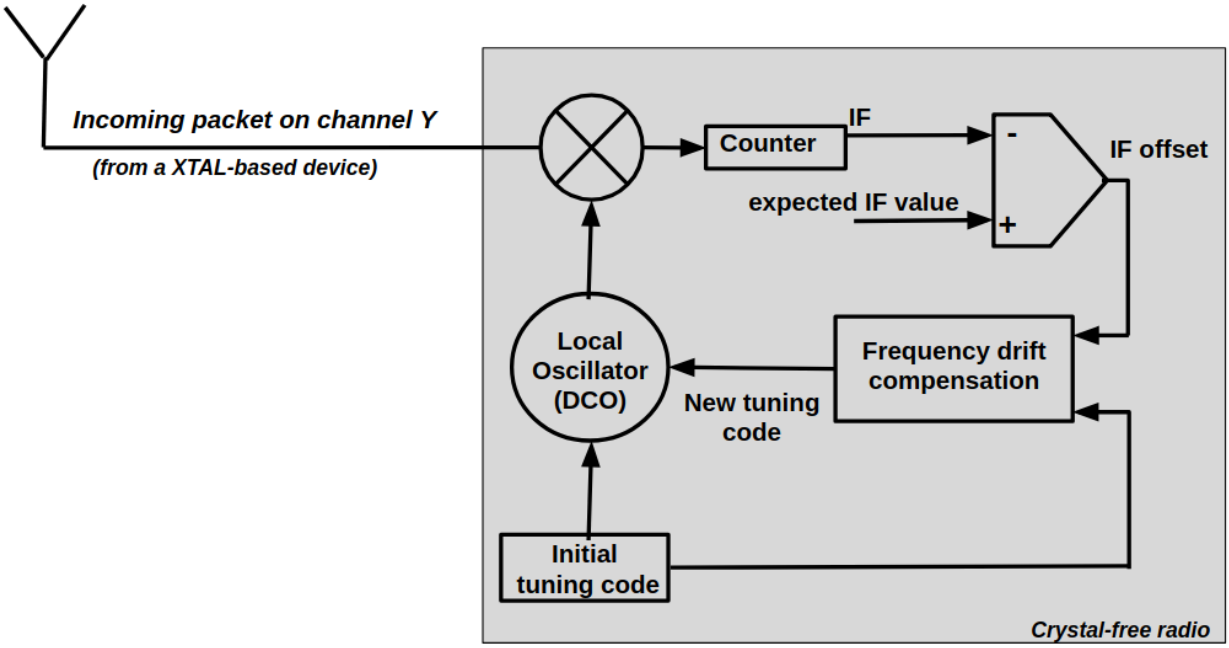


Fig. 6.3 IF-based calibration scheme. The tuning code of the LO is continuously adjusted until the IF value reaches the expected value. This results in a LO frequency as close as possible to the center frequency of channel Y.

oscillations with an accuracy of 40 ppm. The nonlinear nature of an RF mixer means that a frequency difference Δf of 40 ppm at 2.4 GHz turns into the same Δf at 2.5 MHz, where it is now nearly 40,000 ppm of error relative to a 2.5 MHz clock. So the fractional frequency error in the 2.5 MHz measurement is divided by nearly a factor of one thousand when it shows up at 2.4 GHz.

We implement an algorithm that checks the offset (drift) of the IF frequency at each beacon reception, and uses that to finely tune the RF clock in steps given by its tuning resolution ($\pm\Delta F$ [Hz] = ± 1 tuning code, depending on the sign of the offset), as this clock was already calibrated on the beacon frequency and only small corrections are needed as the temperature changes. These temperature drift corrections come at no cost, as the IF information is available at every packet reception. If at startup temperature a tuning code x was used for communicating on IEEE802.15.4 channel 11 for example, as the temperature evolves, the tuning code used for the same channel has to change its value to $y, z, ..$ (using IF-based calibration) depending on how much of a change there is. If we do not adjust the tuning code value, and keep it x, as the temperature changes the crystal-free device will not communicate on channel 11, but on another frequency depending on the experienced drift. We will discuss more about this in the next chapter, as the focus of the current chapter is to maintain communication on the same channel even if the temperature changes, without interest in what actually are the values of $x, y, z...$

This IF-based correction algorithm is packet-loss tolerant. If there is a sudden change in temperature that causes the RF clock to drift directly outside the communication channel and does not change back to the initial temperature, or to a value characterized by a drift within the communication channel ($\pm$ 1 MHz), then communication is completely lost. In this case the crystal-free

radio has to start a search process for finding a communication channel, that will be discussed in the next chapter. Otherwise, as long as the temperature variation causes a drift within the communication channel limits, there can be packet loss, as we showed in Section 6.3.1 that the on-chip oscillators maintain their stability over time and do not accumulate errors over more than 7h of run (see Fig. 6.4b for experimental results showing the recovery from packet loss). After packet loss, the next received packet will be used to update the tuning code of the RF clock so as to ensure that the crystal-free radio remains calibrated on the communication channel at the environmental temperature.

6.4.2 Drift correction of other on-chip oscillators

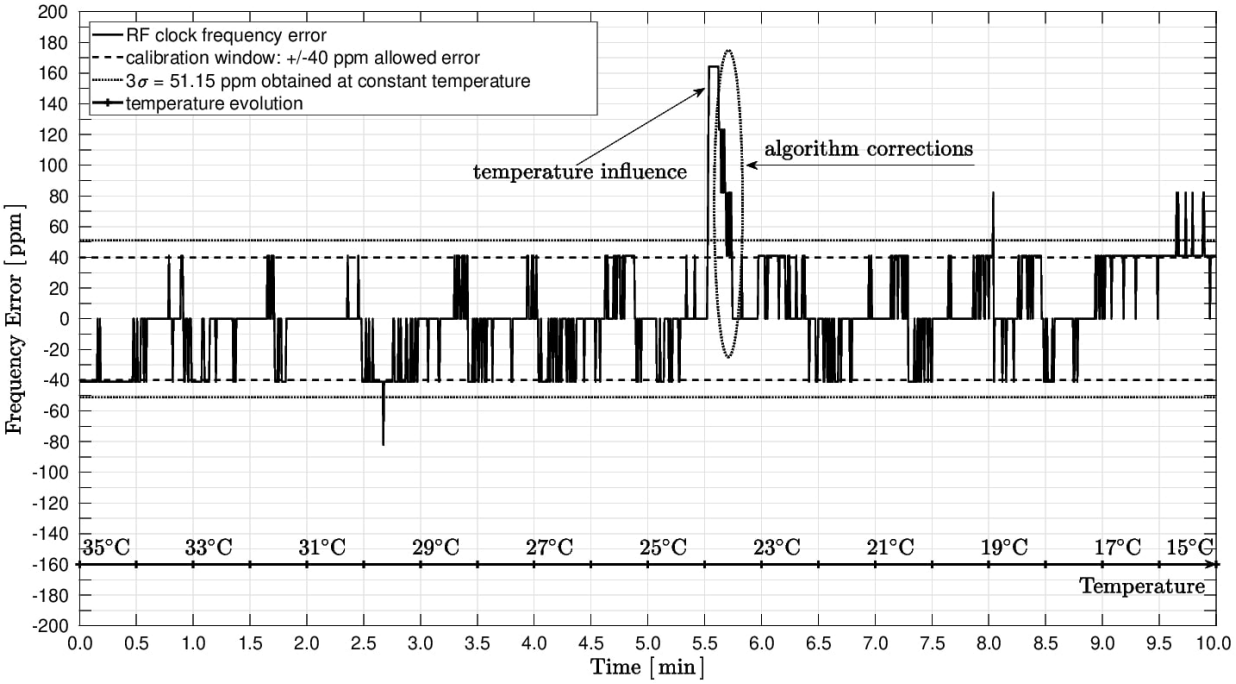
As long as the RF clock is corrected based on the IF offset and the crystal-free platform is able to receive beacons against temperature changes in the environment, the 2 MHz chipping clock (and other on-chip oscillators) errors due to temperature can be corrected using a fine calibration procedure. Again, this calibration is packet loss tolerant, depending completely on the RF clock ability of receiving packets.

We assume that the on-chip oscillators were calibrated at the point when the temperature started changing. How to achieve this initial calibration is detailed in Section 7.2.2. The fine corrections that we will apply in order to update the tuning codes of the oscillators so as to keep calibration as temperature changes, consist in adjusting the settings of the oscillator of interest with $\pm$ 1. These fine corrections can be applied at each received beacon, or after the average number of ticks counted during the last N received beacons is compared to a threshold value (a calibration window defining the acceptable error of the average counted ticks as $\pm x$ ppm). This latter method is discussed in [58] and proves to have better accuracy than the former.

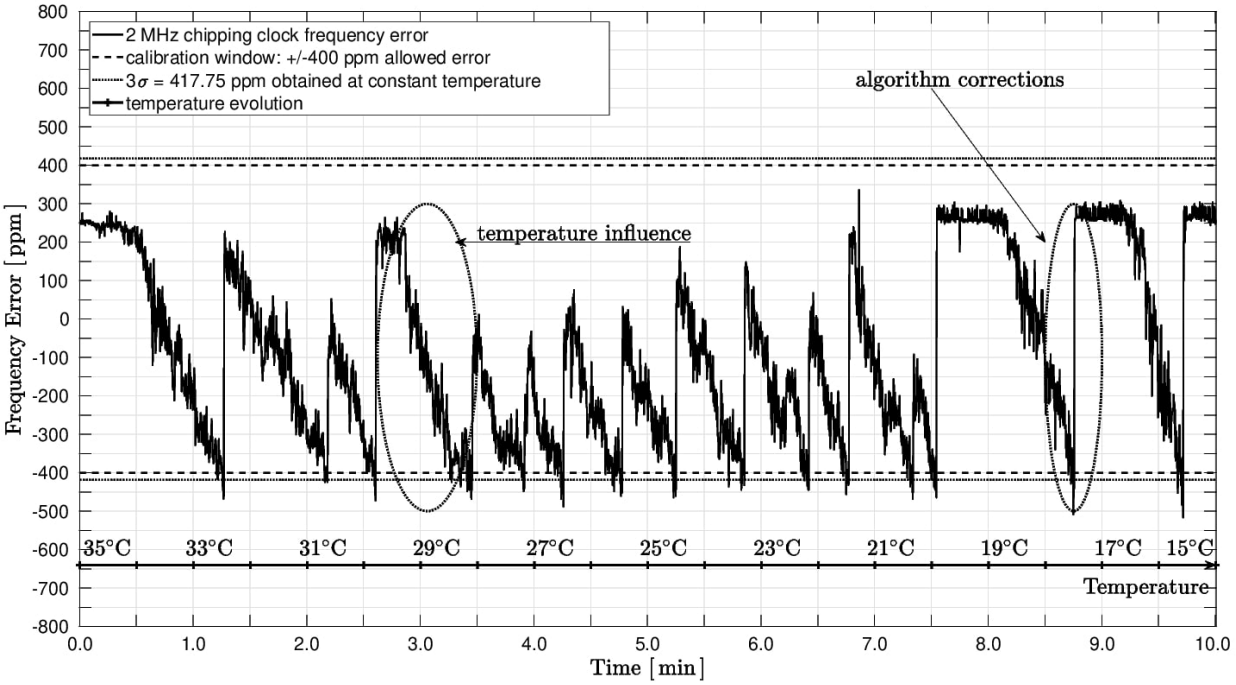
6.4.3 Experimental validation

Fig. 6.4a shows the evolution of the RF clock frequency error while the crystal-free mote is placed inside a temperature chamber and is subject to a $15°C$ temperature change. During this time, the crystal-free platform finely tunes the RF clock based on the measured IF every time a beacon from the OpenMote is received ($T_B = 125ms$). The obtained RF frequency error is as good as in constant temperature environment. The frequency error spike in Fig. 6.4a is a consequence of several lost beacons, but it is corrected as soon as new beacons are received. This is possible because, even if the receiver's RF clock may have accumulated more than 40 ppm of error, the beacons are still within the bandwidth of the receiver.

Fig. 6.4b shows the performance of the applied fine calibration using the beacon periodicity: the frequency error of the 2 MHz chipping clock is as good as at constant temperature, even when the crystal-free platform is subjected to a $15°C$ temperature variation.



(a) RF clock



(b) 2MHz clock

Fig. 6.4 RF (a) and chipping (b) clock corrections when subjected to a $2°C/min$ temperature variation. The RF clock is kept within $\pm$ 40 ppm. The chipping clock is kept within the $\pm 400 ppm$ calibration window. Beacons are sent periodically by an OpenMote.

6.5 Conclusions

In the context of enabling the use of crystal-free radios so as to further enhance the energy efficiency of IoT networks, this chapter analyzes the frequency stability with respect to time and temperature of two fundamental on-chip oscillators. These oscillators drive the wireless communication capabilities of a radio in typical IEEE802.15.4 networks. While the frequency stability in time is good enough to meet the specification demands, these oscillators experience very significant drifts over temperature.

We presented mechanisms to dynamically compensate that drift and we show that using the IEEE802.15.4 signaling we can maintain the frequency calibration even as the temperature changes. The obtained accuracy meets the IEEE802.15.4 requirements of $\pm 40ppm$ frequency stability. The presented strategies are accompanied by experimental results obtained with a crystal-free platform and a crystal-based standards-compliant OpenMote acting as join proxy node of the IEEE802.15.4 network.

While in this chapter we focused only on maintaining communication on one channel as temperature changes, in the next chapter we will show how to obtain this initial communication in the first place and then we will treat the challenge of enabling the multi-channel communication characteristic to IEEE802.15.4 standard in the 2.4GHz band with SCuM, an ultra low power crystal-free and PLL-free platform.

CHAPTER 7

Frequency synchronization techniques for crystal-free radios

7.1 Introduction

This chapter covers techniques for synchronizing a crystal-free radio to an IEEE802.15.4 network in which there is at least one crystal-based device that acts as a gateway or join proxy node, sending periodic beacons, as per IEEE802.15.4 specification [121]. As we also mentioned in the previous chapter, when joining a TSCH network, a node first listens for a beacon frame sent by nodes already part of the network. Once the joining node has received a beacon, it is synchronized to the network, and initiates a secure join handshake to ensure mutual authentication between the joining node and the network [130]. TSCH devices are typically clocked by a low-power 32 kHz crystal with a drift of 10-30 ppm. Because of this drift, a node needs to regularly re-synchronize to its time source neighbor in order to remain synchronized to the network. Assuming a maximum de-synchronization error of 1 ms and a 30 ppm drift, this re-synchronization needs to happen at least every 30 s.

The drift of the internal oscillators of a crystal-free device can be thousands of times higher than that of crystals. This impacts not only the rate at which nodes de-synchronize, but also their capability to tune the communication frequency and communication data rate. Moreover, when a crystal-based end device looks for beacons in order to get synchronized to the network, it can synthesize the beacon channel frequency with a very small error, allowing it to almost immediately receive beacons and process them. This happens because the crystals used as reference for the RF clock oscillate at a determined value with a very good stability ($<$10 ppm). For a crystal-free end device, this process is more complicated and time consuming and, as there is no calibrated reference on-chip, at startup, all on-chip oscillators can be for example 10,000 ppm off their nominal value [126]. For this reason, for using crystal-free radios in an IEEE802.15.4 network, we need additional techniques that allow us to:

1. synthesize the frequency of a communication channel so as to enable the reception of a first network beacon (initial calibration);

2. synthesize the frequencies of the remaining 15 IEEE802.15.4 communication channels to enable the reception of packets on any channel (startup phase);

3. re-calibrate the crystal-free radio to maintain reception on the achieved channels as temper-

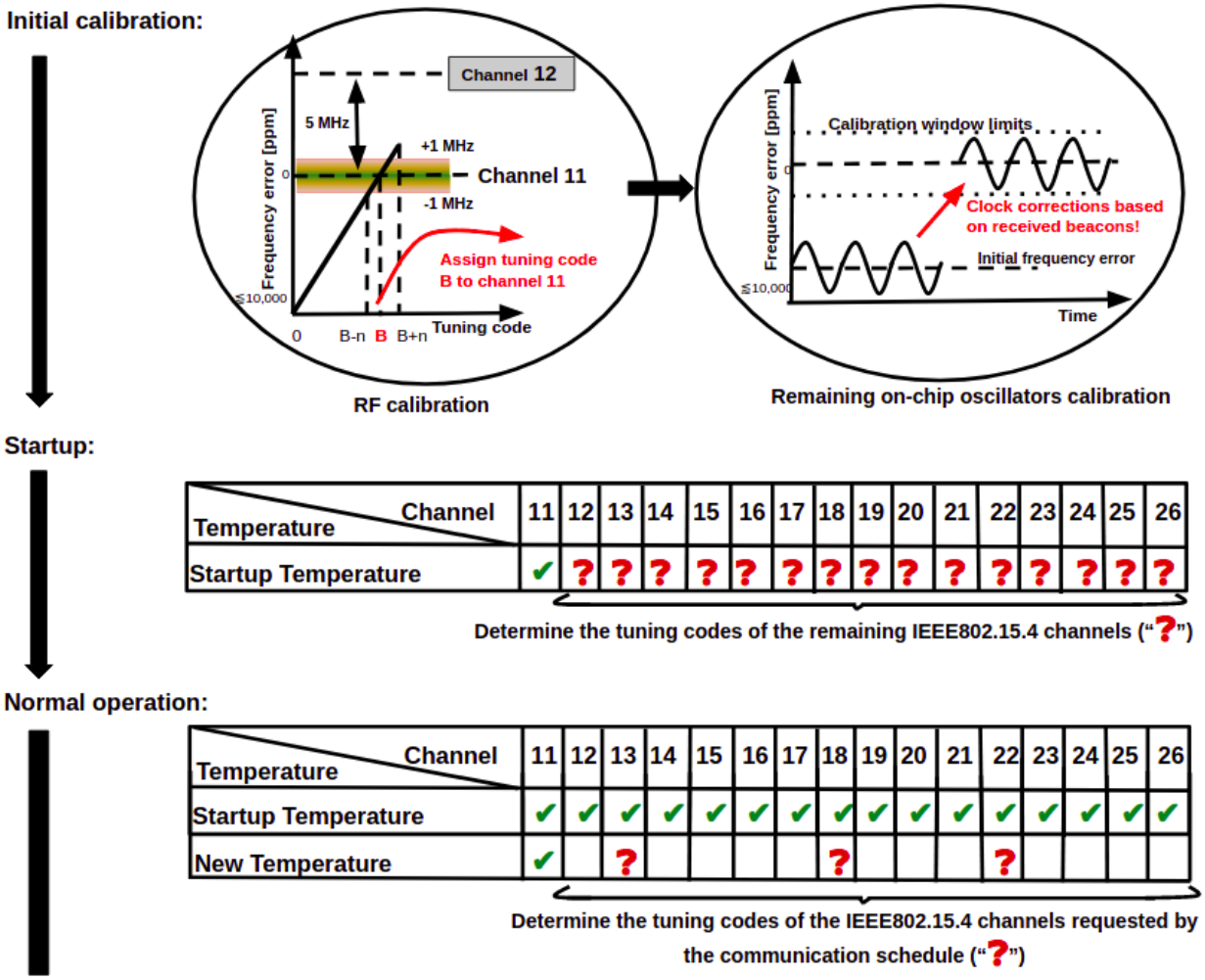


Fig. 7.1 Crystal-free radio. Steps for establishing communication on all channels, at any environmental temperature.

ature evolves (normal operation);

4. achieve and maintain the capability of transmitting packets on any of the 16 IEEE802.15.4 communication channels at constant and variable temperature (steps 1-3 for transmission mode).

In order to enable a crystal-free device to operate in an IEEE802.15.4 network in any environment, we defined three phases that have to run during the lifetime of the device and that are depicted in Fig. 7.1. The initial calibration phase is needed only once, when the device boots for the first time in a given environment, or if for some reasons (very fast change of temperature and lost packets), communication is lost and the device has to start looking for a network beacon again. During this phase the device gets calibrated for receiving network beacons on IEEE802.15.4 channel 11 (Section 7.2) and then enters the startup phase. The startup calibration phase is needed for learning the tuning codes for synthesizing the communication channels in receive mode at the current environmental temperature. Once the tuning codes for the 16 communication channels are known, the device enters the normal operation phase. In this phase, the device is able to receive on any channel, while tracking the drifts caused by any temperature change using IF calibration.

The device can start learning the tuning codes for being able to transmit a packet only if it already knows the tuning codes for being able to receive the acknowledgement on the respective channel.

The following sections will describe the three defined phases. The goal is to be able to synthesize the communication channels and maintain their accuracy within the ± 40 ppm limit demanded by the IEEE802.15.4 standard. This would enable the use of crystal-free radios in IEEE802.15.4 networks, that will further enable more energy efficient communication and longer network lifetime. We will describe the calibration methods we propose for each phase, and we will evaluate the obtained frequency accuracy both through simulations and experimental validation.

7.2 Initial calibration

The crystal-free radio uses a Digitally-Controlled Oscillator (DCO) as time reference. When a crystal-free radio is to join a network, since its DCO can be 10,000 ppm off its target frequency [126], it doesn't know which DCO setting corresponds to the needed frequency. The goal of the initial calibration is to find that relationship for the RF clock, in charge of synthesizing the communication channel frequency, and for the remaining on-chip oscillators, in charge of packet modulation, time keeping, or other functions. It is a process that is only needed once during the lifetime of the crystal-free device. As we specified in Section 7.1, if for some reasons, the device looses the communication capability, this process would have to be repeated.

7.2.1 RF calibration

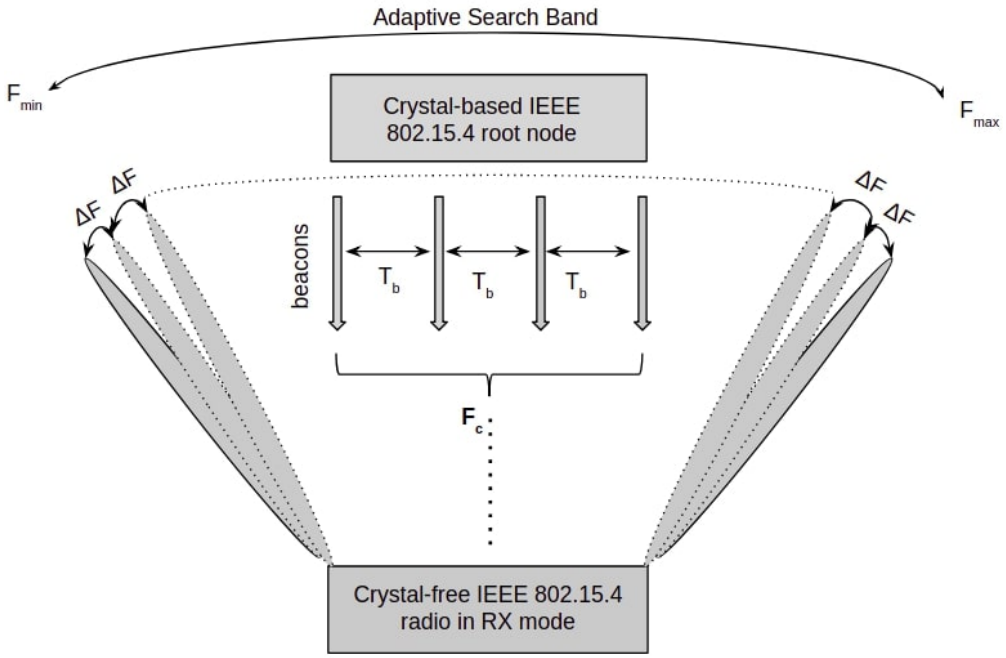
During the initial calibration, a node turns on its radio in **reception mode** and sweeps the tuning code of its DCO across all possible values. It does so until it successfully receives a frame.

$$\begin{aligned} duration_{RF\_cal} &\approx (B+n) \times t_L \\ t_L &\approx 10 \times T_B \\ T_B &= 160\text{ms} \end{aligned} \tag{7.1}$$

Eq. (7.1) expresses the duration of the initial calibration phase. $(B+n)$ is the number of tuning code adjustments until reaching the upper limit of channel 11 (see Fig. 7.1). t_L is the amount of time the radio listens for each tuning code setting. t_L duration ($t_L \approx 10 \times T_B$) is large enough for the radio to be able to receive multiple beacons, which is needed in case the wireless link from the node sending the beacons is lossy. $T_B = 160$ ms is the time it takes for the node sending the beacons to hop across all 16 IEEE802.15.4 frequencies (in TSCH, a 10 ms timeslot is typical).

The IEEE802.15.4 standard uses 16 frequencies – numbered 11-26 – in the 2.4 GHz band. The beacons are sent on all 16 frequencies using a known channel hopping sequence that lasts for T_B. This means that the joining node, when listening on a single frequency, receives at most one beacon every T_B.

The search process starts on the minimum supported code (tuning code 0). The device listens for t_L seconds on the synthesized frequency (F_{min}, see Fig. 7.2a), then increments the tuning code by 1 (which changes the radio frequency to $F_{min} + \Delta F$, where ΔF is the RF oscillator tuning resolution). The radio does not receive any beacon until its tuning code reaches $B - n$ (as shown in Fig. 7.1), which corresponds to a synthesized frequency at the inferior limit of IEEE802.15.4 channel 11. The number of beacons received with a correct Cyclic Redundancy Check (CRC), CRC_OK, increases as the tuning code reaches value B, corresponding to a synthesized frequency close to the center frequency of channel 11. The number of received beacons decreases as the synthesized frequency approaches the upper limit of the channel, as the error between the center frequency of the channel and the synthesized frequency increases. The search process stops after the synthesized radio frequency is outside the communication channel (tuning code value $B + n$),



(a) The process of searching for the beacon channel

Algorithm Beacon Channel Acquisition

```
1: turn on receiver on F_min
2: F = F_min
3: loop:
4:     wait for IEEE 802.15.4 beacons for t_L seconds
5:     if t_L expired
6:         turn off receiver
7:         if number of received beacons == 0
8:             if (beacons_received == true & radio_silence_band > 1MHz)
9:                 beacon_channel_at = F(maximum_performance)
10:                F_max = F
11:            else
12:                if (beacons_received == true)
13:                    update radio_silence_band
14:                end
15:                F = F + ΔF
16:                turn on receiver on F
17:                goto loop
18:            end
19:        else
20:            if (beacons received for the first time)
21:                beacons_received = true
22:                maximum_performance = (number of CRC_OK beacons, F)
23:            else if (performance > maximum_performance)
24:                update maximum_performance
25:            end
26:            F = F + ΔF
27:            turn on receiver on F
28:            goto loop
29:        end
30:    end
```

(b) Algorithm for beacon channel acquisition

Fig. 7.2 *(a)* The process of searching for the beacon channel with the center frequency F_c. *(b)* Algorithm for beacon channel acquisition from cold start.

at which point no more beacons are received ("radio silence"). The crystal-free device tunes its radio using tuning code B (corresponding to the maximum reception performance) and is ready for receiving the periodic beacons sent on channel 11 with an accuracy within $\pm$40 ppm. It stores B as the tuning code needed for synthesizing channel 11 at startup temperature, which can be measured by an on-chip sensor. The algorithm corresponding to the search process, that ends with finding the beacon channel and tuning on the setting that returned the maximum performance (in terms of number of received beacons and number of CRC_OK beacons), is represented in Fig. 7.2b.

7.2.2 Calibration of other on-chip oscillators

The time interval between successive beacons received on the now-calibrated RF clock on channel 11 is used to calibrate the remaining on-chip oscillators, as shown in Fig. 7.1 as the step to follow after RF calibration. These oscillators generate other standard-specified rates, such as the communication bit rate and the real-time clock. The correction (in tuning codes) to be applied to an on-chip oscillator is given by (7.2), where $\Delta F_{oscillator[Hz]}$ is the tuning resolution of that oscillator.

$$correction = -\frac{ticks_{oscillator} - ticks_{idealF[Hz]}}{\Delta F_{oscillator[Hz]}} \tag{7.2}$$

To correct the frequency error, and assuming that the oscillator should have a nominal value of F Hz, we compare the number of ticks this oscillator counts with the value an ideal F Hz clock would count between two successive beacons (T_B).

Eq. (7.2) corresponds to a "fast calibration", as it allows one to quickly bring the clock of interest closer to its ideal value. After this is done, more fine corrections can be applied to the respective clock by adjusting its oscillator setting with $\pm$ 1. These finer corrections have been discussed in Section 6.4.2.

The duration of the complete initial calibration phase depends on both $duration_{RF\_cal}$ and on how restrictive the set calibration window for other on-chip oscillators is. Still, compared with the time consuming search for a beacon ($duration_{RF\_cal}$), the latter is a negligible factor in determining this duration. The initial calibration phase is valid for any crystal-free device starting its operation at any temperature. At the end of this phase, each device has all of its on-chip oscillators calibrated and has the knowledge of one communication channel, which is primarily channel 11 in IEEE802.15.4 networks. Tuning code value B, corresponding to channel 11, has a different value for each device and for each startup temperature. Each device stores B as the tuning code needed for synthesizing channel 11 at startup temperature, which can be measured by an on-chip sensor.

7.2.3 Experimental validation

For the experimental validation of the RF calibration algorithm (Fig. 7.2b) an OpenMote was configured to send beacons on channel 11 (2405 MHz) with a periodicity $T_B = 125ms$. The tuning resolution of SCuM is $\Delta F \approx 90kHz$ (design parameter). The listening duration on each

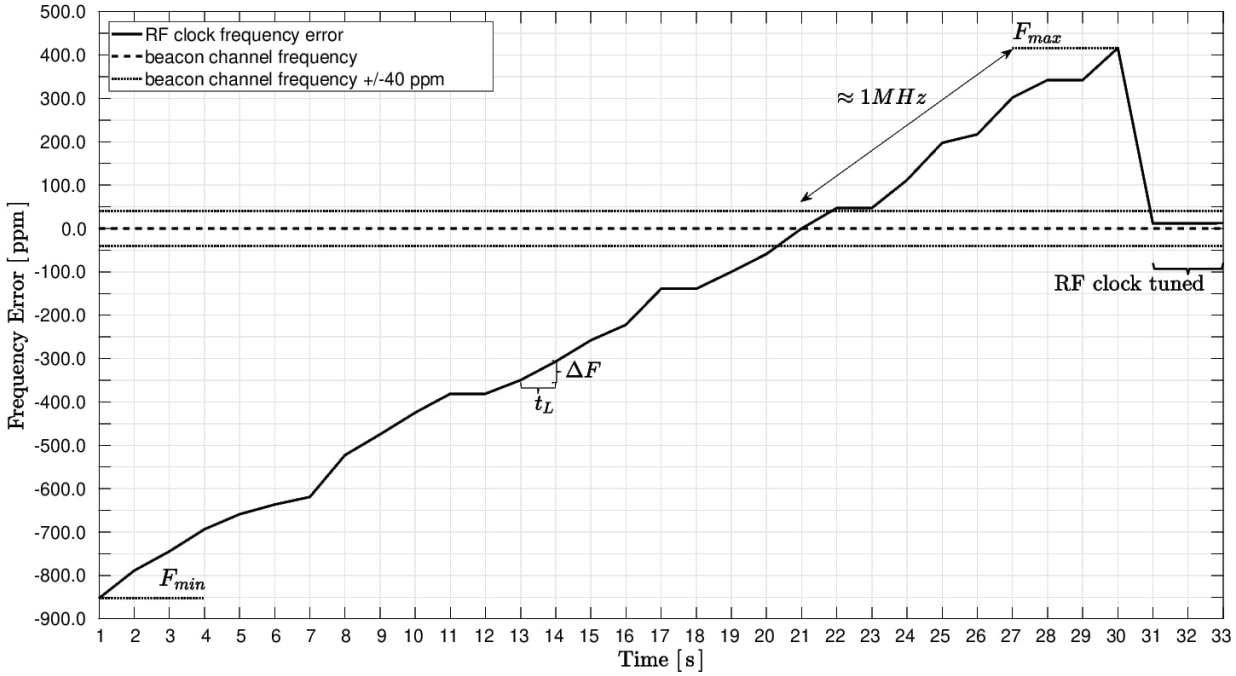


Fig. 7.3 The RF clock frequency changes as the algorithm sweeps through frequencies to find that of the beacon. Beacons are periodically sent by an OpenMote.

tuning code was set to $t_L = 1s$, long enough to ensure the reception of a few beacons while compensating for the uncalibrated time keeping clock ($\pm$ 10000 ppm). As explained in Section 7.2.1, the device will only start receiving the beacons when its tuning code ($B - n$) corresponds to a synthesized frequency value inside the beacon channel bandwidth (2405MHz $\pm$ 1 MHz).

The search process ends when beacons are not received anymore: we set the "radio silence" period to be more than 1 MHz after the reception of beacons stopped, to make sure that the beacon channel bandwidth (2 MHz) has been completely scanned and that the reason for not receiving beacons is that the channel limit was exceeded and not that the radio link is lossy.

Fig. 7.3 shows the evolution of the RF clock frequency while running the beacon channel search algorithm (Fig. 7.2b) on the crystal-free platform, measured with a frequency counter. We can see that, at startup, using the tuning code 0 returns a frequency F_{min} that is -850 ppm away from the center frequency of the beacon channel. The RF frequency is increased gradually ($\Delta F \approx 90kHz \approx 40ppm$) and after 1 MHz of radio silence the frequency sweep ends and the RF clock is tuned to the setting that yielded the best performance. The resulting RF frequency is within $\pm$ 40 ppm of the beacon center frequency. By designing crystal-free radios with even a finer tuning resolution ($\Delta F \ll 40ppm$), the algorithm would perform better, as it would sweep through more intermediate frequencies inside the channel bandwidth, allowing for a synthesized frequency closer than 40 ppm of the channel center frequency.

For validating the technique for calibrating the remaining on-chip oscillators, we implemented the fast calibration strategy on the crystal-free platform, as well as the fine corrections based on the average number of ticks counted by the clock of interest. Specifically, the oscillator we aim to calibrate is the "chipping clock" (2 MHz) used to modulate the packets. This clock will only be

needed when in TX mode. The reason why we did not calibrate a similar clock when in RX mode in order to be able to demodulate the incoming network beacons (Section 7.2.1) is that the chipping clock for RX mode can be recovered from the incoming IEEE802.15.4 frames by a clock and data recovery module (CDR). This means that in RX mode, the closer the RF clock is to the center frequency of the channel, the better the quality of the received frames and the better the quality of the recovered chipping clock needed for packet demodulation. For this reason, in the remaining sections of this work we will refer to the "chipping clock" as being the 2MHz oscillator needed for packet modulation in TX mode only. This oscillator, as well as others on-chip oscillators, will be calibrated with each received network beacon in order to ensure that their accuracy stays within their respective set calibration windows.

For the chipping clock we set $N = 10$ received beacons for fine calibration, and a calibration window of $\pm x = \pm 400 ppm$ accepted frequency error. We extract data with a frequency counter, in a laboratory environment. Fig. 7.4 shows how at startup (while the RF clock looks for the beacon channel), the chipping clock is 8000 ppm away from the nominal 2 MHz value. After fast calibration, it gets within 1000 ppm of the nominal value. After fine calibrations, the chipping clock reaches a frequency error variation inside the defined calibration window. The time interval between fast calibration and fine calibration is influenced by N and by the number of lost beacons: the more beacons are lost, the slower the algorithm reaches the N needed packets before the next calibration round.

An important observation to make is that we should not set a calibration window smaller than the accuracy of the oscillator at constant temperature: in Section 6.3.1, Fig. 6.1b, we saw that naturally at constant temperature, the 2MHz chipping clock has an accuracy $2\sigma = 278.50$ ppm, so in this case, a calibration window of ± 400 ppm is acceptable. In the same way, we cannot obtain a better accuracy of the RF clock than that obtained when in controlled temperature environment ($2\sigma = 34.10$ ppm, Fig. 6.1a). There might appear drifts with time that have to be corrected, but still, the obtained frequency accuracy will not be better than the inherent accuracy of the oscillator. As for energy efficiency reasons we eliminated the XTAL reference and PLL from the chip architecture, the only way to improve the frequency accuracy below this threshold is by improving the oscillators from the design phase.

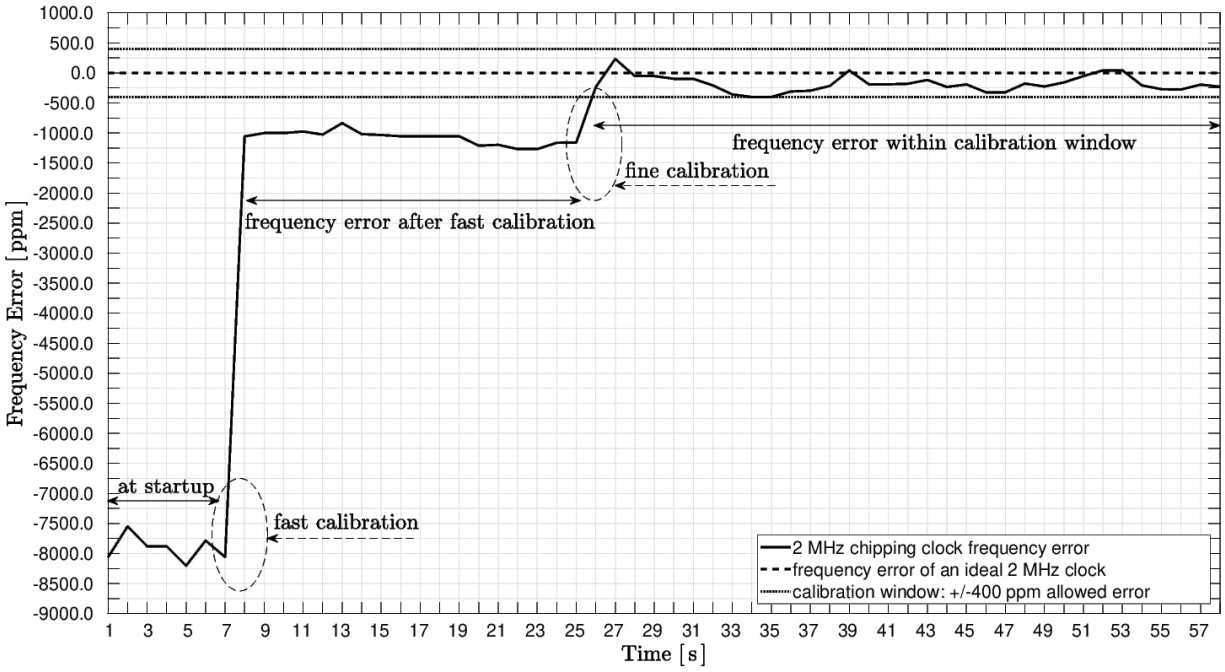


Fig. 7.4 2 MHz chipping clock frequency evolution while running the fast calibration strategy followed by fine clock corrections, laboratory environment. Beacons are periodically sent by an OpenMote.

7.3 Startup phase

The IEEE802.15.4 standard at 2.4 GHz mandates the use of at most 16 channels (numbered 11-26), spaced at 5 MHz, occupying the 2405-2480 MHz band [121]. The term ΔF^{approx} [Hz] defines the frequency resolution of a tuning code, a value chosen in the design phase of the chip. This value may slightly vary from chip to chip and can be influenced by the environmental temperature and operational frequency.

As shown in Fig. 7.1, the startup phase is dedicated to determining the tuning codes of the remaining IEEE802.15.4 channels at startup temperature, once the tuning code B corresponding to channel 11 is found and all the remaining on-chip oscillators are calibrated. As the next IEEE802.15.4 channel, channel 12, has a center frequency spaced 5 MHz above channel 11, the first guess for synthesizing channel 12 would be to use a tuning code C, which can be inferred using (7.3).

$$\begin{aligned} C &= B + offset \\ offset &= \lceil \frac{5MHz}{\Delta F^{approx}\ Hz} \rceil \end{aligned} \tag{7.3}$$

The oscillation frequency of an LC oscillator is given by (7.4) [131]:

$$f = \frac{1}{2\pi\sqrt{L \times C_e}} \tag{7.4}$$

L is the inductance ([H]); C_e is the capacitance ([F]). A linear change in C_e (determined by the

change in tuning code value) around a quiescent point has second order influence on frequency. By analyzing the frequency response to the linear change in tuning codes of the RF oscillator through measurements, we observed that a second degree function approximates f better than a linear function and that this is valid for all the tested temperatures ($5 - 55°$ C). As a consequence, using (7.3) to synthesize the center frequency of channel 12 will introduce three sources of error:

- ΔF^{approx} is an approximation of the particular ΔF of the chip (error ϵ_{Δ})
- $offset$ is an integer approximation of a real number (error $\epsilon_{integer}$)
- C is a linear approximation of a second degree function (error ϵ_{linear})

The total frequency error with respect to center frequency of channel 12 obtained when using tuning code C is then expressed by (7.5):

$$\epsilon^1 = \epsilon_{\Delta} + \epsilon_{integer} + \epsilon_{linear} \tag{7.5}$$

Extrapolating from tuning code B used for channel 11, to synthesizing directly channel $11 + i$ ($i \leqslant 15$) using a linear approximation code $Q = B + i \times offset$, would mean synthesizing a frequency that has an error E with respect to the center frequency of the channel $11+i$: $E \approx i \times \epsilon^1$. The value of E could become so significant that the synthesized frequency is more than one channel apart from the desired communication channel.

In order to determine the tuning codes needed for synthesizing the remaining 15 IEEE802.15.4 channels, and not to deal with cumulative errors, we propose to synthesize channels in consecutive order and refine the channel's tuning code at each step by using IF based calibration (detailed in Chapter 6.4.1), as expressed by (7.6).

$$\begin{aligned} B \text{ (channel 11)} \\ C = B + offset_i \text{ (channel 12)} \\ D = C_{IF} + offset_j \text{ (channel 13)} \\ E = D_{IF} + offset_k \text{ (channel 14)} \\ \dots \end{aligned} \tag{7.6}$$

We see that the tuning code needed for each channel will be determined using the adjusted tuning code after (IF calibration) of the previous channel. This approach synthesizes each channel frequency with an initial error $\epsilon \leqslant \epsilon^1 \ll 1$ MHz (channel edge), that is further reduced to $\epsilon_{integer}$ after IF-based calibration. As an example, when synthesizing a channel's center frequency (F_c) using the tuning function of the radio oscillator f with the tuning codes C and C_{IF} (after IF calibration), the returned errors are ϵ and $\epsilon_{integer}$ respectively, expressed by (7.7).

$$\begin{aligned} f(C) = F, \quad f(C_{IF}) = F_{IF} \\ |F - F_c| = \epsilon, \quad |F_{IF} - F_c| = \epsilon_{integer} \\ \epsilon_{integer} \ll \epsilon \end{aligned} \tag{7.7}$$

This means that, for determining with a good accuracy all 16 IEEE802.15.4 channels in the startup phase, at least one beacon has to be received on each channel, which enables the use of IF calibration and ensures that the radio frequency accuracy meets the $\pm$40 ppm accuracy requested by the standard. We will discuss two solutions for determining the tuning codes of the remaining IEEE802.15.4 channels at startup temperature, assuming that channel 11 is synthesized using tuning code B: the first solution is based on Recursive Least Squares (RLS) and the second one on Moving Average (MA). Our results will show the initial frequency errors of synthesizing new channels (ϵ, before applying IF corrections) of the two approaches through simulations and experimental validation. These initial errors are representative for evaluating how far the tuning tried by the crystal-free device is from the actual channel frequency that is trying to synthesize. The smaller the initial errors will be, the better the algorithms perform. The higher they are, the smaller the chances to correctly receive a network beacon and perform IF calibration. As soon as a network beacon is received on the new synthesized frequency, the initial error (ϵ) is reduced by the means of IF calibration to a value within $\pm$ 40 ppm, as showed in the previous chapter. As all the channel frequency errors will be within $\pm$ 40 ppm after IF calibration, we will only show plots of the evolution of the initial error (ϵ) when using the proposed algorithms.

7.3.1 RLS channel tuning

We propose the use of a Recursive Least Squares-based algorithm for determining the tuning codes of each IEEE802.15.4 channel at startup temperature. This algorithm outputs parameters a of a fit function y_{fit} that approximates f, the oscillator's tuning function. RLS is known to perform well when input data is provided sequentially [132], improving the fitting with every data point received. The RLS procedure is illustrated in Fig. 7.5.

The RLS-based algorithm requires as inputs the tuning code B for synthesizing channel 11, the approximate frequency resolution of a tuning code ΔF^{approx} and the forgetting factor λ. The algorithm is already set to fitting the measured frequency data on a second degree function:

$$y_{fit}(x) = a_3 \times x^2 + a_2 \times x + a_1 \tag{7.8}$$

where $y_{fit}(x)$ represents the synthesized frequency when using tuning code value x. The value of the forgetting factor λ defines the system memory and has an impact to the ability to track the changes in the input sequence and to the stability of the coefficients [133]:

$$\lambda = \frac{N}{N+1} \tag{7.9}$$

where N is the number of iterations before the effect of a disturbance has decayed to e^{-1} of its initial value [133].

The algorithm will predict the tuning code to be used for synthesizing each IEEE802.15.4 channel, from channel 12 to channel 26. The initial values of the coefficients a are null, but the a_1

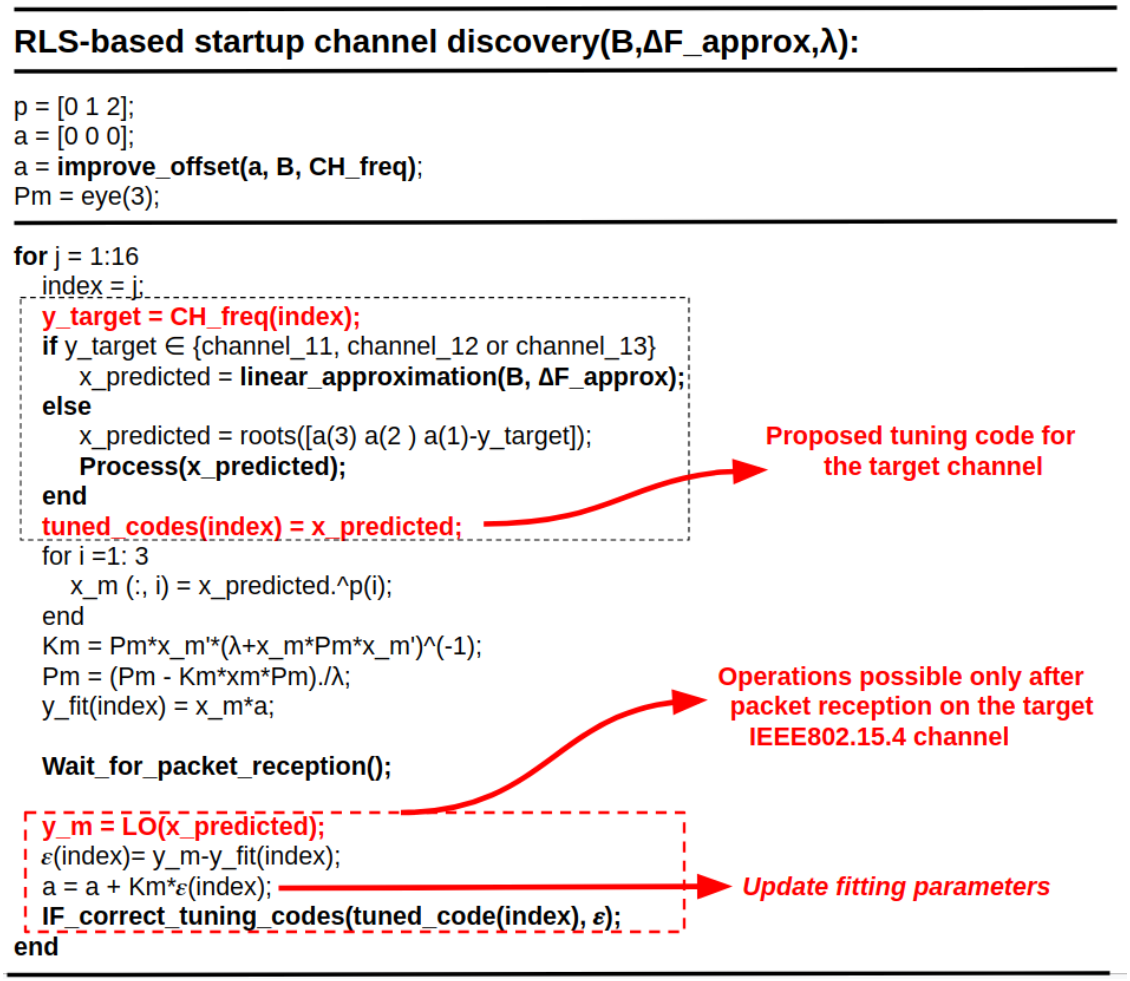


Fig. 7.5 RLS-based approach for consecutively determining the tuning code of each IEEE802.15.4 channel at startup temperature. At each step, a prediction of the tuning code to be used is made, $tuned\_codes$. After a packet reception, the fitting function coefficients a are improved and IF correction is applied to the tuning code prediction.

term (the offset) guess can be improved ($improve\_offset$), to speed up the fitting process:

$$a_1 \approx 2405 - B \times \Delta F^{approx} \text{ [MHz]} \tag{7.10}$$

The tuning codes for the channels (11-known) 12 and 13 will be determined using an approximation function ($linear\_approximation$), as expressed in equation (7.6), because the algorithm needs at least 3 (m) data points before fitting to a second ($m-1$) degree function. After this, the tuning code of target channel j is the solution ($x\_predicted = x$) of the fitting provided up to that point ($a_3(j)$, $a_2(j)$, $a_1(j)$):

$$y\_target(j) = a_3(j) \times x^2 + a_2(j) \times x + a_1(j) \tag{7.11}$$

where $y\_target(j)$ is the frequency of the target channel and j is the algorithm iteration ($j \in 1..16$, one iteration per channel). A problem with this approach is that the solution of a second degree polynomial is not unique. We tackled this issue in $Process()$ function, by providing positive

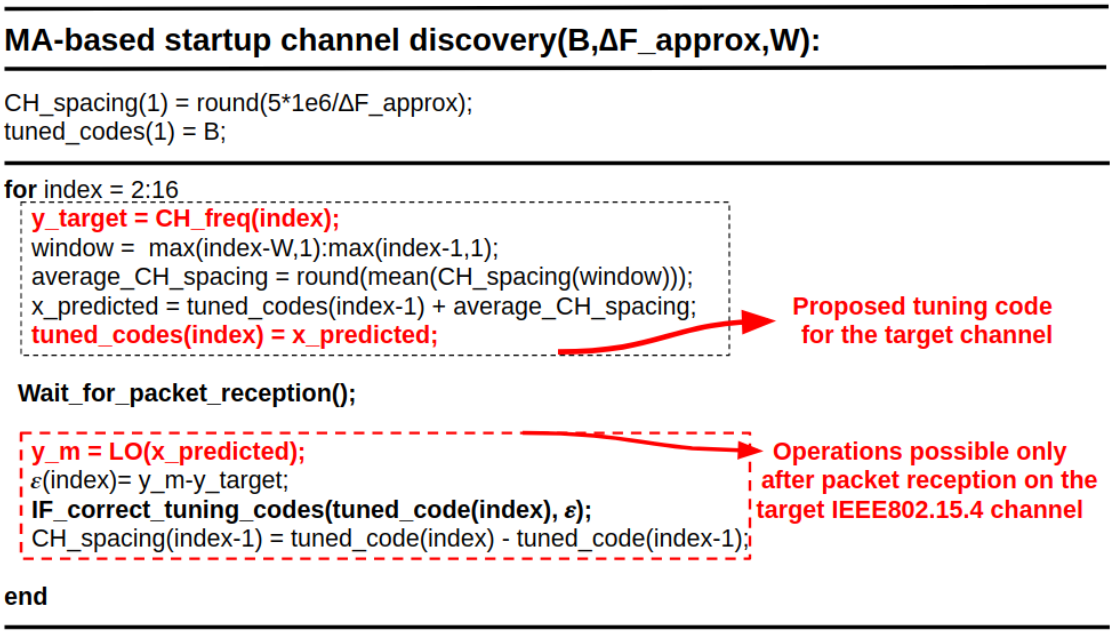


Fig. 7.6 MA-based approach for consecutively determining the tuning code of each IEEE802.15.4 channel at startup temperature. At each step, a tuning code to be used is proposed based on previous channel spacing, $tuned\_codes$. After packet reception, IF correction is applied to the current tuning code.

(integer) solutions that fit with respect to the tuning codes of already discovered channels, provided the approximate frequency resolution of a code, ΔF^{approx}.

After a beacon is received on the target channel, the synthesized frequency can be measured,

$$y_m = LO(x\_predicted) = y\_target + IF\_offset \tag{7.12}$$

as well as the error with respect to the target frequency, $\epsilon = IF\_offset$. This error will be further used to adjust the predicted tuning code, $tuned\_codes(j)$ (using IF calibration) and to improve the fitting coefficients, a. At this point the algorithm continues by synthesizing the next target channel ($j = j + 1$). The algorithm ends when the tuning code needed for synthesizing all 16 channels are determined ($j = 16$, corresponding to channel 26).

7.3.2 MA channel tuning

The Moving Average based algorithm (Fig. 7.6) returns a value for the tuning code to be used for each IEEE802.15.4 channel, consecutively, having as inputs the tuning code value B (channel 11), the approximate frequency resolution of a tuning code ΔF^{approx} and the window size W. For determining the tuning code of a new channel, the algorithm will use the closest integer value to the average spacing between the channels inside the window W. After tuning on a channel using the proposed tuning code, the algorithm will wait for receiving a packet on that channel.

After a packet reception, the error ϵ with respect to the channel's center frequency is computed and the tuning code value is adjusted, as well as the channel spacing between the current channel

and the previous. We assume that a packet can be received on each channel because the approximation ΔF^{approx} is good enough so that the errors in synthesizing a channel are below 1MHz (and we also synthesize the channels in consecutive order and apply IF calibration after each step). What we cannot control is the channel quality, or how much we have to wait in order to receive a beacon on the respective channel.

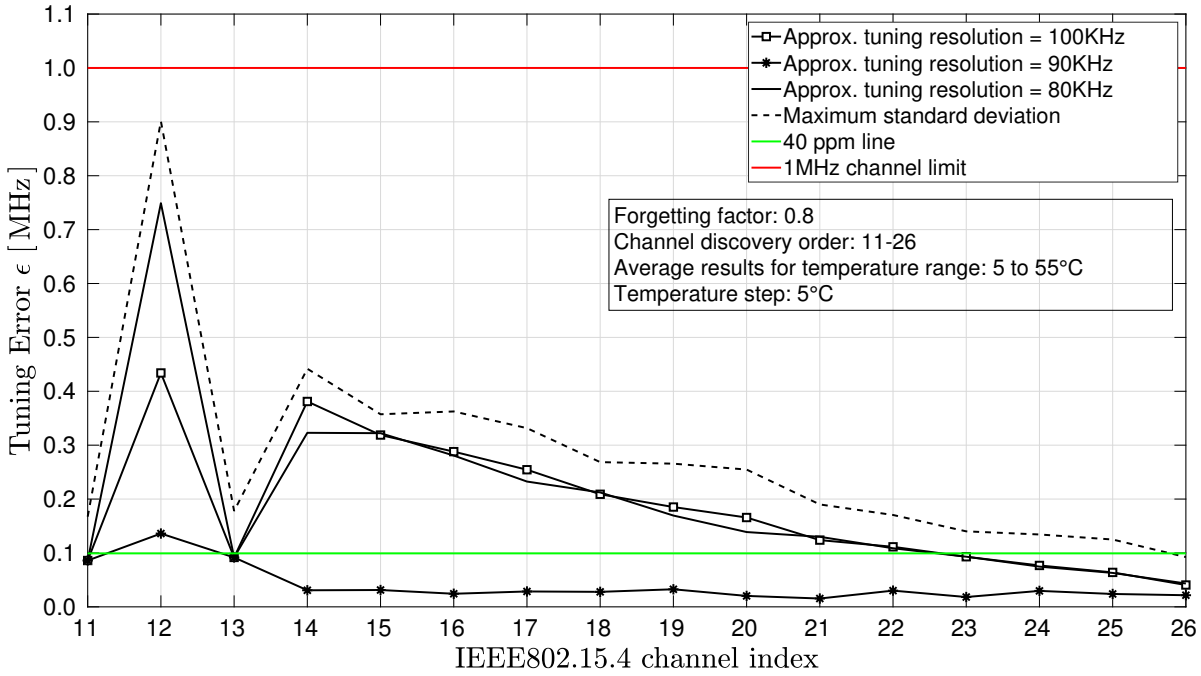
7.3.3 Performance evaluation: simulations

For evaluating the performance of the proposed algorithms in an ideal case, we run the algorithms on simulation data. This data was created by smoothing the irregularities (see Fig. 7.9b) from experimentally collected frequency data for each tuning code, when the crystal-free radio was placed in a temperature chamber. The temperature was changed from 5°C to 55°C in 5°C steps. The performance of the algorithms with experimental data is discussed in Section 7.3.4.

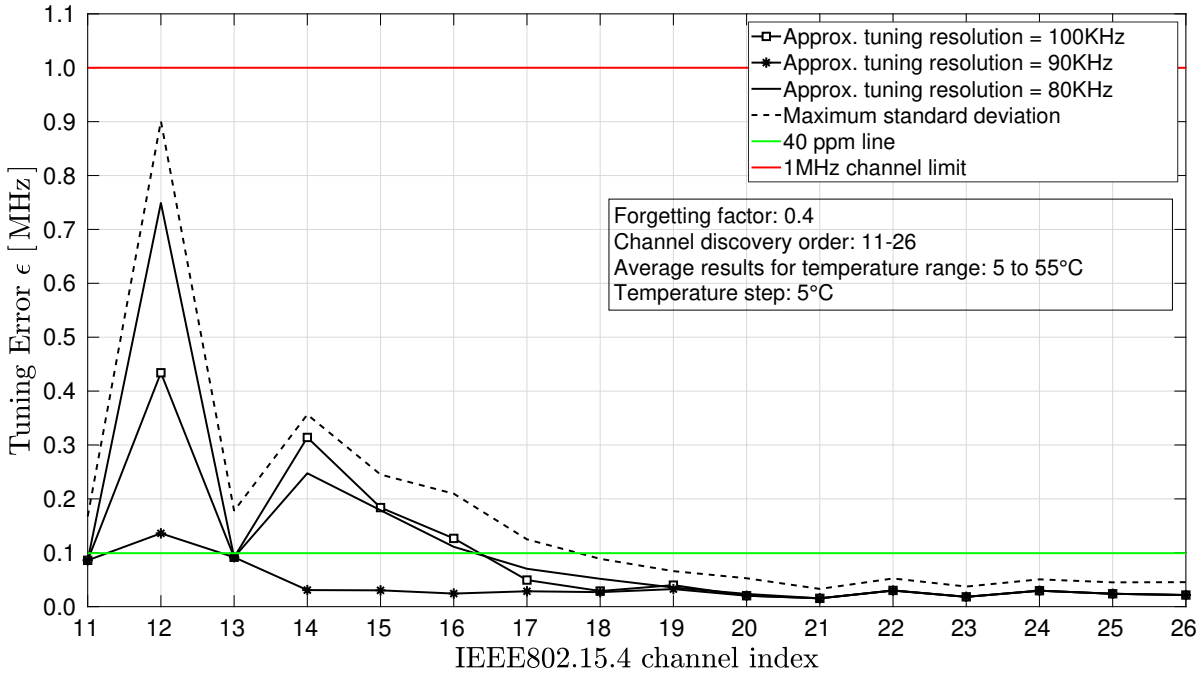
Figs. 7.7 and 7.8 show the evolution of ϵ (the frequency error before IF calibration) when using the RLS-based and MA-based algorithms respectively for discovering each IEEE802.15.4 channel at startup temperature. The algorithms were applied to startups at each of the evaluated temperatures and the average performance was plotted. The red line represents the 1 MHz error limit, above which the packets sent on a channel cannot be detected by the receiver. The green line represents the 40 ppm error, as imposed to the RF accuracy at all times, by the IEEE802.15.4 standard. All presented results represent the frequency synthesizing error before IF corrections.

For the RLS-based algorithm, the impact of the forgetting factor λ can be seen when comparing Fig. 7.7a ($\lambda = 0.8$) to Fig. 7.7b ($\lambda = 0.4$). A higher λ means that the system has a larger memory and so, the function coefficients are updated slower and the convergence rate of ϵ is slower. A lower value of λ gives more weight to recent data samples, improving the tracking ability of the algorithm. Still, when the approximation of ΔF^{approx} is closer to reality (see results for $\Delta F^{approx} = 90kHz$, as $\Delta F^{real} = 93kHz$), the value of λ does not impact the obtained results, as the algorithm starts with small initial error ϵ. For this case, starting with channel 13, all following channels can be synthesized with an error below 40 ppm. When startup errors are high (as it is the case for $\Delta F^{approx} = 80kHz$), a smaller λ is able to faster bring the tuning errors below the 40 ppm limits imposed by the IEEE802.15.4 standard. For this case, we see in Fig. 7.7b that channels following channel 16 are synthesized with very good accuracy, no matter the high startup errors (as long as they are below the red 1MHz limit line).

The Moving Average based algorithm is analyzed for window sizes of 1, 4 and 16, in Fig. 7.8. For this algorithm, the initial approximation of the frequency resolution of a tuning code (ΔF^{approx}) only impacts the accuracy of synthesizing channel 12, as seen in Fig. 7.8 with blue markers. After IF correction, the spacing between channel 11 and channel 12 is adjusted and used to determine the tuning code of the next channel, so ΔF^{approx} value has no further impact on the results. For determining the tuning code value of each new channel, the average of the past W channel spacing values is used. The simulation results in Fig. 7.8 show that smaller window sizes W tend to approximate better the real channel spacing values and are able to synthesize new channels with



(a) $\lambda = 0.8$



(b) $\lambda = 0.4$

Fig. 7.7 Startup phase: RLS algorithm evaluation for a) $\lambda = 0.8$ and b) $\lambda = 0.4$. Tuning error vs discovered channel for three approximations of ΔF^{approx}: 100kHz, 90KHz and 80 kHz. Performance averaged over startups at environmental temperature between $5 - 55°$ C.

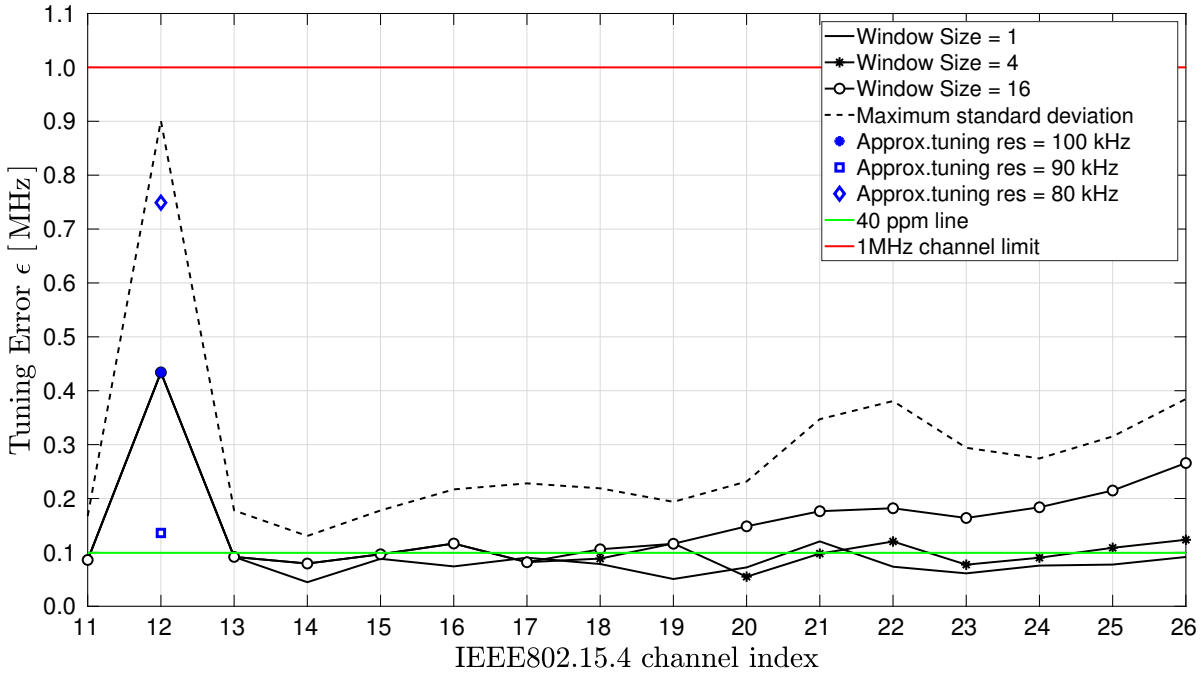


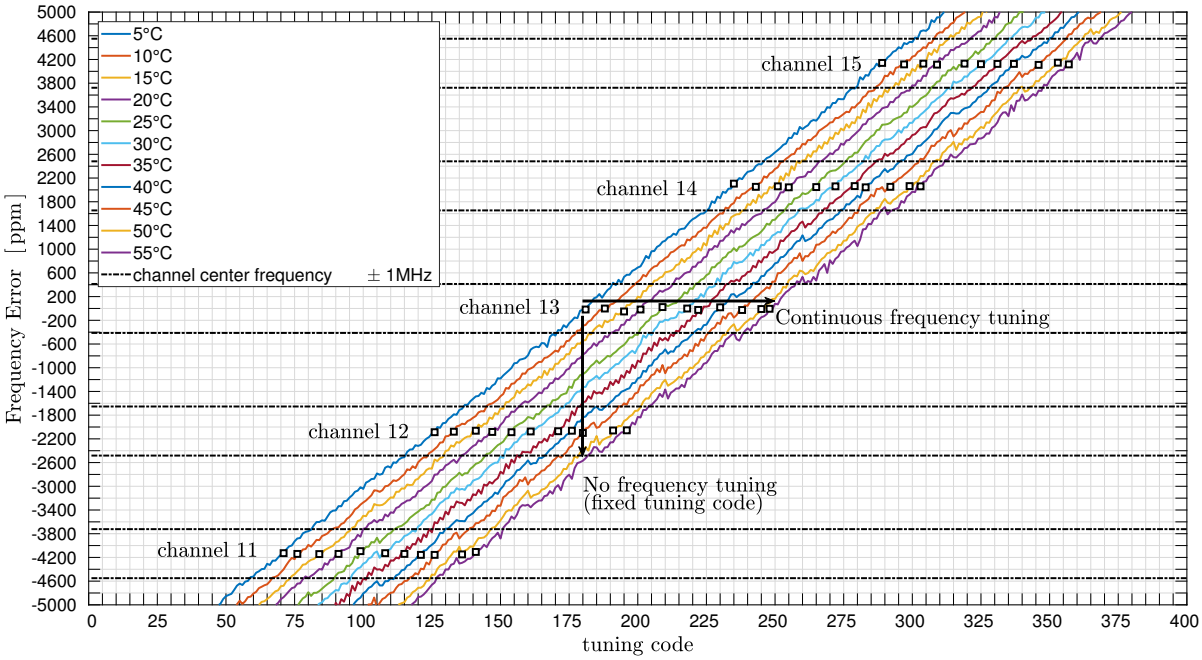
Fig. 7.8 Startup phase: MA algorithm evaluation for window sizes of 1, 4 and 16. Tuning error vs discovered channel for three approximations of ΔF^{approx} : 100kHz, 90KHz and 80 kHz (blue markers). Performance averaged over startups at environmental temperature between $5 - 55$°C.

an error $\epsilon \approx 40$ ppm. As for the RLS-based algorithm, the presented results are the average performance of the algorithm when applied to startups at temperatures between 5°C and 55°C, in 5°C steps.

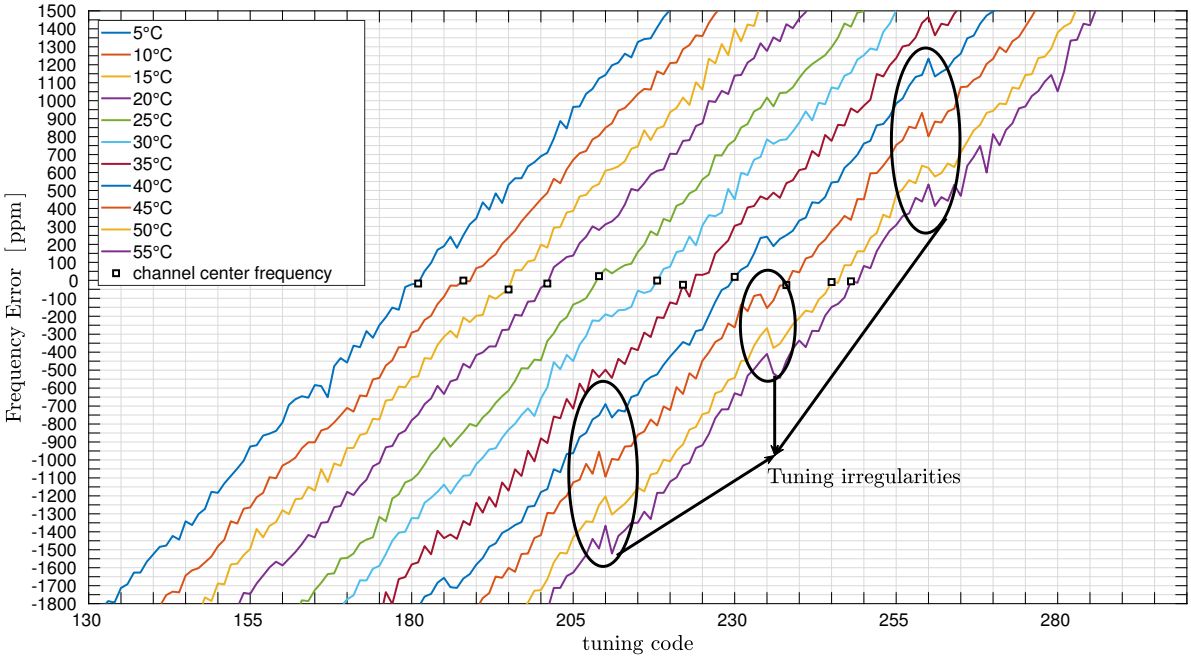
In Fig. 7.7a, Fig. 7.7b and Fig. 7.8, the error ϵ corresponding to channel 11 has already the value determined after IF calibration, as it is the channel that is discovered in the Initial Calibration phase (Section 7.2). The initial error corresponding to synthesizing channel 12 depends on the accuracy of ΔF^{approx}: the closer this approximation is to the real value (in our case $\Delta F^{real} = 93kHz$), the smaller the initial error. For determining channel 13, a new estimation of ΔF^{approx} is made based on the spacing between channels 11 and 12, so that the frequency error corresponding to this channel is improved. The tuning codes for the remaining channels (14 to 26) are determined as specified by each of the two algorithms.

7.3.4 Experimental validation

We place the crystal-free platform in a temperature chamber and vary the temperature between 5°C and 55°C, in 5°C increments. When measuring the output frequency returned by each tuning code of the crystal free platform, at each temperature, we obtain plots which are not perfectly smooth, as shown in Fig. 7.9b. These irregularities are the combined result of unstable temperature, limited frequency counter accuracy, oscillator phase noise and tuning function limitations (fine tuning with 5-bit codes). The immediate consequence of these irregularities is that ϵ errors are higher when applying the RLS and MA algorithms to this data.

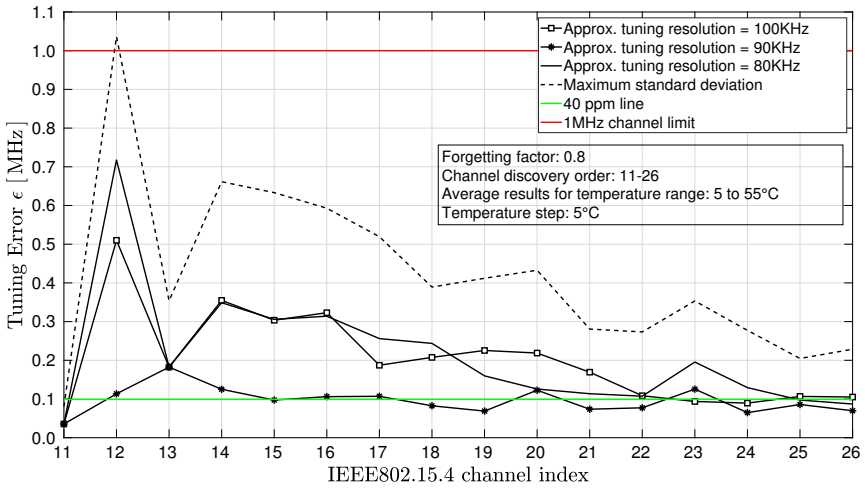


(a) Radio tuned on channel 13 center frequency

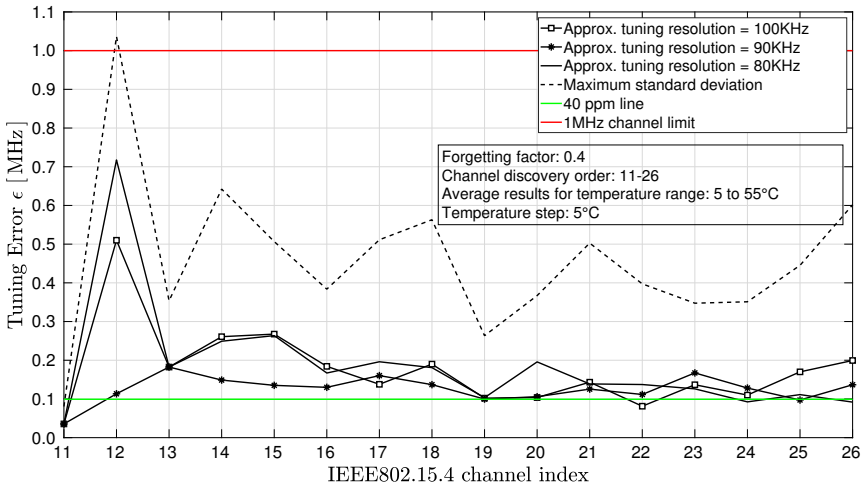


(b) Frequency irregularities on experimental data

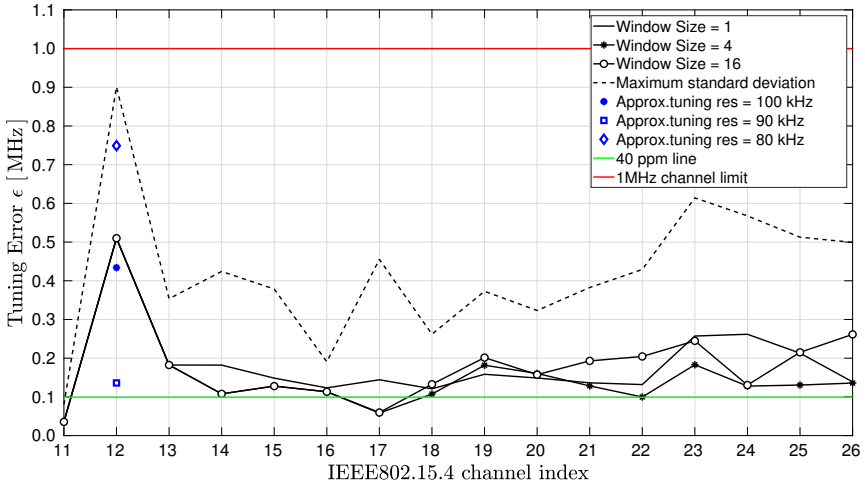
Fig. 7.9 a) Radio tuned on channel 13 center frequency. The temperature effect can be compensated by continuously adjusting the tuning codes of a channel. Experimental data. b) Zoom in: Output frequency irregularities obtained with experimental data.



(a) RLS, $\lambda = 0.8$



(b) RLS, $\lambda = 0.4$



(c) MA, window size of 1, 4 and 16

Fig. 7.10 Tuning error vs discovered channel for three approximations of ΔF^{approx}: 100kHz, 90KHz and 80 kHz. Performance averaged over startups at environmental temperature between $5 - 55°$ C. Startup phase on measurement data: a) RLS algorithm evaluation for $\lambda = 0.8$; b) RLS algorithm evaluation for $\lambda = 0.4$; c) MA algorithm evaluation for window sizes of 1, 4 and 16.

For the RLS-based algorithm in Fig. 7.7, the tuning errors are well below the 40 ppm threshold. For the startup phase with experimental data (Figs. 7.10a and 7.10b) the obtained tuning errors evolve close to this limit, with high standard deviations caused by the irregularities. Higher λ values are equivalent to less influence of new input disturbances. In Fig. 7.10a, the standard deviations of the error has a decreasing trend when compared to Fig. 7.10b. The same happens for the MA-based algorithm (Fig. 7.10c), that shows reduced errors for higher W values. Still, the errors evolve slightly above the green line, with higher standard deviations that the ones obtained with smoothed simulation data (Fig. 7.8).

7.4 Normal operation phase

Once the tuning codes of all 16 IEEE802.15.4 channels are known at startup temperature, the crystal-free device enters the normal operation phase. In this phase, the device will communicate according to a dynamic communication schedule and it has to maintain the communication ability even if the environmental temperature evolves. Fig. 7.9a shows the effect of environmental temperature on the communication accuracy (with experimental data). If the crystal-free radio is tuned on channel 13 for example, if the tuning code is not adjusted, the frequency will drift outside the channel. As seen in Fig. 7.9a, for a 50°C temperature difference, the synthesized frequency has drifted more than one communication channel ($>$ 5MHz), to a value below channel 12.

This section will analyze the ability of keeping communication accuracy on all 16 IEEE802.15.4 channels, when the environmental temperature changes in a range of 5 to 55°C. We assume that at each temperature, the crystal-free device will communicate on a random sequence of channels and that the temperature change rate is slow enough so that at least one channel can be accurately known (synthesized) as the temperature changes, by the means of IF-based calibration, described in Section 6.4.1. The challenge treated in this section is tuning to the communication channels in random order (as the communication schedule will require) with an as-low-as-possible frequency error, at any temperature, when using techniques based on RLS and MA, respectively.

7.4.1 RLS channel tunning

The RLS-based algorithm in normal operation mode makes use of the tuning function approximation y_{fit} determined at startup temperature T_0 (Section 7.3.1), and of the tuning code value U of any channel Y at the new environmental temperature, T_1, expressed by (7.13).

$$\begin{aligned} y_{fit}(x, T_0) &= a_3 \times x^2 + a_2 \times x + a_1 \\ x &= U \text{ :tuning code for channel Y at } T_1 \end{aligned} \tag{7.13}$$

The tuning function is further adapted for the new environmental temperature, T_1, using (7.14).

$$\begin{aligned} y_{fit}(x, T_1) &= a_3 \times x^2 + a_2 \times x + a_{1T_1} \\ a_{1T_1} &\approx a_1 - f\_shift \\ f\_shift &= y_{fit}(U, T_0) - F_Y \end{aligned} \tag{7.14}$$

Where a_{1T_1} represents the new coefficient approximation at temperature T_1, F_Y represents the frequency value of channel Y, and f_shift is an approximation of the frequency shift caused by T_1 with respect to the function approximation at T_0.

The new function approximation $y_{fit}(x, T_1)$ and its coefficients a_3, a_2 and a_{1T_1} are used by the RLS-based algorithm (Fig. 7.5) to predict the new tuning codes to be used by any channel demanded by the communication schedule (with frequency value y_{target}), in any order. Coefficients a_3 and a_2 are updated by the algorithm to the values corresponding to the new environmental temperature, a_{3T_1} and a_{2T_1}, respectively. The tuning function for T_1 thereby changes as expressed

in (7.15):

$$y_{fit}(x, T_1) = a_{3T_1} \times x^2 + a_{2T_1} \times x + a_{1T_1} \tag{7.15}$$

The same technique is used as the temperature evolves from T_1 to T_2 and to any other temperature. As a good function approximation $y_{fit}(x, T_0)$ is provided after the startup phase, the impact of the forgetting factor λ is limited.

7.4.2 MA approach

The MA-based algorithm enters the normal operation phase with the complete knowledge of vector $tuned\_codes(channel, T_0)$, which holds the values of the tuning codes for synthesizing each IEEE802.15.4 channel at startup temperature T_0. As for the RLS case, by the means of IF calibration, the tuning code value U for synthesizing channel Y at T_1 is known and is stored as $tuned\_codes(Y, T_1)$. For channel Y, the shift in tuning code caused by the temperature changing from T_0 to T_1 is expressed by (7.16).

$$shift = tuned\_codes(Y, T_0) - tuned\_codes(Y, T_1) \tag{7.16}$$

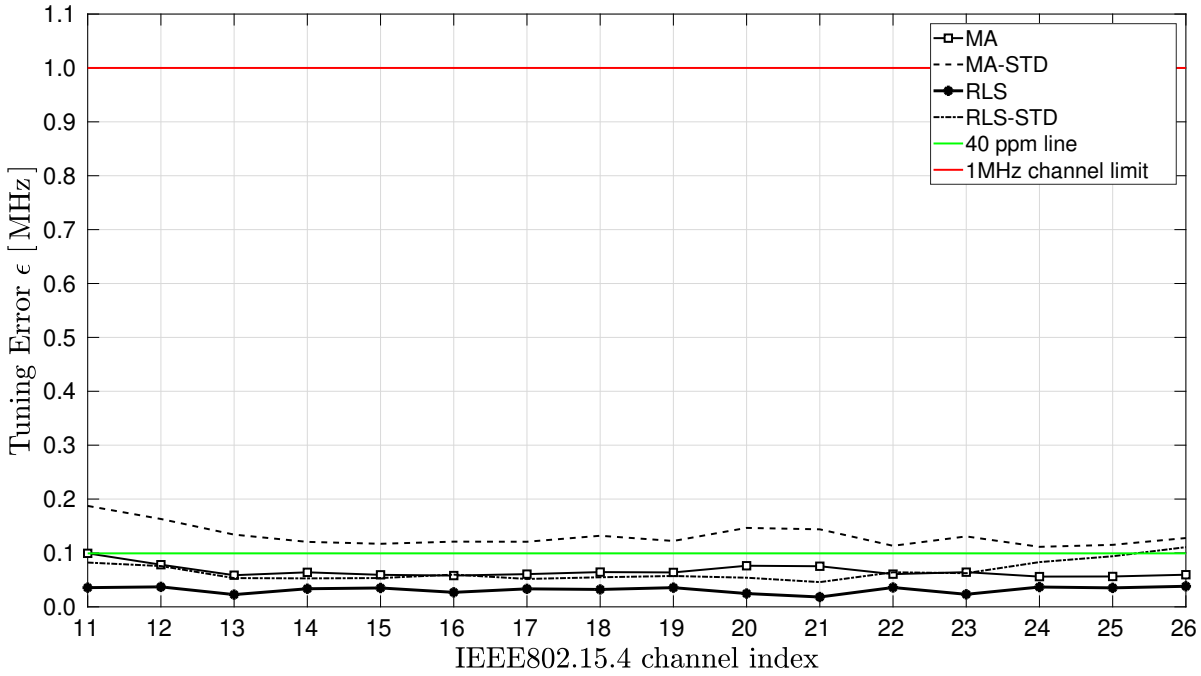
The algorithm detailed in Fig. 7.6 is still valid, but in normal operation mode, the variables inside the averaging window W are shifts in tuning code caused by the changing temperature, and not the channel spacing as in startup phase. The window contains the shifts undergone by the closest W known channels to the channel of interest, y_target. The tuning code predicted by the MA algorithm for a channel W at T_1 is expressed by (7.17).

$$tuned\_codes(W, T_1) = tuned\_codes(W, T_0) + avg\_shift\_W \tag{7.17}$$

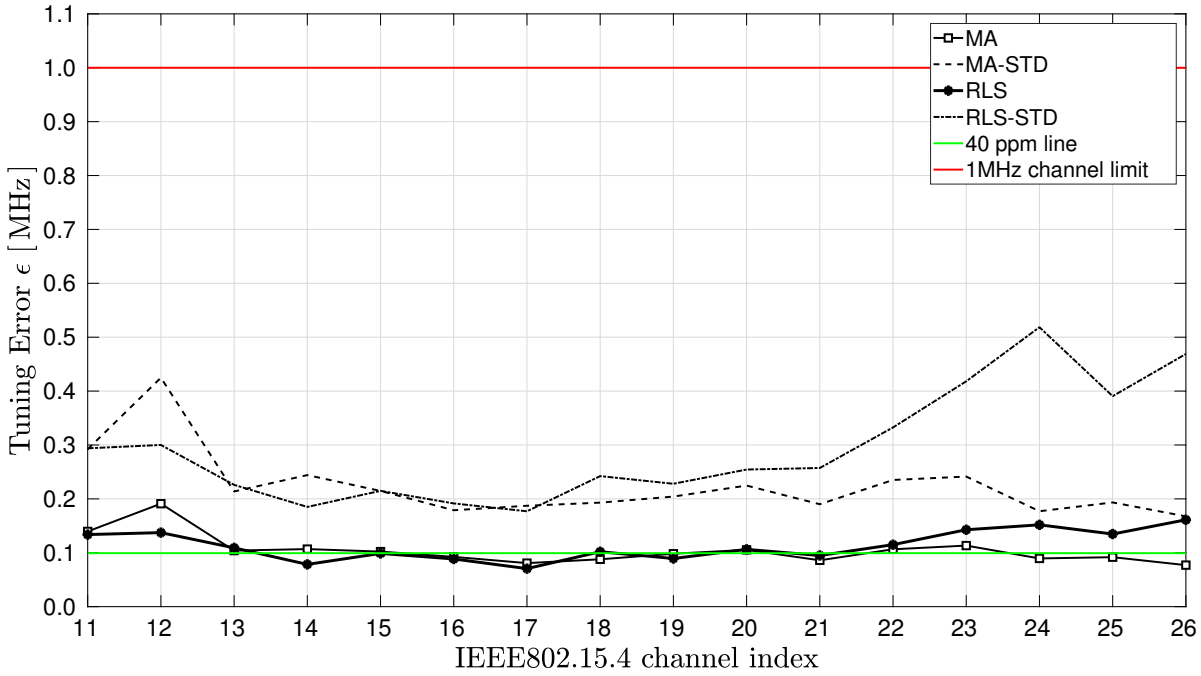
7.4.3 Performance evaluation: simulations

We evaluate the performance of both algorithms by simulating a change in temperature from 5°C to 55°C, in steps of 5°C, using the simulation data described in Section 7.3.3. At each temperature value, all channels are synthesized in random order, as dictated by the communication schedule. As in the previous sections, the error ϵ represents the error in the synthesized frequency with respect to the frequency value of the channel of interest (before IF-correction). The IEEE802.15.4 standard mandates to be below 40 ppm at all times.

Fig. 7.11a shows the average performance of the RLS and MA based algorithms in normal operation mode, obtained by Monte Carlo simulation over the communication schedule at each temperature. In the legend of Fig. 7.11a, the notation "STD" stands for "standard deviation" of the average performance of each algorithm. We see that the average frequency synthesizing error ϵ obtained with the proposed algorithms is below the 40 ppm limit for each channel. Still, for the RLS algorithm, the standard deviation of the average performance is also below 40 ppm, which is not the case for the MA algorithm.



(a) Simulation data



(b) Measurement data

Fig. 7.11 Normal operation on a) simulation data and b) measurement data. Monte Carlo simulation over the performance of RLS and MA based algorithms when synthesizing channels in random order as the environmental temperature slowly changes from 5 to 55°C.

7.4.4 Experimental validation

In normal operation phase, the environmental temperature changes and the channel tuning needs to be adjusted with it. Fig. 7.11b shows the average tuning error ϵ obtained with experimental data, when using the RLS-based and MA-based algorithms. Even if the average error lies on the 40ppm line, the obtained accuracy and standard deviations are worse than for the simulation data showed in Fig. 7.11a. The standard deviation of the error obtained with the RLS algorithm tends to be worse than that of the MA algorithm for experimental data. A cause for this may be that the RLS based algorithm continuously tries to fit data to a second degree function and the irregularities shown in Fig. 7.9b (that worsen with the temperature) lead to fitting on a function different from the real tuning function.

7.5 Frequency synthesis in transmission mode

This section characterizes the behaviour of the RF oscillator in transmission mode and describes the steps to follow in order to be able to reliably transmit packets on IEEE802.15.4 channels using a crystal-free radio. From the best of our knowledge, this work is the first one to address this challenge.

Up until this point in the chapter we treated techniques to initially calibrate the RF oscillator and the remaining on-chip oscillators of a crystal-free radio, followed by methods for synthesizing the IEEE 802.15.4 channels in both constant and variable temperature environment. During all these phases, the crystal-free radio was set to reception mode and RF oscillator tuning corrections were possible while measuring the IF offset during reception of network beacons. The crystal-free radio is able to store the tuning codes needed for being able to receive on any of the 16 IEEE802.15.4 channels.

However, when switching to TX mode, the tuning codes used for synthesizing channels in RX mode are not of use anymore. As different circuitry is involved when transmitting data, the same RF oscillator in TX mode behaves differently than in RX mode. This is best shown in Fig. 7.13, where the tuning codes needed for synthesizing channels 11 to 26 when the RF oscillator is in RX mode and then TX mode are plotted (experimental data). Knowing the tuning code difference when switching from RX mode to TX mode for channel 11 for example (50 codes in Fig. 7.13a), cannot be extrapolated to the remaining channels, as this difference tends to increase with frequency (75 codes for channel 26 in Fig. 7.13a). This difference caused by switching from RX to TX mode is also influenced by environmental temperature. We can see in Fig. 7.13b how at $50°$C the tuning code difference that needs to be applied when switching from RX mode to TX mode for synthesizing the same channel is now smaller for all channels when compared to Fig. 7.13a: only 28 codes for channel 11 and 55 codes for channel 26.

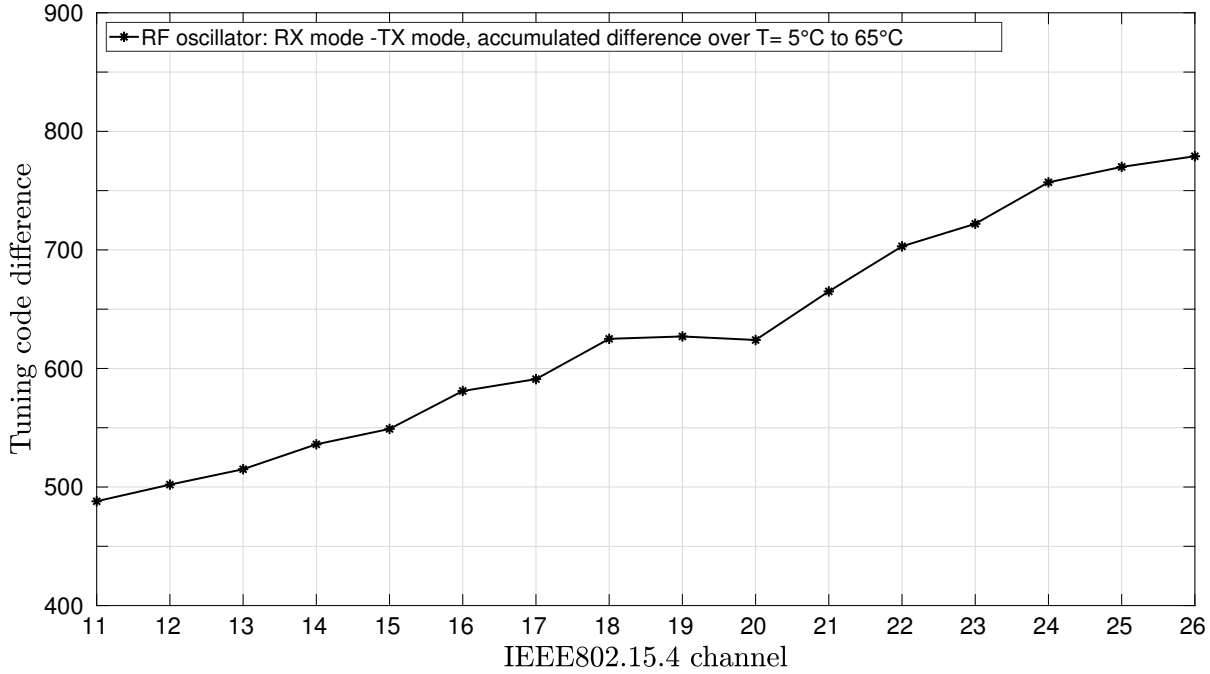
As a generalization for the difference in tuning codes for synthesizing the same channel in RX mode and then TX mode, we plotted in Fig. 7.12a this trend across channels 11-26, accumulated over a 60 °C temperature change. The RX mode - TX mode tuning code difference is higher for higher frequency channels.

Fig. 7.12b shows the sum of differences (RX mode - TX mode) in tuning codes for all 16 channels and how it evolves with temperature. We see that as the temperature increases, the RX mode - TX mode gap decreases for all channels, as also noticed in Fig. 7.13b.

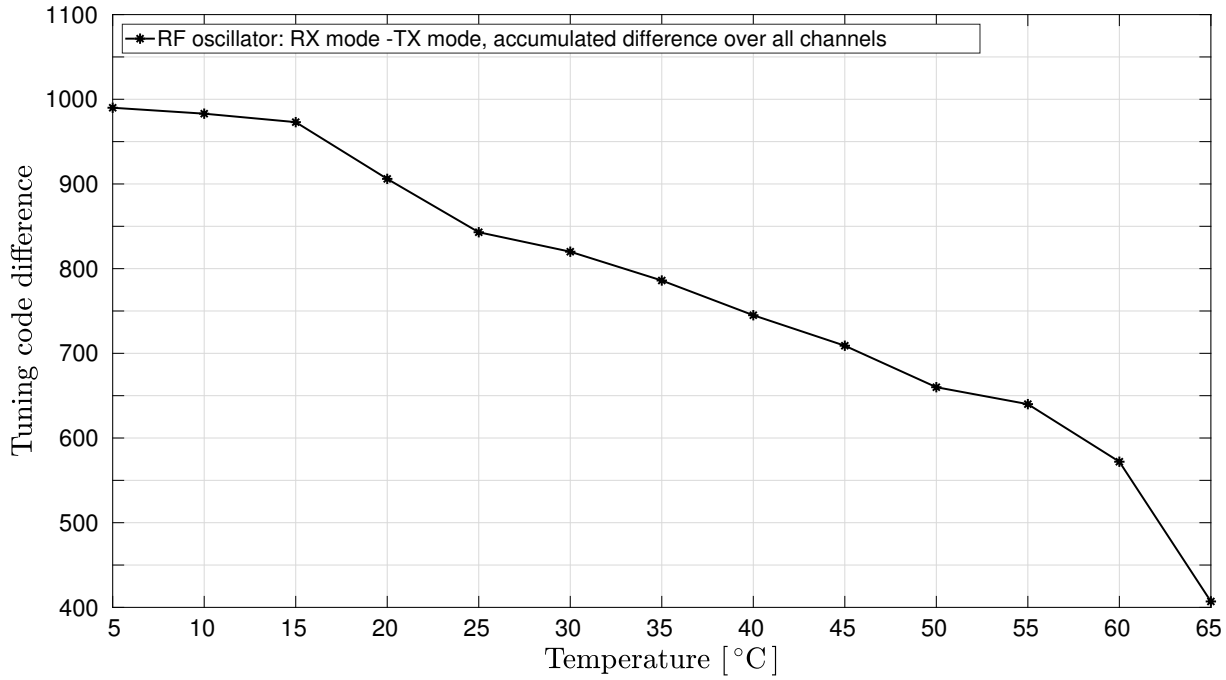
7.5.1 Proposed approach

Because the RF oscillator in TX mode is characterized by a tuning function with different coefficients than in RX mode, both with frequency and temperature, a similar channel discovery as done in RX mode is advisable (Sections 7.3 - 7.4), with a few differences:

1. the RF oscillator needs to be calibrated in RX mode (Startup phase completed) so as to ensure ACK reception in case of successful transmission attempt (see Fig. 7.14).

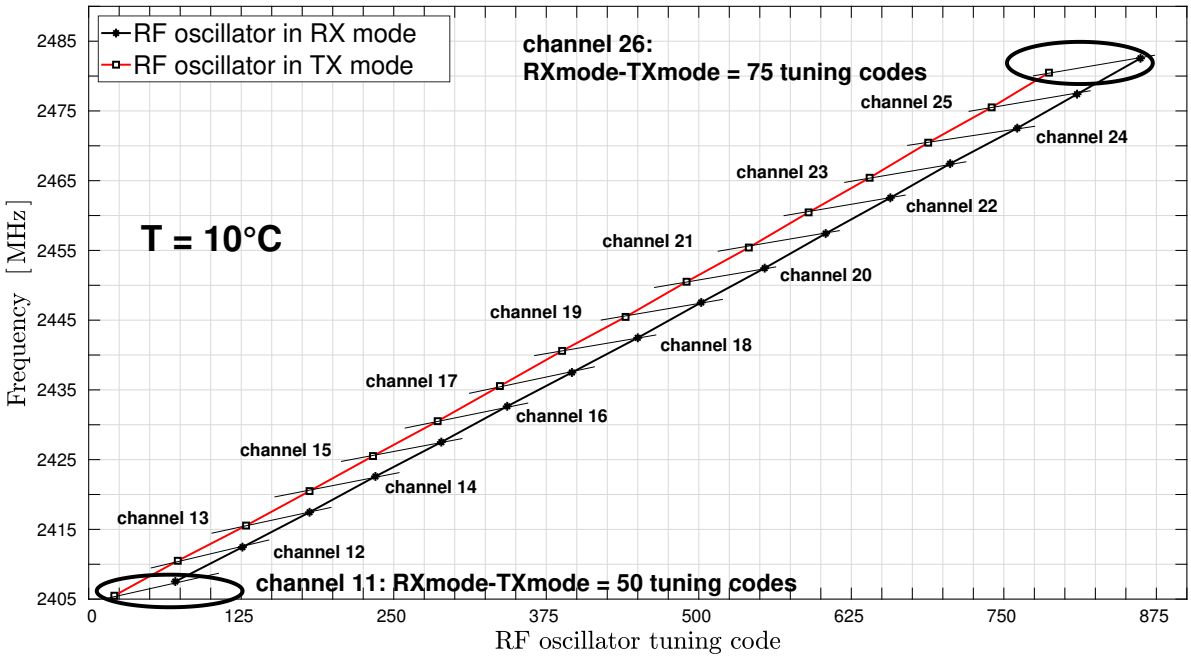


(a) RX mode- TX mode tuning code difference over channels

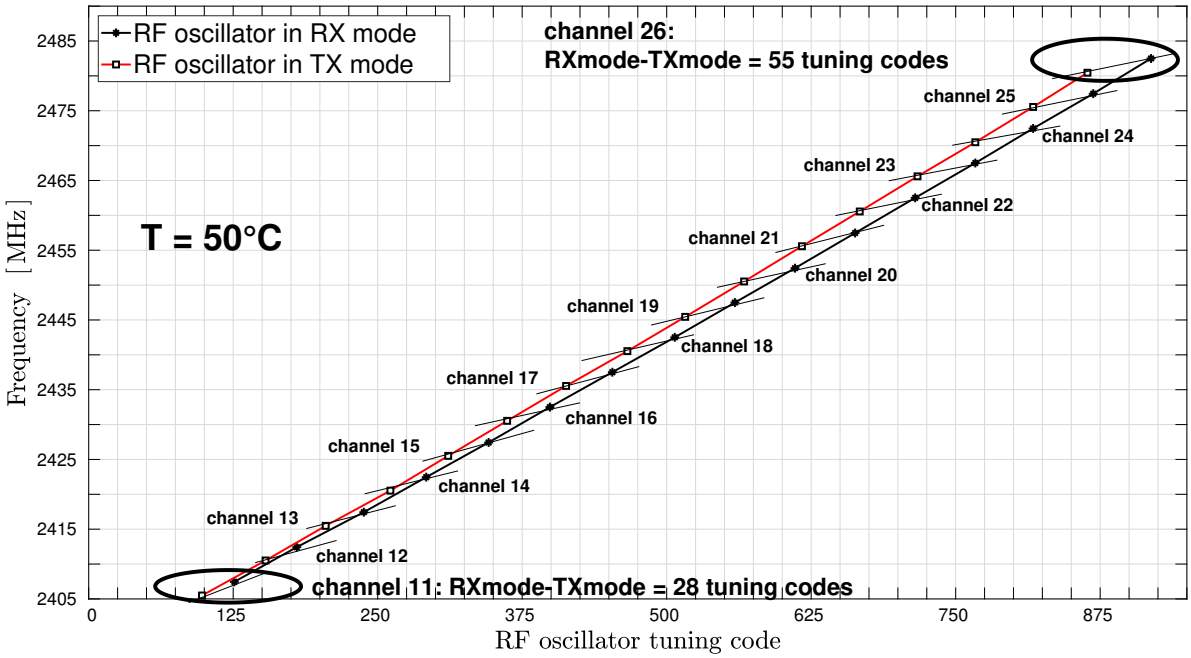


(b) RX mode- TX mode tuning code difference over temperature

Fig. 7.12 Crystal-free radio, RF oscillator, experimental data. a) For higher frequency channels, the tuning code difference needed when switching from RX mode to TX mode is higher than for lower frequency channels, at all temperatures. b) The trend with temperature of the aforementioned switching difference.



(a) T=10°



(b) T=50°C

Fig. 7.13 Crystal-free radio: RF oscillator tuning codes for synthesizing the 16 IEEE802.15.4 channels in RX mode and TX mode at a) 10° C and b) 50°C. Experimental data.

2. a good starting value for ΔF^{approx} is already provided by analyzing the channel spacing in RX mode (specifically, for RLS, the a_2 term returned after running the algorithm and for MA the average of $CH\_spacing$ vector).

3. the tuning code B^{TX} corresponding to channel 11 in TX mode is to be found using a calibrated on-chip frequency reference.

4. further corrections to be applied to the frequency synthesized using the tuning code B^{TX} in TX mode are based on the frequency offset (signed) value included in the ACK received from the destination device (replacing the function $IF\_correct\_tuning\_codes$ in Section 7.3).

We see that the above mentioned differences impact only the initialization of the Startup phase for TX mode. The Normal Operation phase is as described in Section 7.4 for RX mode, but with using the frequency offset information included in ACKs for frequency corrections, and not the IF value (available only at packet reception- RX mode). For these reasons, we will include in this section the results of the algorithms running on simulation data and measurement data for the Startup Phase only. We will see that these results match the ones obtained for RX mode, for the case of good approximations of ΔF^{approx} (see Section 7.3).

The flowchart represented in Fig. 7.14 places the right moment to start calibrating the RF oscillator for transmission to be only after the on-chip oscillators are calibrated at current temperature and the device is able to receive network beacons and acknowledgements.

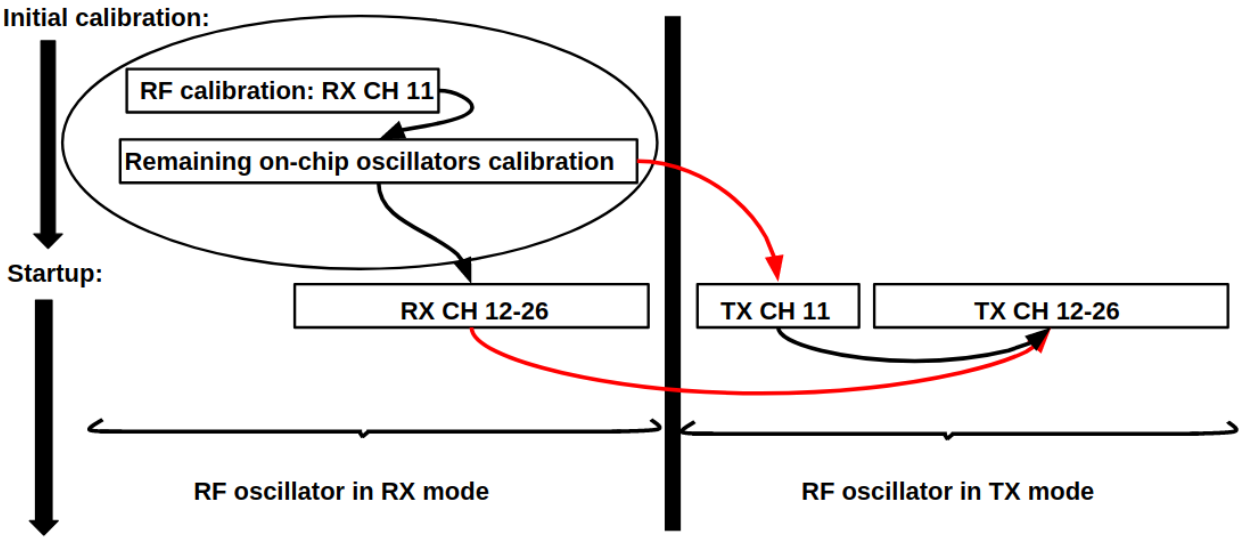


Fig. 7.14 Crystal-free radio: determining the tuning codes for transmission on channels 11-26 is possible after the Initial Calibration phase has been completed and ACK reception is possible (RF oscillator calibrated in Startup Phase for RX mode).

In order to determine the tuning code B^{TX} corresponding to channel 11 when the RF oscillator

is in TX mode, the following approximations are used:

$$f^{TX\_mode(B)} \approx \frac{ticks_{RF\_oscillator}}{ticks\_reference} \times f_{reference}[\text{Hz}]$$
$$B^{TX} \approx B - \frac{f^{TX\_mode(B)} - 2405\ [\text{MHz}]}{\Delta F^{approx}} \tag{7.18}$$

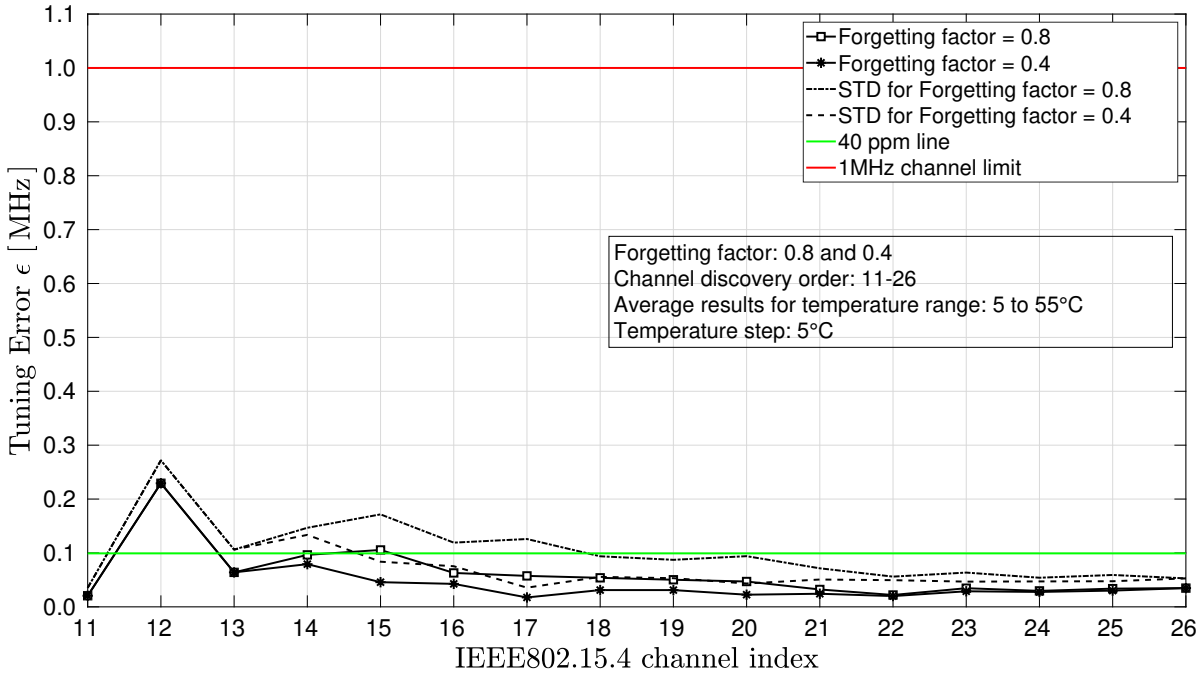
where $f^{TX\_mode(B)}$ is an approximation of the frequency output of the RF oscillator in TX mode when using tuning code B. The accuracy of this approximation is directly influenced by the accuracy of the on-chip reference used, $f_{reference}$. The tuning code B^{TX} to be used for attempting transmission on channel 11 is determined by adjusting the tuning code B so that $f^{TX\_mode(B)}$ matches the frequency of channel 11 (2405MHz).

7.5.2 Performance evaluation: simulations

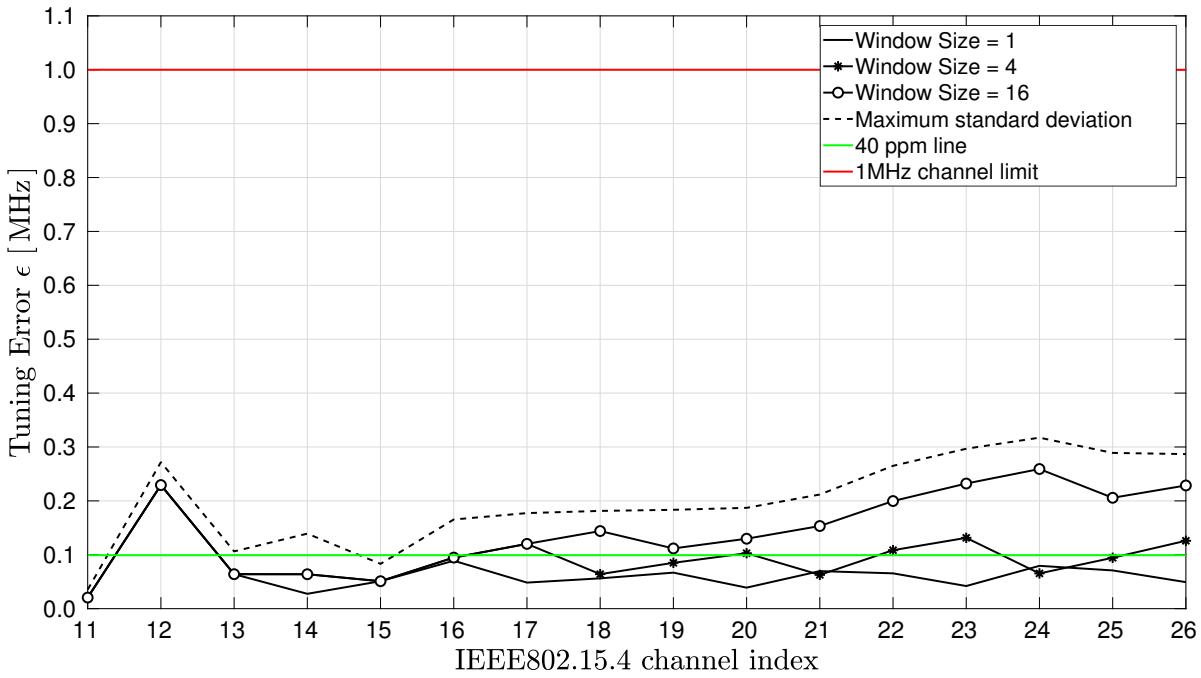
Fig. 7.15 shows the simulation results for the average frequency error when synthesizing the 16 IEEE802.15.4 channels with the RF oscillator in TX mode (Startup Phase). Even if in TX mode the RF oscillator is characterized by different function coefficients and a different frequency resolution of a tuning code, using ΔF^{approx} from the RX mode is a good startup approximation.

7.5.3 Experimental validation

While the results returned by the RLS and MA algorithms applied to simulation data are promising, with errors dropping below 40 ppm, when applying them on measurement data, the average frequency error is above the green 40 ppm line (Fig. 7.16). The average performance and the standard deviations obtained match the ones obtained for RX mode and detailed in Section 7.3.4.

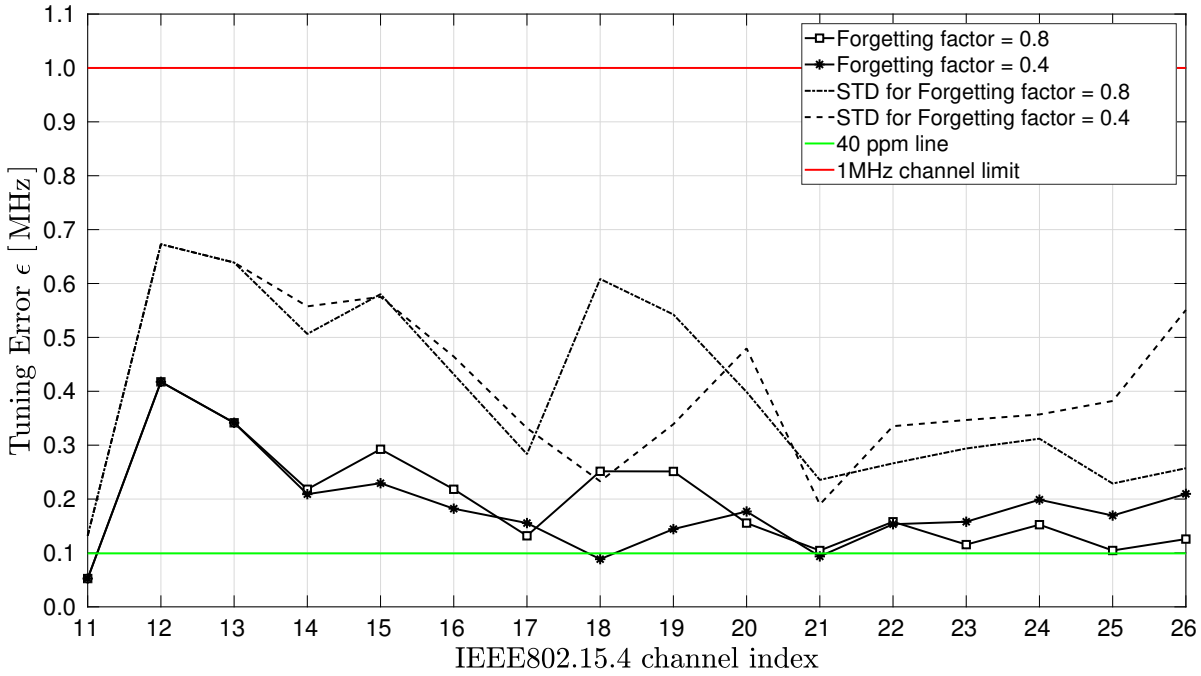


(a) RLS

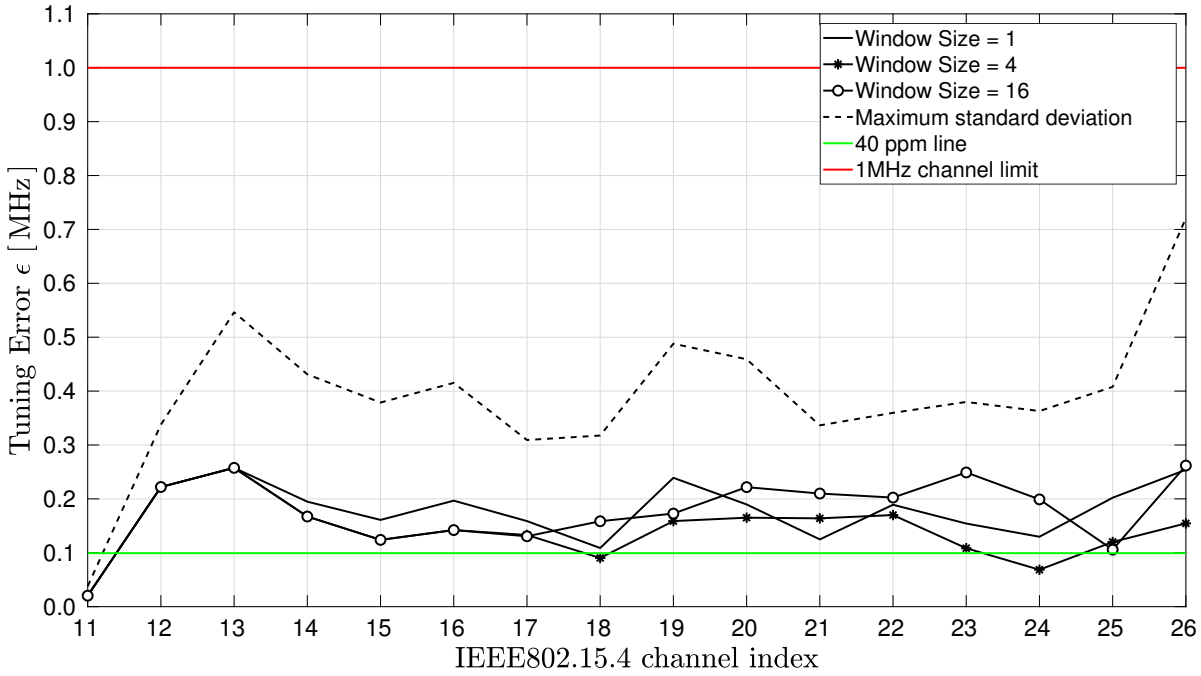


(b) MA

Fig. 7.15 Startup phase applied to RF oscillator in TX mode (simulation data): a) RLS algorithm evaluation for $\lambda = 0.8$ and $\lambda = 0.4$ and b) MA algorithm for window sizes of 1, 4 and 16. Performance averaged over startups at environmental temperature between $5 - 55$°C.



(a) RLS



(b) MA

Fig. 7.16 Startup phase applied to RF oscillator in TX mode (measurement data): a) RLS algorithm evaluation for $\lambda = 0.8$ and $\lambda = 0.4$ and b) MA algorithm for window sizes of 1, 4 and 16. Performance averaged over startups at environmental temperature between $5 - 55°$C.

7.6 Conclusions

This chapter addresses the challenge of RF local oscillator calibration on a single chip, crystal-free, IEEE802.15.4 device. It presents a full solution for compensating the inherent drift of the internal oscillators, and making it possible for the IEEE802.15.4 device to tune its radio at the specific channels, keeping the drift boundaries at the values imposed by the standard. The proposed method is divided in three phases: initial calibration, startup phase and normal operation phase. The on-chip oscillator settings for synthesizing the 16 IEEE802.15.4 communication channels are discovered at a constant temperature during the startup phase, for which we evaluate the use of RLS or MA-based algorithms. Once all the communication channels settings are determined and stored, the device switches to normal operation phase. This phase deals with maintaining the communication accuracy on all channels taking into account temperature variation and other inherent clock drifts.

The proposed algorithms are evaluated through both simulations and experimentation. While the obtained accuracy on simulation data satisfies the 40 ppm accuracy required by the IEEE802.15.4 standard, the results obtained on experimental data show higher errors, performing in average close to the 40 ppm limit. The higher errors are the consequence of irregularities in the tuning function across frequencies and increasing temperatures.

It is important to keep in mind that these frequency errors (obtained during startup phase or normal operation phase) impact the accuracy with which a channel will be synthesized for the first time at a certain temperature and will be immediately corrected to be below 40 ppm after the reception of a beacon (for IF-based frequency correction in RX mode) or of an ACK (for frequency correction in TX mode).

CHAPTER 8

Conclusions

The first part of this thesis explored energy efficiency techniques for wireless sensor networks that can be applied at frame level. We studied the use of preamble authentication to cope with energy exhaustion attacks in LoRaWAN networks. A short 4B authentication preamble incurs an energy consumption overhead of less than 4% in operational networks and reduces with 91% the energy exhaustion that a malicious attacker could cause to a network. These values were verified experimentally using Loadsensing sensor nodes, commercially used for critical infrastructure control and monitoring. This technique is extensible to any wireless protocol.

We analyzed the impact of packet fragmentation in LPWANs, following standardization directions defined by the LPWAN working group at the IETF. We focused on the use of fragmentation despite longer packets do fit in the frame, in order to identify the advantages and disadvantages of sending the data in smaller fragments when deploying duty-cycle restricted LPWAN networks. The results of our analysis show that packet fragmentation can increase the communication reliability for the case of duty-cycle restricted networks: important goodput improvement was obtained with fragmentation, with higher impact in denser and slower networks. This makes packet fragmentation a technique to be considered when deploying dense and low data rate industrial LPWAN networks.

Combining packet fragmentation with group NACK in duty cycle restricted LPWANs is a strategy that we show to provide increased network goodput and energy efficiency for dense networks. The retransmission policy is more efficient for smaller fragment sizes, where the probability of successful NACK request is higher. We showed that for small networks, it is better not to use packet fragmentation, but to use an Aloha-like protocol, which provides similar goodput at increased application capacity and energy efficiency. This is true also for IoT applications that only need to send packets of very small payloads. As the network size increases, the aggressive fragmentation strategy provides better network performance. The number of fragments/packet to be used could be dynamically adapted so as to provide the best network performance: goodput, application capacity or energy efficiency. The gateway could control the number of fragments/packet that the nodes use by issuing a MAC command.

In the context of enabling the use of crystal-free radios so as to further enhance the energy efficiency of IoT networks, the second part of this thesis analyzes the communication accuracy that can be obtained when eliminating the off-chip frequency references. For experimental evaluation, a

prototype implementation of a crystal-free radio incorporating a IEEE802.15.4 radio was used. Our analysis shows that while the frequency stability in time is good enough to meet the IEEE802.15.4 specification demands, the on-chip oscillators experience very significant drifts over temperature, that affect the communication performance. We presented mechanisms to dynamically compensate these drifts and we show that using the IEEE802.15.4 signaling, the communication accuracy can be maintained even as the temperature changes. The obtained accuracy meets the IEEE802.15.4 requirements of $\pm 40ppm$ frequency stability.

The final chapter of the thesis proposes three phases for enabling a very energy-efficient crystal-free radio to be part of an IEEE802.15.4 network: initial calibration, startup phase and normal operation phase. These phases define how a crystal-free radio can synchronize to a network, synthesize all communication channels at constant temperature and in the end, how to maintain communication accuracy on all channels as the temperature in the environment evolves. These phases include synchronization algorithms that are evaluated using MATLAB simulations and are experimentally evaluated on the SCuM platform. Even if we evaluate these algorithms on SCuM, they are generic and can be applied to any crystal-free radio. While the obtained accuracy on simulation data satisfies the 40 ppm accuracy required by the IEEE802.15.4 standard, the results obtained on experimental data show higher errors, performing in average close to the 40 ppm limit. The higher errors are the consequence of irregularities in the tuning function across frequencies and increasing temperatures. These results represent an important milestone in the way of obtaining very energy-efficient hardware for wireless networks, as we show that even if removing the off-chip frequency references is challenging, communication with an accuracy very close to meeting the standards is possible. By continuing this research, on-chip oscillators of even higher accuracy and stability can be obtained, paving the way to very cheap, dust-size, even disposable sensors that operate within the accuracy defined by standards.

Overall, this thesis has contributed to the development of greener communication technologies, suited for IoT applications. The thesis has explored different areas of a low power wireless system, considering the constraints imposed by current technologies and standards. It aims to improve the performance of low power wireless systems, without eliminating underlying assumptions, such as standardization limits or hardware features.

We have seen that current wireless sensor networks have to make a trade-off between various factors: energy efficiency, network performance, amount of data and costs. High energy efficiency and low costs usually translate into low amount of data or poor network performance. While many of the current energy efficiency approaches in the literature are defined by the limitations of existent technologies, in this thesis we identified that a change in technology could be beneficial. We showed that significant results can be achieved by re-thinking the technology in order to satisfy the demands for very low power consumption and cheap price, while being able to increase the amount of data sent by the sensor nodes. As stated in the second part of this thesis, a ten times more energy efficient chip can already be obtained with just removing the external frequency reference from the chip. In order to obtain significant gains, it is important to continue the research for

energy efficiency techniques, starting at chip level.

The trend for future wireless sensor networks corresponds to an intelligent and fully connected world: this means, a technology that is accessible to everyone and everywhere, very energy efficient and reliable. Step by step, complexity is removed from the sensor node to the gateway, and network server. The less complex the sensor nodes, the smaller, the cheaper and more energy efficient. Becoming more energy efficient could lead to eliminating the need for batteries, as more solutions for energy harvesting are found, enabling infinite lifetime for these devices. Becoming cheaper, allows for denser networks to be deployed, using less transmission power and higher frequency bands, non duty-cycle restricted, offering finer grained information. Becoming smaller, opens unlimited possibilities for medical applications, such as in-body sensors, or other industrial use cases for miniature devices.

BIBLIOGRAPHY

[1] "METIS: Mobile Communications for 2020 and beyond," https://metis2020.com/wp-content/uploads/publications/VDE_ITG_2013_Brahmi_Mobile_Communications-.pdf, [Online, accessed August 15, 2019].

[2] M. P. Millis, "The Cloud Begins with Coal: Big Data, Big Networks, Big Infrastructure, and Big Power," http://www.ict-21.ch/l4d/pg/file/read/878855/the-cloud-begins-with-coal-big-data-big-networks-big-infrastructure-and-big-power-an-overview-of-the-electricity-used-by-the-global-digital-ecosystem-mark-p-mills, [Online, accessed August 15, 2019].

[3] "5G PPP," https://5g-ppp.eu/, [Online, accessed June 5, 2019].

[4] Intel, "The 5G revolution," https://worldin2019.economist.com/transformbusinessesandtheworld, [Online, accessed September 6, 2019].

[5] S. Evanczuk, "Speed Development of Secure Cellular Connected IoT Applications," https://www.digikey.com/en/articles/techzone/2018/jun/speed-development-secure-cellular-connected-iot-applications, [Online, accessed June 5, 2019].

[6] D. Gislason, *Zigbee Wireless Networking*, pap/onl ed. Newton, MA, USA: Newnes, 2008.

[7] N. Erasala and D. C. Yen, "Bluetooth technology: A strategic analysis of its role in global 3g wireless communication era," *Comput. Stand. Interfaces*, vol. 24, no. 3, pp. 193–206, Jul. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0920-5489(02)00018-1

[8] P. Thubert, T. Watteyne, R. Struik, and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4," *Working Draft, IETF Secretariat, Internet-Draft draft-ietf-6tisch-architecture-08*, 2015.

[9] A. Al-Alawi, "Wifi technology: Future market challenges and opportunities," *Journal of Computer Science*, vol. 2, 01 2006.

[10] LoRa Alliance, "LoRaWAN Specification," 2017.

[11] "Sigfox Technology Overview," https://www.sigfox.com/en/sigfox-iot-technology-overview, [Online; accessed 15-July-2018].

[12] "Weightless-P System Specification," version 1.0.

[13] Ingenu, “The making of RPMA,” [ebook].

[14] “ETSI EN 300 220-2 V3.1.1 ,” 2007.

[15] “Loadsensing,” https://www.worldsensing.com/product/loadsensing/, [Online; accessed 16-July-2018].

[16] S. Mesri, “Design and user guide for the single chip mote digital system,” Master’s thesis, EECS Department, University of California, Berkeley, May 2016.

[17] F. Maksimovic, B. Wheeler, D. Burnett, O. Khan, S. Mesri, I. Suciu, L. Lee, A. Moreno, A. Sundararajan, B. Zhou, R. Zoll, A. Ng, T. Chang, X. Vilajosana, T. Watteyne, A. Niknejad, and K. Pister, “A Crystal-Free Single-Chip Micro Mote with Integrated 802.15.4 Compatible Transceiver, Sub-mW BLE Compatible Beacon Transmitter, and Cortex M0,” in *2019 Symposium on VLSI Circuits*, 2019.

[18] “Worldsensing,” https://www.worldsensing.com/, [Online; accessed 16-July-2018].

[19] S. W. . Sensing, “SX1272/3/6/7/8 : LoRa Modem Designers Guide-AN1200.13,” 2013, [Online; accessed 31-October-2017].

[20] I. Suciu, F. Maksimovic, B. Wheeler, D. C. Burnett, O. Khan, T. Watteyne, X. Vilajosana, and K. S. J. Pister, “Dynamic channel calibration on a crystal-free mote-on-a-chip,” *IEEE Access*, vol. 7, pp. 120 884–120 900, 2019.

[21] Dust Networks, “LTC5800-IPM:SmartMesh IP Node 2.4GHz, 802.15.4e Wireless Mote-on-Chip,” https://www.analog.com/media/en/technical-documentation/data-sheets/5800ipmfa.pdf, [Online, accessed April 3, 2019].

[22] “Energy Effciency of the Internet of Things,” https://www.iea-4e.org/document/388/energy-efficiency-of-the-internet-of-things-policy-options, policy Options Prepared for IEA 4E EDNA JULY 2016. [Online; accessed 1-July-2019].

[23] T. Rault, A. Bouabdallah, and Y. Challal, “Energy efficiency in wireless sensor networks: a top-down survey,” *Computer Networks*, vol. 67, p. 104122, 07 2014.

[24] Z. Abbas and W. Yoon, “A survey on energy conserving mechanisms for the internet of things: Wireless networking aspects,” *Sensors*, vol. 15, pp. 24 818–24 847, 09 2015.

[25] S. S. Sebastian, “Energy efficiency in internet of things: An overview,” *International Journal of Recent Trends in Engineering and Research (IJRTER)*, vol. 02, pp. 475–482, 06 2016.

[26] H. Bello, Z. Xiaoping, R. Nordin, and J. Xin, “Advances and opportunities in passive wake-up radios with wireless energy harvesting for the internet of things applications,” *Sensors*, vol. 19, no. 14, p. 3078, Jul 2019. [Online]. Available: http://dx.doi.org/10.3390/s19143078

[27] R. Piyare, A. Murphy, M. Magno, and L. Benini, "On-demand lora: Asynchronous tdma for energy efficient and low latency communication in iot," *Sensors*, vol. 18, p. 3718, 11 2018.

[28] P. P. Wang, H. Jiang, L. Gao, P. Sen, Y. Kim, G. M. Rebeiz, P. P. Mercier, and D. A. Hall, "A 6.1-nw wake-up receiver achieving 80.5-dbm sensitivity via a passive pseudo-balun envelope detector," *IEEE Solid-State Circuits Letters*, vol. 1, no. 5, pp. 134–137, May 2018.

[29] R. Piyare, Amy.L.Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Communications Surveys and Tutorials*, vol. PP, 06 2017.

[30] J. Blobel, J. Krasemann, and F. Dressler, "An architecture for sender-based addressing for selective sensor network wake-up receivers," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–7.

[31] B. Zebbane, M. Chenait, and N. Badache, "A distributed lightweight redundancy aware topology control protocol for wireless sensor networks." *Wireless Networks*, vol. 23, pp. 1779–1792, 07 2017.

[32] G. Jaber, R. Kacimi, and Z. Mammeri, "Exploiting redundancy for energy-efficiency in wireless sensor networks," in *2016 9th IFIP Wireless and Mobile Networking Conference (WMNC)*, July 2016, pp. 180–186.

[33] S. Al-Omari and W. Shi, "Redundancy-aware topology control in wireless sensor networks," 08 2009.

[34] D. Sharma, K. Kavitha, and R.Gururaj, "Estimating node density for redundant sensors in wireless sensor network," *International Journal of Sensor Networks and Data Communications*, vol. 4, 11 2015.

[35] X. Fan, W. Wei, M. Wozniak, and Y. Li, "Low energy consumption and data redundancy approach of wireless sensor networks with bigdata," *Information Technology And Control*, vol. 47, 09 2018.

[36] H. Zayed, M. Taha, and A. H Allam, "Energy efficient routing in wireless sensor networks: A survey," 01 2018.

[37] X. Yang, "The application of rpl routing protocol in low power wireless sensor and lossy networks," *Sensors and Transducers*, vol. 170, pp. 107–111, 05 2014.

[38] M. N. Jambli, M. I. Bandan, K. S. Pillay, and S. M. Suhaili, "An analytical study of leach routing protocol for wireless sensor network," in *2018 IEEE Conference on Wireless Sensors (ICWiSe)*, Nov 2018, pp. 44–49.

[39] P. Kamgueu, E. Nataf, T. Djotio, and O. Festor, "Energy-based routing metric for rpl," 01 2013.

[40] O. Iova, F. Theoleyre, and T. Noel, "Improving the network lifetime with energy-balancing routing: Application to rpl," in *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, May 2014, pp. 1–8.

[41] Y. Lu and V. Wong, "An energyefficient multipath routing protocol for wireless sensor networks," *International Journal of Communication Systems*, vol. 20, pp. 747 – 766, 07 2007.

[42] Y. Arora and H. Pande, "Energy saving multipath routing protocol for wireless sensor networks," 2013.

[43] D. Raj and P. Sumathi, "Enhanced energy efficient multipath routing protocol for wireless sensor communication networks using cuckoo search algorithm," *Wireless Sensor Network*, vol. 06, pp. 49–55, 01 2014.

[44] B. Risteska Stojkoska and Z. Nikolovski, "Data compression for energy efficient iot solutions," 11 2017.

[45] A. Orsino, G. Araniti, L. Militano, J. Alonso-Zarate, A. Molinaro, and A. Iera, "Energy efficient iot data collection in smart cities exploiting d2d communications," *Sensors*, vol. 16, 06 2016.

[46] S. Randhawa and S. Jain, "Data aggregation in wireless sensor networks: Previous research, current status and future directions," *Wireless Personal Communications*, vol. 97, pp. 1–71, 07 2017.

[47] A. V. Sisal and S. R. Khiani, "Data aggregation techniques in wireless sensor network: Survey," 2015.

[48] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, pp. 335 – 344, 03 2010.

[49] B. Martinez, X. Vilajosana, I. Vilajosana, and M. Dohler, "Lean sensing: Exploiting contextual information for most energy-efficient sensing," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1156–1165, Oct 2015.

[50] P. Ostovari, J. Wu, and A. Khreishah, *Network Coding Techniques for Wireless and Sensor Networks*, 12 2014.

[51] G. Peralta, R. G. Cid-Fuentes, J. Bilbao, and P. Crespo, *Network Coding-Based Next-Generation IoT for Industry 4.0*, 08 2018.

[52] C. Han, Y. Yang, and X. Han, "A fast network coding scheme for mobile wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 2, p. 1550147717693241, 2017. [Online]. Available: https://doi.org/10.1177/1550147717693241

[53] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *Communications Letters, IEEE*, vol. 12, pp. 411 – 413, 07 2008.

[54] M. Mahmoud and A. Mohamad, "A study of efficient power consumption wireless communication techniques/ modules for internet of things (iot) applications," *Advances in Internet of Things*, vol. 06, pp. 19–29, 01 2016.

[55] D. Griffith, P. T. RŁine, T. Kallerud, B. Goodlin, Z. Hughes, and E. T. Yen, "A 10ppm 40 to 125c baw-based frequency reference system for crystal-less wireless sensor nodes," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[56] C. Lam, "A review of the recent development of MEMS and crystal oscillators and their impacts on the frequency control products industry," *2008 IEEE Ultrasonics Symposium*, pp. 694–704, 2008.

[57] B. Wheeler, F. Maksimovic, N. Baniasadi, S. Mesri, O. Khan, D. Burnett, A. Niknejad, and K. Pister, "Crystal-free narrow-band radios for low-cost IoT," in *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, June 2017, pp. 228–231.

[58] O. Khan, D. Burnett, F. Maksimovic, B. Wheeler, S. Mesri, A. Sundararajan, B. L. Zhou, A. Niknejad, and K. Pister, "Time keeping ability of crystal free radios," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[59] Song Li (Microchip Technology), "Revolutionary Timing for Auto-Qualified MEMS Oscillators," https://www.sensorsmag.com/components/revolutionary-timing-for-auto-qualified-mems-oscillators, [Online, accessed April 2, 2019].

[60] Epson, "Comparison of Crystal Oscillator and Si-MEMS Oscillators," https://www5.epsondevice.com/en/information/technical_info/pdf/wp_e20140911_osc.pdf, [Online, accessed April 2, 2019].

[61] Paul Pickering (Mouser Electronics), "MEMS Oscillators Make Inroads," https://eu.mouser.com/applications/timer-mems-oscillators/, [Online, accessed April 3, 2019].

[62] SiTime, "MEMS Oscillators: Enabling Smaller, Lower Power IoTand Wearables," https://www.sitime.com/sites/default/files/_legacy/images/stories/applications/SiTimes-MEMS-Oscillators-Lower-Power--Reduce-Size-2016.pdf, [Online, accessed April 2, 2019].

[63] "Texas Instruments," http://www.ti.com/, [Online, accessed April 5, 2019].

[64] "Texas Instruments: CC2652RB (PREVIEW) SimpleLink crystal-less BAW wireless MCU ," http://www.ti.com/product/CC2652RB, [Online, accessed April 5, 2019].

[65] Baker Lawley (All About Circuits), "The End of the Crystal? TI Introduces Two New Products Using Breakthrough BAW Resonator Technology," https://www.allaboutcircuits.com/news/ti-introduces-two-new-products-using-breakthrough-baw-resonator-technology/, [Online, accessed April 5, 2019].

[66] Y. Shih and B. Otis, "An on-chip tunable frequency generator for crystal-less low-power wban radio," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 4, pp. 187–191, April 2013.

[67] X. Zhang, I. Mukhopadhyay, R. Dokania, and A. B. Apsel, "A 46-μWself-calibrated gigahertz vco for low-power radios," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 12, pp. 847–851, Dec 2011.

[68] D. Burnett, B. Wheeler, L. Lee, F. Maksimovic, A. Sundararajan, O. Khan, and K. Pister, "CMOS oscillators to satisfy 802.15.4 and bluetooth LE PHY specifications without a crystal reference," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (IEEE CCWC 2019)*, Las Vegas NV, USA, Jan. 2019.

[69] A. Paidimarri, N. Ickes, and A. P. Chandrakasan, "A 0.68V 0.68mW 2.4GHz PLL for ultra-low power RF systems," in *2015 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, May 2015, pp. 397–400.

[70] I. Suciu, F. Maksimovic, D. Burnett, O. Khan, B. Wheeler, A. Sundararajan, T. Watteyne, X. Vilajosana, and K. Pister, "Experimental clock calibration on a Crystal-Free Mote-on-a-Chip," in *2019 IEEE INFOCOM WKSHPS: CNERT 2019: Computer and Networking Experimental Research using Testbeds (INFOCOM 2019 WKSHPS - CNERT 2019)*, Paris, France, Apr. 2019.

[71] A. Y. Jou, H. Shan, H. Pajouhi, J. Peterson, and S. Mohammadi, "A Single-Chip Wireless Powered RFID Antenna and Transceiver," *IEEE Journal of Radio Frequency Identification*, vol. 1, no. 3, pp. 219–227, Sep. 2017.

[72] H. Bhamra, Y. Kim, J. Joseph, J. Lynch, O. Z. Gall, H. Mei, C. Meng, J. Tsai, and P. Irazoqui, "A24 μW, Batteryless, Crystal-free, Multinode Synchronized SoC "Bionode" for Wireless Prosthesis Control," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 11, pp. 2714–2727, Nov 2015.

[73] M. Tabesh, N. Dolatsha, A. Arbabian, and A. M. Niknejad, "A Power-Harvesting Pad-Less Millimeter-Sized Radio," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 962–977, April 2015.

[74] L. Chuo, Y. Shi, Z. Luo, N. Chiotellis, Z. Foo, G. Kim, Y. Kim, A. Grbic, D. Wentzloff, H. Kim, and D. Blaauw, "A 915MHz asymmetric radio using Q-enhanced amplifier for a

fully integrated $3 \times 3 \times 3mm^2$ wireless sensor node with 20m non-line-of-sight communication," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2017, pp. 132–133.

[75] D. Shaviv, A. Ozgur, and A. Arbabian, "Communication With Crystal-Free Radios," *IEEE Transactions on Communications*, vol. 66, no. 10, pp. 4513–4520, Oct 2018.

[76] K. S. Deepak and A. V. Babu, "Energy consumption analysis of modulation schemes in ieee 802.15.6-based wireless body area networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, 12 2016.

[77] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and A. Guillaume, "Energy consumption model for sensor nodes based on lora and lorawan," *Sensors*, vol. 18, p. 2104, 06 2018.

[78] Semtech, "What is LoRa?" http://www.semtech.com/wireless-rf/internet-of-things/what-is-lora/, [Online; accessed 31-October-2017].

[79] K.-H. Hsia, C.-W. Hung, H. T. Chang, and Y.-H. Lai, "Transmission power control for wireless sensor network," *Journal of Robotics, Networking and Artificial Life*, vol. 3, no. 4, pp. 279–282, 2017.

[80] Yong Fu, Mo Sha, G. Hackmann, and C. Lu, "Practical control of transmission power for wireless sensor networks," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*, Oct 2012, pp. 1–10.

[81] X. Chu and H. Sethu, "Cooperative topology control with adaptation for improved lifetime in wireless sensor networks," *Ad Hoc Networks*, vol. 30, 09 2013.

[82] N. Kumari, R. Kumar, and R. Bajaj, "Energy efficient communication using reconfigurable directional antenna in manet," *Procedia Computer Science*, vol. 125, pp. 194–200, 01 2018.

[83] Q. Wang, H.-N. Dai, Z. Zheng, M. Imran, and A. Vasilakos, "On connectivity of wireless sensor networks with directional antennas," *Sensors*, vol. 17, p. 134, 01 2017.

[84] R. Choudhury and N. Vaidya, "Deafness: a mac problem in ad hoc networks when using directional antennas," 11 2004, pp. 283– 292.

[85] G. Tarter, L. Mottola, and G. P. Picco, "Directional antennas for convergecast in wireless sensor networks: Are they a good idea?" in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2016, pp. 172–182.

[86] M. Shirvanimoghaddam, K. Shirvanimoghaddam, M. M. Abolhasani, M. Farhangi, V. Zahiri Barsari, H. Liu, M. Dohler, and M. Naebe, "Paving the path to a green and self-powered internet of things," 12 2017.

[87] B. Martinez, M. Montn, I. Vilajosana, and X. Vilajosana, "Early scavenger dimensioning in wireless industrial monitoring applications," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 170–178, April 2016.

[88] B. Martinez, X. Vilajosana, F. Chraim, I. Vilajosana, and K. S. J. Pister, "When scavengers meet industrial wireless," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2994–3003, May 2015.

[89] Q. Liu, J. Wu, P. Xia, S. Zhao, W. Chen, Y. Yang, and L. Hanzo, "Charging unplugged: Will distributed laser charging for mobile wireless power transfer work?" *IEEE Vehicular Technology Magazine*, vol. 11, no. 4, pp. 36–45, Dec 2016.

[90] M. Liu, M. Xiong, H. Deng, Q. Liu, J. Wu, and P. Xia, "Mobile energy internet," 02 2018.

[91] W.-Y. Lai and T.-R. Hsiang, "Wireless charging deployment in sensor networks," *Sensors*, vol. 19, p. 201, 01 2019.

[92] "Weightless," http://www.weightless.org/, [Online; accessed 15-July-2018].

[93] "Most common attack vector over Critical Infrastructures," http://www.cipsec.eu/content/most-common-attack-vector-over-critical-infrastructures, [Online; accessed 15-July-2018].

[94] Emekcan Aras , Nicolas Small , Gowri Sankar Ramachandran , Stphane Delbruel , Wouter Joosen and Danny Hughes, "Selective Jamming of LoRaWAN using Commodity Hardware," 2017. arXiv:1712.02141v1 [cs.NI].

[95] "Energy and information sabotage: The threats facing our smart cities," https://www.zdnet.com/article/energy-and-information-sabotage-the-threats-facing-our-smart-cities/, [Online; accessed 15-July-2018].

[96] S. Tomasin, S. Zulian, and L. Vangelista, "Security analysis of lorawan join procedure for internet of things networks," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, March 2017, pp. 1–6.

[97] Bhupjit Singh and Bipjeet Kaur, "Comparative study of Internet of Things infrastructure and security," abstract from Global Wireless Submit 2016, Aarhus, Denmark.

[98] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[99] Orestis Georgiou and Usman Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" 2017. arXiv:iv:1610.0479 [cs.NI].

[100] "AN1200.22, LoRa Modulation Basics," https://www.semtech.com/uploads/documents/an1200.22.pdf, [Online; accessed 15-July-2018].

[101] Kamil Staniec and Micha Kowal, "LoRa Performance under Variable Interference and Heavy-Multipath Conditions," Wireless Communications and Mobile Computing, vol. 2018, Article ID 6931083, 9 pages, 2018.

[102] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," in *Advances in Cryptology - CRYPTO 2003*, D. Boneh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 617–630.

[103] "PowerScale with ACM technology," https://www.hitex.com/tools-components/test-tools/analyzer/energy-optimization/powerscale/, [Online; accessed 16-July-2018].

[104] A. Minaburo, L. Toutain, and C. Gomez, "LPWAN Static Context Header Compression (SCHC) and fragmentation for IPv6 and UDP," Internet Engineering Task Force, Internet-Draft draft-ietf-lpwan-ipv6-static-context-hc-07, Oct. 2017, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-lpwan-ipv6-static-context-hc-07

[105] F. Adelantado, X. Vilajosana, P. Tuset-Peiró, B. Martínez, J. Melià-Seguí, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017. [Online]. Available: https://doi.org/10.1109/MCOM.2017.1600613

[106] C. Gomez, J. Paradells, and J. Crowcroft, "Optimized 6lowpan fragmentation header for lpwan," Working Draft, IETF Secretariat, Internet-Draft draft-gomez-lpwan-fragmentation-header-00, March 2016, http://www.ietf.org/internet-drafts/draft-gomez-lpwan-fragmentation-header-00.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-gomez-lpwan-fragmentation-header-00.txt

[107] G. Montenegro, J. Hui, D. Culler, and N. Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, Sep. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4944.txt

[108] P. Thubert and J. Hui, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282, Sep. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6282.txt

[109] "IPv6 over Low Power Wide-Area Networks." [Online]. Available: https://datatracker.ietf.org/doc/charter-ietf-lpwan/

[110] J. Yoon, H. Kim, and J. Ko, "Data fragmentation scheme in IEEE 802.15.4 wireless sensor networks," in *Proceedings of the 65th IEEE Vehicular Technology Conference, VTC Spring 2007, 22-25 April 2007, Dublin, Ireland.* IEEE, 2007, pp. 26–30. [Online]. Available: https://doi.org/10.1109/VETECS.2007.18

[111] Y. T. Park, P. Sthapit, and J.-Y. Pyun, "Energy efficient data fragmentation for ubiquitous computing," *The Computer Journal*, vol. 57, no. 2, pp. 263–272, 2014. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxt080

[112] Y. Sankarasubramaniam, I. F. Akyildiz, and S. W. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, 2003, pp. 1–8.

[113] M. DENER, "Optimum packet length over data transmission for wireless sensor networks," 2013.

[114] O. W. W. Y. Rui Lint and S. A. Mahmoudt, "Packet length optimizations in meteor burst communication systems," 1991.

[115] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range and low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/9/1466

[116] "The Internet of Everything," http://www.neul.com, [Online; accessed 18-December-2017].

[117] "RPMA Technology," https://www.ingenu.com/technology/rpma/, [Online; accessed 18-December-2017].

[118] "3GPP Low Power Wide Area Technologies," https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf, [Online; accessed 18-December-2017].

[119] Ioana Suciu, Xavier Vilajosana, Ferran Adelantado, "An Analysis of Packet Fragmentation Impact in LPWAN," 2017. arXiv:1712.06878 [cs.NI].

[120] Alexandru-Ioan Pop, Usman Raza, Parag Kulkarni, Mahesh Sooriyabandara, "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?" 2017. arXiv:1704.04174 [cs.NI].

[121] N. Salman, I. Rasool, and A. H. Kemp, "Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks," in *Symposium on Wireless Communication Systems*, Sep. 2010, pp. 701–705.

[122] Freescale Semiconductor, Inc., "Reference oscillator crystal requirements for mkw40 and mkw30 device series," https://www.nxp.com/docs/en/application-note/AN5177.pdf, august, 2015, [Online, accessed November 21, 2018].

[123] Thomas Watteyne, Branko Kerkez, Kris Pister, Steven Glaser, "Crystal-free network synchronization," trans. Emerging Tel. Tech. 2016.

[124] X. Vilajosana, P. Tuset-Peiro, T. Watteyne, and K. Pister, "Openmote: Open-source prototyping platform for the industrial iot," 09 2015.

[125] A. M. Mehta and K. S. J. Pister, "Frequency offset compensation for crystal-free 802.15.4 communication," in *The 2011 International Conference on Advanced Technologies for Communications (ATC 2011)*, Aug 2011, pp. 45–47.

[126] O. Khan, B. Wheeler, D. Burnett, F. Maksimovic, S. Mesri, K. Pister, and A. Niknejad, "Frequency reference for crystal free radio," in *2016 IEEE International Frequency Control Symposium (IFCS)*, May 2016, pp. 1–2.

[127] Texas Instruments, "CC2538 powerful wireless microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee applications datasheed (Rev. D)."

[128] H. Foundation, "IEC 62591:2016 Industrial networks - Wireless communication network and communication profiles - WirelessHART," HART Foundation, Tech. Rep., 2016.

[129] X. Vilajosana, T. Watteyne, M. Vucinic, T.Chang, and K. Pister, "6TiSCH: Industrial Performance for IPv6 Internet of Things Networks," *Special issue Real-Time Networks and Protocols for Factory Automation and Process Control Systems. IEEE Proceedings Journal*, 2019.

[130] M. Vučinić, J. Simon, K. Pister, and M. Richardson, "Minimal Security Framework for 6TiSCH," Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-minimal-security-10, Apr. 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-minimal-security-10

[131] "LC Oscillator Basics," https://www.electronics-tutorials.ws/oscillator/oscillators.html, [Online, accessed May 20, 2019].

[132] H. Gavin, "Fitting Models to Data, Generalized Linear Least Squares, and Error Analysis," *CEE 629. System Identification, Duke University, Spring 2019*, 2019.

[133] MIT, "Introduction to Recursive-Least-Squares (RLS) Adaptive Filters," *Massachusetts Institute of Technology, Department of Mechanical Engineering, 2.161 Signal Processing - Continuous and Discrete*, 2008.