UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Software-defined and multi-user LPWAN radios: towards preventing the obsolescence of IoT devices

Mathieu Xhonneux

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Prof. David Bol	*Advisor*	UCLouvain, Belgium
Prof. Jérôme Louveaux	*Advisor*	UCLouvain, Belgium
Prof. Laurent Francis	*Chair*	UCLouvain, Belgium
Prof. Jean-Didier Legat	*Secretary*	UCLouvain, Belgium
Dr. Carolynn Bernier		CEA, France
Prof. Andreas Burg		EPFL, Switzerland
Prof. Liesbet Van der Perre		KULeuven, Belgium

Contents

II Multi-User Receivers for LoRa Gateways

Abstract

With the exponential deployment of Internet-of-Things (IoT) connected devices, massive networks relying on first-generation low-power wide-area network (LPWAN) protocols begin to reach their limits. Although more scalable LPWAN protocols are expected to be commercialized, LPWAN radios are typically implemented in hardware and tied to a single protocol. Upgrading an IoT network to a newer protocol hence requires the replacement of functional devices, which implies a high environmental cost. We study in this thesis two approaches to prevent the disposal of existing IoT devices that could arise from technological obsolescence.

The first part of this thesis tackles the challenge of designing LPWAN software-defined radios (SDRs) on ultra low-power microcontrollers. In particular, we target SDR implementations of the LoRa and Sigfox protocols. We initially design, thanks to an accurate analytical model of unsynchronized LoRa signals, a Nyquist-rate LoRa receiver chain with a low-complexity iterative synchronization procedure. We then conduct a hardware/software co-design of a microcontroller architecture for LPWAN SDRs on IoT devices. This architecture includes a general-purpose low-power processor for the protocol-specific baseband computations and a hardware digital front-end for the generic signal processing. The LoRa and Sigfox SDRs are evaluated with an ultra low-power prototype of the microcontroller architecture and attain a sub-mW power consumption for the baseband processing.

The second part of this thesis studies the design of multi-user receivers for LoRa gateways. Since LoRa devices are not synchronized in time, collisions between uplink frames limit the scalability of LoRaWAN networks. As a first attempt, we derive from the maximum-likelihood criterion a two-user detector capable of decoding two colliding frames. We demonstrate the practicality of this approach thanks to an SDR implementation of the proposed detector, along with an interference-robust synchronization algorithm. To overcome the intrinsic limitations of this first detector, we then design a successive interference cancellation two-user receiver that explicitly leverages the interleaving and channel code of LoRa with soft decisions. Network-level simulations show that a LoRa gateway employing our two-user receiver may serve 4.7 times more devices than a conventional gateway.

Author's publication list

Related journal papers

JP1. M. Xhonneux, O. Afisiadis, D. Bol, and J. Louveaux, "A Low-Complexity LoRa Synchronization Algorithm Robust to Sampling Time Offsets," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3756–3769, Mar. 2022.

JP2. M. Xhonneux, J. Louveaux, and D. Bol, "A sub-mW Cortex-M4 Microcontroller Design for IoT Software-Defined Radios," submitted to *IEEE Transactions on Computers*.

JP3. M. Xhonneux, J. Tapparel, A. Balatsoukas-Stimming, A. Burg, and O. Afisiadis, "A Maximum-Likelihood-based Two-User Receiver for LoRa Chirp Spread-Spectrum Modulation," *IEEE Internet of Things Journal*, early access, Jun. 2022.

JP4. J. Tapparel, M. Xhonneux, D. Bol, J. Louveaux, and A. Burg, "Enhancing the Reliability of Dense LoRaWAN Networks With Multi-User Receivers," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2725–2738, Dec. 2021.

Related conference papers

CP1. M. Xhonneux, J. Louveaux, and D. Bol, "Implementing a LoRa software-defined radio on a general-purpose ULP microcontroller," in *Proc. 2021 IEEE Workshop on Signal Processing Systems*, 2021, pp. 105–110.

CP2. R. Dekimpe, M. Schramme, M. Lefebvre, A. Kneip, R. Saeidi, M. Xhonneux, L. Moreau, M. Gonzalez, T. Pirson, and D. Bol, "SleepRider: a 5.5μW/MHz Cortex-M4 MCU in 28nm FD-SOI with ULP SRAM, Biomedical AFE and Fully-Integrated Power, Clock and Back-Bias Management," in *Proc. 2021 IEEE Symposium on VLSI Circuits*, 2021.

CP3. M. Xhonneux, J. Tapparel, O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A Maximum-Likelihood-based Multi-User LoRa Receiver Implemented in GNU Radio," in *Proc. 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 1106–1111.

CP4. M. Xhonneux, J. Tapparel, P. Scheepers, O. Afisiadis, A. Balatsoukas-Stimming, D. Bol, J. Louveaux, and A. Burg, "A Two-User Successive Interference Cancellation LoRa Receiver with Soft-Decoding," in *Proc. 55th Asilomar Conference on Signals, Systems, and Computers*, 2021.

Unrelated conference papers

UCP1. M. Xhonneux, F. Duchene, and O. Bonaventure, "Leveraging eBPF for programmable network functions with IPv6 segment routing," in *Proc. of the 14th International Conference on emerging Networking EXperiments and Technologies*, 2018, pp. 67–72.

UCP2. M. Xhonneux, and O. Bonaventure, "Flexible failure detection and fast reroute using eBPF and SRv6," in *Proc. 14th International Conference on Network and Service Management*, 2018, pp. 408–413.

Acknowledgments

Beginning a PhD thesis is much like embarking on a journey towards a known destination, but whose roads and paths are still uncharted. Although this journey requires a fair amount of efforts, perseverance and faith, it certainly is impossible to achieve without the help of a devoted and thoughtful team. In this sense, I feel extremely thankful for the support I received over these four years.

First of all, I heartfully want to thanks David and Jérôme for steering me all along the way. Beside providing me the opportunity to start this project, I am also very grateful to my advisors for always lending me an attentive ear, bringing ideas to the table, and encouraging me whenever needed. I would also like to thank my jury members, whose comments and ideas helped to improve this manuscript.

I am also particularly grateful to my Swiss colleagues for the very enriching collaboration we had over the past two years. I am foremost thankful to Andy for hosting me several times at EPFL and supervising me on his insightful research projects. I also want to thank Orion and Alex for their numerous advices, which undoubtedly helped me to become a better researcher. Last but not least, thanks to Joachim for being an always amusing and enthusiastic colleague, even during our late and tedious debugging sessions!

Of course, these four years would not have been so fun and instructive without all the UCLouvain colleagues. Thanks to the entire ECS group (Adrian, Charlotte, Julien, Louis, Ludovic, Marco, Martin, Maxime, Nicolas, Pengcheng, Pol, Rémi, Roghayeh, Thibault) for all the great time we spent together, let it be at a coffe break, in the rush of the SleepRider tapeout or during social events. In addition, my daily life at UCLouvain would certainly not have been so pleasant if it were not for the always cheer-

ful technical and administrative staff of the electrical department: many thanks to Anne, Benoit, Brigitte, Isabelle, Pascal, Souley, Valeriya and Viviane for always kindly lending a hand. On a special note, I also feel particularly grateful to Olivier Bonaventure, whose attentive guidance during my studies gave me the taste for research and the impulse to start this thesis.

Finally, I want to thank all my friends, family members, and particularly Ysaline, for their unconditional support throughout this journey. You all rock my world, thanks!

1

Introduction

The last decade witnessed the ever-increasing digitalization of tools and infrastructure in our daily lives. From smart wearables for healthcare to intelligent electrical meters, the Internet of Things (IoT) aims to revolutionize both consumer and industrial sectors by integrating Internet-connected sensors to physical objects [1, 2]. These sensors, usually referred to as *end nodes*, generate data from their environment and transfer it wirelessly over the Internet to the cloud. This data is then analyzed by an application to extract insights and actionable information to the final user.

At the core of the IoT paradigm lies the wireless connectivity of the sensors to the Internet. Compared to traditional connected devices such as laptops or smartphones, many IoT end nodes rely on a much more stringent energy supply (e.g., small batteries or an energy harvester). To avoid a too frequent and cumbersome maintenance of the nodes, users expect a long autonomy of several years. Yet, the wireless transmissions represent an important part of the energy budget of the devices [3]. A major trade-off hence opposes the autonomy of the end nodes and the quantity of data they transmit.

In particular, the prohibitive energy consumption of conventional 3GPP cellular radios has led to the emergence of new low-power wide-area network (LPWAN) technologies [5, 6]. As illustrated in Fig. 1.1, LPWANs aim to connect IoT devices to gateways over large distances, while minimizing for the end nodes the energy overhead of the transmissions. The gate-

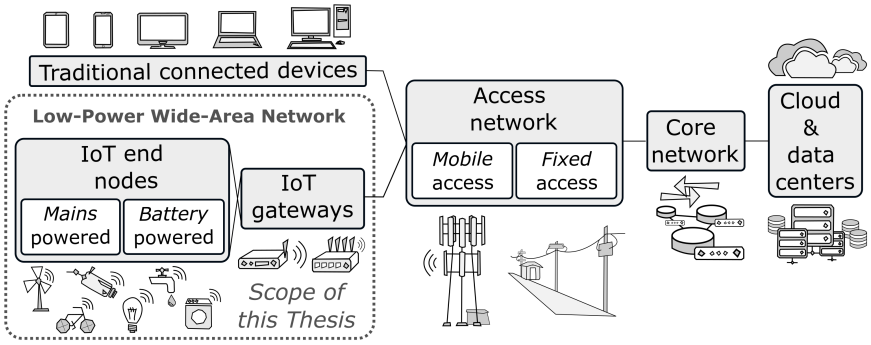


Fig. 1.1 Illustration of the network architecture of IoT applications, adapted from [4]. This thesis focuses on the LPWAN subsystem.

way is connected to the Internet through an access network, and relays the messages from the end nodes to an application back-end hosted in a data center. The wireless communications between the end nodes and the gateway are defined by a protocol, which comprises a physical (PHY) layer and a medium access control (MAC) layer. Since end nodes are used across a large variety of use-cases, no single wireless technology matches the broad range of requirements of all IoT applications [7]. Several LPWAN protocols hence co-exist together, as each favors a different set of characteristics (e.g., payload size, communication range, energy efficiency, cost, ...).

The recent popularity of IoT applications has driven an exponential increase of the number of deployed end nodes [4]. Even though these applications provide potential benefits to the final users, their infrastructure also causes inevitable environmental burdens [8]. For instance, an end node typically embeds several electronic circuits and components (e.g., sensor, radio, battery, ...). As for every electronic product, the life cycle of an IoT end node implies the extraction and refining of metals, its manufacturing in energy-intensive cleanrooms, the transport to the customer, the usage of the device itself and finally its end-of-life treatment [9, 10]. These stages of the life cycle all generate direct environmental impacts such as the release of greenhouse gases (GHG) into the atmosphere, water and soil pollution, and many other ecosystem damages.

As of today, the environmental burdens generated by IoT end nodes are usually overlooked, since it is implicitly assumed that these are negligible compared to the positive impacts they enable. Yet, the number of IoT devices worldwide, albeit difficult to estimate, is expected to continue

increasing from 5 billion in 2020 up to 200+ billion in 2030 [4]. Such an exponential growth of the number of manufactured IoT devices also implies a strong increase of the associated direct environmental impacts. For instance, GHG emissions due to the production of end nodes could range from 22 to 1124 MtCO2-eq/year in 2027 depending on the growth scenarios [4], whereas the annual carbon footprint of the global ICT production in 2020 has been evaluated between 281 and 543 MtCO2-eq [4].

From an engineering standpoint, several actions could however be taken to curb the increasing environmental impacts of IoT devices. An important shortcoming lies in the lifetimes of the end nodes. For small consumer electronics such as smartphones, the carbon footprint of the manufacturing often exceeds the carbon footprint linked to the electricity consumption of the device [11]. Increasing the lifetime of the device is thus a straightforward way to decrease its associated environmental burdens [12]. In spite of that, most end nodes are commercially intended to have short lifespans, since their designs very rarely incorporate means for repair, upgrade or recycling [13]. In addition to the regular wear-out that defines the expected lifetime of a device, IoT equipment is especially subject to technological obsolescence because of the fast-paced innovation in the sector. This obsolescence, which is driven by user motivations to benefit from a newer technology, further shortens the effective lifetime of the devices [14].

This work focuses on a key functionality of IoT devices that is particularly exposed to technological obsolescence: the wireless connectivity. Nowadays, three competing technologies share the LPWAN market: NB-IoT, LoRaWAN and Sigfox [6]. Nonetheless, new protocols are constantly designed and put on the market, especially with the advent of satellite IoT communications [15]. Because the end nodes cannot be upgraded to support new radio standards, the arrival of more efficient protocols may trigger a technological obsolescence and incite users to replace their existing devices, which is both economically and environmentally unsound.

To overcome this environmental conundrum, we study in this thesis two complementary approaches whose common goal is to prevent the technological obsolescence of wireless IoT networks. The first approach targets the low-power end nodes and consists in the design of a software-defined radio (SDR) for LPWAN protocols, whose PHY layer implementation can be reprogrammed by means of a software update. The second approach aims specifically at improving the scalability and robustness of existing LoRaWAN networks by designing multi-user receivers for the gateways.

In the remainder of this introduction, we first present in more detail the

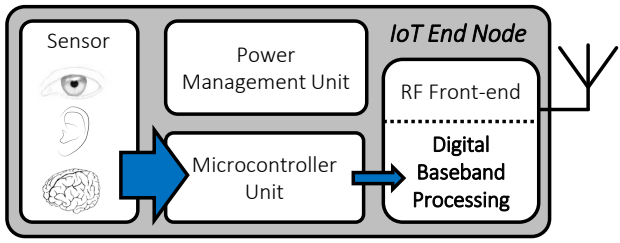


Fig. 1.2 Illustration of an IoT end node. The radio includes a digital baseband processing unit and an RF front-end. The first part of this thesis focuses on software implementations of the digital baseband processing.

motivations for each approach separately. We then conclude by describing the contributions and the outline of this thesis.

1.1 Reprogrammable Software-Defined LPWAN Radios

IoT end nodes are designed following the architecture depicted in Fig. 1.2, whose central component is a low-power microcontroller unit (MCU). This architecture generally also contains an embedded sensor, a power management unit and a wireless radio. The sensor generates data based on its physical environment and feeds it to the MCU. The MCU then processes the data according to application needs, and eventually forwards the processed information to the radio for a wireless transmission to the gateway.

The wireless radio consists of a digital baseband (DBB) processing unit, which implements the PHY layer of a protocol, and a radiofrequency (RF) front-end. The DBB processing of LPWAN radios is often implemented in hardware and tied to a single standard. Since this architecture provides no reconfigurability at the PHY layer, LPWAN networks are particularly subject to technological obsolescence. Indeed, these networks cannot be upgraded to another protocol without disposing of the already deployed infrastructure and deploying new devices. Similarly, the re-use of end nodes for another application with different communication requirements (e.g., greater range or greater throughput) is also impossible.

To enhance the interoperability of IoT end nodes, the first part of this thesis considers the design of software-defined radios (SDRs) for LPWANs. In an SDR, the RF front-end is designed to be parameterizable while most of the DBB computations of the PHY layer are performed in software by a

processor. This reconfigurable architecture enables over-the-air updates of the communication protocol, and therefore allows a radio to switch from one standard to another throughout its entire lifetime. The flexibility offered by SDRs could therefore help to both extend the effective lifetimes of end nodes, but also to foster innovation at the physical layer of LPWANs.

However, for energy-constrained end nodes, the underlying challenge of this approach is to successfully implement the DBB processing in software while fitting in the 1–10 mW budget of low-power transceivers [16]. Since all DBB computations of an LPWAN radio cannot be carried out on a general-purpose processor in the tight energy budget of an end node, an efficient hardware/software partitioning of these computations is needed. Therefore, the first question that this thesis focuses on is

Question 1 *How can we design an efficient ultra low-power MCU architecture that is suitable for a software implementation of several LPWAN physical layers?*

1.2 Improving the Scalability of LoRaWAN Networks

NB-IoT, Sigfox and LoRaWAN are today the principal LPWAN protocols used in IoT networks [6]. NB-IoT is a low-power cellular standard derived from LTE that uses licensed frequency bands. LoRaWAN and Sigfox are two technologies operating in the unlicensed industrial, scientific and medical (ISM) frequency bands. Whereas the Sigfox network is managed worldwide by a single company, LoRaWAN networks are deployed locally and independently by both commercial operators and non-profit organizations. This decentralized organization, in addition to a simple and efficient PHY layer, has fostered the popularity of LoRaWAN in the IoT ecosystem.

Despite its wide adoption, it is well-known that LoRaWAN suffers from scalability limitations [17, 18]. The LoRaWAN protocol stack combines a PHY layer, usually named LoRa, and a MAC layer. To achieve long range communications in a stringent energy budget, the PHY layer uses a chirp spread spectrum modulation in a bit-interleaved coded modulation scheme [19, 20]. At the MAC layer, LoRaWAN uses a pure (non-slotted) ALOHA multiple access scheme to avoid an energy-intensive synchronization mechanism between the end node and the gateway.

Because the end nodes are not synchronized, collisions between network users may occur and even cause the network throughput to rapidly

decrease for an increasing number of users [21, 22]. The scalability of LoRaWAN networks is thus inherently limited by the interference, as for ALOHA networks. In particular, it is often assumed that LoRaWAN cannot cope with the deployment of massive IoT networks, such as smart city scenarios with more than $1000 - 10,000$ nodes per gateway [21, 23]. The continuous densification of LoRaWAN networks thus poses a risk to the functioning of the already deployed end nodes, since they may become inoperative when the network becomes interference-limited.

A straightforward solution to this problem would be to replace the existing IoT infrastructure with a newer protocol better suited for dense networks. For instance, a new PHY layer named LR-FHSS has recently been announced [23]. This PHY layer, which uses a frequency hopping spread spectrum modulation, is compatible with the LoRaWAN MAC layer. A preliminary study indeed shows that, for the same network goodput, LoRaWAN networks using the LR-FHSS PHY allow to serve about 100 times more devices than with the conventional LoRa PHY [23].

Yet, as previously discussed, the existing LoRaWAN end nodes cannot be upgraded to LR-FHSS since the radios only support the LoRa PHY. To avoid the disposal of the already deployed devices and its associated environmental cost, we propose instead to improve the performance of the LoRa PHY. More specifically, we study how multi-user receivers can increase the throughput of LoRa gateways. Multi-user receivers allow to recover separately colliding signals from different users by jointly demodulating them [24]. Instead of losing two colliding frames because of the interference, a gateway with a multi-user detector can successfully demodulate both of them, thereby increasing the network throughput.

Nonetheless, the LoRa PHY has not been designed for multi-user detection. Several practical problems hence need to be addressed, and in particular, the concurrent synchronization of the gateway to multiple users. This challenge leads us the second question on this thesis, which is

Question 2 *How can we design a practical multi-user receiver for LoRa gateways that is capable of jointly demodulating two colliding users?*

By trying to answer to this question, we derive in this thesis two different designs of multi-user receivers for LoRa. However, beyond their practicality, we also need to evaluate the benefits of such receivers at the network level. Hence, the third question that this thesis investigates is

Question 3 *To which extent do gateways with two-user receivers improve the*

performance of LoRaWAN networks, compared to the existing single-user receiver?

1.3 Contributions and Outline of this Thesis

In order to answer the questions outlined above, this thesis is organized as follows. Since LoRaWAN is a recurrent subject of this work, **Chapter 2** provides a self-contained description of the protocol. In this chapter, we exhaustively describe its PHY layer, and we also introduce the architecture of LoRaWAN networks, including some relevant aspects of the MAC layer. The remainder of this thesis is then split into two parts.

The first part of this thesis studies the implementation of low-power LPWAN SDRs, and in particular for the LoRa and Sigfox PHY layers. However, such SDRs require DBB algorithms with a sufficiently low complexity to be implementable on the general-purpose processors of IoT MCUs. Although the Sigfox PHY layer is substantially simple, many aspects of the implementation of LoRa transceivers had not been examined in the scientific literature when we started this research. Notably, a low-complexity synchronization algorithm for LoRa receivers was missing. **Chapter 3** hence studies the design of a low-complexity, yet efficient LoRa synchronization algorithm capable of correcting both sampling time and carrier frequency offsets. We start by deriving an accurate Nyquist-rate analytical model of a LoRa signal sampled with these two offsets. Using this model, we propose a new estimator for the fractional sampling time offset. We then design a complete synchronization procedure with a very low computational overhead that induces only 1 or 2 dB penalty compared to an ideally synchronized receiver. The contents of this chapter are adapted from [JP1].

In **Chapter 4**, we address the challenge of designing an ultra low-power (ULP) MCU architecture suitable for a software implementation of the DBB processing of LPWAN protocols. By means of a hardware/software partitioning of the DBB computations, we propose an MCU architecture with an ARM Cortex-M4 processor for the protocol-specific computations and a hardware digital front-end for the generic signal processing. The proposed architecture has been prototyped in a 28 nm FDSOI ULP design and the LoRa and Sigfox PHY layers have been implemented in software. Experimental evaluations of the MCU prototype for both SDRs show a power consumption between 32 and 332 µW for the DBB processing, and near-state-of-the-art receiver sensitivities below -120 dBm. To the best of our

knowledge, this work is the first to actually demonstrate SDR implementations of two commercial LPWAN PHY layers running on an ULP MCU. The contents of this chapter are adapted from [CP1], [CP2] and [JP2].

The second part of this thesis focuses on the design of multi-user receivers for LoRa gateways, and has been conducted in collaboration with the Telecommunications Circuits Laboratory of EPFL, Switzerland. Although several multi-user receivers have been proposed in the literature, they all fail to attain useful error rates at the low signal-to-noise ratios (SNRs) that are characteristic of LoRa communications. In **Chapter 5**, we hence derive a two-user detector from the maximum-likelihood principle. However, as the complexity of the maximum-likelihood sequence estimation is prohibitive, complexity-reduction techniques are introduced to enable practical implementations of the receiver. An in-depth performance analysis highlights that the proposed two-user detector inherently leverages the differences in received power, time offsets, and frequency offsets between the users to separate and demodulate their respective signals. To demonstrate the practicality of the proposed detector, we then design an interference-robust synchronization algorithm and implement it along the proposed two-user detector in the GNU Radio SDR environment. Experimental results show that our SDR implementation is capable of detecting and demodulating two interfering users with symbol error rates below 10^{-3}, even at low SNRs. The contents of this chapter are adapted from [CP3] and [JP3].

By leveraging the outcomes of the previous chapter, we propose in **Chapter 6** a successive interference cancellation LoRa soft-receiver capable of decoding frames from two colliding users. The proposed two-user detector explicitly leverages the bit-interleaved coded modulation scheme of LoRa to improve the detection and cancellation of the strongest interfering user. The weakest user is then demodulated with the matched filters derived in Chapter 5. We show that in the presence of two interfering users, the usage of a low coding-rate and an iterative soft-detection are essential to attain error rates sufficiently close to the single-user scenario. Thanks to network-level simulations carried out with network simulator ns-3, we subsequently evaluate the gains of our proposed two-user receiver in a realistic LoRaWAN network. For an overall frame error rate of 1%, simulation results show that a LoRaWAN network employing our two-user receiver may serve 4.7 times more devices than a network with only a single-user receiver at the gateway. The contents of this chapter are adapted from [CP4] and [JP4].

2

Introduction to LoRa and LoRaWAN

LoRaWAN is nowadays the most popular non-cellular LPWAN protocol for IoT communications. Unlike the energy-intensive cellular standards NB-IoT and LTE-M, LoRaWAN relies on low-complexity PHY and MAC layers operating in the unlicensed ISM frequency bands. The LoRa PHY layer is a proprietary technology owned by Semtech [25, 26], whereas the MAC layer is an open standard defined by the LoRa Alliance [27].

Because the PHY layer is patented, the only transceivers compatible with LoRa are commercialized in agreement with Semtech, and no precise specifications are publicly available. Yet, thanks to reverse-engineering efforts, the chirp spread spectrum modulation and other aspects of the LoRa PHY have extensively been studied in the scientific literature [28, 29, 30, 31, 32]. The optimization of the LoRaWAN MAC layer has also become a very popular research topic over the last years [33].

The objective of this chapter is to serve as a self-contained introduction to LoRa and LoRaWAN. We provide first in Section 2.1 a detailed description of the LoRa PHY layer, and we then discuss the baseband implementations of several LoRa transceivers. Subsequently, we explain in Section 2.2 the LoRaWAN network architecture and specific aspects of the MAC layer that play a role in the digital baseband processing of LoRa transceivers.

2.1 The LoRa Physical Layer

The LoRa PHY operates within the ISM radio bands, mainly around 868 MHz in Europe and 915 MHz in North America [34]. The protocol uses a chirp spread spectrum (CSS) modulation, that provides several key benefits for IoT communications. First, LoRa, whose name stands for *long range*, achieves wireless transmissions over distances up to 15–30 km [35, 36]. By configuring the spreading gain of the CSS modulation, the PHY layer allows to trade off transmission time and data rate for longer communication ranges [33], [37]. Also, the modulation is more robust to frequency selective channels than conventional modulation schemes such as FSK [30].

In this section, we provide a thorough explanation of the LoRa PHY layer. We first describe the CSS modulation and how LoRa integrates this modulation with a Hamming code following the bit-interleaved coded modulation (BICM) principle. Next, we explain the general digital baseband architecture of a LoRa transmitter and the implementations of several LoRa receivers.

2.1.1 Chirp Spread Spectrum Modulation

The CSS modulation of the LoRa PHY is a continuous phase modulation (CPM) that relies on chirps, i.e., waveforms whose instantaneous frequency increases linearly over time and spans a bandwidth B. Typical passband bandwidth values, such as advertised by commercial LoRa transceivers, are $B \in \{125, 250, 500\}$ kHz. Every symbol has a duration $T = \frac{2^{\mathrm{SF}}}{B}$ and carries SF bits of information, where $\mathrm{SF} \in \{7, \ldots, 12\}$ is called the *spreading factor*. For a symbol $s \in \{0, \ldots, 2^{\mathrm{SF}} - 1\}$, the signal starts at an initial frequency of $B\left(\frac{s}{2^{\mathrm{SF}}} - \frac{1}{2}\right)$. The instantaneous frequency of the modulated signal increases linearly over time, until the moment $t_{\mathrm{fold}} = \frac{2^{\mathrm{SF}} - s}{B}$, when the instantaneous frequency is decreased by B (i.e., folded) to keep the signal in the allocated bandwidth [32]

$$x_s(t) = \begin{cases} e^{j2\pi\left(\frac{B}{2T}t^2 + B\left(\frac{s}{2^{\mathrm{SF}}} - \frac{1}{2}\right)t\right)} & \text{for } 0 \leq t < t_{\mathrm{fold}}, \\ e^{j2\pi\left(\frac{B}{2T}t^2 + B\left(\frac{s}{2^{\mathrm{SF}}} - \frac{3}{2}\right)t\right)} & \text{for } t_{\mathrm{fold}} \leq t < T. \end{cases} \tag{2.1}$$

Commercial LoRa receivers such as the Semtech SX1276 carry out the

demodulation of a LoRa symbol at a sampling frequency $f_s = B$ [26, 38]. When sampled at this frequency, the discrete-time signal of a symbol consists of $N = 2^{\mathrm{SF}}$ samples. It has however been shown that the actual bandwidth of LoRa signals slightly exceeds the advertised bandwidth B [32]. Yet, for a sufficiently large N, the discrete-time baseband-equivalent model of a LoRa symbol s sampled with $f_s = B$ can still be expressed as

$$x_s[n] = \begin{cases} e^{j2\pi\left(\frac{n^2}{2N}+\left(\frac{s}{N}-\frac{1}{2}\right)n\right)}, & 0 \leq n < n_f, \\ e^{j2\pi\left(\frac{n^2}{2N}+\left(\frac{s}{N}-\frac{3}{2}\right)n\right)}, & n_f \leq n < N, \end{cases} \tag{2.2}$$

where $n_f = N - s$ is the sample index at which the frequency folding occurs. In the remaining of this thesis, we identify for convenience as a *Nyquist-rate receiver* any LoRa receiver that performs both the synchronization and the demodulation stages at the sampling frequency $f_S = B$, despite the fact such receivers do not strictly fulfill the Nyquist-Shannon sampling theorem [39].

Moreover, we always assume in this thesis that the transmission takes place over an additive white Gaussian noise (AWGN) channel with a complex-valued channel gain $h \in \mathbb{C}$, unless stated otherwise. A received LoRa symbol s after sampling at $f_s = B$ is then represented by

$$y[n] = hx_s[n] + z[n] \tag{2.3}$$

where $z[n] \sim \mathcal{CN}(0, \sigma^2)$ is complex AWGN with variance $\sigma^2 = \frac{N_0}{2}$ and N_0 is the single-sided noise power spectral density.

The maximum likelihood (ML) detection of a LoRa symbol can be efficiently implemented at the Nyquist rate $f_s = B$ with a discrete Fourier transform (DFT). To this end, the sampled signal $y[n]$ is first multiplied point-wise with $x_0^*[n]$, the complex conjugate of an unmodulated symbol $s = 0$. This multiplication yields the *dechirped* signal

$$\tilde{y}[n] = y[n]x_0^*[n] = \sqrt{P}e^{j2\pi n\frac{s}{N}+\theta} + \tilde{z}[n], \tag{2.4}$$

where $P = |h|^2$ is the power of the signal at the receiver, $\theta = \angle h$ represents the phase introduced by the channel, and $\tilde{z}[n] = z[n]x_0^*[n]$. For a perfectly synchronized receiver, $\tilde{y}[n]$ hence contains a single complex tone

of frequency $\frac{s}{N}$ and AWGN. Let

$$Y[i] = \sum_{n=0}^{N-1} \tilde{y}[n] e^{-j2\pi \frac{ni}{N}} \tag{2.5}$$

be the i-th bin from the N-point discrete Fourier transform (DFT) of the dechirped signal. Computing the DFT on the dechirped signal yields

$$Y[i] = N \cdot \Pi(i - s, 0) + Z[i] \tag{2.6}$$

where $\Pi(k, \xi) = \dfrac{1 - e^{-j2\pi(\xi - k)}}{1 - e^{-j2\pi(\xi - k)/N}}$ is the discrete *sinc* function centered around the fractional frequency ξ for an N-point DFT and $Z[i]$ is the DFT of the dechirped AWGN $\tilde{z}[n]$. Assuming ideal synchronization, we have $\xi = 0$ and the *sinc* function becomes a Kronecker delta at position $i = s$.

LoRa receivers typically perform a non-coherent detection of the received symbols to avoid the tracking of the phase θ [30, 31, 40]. For a non-coherent detection, the ML symbol candidate $\hat{s}$ corresponds to the index of the DFT bin $Y[i]$ with the largest magnitude

$$\hat{s} = \arg\max_{\bar{s}} |Y[\bar{s}]| . \tag{2.7}$$

In the presence of AWGN, the non-coherent detection incurs a 0.7 dB loss compared to the ML coherent detection [41, 42].

We finally present two properties of the CSS modulation that play an important role in LoRaWAN networks. First, increasing the spreading factor SF by one increases the spreading gain of the modulation by 2.5 dB, but at the expense of halving the data rate [38, 43]. The MAC layer uses this property to allocate for each device the lowest possible SF while ensuring a minimal quality of service. This optimization minimizes the airtime and the energy consumption of the radio [27]. Secondly, LoRa waveforms using different SFs are quasi-orthogonal to each other [44]. This property, further described in Chapter 5, allows to increase the throughput of a LoRaWAN network if the SFs are evenly distributed among the end nodes [18].

2.1.2 Bit-Interleaved Coded Modulation

To improve its robustness against noise, fading channels, and residual time and frequency offsets, LoRa combines its CSS modulation with an error-

Fig. 2.1 Architecture of a Bit-Interleaved Coded Modulation encoder.

correcting code following the bit-interleaved coded modulation (BICM) strategy [45, 46]. At the transmitter, BICM consists in the concatenation of a binary encoder, an interleaver and a binary labeling (i.e., a symbol mapper), as shown in Fig. 2.1 [47]. This simple and flexible scheme allows to independently select a modulation and an error correcting code with any code rate. The interleaver acts as a channel interleaver and provides a time diversity gain to the receiver, thus improving the robustness of transmissions in fading channels. Efficient receiver implementations for BICM systems compute soft-information for each bit in the demodulation stage in the form of log-likelihood ratios, then de-interleave these soft values and eventually feed them to a soft-input channel decoder.

Although one of the original Semtech LoRa patents does mention the concept of soft-demodulation for LoRa receivers [26], the assumption that LoRa actually implements a BICM scheme has been raised in the scientific literature only in 2019 [46]. Very few papers hence consider and evaluate soft-decisions for the demodulation and decoding stages of LoRa receivers [46, 48, 49]. Instead, most works in the literature either ignore the coding or implement a hard-decision decoding.

We introduce next the different elements of the LoRa BICM strategy (the Hamming code, a diagonal interleaving scheme and Gray mapping) and explain their implementation in a LoRa transmitter. The implementation of a single-user LoRa soft-receiver is later explained in Chapter 6.

Error-Correction Coding

LoRa uses two simple schemes for error detection and error correction [40]. Let $\text{CR} = (k_c, n_c)$ be the code rate, where k_c denotes the data word length and n_c denotes the codeword length. Four different code rates $\text{CR} = \{4/5, 4/6, 4/7, 4/8\}$, and an uncoded mode, are supported. For $\text{CR} = 4/5$, a simple even parity-check code is used. For the other code rates, LoRa uses the well-known Hamming code, with its 1-bit punctured and extended versions. Let d_0, d_1, d_2 and d_3 be the bits of a data word. The codewords are

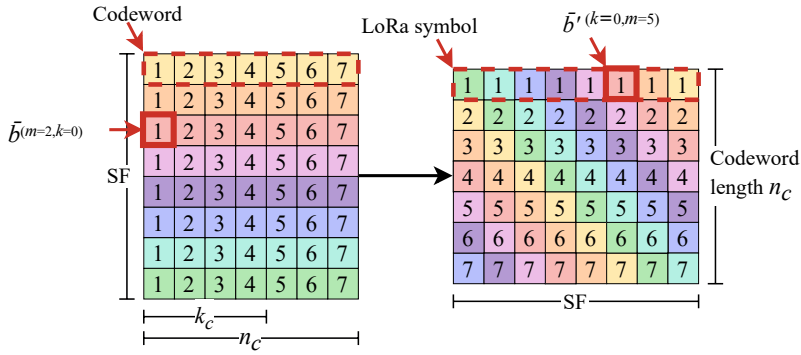


Fig. 2.2 LoRa interleaving for SF $= 8$ and a $(4,7)$ Hamming code.

obtained using the following generator matrix

$$\begin{bmatrix} d_3 & d_2 & d_1 & d_0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 & p_0 & p_1 & p_2 & p_3 \end{bmatrix}, \tag{2.8}$$

where only the first n_c bits are kept, and p_0, p_1, p_2, p_3 are the parity bits.

Interleaving

The interleaving stage is a simple block diagonal interleaver with the arrangement shown in Fig. 2.2. The interleaver transforms an input matrix of SF $\times$ n_c bits to a shuffled matrix of size $n_c \times$ SF. In the remainder of this thesis, we denote $b^{(m,p)}$ as the input data bits before coding and interleaving, where $m \in \{0, \ldots, \text{SF} - 1\}$ and $p \in \{0, \ldots, k_c - 1\}$. Similarly, let $\bar{b}^{(m,k)}$ and $\bar{b}'^{(k,m)}$ ($k \in \{0, \ldots, n_c - 1\}$) be the corresponding coded bits before and after interleaving, respectively. Each interleaved block hence contains SF codewords that are subsequently mapped to n_c symbols. The objective of the interleaving is to provide time diversity at the receiver by spreading demodulation errors from one or several symbols to as many codewords as possible. For instance, by assuming a receiver with a hard-decision Hamming decoder and a code rate of $(4,7)$, if a single symbol demodulation error occurs in an interleaved block, each codeword will contain at most one erroneous bit, which can always be recovered by the decoder.

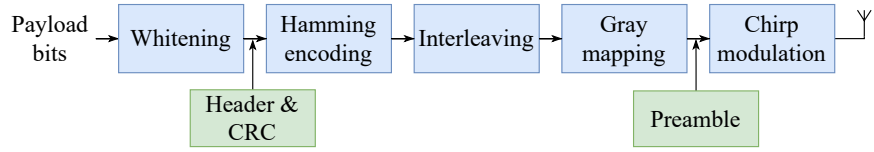


Fig. 2.3 Illustration of the LoRa PHY Tx chain, which implements Bit-Interleaved Coded Modulation.

Binary Labeling

Finally, the n_c rows of $\mathrm{SF} = \log_2 N$ bits in an interleaved block are mapped into symbols $s \in \{0, \ldots, N-1\}$. To this end, a labeling with Gray codes is used. Gray codes are a binary ordering which guarantee that two successive coded values differ in only one bit. In a LoRa receiver chain, this property is particularly useful when the receiver is not perfectly synchronized to the transmitter, as small frequency and time offsets often cause one-off symbol demodulation errors. The Gray coding ensures that if such one-off demodulation error happens, only one bit of the demodulated symbol after Gray decoding will be erroneous. This behavior is further discussed in Chapter 3.

2.1.3 Baseband Transmitter Chain

We now introduce the digital baseband architecture of a LoRa transmitter, such as the Semtech SX1276 [38]. This architecture is depicted in Fig. 2.3. The remaining elements of a LoRa transmitter that have not yet been introduced are also described.

The transmitter chain starts with a whitening stage that adds in $GF(2)$ a pseudo-random binary sequence to the input payload bits [40]. A header (indicating the payload length and the code rate) and a cyclic redundancy check are optionally inserted into the whitened bits. The data then follows the different stages of the bit-interleaved coded modulation (Hamming encoding, interleaving, Gray mapping and CSS modulation) previously explained. A preamble is finally added to the beginning of each frame to facilitate the synchronization of the receiver with the transmitter [25]. The structure of a LoRa preamble is illustrated in Fig. 2.4.

The preamble structure consists of N_{pr} repetitions of an unmodulated upchirp (i.e., $x_0[n]$), succeeded by two identical network identifiers symbols and 2.25 downchirps $x_0^*[n]$ [50]. The network identifiers symbols are

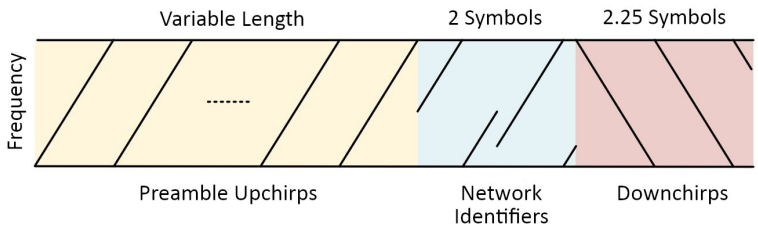


Fig. 2.4 Structure of a LoRa preamble.

used for frame synchronization and to distinguish devices from different networks. We further show in Chapter 3 that the joint usage of upchirps and downchirps in the preamble allows a receiver to estimate its sampling time and carrier frequency offsets. A downchirp can be demodulated similarly to an upchirp by dechirping the sampled signal with an unmodulated upchirp $x_0[n]$, instead of an unmodulated downchirp $x_0^*[n]$. Further details on the usage of the preamble in the synchronization procedure of LoRa receivers are provided in Chapter 3.

2.1.4 Baseband Implementations of LoRa Receivers

We finally describe the baseband implementations of five LoRa receivers. The first three receivers are open source SDR implementations designed for scientific purposes, whereas the latter two are commercial chips fabricated and sold by Semtech.

Software-Defined Radio Implementations

Three implementations of LoRa SDRs have been reported in the literature [40, 51, 52]. All three use the open source GNU Radio environment and have been experimentally validated. Contrary to the low-power embedded SDR implementation described in Chapter 4, these GNU Radio implementations are executed on high-end computers or servers. We provide a brief summary of the characteristics of each implementation:

- *Robyns et al.* provided the first functional implementation of a LoRa SDR receiver [51]. The baseband Rx chain includes a preamble detection and synchronization procedure, a hard-demodulation of the CSS symbols in the time domain and a hard-decoding of the Hamming code. The synchronization procedure requires an oversampling fac-

tor $R = 8$ and implements a modified Schmidl-Cox algorithm to estimate the sampling time and carrier frequency offsets. Yet, the SDR exhibits poor performance as it only reaches frame error rates below 10% for SNRs higher than 15 dB, whereas commercial LoRa receivers attain this error rate for SNRs well below 0 dB.

- *Tapparel et al.* implemented an entire LoRa transceiver with Tx and Rx data paths [40]. The demodulation is implemented using the Nyquist-rate ML non-coherent detector previously described. No oversampling of the signal is required, but the synchronization stage uses a zero-padded DFT of size $2(N_{\text{pr}} - 2)N$. The decoding is achieved using a hard-decision Hamming decoder. For correctly detected frames, an experimental evaluation of the SDR implementation for SF $= 7$ shows good performance (less than 1 dB loss compared to an ideally synchronized receiver).

- *Xu et al.* implemented a complete LoRa receiver [52]. Contrary to *Tapparel et al.*, their Rx data path uses an oversampling factor $R = 2$ for both the synchronization procedure and the demodulation. They propose a new hard-decision demodulation strategy that uses the oversampling to facilitate the receiver synchronization, and perform a hard-decoding of the codewords. An experimental set-up with an USRP N210 running their SDR implementation is capable of receiving 95% of the transmitted frames up to a distance of 3600 m for SF $= 12$ and $B = 125$ kHz.

From the above descriptions, we point out that the SDR receivers from *Robyns et al.* and *Xu et al.* require an oversampling of the received signal ($f_s > B$) for at least the synchronization procedure. *Tapparel et al.* avoid the oversampling by relying on zero-padded DFTs. Moreover, no SDR receiver implements a soft-demodulation and soft-decoding of the LoRa symbols.

Semtech SX1276: End Node Transceiver

The Semtech SX1276 is a low-power LoRa transceiver for end nodes [38]. The chip integrates both a digital baseband unit and an RF front-end. In Tx, the radio has a maximal output power of 20 dBm (or 14 dBm with a high efficiency power amplifier). In Rx, the radio has a power consumption of 38 mW and a minimal sensitivity of -123 dBm for SF $= 7$, $B = 125$ kHz and -136 dBm for SF $= 12$, $B = 125$ kHz.

The implementation of the baseband unit is partly described in two

patents [25, 26]. Contrary to the previous SDR receivers, both the Tx and the Rx data paths are sampled and processed at the Nyquist rate $f_s = B$. The SX1276 datasheet mentions that the coding is particularly efficient to improve the demodulation performance in the presence of interference, and recommends increasing the code rate in such situation. In addition, an implementation of the soft-demodulation and soft-decoding of LoRa symbols is described in [26]. These statements lead us to believe that the SX1276 implements a soft-decision decoding of the Hamming code and thus leverages the BICM strategy of the LoRa PHY.

Semtech SX1301: Macro Gateway Transceiver

The Semtech SX1301 is a digital baseband chip for LoRaWAN macro gateways [53]. In practical gateways, this baseband chip is coupled with an RF front-end chip such as the Semtech SX1250. To allow the reception of concurrent LoRa transmissions using different SFs, the baseband unit features 10 parallel Rx data paths:

- 8 LoRa data paths at $B = 125$ kHz, each with a configurable SF.
- 1 LoRa data path at a configurable $B \in \{125, 250, 500\}$ kHz, with a single predefined SF.
- 1 (G)FSK data path.

In Tx, a single data path is responsible for the transmission of either a LoRa or (G)FSK frame.

More specifically, the LoRa Rx data paths all implement a Nyquist-rate synchronization and demodulation of the acquired LoRa signals. Yet, contrary to the SX1276, the SX1301 datasheet does not mention any specific usage of the coding. It is hence unclear whether the SX1301 actually implements a soft-demodulation and soft-decoding of the received data.

2.2 The LoRaWAN MAC Layer

We now provide a brief introduction to the LoRaWAN MAC layer. We first describe the architecture of LoRaWAN networks. We subsequently explain three aspects of this protocol that play an important role in the network-level evaluation conducted in Chapter 6.

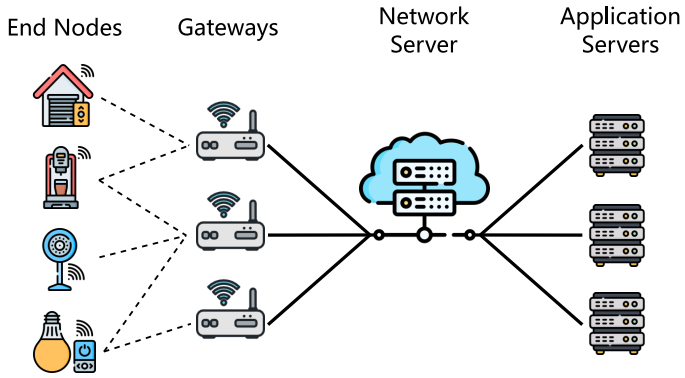


Fig. 2.5 Star-of-stars architecture of a LoRaWAN network. The wireless connections between the end nodes and the gateways use the LoRa PHY, whereas the gateways, network server and application servers communicate over the Internet.

2.2.1 Network Architecture

LoRaWAN networks follow a *star-of-stars* topology, as shown in Fig. 2.5. The central element of the topology is the network server, which manages gateways, end nodes and applications in the entire network.

In uplink, the end nodes transmit a message over the air and unknowingly communicate with one or several gateways. Point-to-point communications between end nodes are not supported. In turn, the gateways are responsible for the reception and the demodulation of incoming LoRa frames. Once decoded, the payloads of the received frames are subsequently transferred through the Internet to the network server. The network server retrieves the messages received by the gateways, identifies the corresponding application, and subsequently transfers them to the appropriate application server.

In downlink, when the network server receives a response from an application server, it schedules the transmission of the corresponding LoRa frame by selecting an appropriate gateway to transfer it to the targeted end node. The network server is thus the real processing unit in the network, whereas the gateways are merely an interface between the end nodes and the network server.

For energy efficiency reasons, LoRaWAN end nodes implement a non-slotted ALOHA multiple access scheme to transmit frames to the gateway.

End nodes are not synchronized in time and may transmit uplink frames at any moment. Collisions between uplink frames may hence arise due to the lack of synchronization between end nodes. Similarly to all ALOHA networks, interference between same-technology devices has been identified as the main obstacle to the scaling of dense LoRa networks [17, 18]. This scalability limitation is described in more detail in Chapter 5.

2.2.2 Duty Cycling and End Nodes Classes

The LoRa PHY uses the unlicensed ISM frequency bands, which enforce fair access policies. In Europe, transmitters using the 868 MHz and 867 MHz sub-bands should either comply with a 1% radio duty cycle or implement a listen-before-talk mechanism. LoRaWAN follows the former policy, which means that if the radio transmitted during one second, it cannot transmit during the following 99 seconds [33]. This policy notably also applies to the gateways. As a consequence, situations in which a gateway cannot respond to an end node may arise due to the duty cycling regulations. LoRaWAN alleviates this constraint at the gateway by defining two downlink receive windows for each uplink message, as described hereafter. Similarly, to minimize the airtime of transmissions and thus to avoid collisions, LoRa radios operating in LoRaWAN networks can only use the code rate $\mathrm{CR} = 4/5$ [27].

Furthermore, the LoRaWAN specifications define three classes of end nodes, namely, Class A, Class B and Class C [27]. We briefly introduce the main characteristics of these classes:

- **Class A**: Communications with class A devices can only be initiated by the end nodes. A device may start an uplink transmission at any time. Two short downlink windows (RX1 and RX2) are subsequently opened for the gateway to transmit an answer or an acknowledgment. The server can either respond during the RX1 or the RX2 window. Downlink communications in window RX1 use the same parameters (frequency band, bandwidth B, SF) as the uplink transmission, whereas RX2 transmissions use a predefined frequency band, bandwidth and SF. The delays between the end of the uplink transmission and the begin of the RX1 and RX2 windows are $1\mathrm{s} \pm 20\mu\mathrm{s}$ and $2\mathrm{s} \pm 20\mu\mathrm{s}$, respectively.
- **Class B**: This class extends the characteristics of Class A. In addition, the end nodes open periodically downlink windows for scheduled

downlink messages. This mode requires a time synchronization between the gateway and the end nodes, which is achieved thanks to periodic beacons sent by the gateway.

- **Class C**: This class extends the characteristics of Class A. In addition, Class C devices must continously listen to the gateway for incoming frames that may arrive at any time.

The Class A is targeted for low-power devices running on batteries or energy-harvesting, whereas Class C end nodes are more energy-intensive but benefit from a very low receive latency. Class B offers an intermediate trade-off between Classes A and C.

2.2.3 Adaptive Data Rate Mechanism

The LoRaWAN MAC layer provides a mechanism, namely the *Adaptive Data Rate* (ADR) mechanism, for optimizing the data rates, airtime and energy consumption of the end nodes. The ADR tunes the transmission power, bandwidth and SF of the end nodes using data from the PHY layer, such as the SNR or the frame error rate. Depending on the configuration, the ADR optimization is conducted either by the end node or the network server. The ADR algorithm on the end nodes is defined by the LoRaWAN specifications, whereas the implementation of the ADR algorithm on the network server can be defined by the network operator. In the latter case, the network server indicates to the end node the radio parameters to use in downlink messages.

Since LoRa gateways are capable of concurrently demodulating frames using different SFs, the ADR algorithms select for each end node the lowest possible SF [54]. We show in Chapter 6 that this strategy enables an efficient repartition of the SFs among the end nodes if their locations are uniformly distributed in the cell. In the end, the ADR mechanism seeks to both minimize the energy consumption of the end nodes and to maximize the network throughput, thanks to the near-orthogonality of inter-SF transmissions.

PART I
Towards ULP IoT Software-Defined Radios

3

A Low-Complexity LoRa Synchronization Algorithm

Although the literature on LoRa has grown rapidly over the years, some aspects of the DBB processing of LoRa radios have scarcely been studied. As discussed in Chapter 2, the CSS modulation is nowadays well-known, but little research has been conducted on the synchronization stage of a LoRa receiver. In particular, an SDR implementation of a LoRa transceiver on an ULP MCU requires an efficient low-complexity synchronization algorithm. Yet, despite the recent efforts on the topic of LoRa synchronization and offsets correction, a complete and low-complexity LoRa synchronization algorithm that results in low packet error rates was still missing in the literature when we designed the ULP MCU architecture presented in Chapter 4.

The first step of this thesis was hence to design a synchronization algorithm suitable for a software execution of the LoRa DBB on ULP MCUs. The principal impairments to correct during the synchronization procedure of a LoRa receiver are the carrier frequency offset (CFO) and the sampling time offset (STO) [31, 50]. An efficient estimation of these offsets is crucial, as the presence of residual offsets prevents a receiver to reach the low sensitivity levels provided by the CSS modulation. However, contrary to the majority of LoRa synchronization algorithms available in the literature, our objective is to keep the overall complexity of the receiver as low as

possible by restraining the sampling frequency at $f_s = B$, i.e., by targeting the implementation of a Nyquist-rate LoRa receiver.

This chapter describes the design and the evaluation of the proposed Nyquist-rate LoRa synchronization algorithm, and is organized as follows. We first explain in Section 3.1 the effects of a CFO and STO on the demodulation stage of a Nyquist-rate LoRa receiver. We conduct a brief analysis of the state-of-the-art and we explain why other synchronization algorithms published in the literature are not appropriate for an implementation on an ULP MCU. Next, we derive in Section 3.2 an analytical Nyquist-rate signal model of LoRa preamble symbols received prior to synchronization, which includes the aforementioned offsets. This model then allows us in Section 3.3 to design a new low-variance estimator capable of efficiently estimating the fractional component of the STO. Thanks to this estimator and the accurate signal model, we design in Section 3.4 a complete Nyquist-rate LoRa synchronization algorithm. We finally evaluate the performance and the computational complexity of this algorithm in Section 3.5 and Section 3.6, respectively.

3.1 Motivations and Related Works

To better understand why previously published LoRa synchronization algorithms are not suitable for an implementation on an ULP MCU, we first provide an intuitive explanation of the impacts of both an STO and a CFO on the demodulation stage of a Nyquist-rate LoRa receiver.

Let us hence consider a LoRa signal received with an STO τ and a CFO Δf_c with respect to the transmitter. The continuous-time baseband-equivalent model of the signal is

$$y(t) = he^{j2\pi t\Delta f_c} x_s(t-\tau) + z(t). \tag{3.1}$$

Throughout this chapter, we assume a constant received power $P = 1$.

Bernier et al. have shown that the effects of a CFO or STO on the non-coherent ML detector of (2.7) can be seen as *approximately* equivalent [50]. When demodulating a repeating LoRa symbol s (such as encountered in the LoRa preamble for $s = 0$) sampled at $f_s = B$, a CFO (resp. STO), displaces the *sinc* function $\Pi(i - s, 0)$ in the frequency domain after dechirping from position s to $s + N\Delta f_c/B$ (resp. $s - B\tau$) [26, 31, 50]. Both offsets

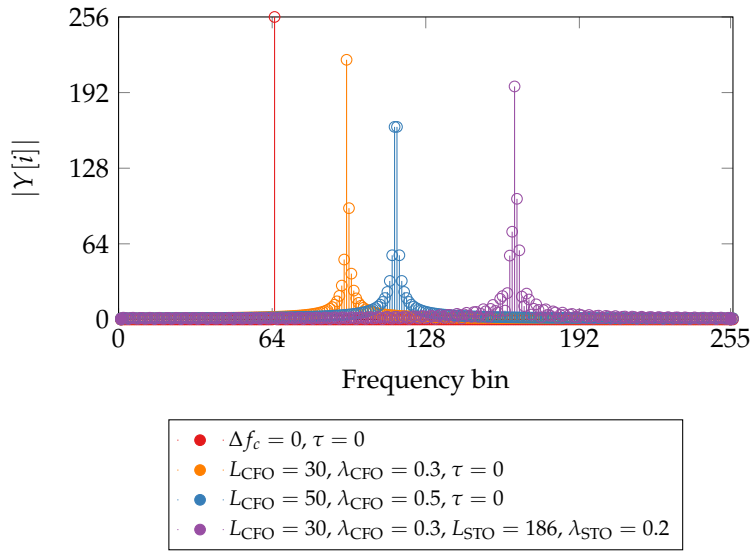


Fig. 3.1 Amplitudes $|Y[i]|$ obtained when demodulating a repeating symbol $s = 64$ (SF $= 8$) without AWGN for different CFO and STO values. Integer offsets displace the Kronecker delta originally located at $k = 64$, whereas fractional offsets scatter the delta on several frequency bins.

may be decomposed into integer and fractional components such as

$$\Delta f_c = B \cdot \frac{L_{CFO} + \lambda_{CFO}}{N}, \tau = \frac{L_{STO} + \lambda_{STO}}{B} \tag{3.2}$$

where L_{CFO}, L_{STO} are integer, and $\lambda_{CFO}, \lambda_{STO} \in]-0.5, 0.5]$ [50]. Since L_{CFO} and L_{STO} are integer, they shift the Kronecker delta in the DFT $Y[i]$ initially located at $i = s$ of an integer number of DFT bins. On the contrary, the fractional offsets λ_{CFO} and λ_{STO} scatter the energy of the symbol previously contained in a single bin over several frequency bins. These behaviors are illustrated in Fig. 3.1.

In the presence of AWGN, a symbol demodulation error occurs if and only if any of the $|Y[i]|$ values for $k \in \{0, \dots, N-1\} \backslash s$ exceeds the value of $|Y[s]|$. We denote the probability of incorrectly demodulating a symbol s under a fractional CFO λ_{CFO} as $P(\hat{s} \neq s | s, \lambda_{CFO})$. The scattering induced by the fractional offset increases $P(\hat{s} \neq s | s, \lambda_{CFO})$, due to the increase of energy in the bins adjacent to $i = s$. It is therefore necessary to accurately estimate and correct both integer and fractional offsets to avoid a severe

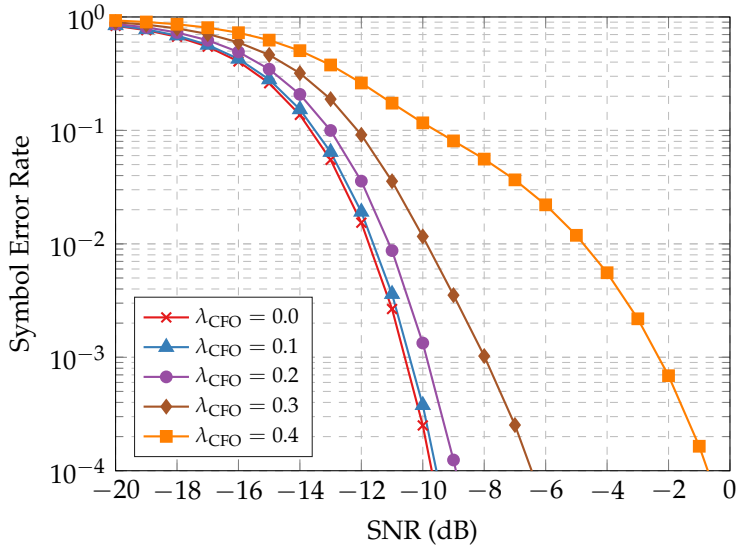


Fig. 3.2 SER of a Nyquist-rate LoRa receiver using (2.7) with a fractional CFO λ_{CFO} ($L_{\text{CFO}} = 0$ and SF $= 8$) for $\tau = 0$. Identical results apply for a receiver with a fractional STO λ_{STO} (for $L_{\text{STO}} = 0$) and $\Delta f_c = 0$.

degradation of the symbol error rate. An accurate analytical approximation for $P(\hat{s} \neq s|s, \lambda_{\text{CFO}})$ has been derived in [45]. This approximation is used to illustrate in Fig. 3.2 the impact of different fractional CFOs λ_{CFO} on the symbol error rate (SER) for uncoded transmissions (i.e., CR $= 4/4$). Obviously, values of λ_{CFO} near -0.5 or 0.5 are much more detrimental to the SER than smaller ones, e.g., $\lambda_{\text{CFO}} = 0.1$ has almost no impact on the receiver.

3.1.1 Related Works

Several LoRa receivers with synchronization algorithms have been published in the literature over the last years. These receivers can be classified into two categories: Nyquist-rate receivers and receivers operating with oversampling, i.e., with a sampling frequency $f_s > B$. The latter receivers are usually more performant than the former ones, but they have a higher computational complexity and are hence more tedious to implement on an ULP SDR platform.

Recent research has highlighted that the signal model of *dechirped* LoRa

symbols strongly differs whether the receiver samples the received signal at the Nyquist frequency or with oversampling. Due to the frequency folding of the LoRa waveforms, the Fourier transform of a continuous-time dechirped LoRa signal actually spans two *sinc* functions, at the frequencies $B\frac{s}{N}$ and $-B + B\frac{s}{N}$ [39]. A sampling frequency $f_s \geq 2B$ is hence required to strictly respect the Nyquist-Shannon theorem in the demodulation stage of a LoRa receiver. Yet, as discussed in Chapter 2, commercial LoRa receivers (which we refer to in this manuscript as *Nyquist-rate* receivers) demodulate signals at $f_s = B$. These receivers however still achieve an optimal demodulation of the received symbols because the two *sinc* functions constructively overlap in the DFT $Y[i]$ into a single Kronecker delta at position $i = s$, thanks to the aliasing [39].

Synchronization Algorithms with Oversampling

Although the aliasing in Nyquist-rate receivers simplifies the demodulation of LoRa symbols, it also erases properties of the signal that simplify the estimations of the CFO and STO [55]. For this reason, most of the works in the literature design synchronization algorithms for LoRa receivers that oversample the received signal. We now provide a brief summary of these works by emphasizing their computational complexity.

The authors in [51] present a preamble synchronization algorithm for LoRa based on the Schmidl-Cox algorithm, which however works only for very high signal-to-noise ratio (SNR) values. Oversampling is used in the synchronization procedure to obtain accurate estimations of the STO.

The authors in [56] present a joint estimator of time and frequency offsets based on the maximum-likelihood principle. Their synchronization algorithm exhibits near-optimal performance, but requires an oversampling rate of at least two and has a stringent complexity of $\mathcal{O}(N^2 \log N)$.

The authors in [55] design a synchronization algorithm working in the time-domain that takes advantage of the specific time-domain properties of the LoRa preamble. This algorithm relies on a matched filtering with convolutions and requires an oversampling rate greater than two to achieve a sufficiently accurate estimation of the STO and CFO, with a complexity of $\mathcal{O}(N^2)$.

Nyquist-Rate Synchronization Algorithms

Unfortunately, the synchronization procedures described in [51, 55, 56] have computational complexities that are prohibitive for ULP MCUs. Similarly to the commercial low-power transceiver SX176 [38], a synchroniza-

tion algorithm running at $f_s = B$ would have a much lower computational complexity. The only papers in the literature that study synchronization algorithms for Nyquist-rate LoRa receivers are [40] and [50]. However, both works do not fully model the impact of the STO on sampled signals.

Tapparel et al. present a functional GNU Radio implementation of a LoRa receiver, but no accurate model of a dechirped Nyquist-rate LoRa signal with CFO and STO is discussed [40]. A Nyquist-rate synchronization procedure is briefly described, but the algorithm uses a zero-padded DFT of size $2(N_{\text{pr}} - 2)N$ whose complexity is prohibitive for ULP MCUs.

Bernier et al. present the first low-complexity synchronization algorithm for LoRa, including estimators for the integer and fractional components of both time and frequency offset [50]. The proposed algorithm is inspired from the Semtech patent of [26] and has a very low asymptotic complexity of $\mathcal{O}(N \log N)$. The authors include the fractional timing offset λ_{STO} in their discussion and suggest a method to estimate it, but an exhaustive analytical model of this offset is not examined since they evaluate their proposed synchronization algorithm only under the non-realistic assumption of $\lambda_{\text{STO}} = 0$. We show in this chapter that their proposed synchronization algorithm estimates the fractional STO with a high-variance estimator, which strongly degrades the overall receiver performance.

3.1.2 Objectives

Due to their high computational complexities, the algorithms from [40, 51, 55, 56] are not appropriate for an ULP SDR implementation of a LoRa receiver. Only the algorithm from [50] exhibits a sufficiently low complexity. This algorithm however strongly degrades the receiver performance because of an inefficient estimation of the fractional STO. The goal of this chapter is hence to design an efficient LoRa synchronization algorithm with a sufficiently low computational complexity that allows its straightforward implementation on the ULP SDR plateform introduced in Chapter 4.

The first objective of this chapter is to derive a complete analytical model of a dechirped Nyquist-rate LoRa signal with both a CFO and STO. Next, leveraging this accurate model, an estimator for the fractional STO with a lower variance than the one of *Bernier et al.* needs to be derived. Finally, the accurate signal model and the low-variance fractional STO estimator can be used altogether to design a novel low-complexity synchro-

nization algorithm for Nyquist-rate LoRa receivers.

3.2 Modeling Nyquist-Rate LoRa Signals with STO and CFO

In this section, we seek to accurately model the impacts of an STO τ and a CFO Δf_c on a LoRa signal sampled by a receiver at the Nyquist frequency $f_s = B$. We first derive continuous-time and discrete-time signal models for the upchirps contained in the LoRa preamble, and we subsequently adapt the discrete-time model for the preamble downchirps.

3.2.1 Continuous-time Signal Model

We first analytically model the impact of an STO on the receiver, in the concurrent presence of a CFO. This requires going back to a continuous-time representation since the discrete-time representation in (2.2) is only valid for a perfectly synchronized receiver. Let $u(t)$ be the continuous-time step function and $c(t) = e^{j2\pi t\Delta f_c}$ be the frequency shift of the CFO. For ease of notation, we omit the AWGN in the following derivations. The continuous-time signal $\underline{x}_0(t)$ ($t \in [0, 2T[$) represents two consecutive upchirps in the preamble

$$\underline{x}_0(t) = e^{j2\pi B\left(\frac{t^2}{2T} - \frac{t}{2} - t\cdot u(t-T)\right)}.$$

Let $y(t)$ be the non-synchronized version of $\underline{x}_0(t)$, i.e., after transmission through the channel defined in (3.1), where $t \in [0, T[$. Because $y(t)$ is defined over only one symbol period, its expression can be written as the product of an upchirp $x_0(t)$, the frequency shift $c(t)$ of the CFO and another frequency shift induced by the STO

$$\begin{aligned} y(t) &= hc(t)\underline{x}_0(t-\tau) \\ &= hc(t)e^{j2\pi B\left(\frac{t^2-2t\tau+\tau^2}{2T} - \frac{1}{2}(t-\tau) - (t-\tau)\cdot u(t-\tau-T)\right)} \\ &= x_0(t)c(t)e^{j\left(2\pi B\left(-\frac{t\tau}{T} - (t-\tau)\cdot u(t-\tau-T)\right) + \theta'\right)}, \end{aligned}$$

where θ' is a phase offset representing the constant phase contributions of the channel and the STO. Since this offset does not influence the non-

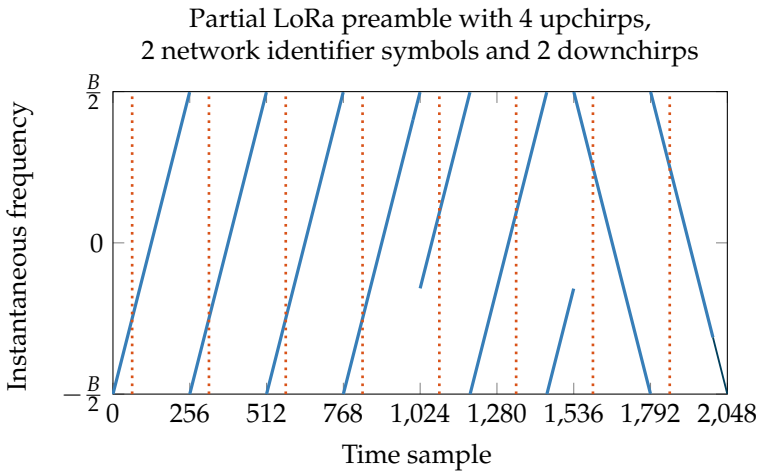


Fig. 3.3 Structure of a preamble for SF $= 8$. The dotted lines delimitate the consecutive windows of N samples received under an integer STO $L_{\text{STO}} = 192$. Only 4 upchirps are shown for the sake of clarity.

coherent ML demodulator of (2.7), we ignore all constant phase offsets in the following derivations.

3.2.2 Discrete-time Signal Model

The continuous-time signal $y(t)$ is sampled at the frequency $f_s = B$. Since a LoRa symbol spans over N samples, a receiver that is not synchronized in time processes windows of samples belonging to two consecutive symbols. For an STO $\tau = (L_{\text{STO}} + \lambda_{\text{STO}})/B$, the first $M = \lceil L_{\text{STO}} + \lambda_{\text{STO}} \rceil$ samples in the window belong to the first symbol and the remaining $N - \lceil L_{\text{STO}} + \lambda_{\text{STO}} \rceil$ samples originate from the second symbol, with $\lceil \cdot \rceil$ being the ceiling function. This behavior is illustrated in Fig. 3.3.

Since the transmitted signal $\underline{x}_0(t)$ consists of two repetitions of the same symbol $x_0(t)$, we can easily analytically model the impact of the CFO and STO in the demodulation stage. Following the definition of Δf_c from (3.2), the discrete-time counterpart of $c(t)$ is

$$c[n] = e^{j2\pi \frac{L_{\text{CFO}} + \lambda_{\text{CFO}}}{N} n}.$$

Let $y[n]$ be the discrete-time signal sampled from $y(t)$. This signal

is first dechirped by the receiver, i.e., multiplied pointwise by $x_0^*[n]$, the complex conjugate of an unmodulated upchirp. The expression of the dechirped signal $\tilde{y}[n]$ is:

$$\tilde{y}[n] = c[n]e^{j2\pi\left(-\frac{L_{\text{STO}}+\lambda_{\text{STO}}}{N}n-(n-L_{\text{STO}}-\lambda_{\text{STO}})u[n-M]\right)},$$

where $u[n]$ is the discrete-time step function. Since $(n - L_{\text{STO}})u[n - M]$ is necessarily integer, this term only induces phases shifts of 2π, and the equation can be simplified to

$$\tilde{y}[n] = c[n]e^{j2\pi\left(-\frac{L_{\text{STO}}+\lambda_{\text{STO}}}{N}n+\lambda_{\text{STO}}u[n-M]\right)}. \tag{3.3}$$

In (3.3), we observe that a window of N points sampled with an STO τ such that $L_{\text{STO}} \neq 0$ and $\lambda_{\text{STO}} \neq 0$ presents an instantaneous phase jump of $2\pi\lambda_{\text{STO}}$ at the sample index M, i.e., at the boundary of the two unmodulated upchirps. By separating the fractional offset λ_{STO} from the integer offset L_{STO}, the impact of λ_{STO} on the dechirped signal can be seen as a single frequency component circularly shifted by M samples

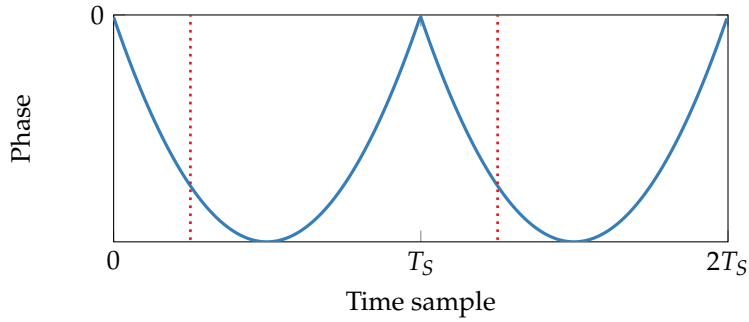
$$\tilde{y}[n] = c[n]e^{-j2\pi\frac{L_{\text{STO}}}{N}n} \cdot \left\langle e^{-j2\pi\frac{\lambda_{\text{STO}}n}{N}}\right\rangle_M$$

where $\langle x[n]\rangle_K$ corresponds to a circular shift of K samples on the signal $x[n]$.

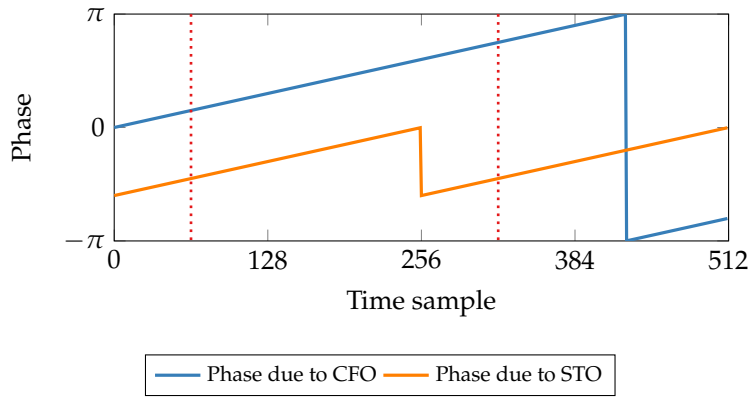
Finally, by expanding the term $c[n]$, we obtain an expression for the dechirped signal depending on all offsets components:

$$\tilde{y}[n] = e^{j2\pi\frac{L_{\text{CFO}}+\lambda_{\text{CFO}}-L_{\text{STO}}}{N}n} \cdot \left\langle e^{-j2\pi\frac{\lambda_{\text{STO}}n}{N}}\right\rangle_M. \tag{3.4}$$

Equation (3.4) shows that the fractional offsets λ_{CFO} and λ_{STO} cannot be considered equivalent. Notably, the CFO adds to the dechirped signal $\tilde{y}[n]$ a residual frequency term continuous across the upchirps boundaries, whereas the fractional STO induces a frequency shift whose phase is reset to $-2\pi\lambda_{\text{STO}}$ after crossing an upchirp boundary. This difference is reflected in Fig. 3.4, where $\lambda_{\text{STO}} = -\lambda_{\text{CFO}}$ such that both phase contributions increase at the same speed. The phase induced by the STO is cyclic between the boundaries of symbols, whereas the phase due to the CFO is continuous across symbols and wraps around $-\pi$ and π. This difference



(a) Unbounded phase of $\bar{x}_0(t)$.



(b) Phase induced by both offsets in $\tilde{y}[n]$.

Fig. 3.4 Phases of two consecutive upchirps from the preamble (a) at the transmitter (b) at the receiver, after sampling at frequency $f_s = \frac{N}{T_s}$ ($N = 256$) and dechirping, with a CFO $L_{\text{CFO}} = 10$, $\lambda_{\text{CFO}} = 0.3$ and an STO $L_{\text{STO}} = 192$, $\lambda_{\text{STO}} = -0.3$. The dotted lines indicate the windows of N samples processed by the receiver.

of behavior is not modeled in *Bernier et al.*, as they consider both fractional offsets to have equivalent effects on the dechirped signal [50].

This absence of equivalence for the fractional offsets leads to a DFT output $Y[i]$ that is no longer similar to a *sinc* function in the concurrent presence of both time and frequency offsets. Let $a[n] \circledast b[n]$ be the convolution between the signals $a[n]$ and $b[n]$. Computing the N-point DFT of (3.4)

yields the following signal

$$Y[i] = N \cdot \Pi(i - L_{\text{CFO}} + L_{\text{STO}}, \lambda_{\text{CFO}}) \circledast \left[e^{-j2\pi \frac{iM}{N}} \cdot \Pi(i, -\lambda_{\text{STO}}) \right]. \quad (3.5)$$

Due to the term $e^{-j2\pi \frac{iM}{N}}$, the magnitude $|Y[i]|$ of the DFT retains a peak around $i = (L_{\text{CFO}} - L_{\text{STO}}) \bmod N$, but the tails of the function do not match the tails of a *sinc* function, as shown in Fig. 3.1. We discuss in Section 3.3 that the dependency of the signal $Y[i]$ on the variable $M = \lceil L_{\text{STO}} + \lambda_{\text{STO}} \rceil$ complicates the estimation of both the STO and CFO.

3.2.3 Differences Between Upchirps and Downchirps

The same development is now used to derive the analytical representation of an unmodulated downchirp before synchronization. It can be shown that the expression of a downchirp $y_d[n]$ sampled at $f_s = B$ with an STO τ and CFO Δf_c corresponds to

$$y_d[n] = c[n] e^{-j2\pi \left(\frac{n^2}{2N} - \frac{L_{\text{STO}} n}{N} - \frac{n}{2} \right)} \cdot \left\langle e^{j2\pi \frac{\lambda_{\text{STO}} n}{N}} \right\rangle_M.$$

After dechirping with an upchirp and expanding $c[n]$, we obtain

$$\tilde{y}_d[n] = e^{j2\pi \frac{L_{\text{CFO}} + L_{\text{STO}} + \lambda_{\text{CFO}}}{N} n} \cdot \left\langle e^{j2\pi \frac{\lambda_{\text{STO}}}{N} n} \right\rangle_M. \quad (3.6)$$

We finally denote the DFT of $\tilde{y}_d[n]$ as $Y_d[i]$, whose expression is given by

$$Y_d[i] = N \cdot \Pi(i - L_{\text{CFO}} - L_{\text{STO}}, \lambda_{\text{CFO}}) \circledast \left[e^{-j2\pi \frac{iM}{N}} \cdot \Pi(i, \lambda_{\text{STO}}) \right]. \quad (3.7)$$

By comparing (3.5) and (3.7), we observe that positive STOs and CFOs shift the Kronecker delta originally at $i = s$ in the opposite direction for upchirps, but in the same directions for downchirps. This property is needed to separately estimate the integer offsets L_{CFO} and L_{STO}, as first discussed in [26, 50] and explained in the following section under our detailed model.

3.3 Interdependencies between Integer and Fractional Offsets

Due to the complexity of the DFT output $Y[i]$ given in (3.5), deriving estimators of the STO and CFO when $L_{\text{CFO}}, L_{\text{STO}}, \lambda_{\text{CFO}}, \lambda_{\text{STO}} \neq 0$ is challenging. The difficulty of the problem can however be reduced by partially correcting the CFO. In this section, we show that estimating λ_{CFO} can be achieved independently of the presence of the other offsets, but that the estimations of λ_{STO}, L_{STO} and L_{CFO} are all intertwined.

After correction of the fractional CFO λ_{CFO}, the *sinc* function $\Pi(i - L_{\text{CFO}} + L_{\text{STO}}, \lambda_{\text{CFO}})$ from (3.5) becomes a Kronecker delta $\delta[i - L_{\text{CFO}} + L_{\text{STO}}]$. Consequently, the DFT of the partially corrected signal has a simpler expression without any convolution

$$Y[i] = N \cdot e^{-j2\pi \frac{iM}{N}} \cdot \Pi(i - L_{\text{CFO}} + L_{\text{STO}}, -\lambda_{\text{STO}}) + Z[i], \tag{3.8}$$

where $Z[i]$ is the DFT of the dechirped AWGN $\tilde{z}[n]$. The energy of the signal in (3.8) is now spread in multiple DFT bins only because of the presence of the fractional STO.

In this section, we derive a new low-variance estimator for λ_{STO} for the signal model of (3.8). Moreover, we show that the correction of λ_{STO} before the estimation of L_{CFO} and L_{STO} is crucial for the correct estimation of these integer offsets, a fact that has been neglected in the literature [40, 50]. To simplify the upcoming equations, we first discuss the estimation and correction of the fractional CFO as the initial step performed by the receiver. All subsequent steps are discussed after this partial correction, thus with $\lambda_{\text{CFO}} = 0$.

3.3.1 Estimating λ_{CFO} is Independent of the Other Offsets

We showed in Section 3.2 that the fractional CFO and STO have distinct effects on the DFT output $Y[i]$. The former induces a linear phase term that is continuous across symbols, whereas the phase due to the latter is cyclic with period N. This cyclical property allows a receiver to estimate λ_{CFO} independently of λ_{STO}. Let $\tilde{y}^l[n]$ be the l-th window of N samples dechirped by the receiver, and $Y^l[i]$ its N-point DFT. Starting from the time-domain representation of unmodulated upchirps from (3.4), and by ignoring the AWGN for now, all upchirps in the preamble are identical, up to a phase

difference of $2\pi\lambda_{\text{CFO}}$

$$\begin{aligned}\tilde{y}^l[n] &= e^{j2\pi\frac{L_{\text{CFO}}+\lambda_{\text{CFO}}}{N}(n+lN)} \cdot \left\langle e^{-j2\pi\frac{L_{\text{STO}}+\lambda_{\text{STO}}}{N}n}\right\rangle_M \\ &= e^{j2\pi\lambda_{\text{CFO}}}\tilde{y}^{l-1}[n].\end{aligned}$$

By linearity of the DFT, the same result holds in the frequency domain: $Y^l[i] = e^{j2\pi\lambda_{\text{CFO}}} \cdot Y^{l-1}[i]$. It has been shown by *Bernier et al.* that this property can be leveraged to estimate λ_{CFO} by multiplying a DFT bin $Y^l[i]$ with $\overline{Y}^{l-1}[i]$, the conjugate of the same bin from the previous upchirp, and taking the phase of the product

$$\widehat{\lambda}_{\text{CFO}} = \frac{1}{2\pi}\,\text{angle}\left(\sum_{l=2}^{N_C}\sum_{p=-2}^{2} Y^l[q+p]\cdot\overline{Y}^{l-1}[q+p]\right), \tag{3.9}$$

where $q = \arg\max_i |Y^l[i]|$ [50]. To improve the accuracy of the estimate, the five DFT bins of each symbol centered around the bin of maximum height are used. In practice, these DFT bins near $i = (L_{\text{CFO}} - L_{\text{STO}}) \bmod N$ benefit from the processing gain of the modulation, and thus exhibit a higher SNR. This operation is repeated over N_C pairs of successive upchirps, and all intermediate products $Y^l[q+p]\cdot\overline{Y}^{l-1}[q+p]$ are summed up before computing the final estimate $\widehat{\lambda}_{\text{CFO}}$.

3.3.2 Estimating L_{CFO} and L_{STO} Depends on λ_{STO}

In Section 3.2, we showed that an STO has an opposite effect on upchirps and downchirps. Following (3.5) and (3.7), in the absence of AWGN and after correction of λ_{CFO}, the non-coherent ML demodulation decision of (2.7) for both an upchirp and a downchirp yields

$$s_{\text{up}} = \arg\max_i |Y[i]| = (L_{\text{CFO}} - L_{\text{STO}}) \bmod N, \tag{3.10}$$

$$s_{\text{down}} = \arg\max_i |Y_d[i]| = (L_{\text{CFO}} + L_{\text{STO}}) \bmod N. \tag{3.11}$$

By respectively adding and subtracting s_{up} and s_{down}, L_{CFO} and L_{STO} can be estimated separately to some extent. The estimation of L_{CFO} is first obtained by adding s_{up} and s_{down}, and correcting for the modulo effect

from the DFT [26], [50]:

$$\widehat{L}_{\text{CFO}} = \frac{1}{2}\Gamma_N\Big[(s_{\text{up}} + s_{\text{down}}) \bmod N\Big], \tag{3.12}$$

where $\Gamma_N[i] = \begin{cases} i & \text{for } 0 \leq i < \frac{N}{2}, \\ i - N & \text{for } \frac{N}{2} \leq i < N. \end{cases}$

Once $\widehat{L}_{\text{CFO}}$ is known, the integer STO can be estimated with the following estimator

$$\widehat{L}_{\text{STO}} = (\widehat{L}_{\text{CFO}} - s_{\text{up}}) \bmod N. \tag{3.13}$$

As a consequence of the wrappings induced by the DFT, it is impossible to recover both integer components in the range $[0, N-1]$ without ambiguity. Owing to the fact that the receiver may start acquiring a symbol at any time of its transmission, i.e., $0 \leq L_{\text{STO}} < N$, and that Δf_c may be positive or negative, the estimation of L_{CFO} is thus bounded to $\left[-\frac{N}{4}, \frac{N}{4} - 1\right]$ [50].

Since the estimators (3.12) and (3.13) rely on the demodulation stage provided by (2.7), the presence of a potential fractional STO increases the probability of incorrectly demodulating s_{up} and s_{down}. To the contrary of the fractional CFO which can be estimated and corrected independently of the other offsets, the estimation of the integer STO and CFO is sensitive to the presence of λ_{STO}. To decrease the probability of a synchronization failure, it is thus crucial to estimate and correct the fractional STO, even partially, before estimating the integer offsets. We note that in the literature [40, 50], the fractional STO λ_{STO} is never corrected before the integer offsets, which strongly increases the probability of wrongly estimating the integer offsets and thus of experiencing a synchronization failure. None of the aforementioned works shows the importance of correcting λ_{STO} before the integer offsets, since none of them presents results for the overall packet error rate in the presence of all integer and fractional offsets. In particular, [40] shows good conditional bit error rates (i.e., only for the packets that managed to synchronize correctly), but the authors do not indicate how many of the overall transmitted packets actually managed to synchronize. Moreover, *Bernier et al.* simulate results only under the assumption of $\lambda_{\text{STO}} = 0$ [50].

3.3.3 Estimating the Fractional STO λ_{STO}

Many estimators in the literature have been designed to estimate a fractional frequency ξ of a *sinc* signal $\Pi(i, \xi)$ with AWGN (e.g., [57, 58, 59]). However, due to the additional phase term $e^{-j2\pi\frac{iM}{N}}$ in (3.8), the expression of the signal $Y[i]$ after correction of λ_{CFO} is not a strict *sinc* function. Estimating λ_{STO} with usual estimators therefore requires *a priori* knowledge of $M = \lceil L_{\text{STO}} + \lambda_{\text{STO}} \rceil$, which is unknown until the reception of a downchirp.

Yet, as previously discussed, the fractional STO should preferably be corrected before estimating L_{STO}. To avoid the complex joint estimation of L_{STO} and λ_{STO} over several symbols, we suggest instead to design an estimator $\widehat{\lambda}_{\text{STO}}$ that relies on a prior estimate $\widehat{M}$. This estimator $\widehat{\lambda}_{\text{STO}}$ will then be used iteratively in the synchronization algorithm, with increasingly accurate estimates of M.

Regarding first the specific case when $M = 0$ (i.e., when $Y[i]$ is a *sinc* function with AWGN), the following estimator from [58] is valid

$$\widehat{\lambda}_{\text{STO}} = -\operatorname{Re}\left[\frac{Y[q+1] - Y[q-1]}{2Y[q] - Y[q+1] - Y[q-1]}\right], \tag{3.14}$$

where $q = \arg\max_i |Y[i]|$. We now extend (3.14) for the general case when the receiver is affected by an STO such that $M \neq 0$. This requires correcting the phases of $Y[i]$ to remove the phase term $e^{-j2\pi\frac{iM}{N}}$. By assuming the knowledge of a prior estimate of L_{STO} (e.g., in an iterative scheme as later described), and using the approximation $\widehat{M} \approx L_{\text{STO}}$ to correct the phases of $Y[i]$, we obtain an estimator relying on the complex values $Y[i]$ of a demodulated upchirp:

$$\widehat{\lambda}_{\text{STO}} = -\operatorname{Re}\left[\frac{e^{j2\pi\frac{\widehat{M}}{N}}Y[q+1] - e^{-j2\pi\frac{\widehat{M}}{N}}Y[q-1]}{2Y[q] - e^{j2\pi\frac{\widehat{M}}{N}}Y[q+1] - e^{-j2\pi\frac{\widehat{M}}{N}}Y[q-1]}\right]. \tag{3.15}$$

The precision of this estimator depends on the accuracy of the estimate $\widehat{M}$. The accuracy of the estimate $\widehat{\lambda}_{\text{STO}}$ can also be improved by averaging the complex values $\{Y[q-1], Y[q], Y[q+1]\}$ over several upchirps.

To estimate the fractional STO without an estimate $\widehat{M}$, i.e., without having to wait on a downchirp, another estimator $\widehat{\lambda}_{\overline{\text{STO}}}$ that relies only on the magnitudes $|Y[i]|$ is proposed in [26] and used in [50]. This estimator lever-

ages the intermediate function

$$T(\lambda_{\text{STO}}) = \frac{|\Pi(1, \lambda_{\text{STO}})| - |\Pi(-1, \lambda_{\text{STO}})|}{|\Pi(0, \lambda_{\text{STO}})|}.$$

After demodulating an upchirp, the receiver estimates λ_{STO} by inverting $T(\lambda_{\text{STO}})$ as follows:

$$\widehat{\lambda}_{\overline{\text{STO}}} = T^{-1}\left(\frac{|Y[q+1]| - |Y[q-1]|}{|Y[q]|}\right), \tag{3.16}$$

with $q = \arg\max_i |Y[i]|$. Averaging the magnitudes $|Y[q]|$ of consecutive upchirps also enhances the accuracy of (3.16), a fact that is however not evaluated in the aforementioned works. Although the estimator $\widehat{\lambda}_{\overline{\text{STO}}}$ of (3.16) is independent of L_{STO}, this estimator presents a very high variance due to the derivative of $T^{-1}(x)$ being close to 0 for small values of λ_{STO} [26]. We show in Section 3.5 that the high variance of this estimator degrades the receiver performance when used naively in a symbol-by-symbol tracking loop as proposed in [50].

The Root Mean Square Errors (RMSE) of all three estimators (3.9), (3.15) and (3.16) have been obtained for SF $= 8$ using Monte-Carlo simulations, and are presented in Fig. 3.5. The estimators $\widehat{\lambda}_{\text{STO}}$ and $\widehat{\lambda}_{\overline{\text{STO}}}$ are evaluated using upchirps received with an STO τ uniformly distributed in the range $[0, \frac{N}{B}[$. The simulation assumes that the receiver has an ideal knowledge of the integer offset L_{STO} to estimate the fractional STO $\widehat{\lambda}_{\text{STO}}$. To evaluate the RMSE of the estimator $\widehat{\lambda}_{\text{CFO}}$ from (3.9), a fractional CFO λ_{CFO} uniformly distributed in the range $]-0.5, 0.5]$ is added to the delayed signal.

We observe in Fig. 3.5 that, for a single upchirp and an SNR of -9 dB[1], the proposed estimator $\widehat{\lambda}_{\text{STO}}$ is one order of magnitude more accurate than the estimator $\widehat{\lambda}_{\overline{\text{STO}}}$ from [26]. We also point out that the estimator of the fractional CFO exhibits a lower RMSE than both estimators of the fractional STO. The estimators are also evaluated with two or three preamble upchirps. The averaging of these upchirps is performed differently depending on the estimator, as previously described for each estimator. We note that using three upchirps instead of one improves the precision of the estimates by a factor 1.9 for the proposed estimator $\widehat{\lambda}_{\text{STO}}$, but only by a factor 1.3 for the estimator $\widehat{\lambda}_{\overline{\text{STO}}}$ used by *Bernier et al*.

[1]This SNR corresponds to a symbol error rate of 10^{-4} for a perfectly synchronized receiver using the non-coherent ML demodulator of (2.7) and uncoded LoRa symbols.

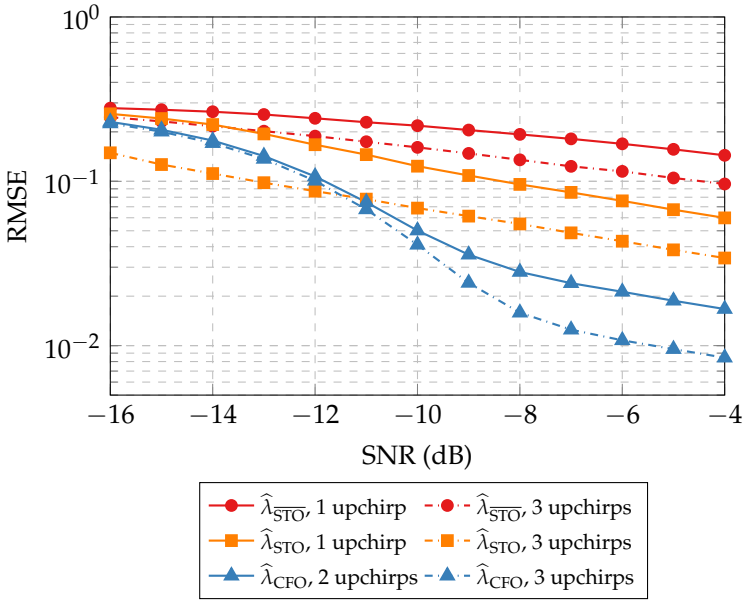


Fig. 3.5 RMSE of the three studied fractional offsets estimators, for SF = 8 and one or several upchirps.

3.4 Designing a Nyquist-Rate Synchronization Algorithm

We showed in the previous section that the estimation of the offsets components L_{STO}, L_{CFO} and λ_{STO} are intertwined. Only the estimation of the fractional CFO λ_{CFO} can be performed independently of the other offsets. Moreover, since the estimation of the integer offsets requires the demodulation of a downchirp, which is only available at the end of the preamble, the interdependencies between offsets complexify the design of a synchronization scheme. To solve this issue while keeping the implementation of the receiver as simple as possible, we now design an iterative low-complexity synchronization algorithm capable of accurately estimating all offsets components. The proposed synchronization algorithm is then compared to the synchronization algorithm from *Bernier et al.* [50]. Both algorithms have as input a non-synchronized LoRa packet sampled at the Nyquist-rate $f_s = B$ that follows the model described in Section 3.2.

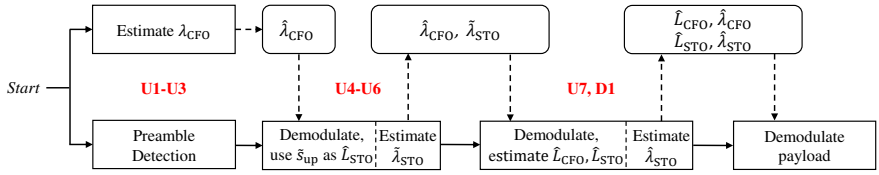


Fig. 3.6 Flowgraph of the synchronization algorithm. For each step, the related parts of the preamble are indicated in red. Un refers to the n-th full upchirp in the preamble, and D1 to the single full downchirp. Dashed arrows indicate the CFO and STO components estimated or corrected by each stage.

3.4.1 Proposed Synchronization Algorithm

The proposed algorithm, whose pseudocode is given in Algorithm 1, leverages sequentially the different parts of the preamble, by processing a single symbol at a time. The procedure uses the estimators derived in Section 3.3, and estimates the fractional STO λ_{STO} iteratively in two steps. The algorithm is split in three stages, as shown in Fig. 3.6: first the estimation of λ_{CFO}, then a preliminary estimation of the fractional STO, and finally the definitive estimation of L_{CFO}, L_{STO} and λ_{STO}.

Preamble Detection and Estimation of the Fractional CFO

Before delving into the synchronization algorithm itself, the receiver first implements a preamble detection scheme. It is well-known that the presence of several consecutive upchirps can be used to detect the start of a frame [31, 51]. A preamble is detected by a receiver when the same symbol s, or its neighbors $s \pm 1$, are consecutively demodulated [40]. Let $y^l[n]$ be the l-th window of N samples processed by the receiver. Upon a preamble detection on three consecutive upchirps $y^l[n]$, $y^{l-1}[n]$ and $y^{l-2}[n]$, these symbols are also used to compute the estimate $\widehat{\lambda}_{CFO}$. This estimate is used to correct the fractional CFO when demodulating the remaining symbols of the preamble.

First Correction of the Fractional STO

The next three upchirps of the preamble are dedicated to the estimation of the fractional STO. The demodulated value $\tilde{s}_{up}$ of the fourth upchirp of the preamble is used as an initial estimation of s_{up}. A temporary estimate $\tilde{\lambda}_{STO}$ of the fractional STO is then computed using the estimator $\widehat{\lambda}_{STO}$ from (3.15) with the averaged DFTs of the three upchirps and $\tilde{s}_{up}$ as an approximation

Algorithm 1 Proposed synchronization algorithm

$l = 0$

Preamble Detection and Fract. CFO Estimation:

while preamble not detected **do**

  $l = l + 1$

  $Y^l[i] = \text{DFT}\left(y^l[n] \odot x_0^*[n]\right)$, $\hat{s}^l = \arg\max_i \left|Y^l[i]\right|$

  $z^l = \sum_{p=-2}^{2} Y^l[\hat{s}^l + p] \cdot \overline{Y}^{l-1}[\hat{s}^l + p]$

  PreambleDetection $\left(\hat{s}^l, \hat{s}^{l-1}, \hat{s}^{l-2}\right)$

end while

$\widehat{\lambda}_{\text{CFO}} = \frac{1}{2\pi} \text{angle}\left(\sum_{p=0}^{1} z^{l-p}\right)$

First Correction of the Fractional STO:

for $m = 1$ to 3 **do**

  $Y^{l+m}[i] = \text{DFT}\left(y^{l+m}[n] \odot x_0^*[n] \odot e^{-j2\pi\widehat{\lambda}_{\text{CFO}}\frac{mN+n}{N}}\right)$

end for

$\tilde{s}_{\text{up}} = \arg\max_i \left|Y^{l+1}[i]\right|$, $\widetilde{M} = \left(N - \tilde{s}_{\text{up}}\right) \bmod N$

$Y^{\text{avg}}[i] = \sum_{m=1}^{3} Y^{l+m}[i]$

Compute $\tilde{\lambda}_{\text{STO}}$ with $\{Y^{\text{avg}}_{\tilde{s}_{\text{up}}-1}, Y^{\text{avg}}_{\tilde{s}_{\text{up}}}, Y^{\text{avg}}_{\tilde{s}_{\text{up}}+1}\}$, $\widetilde{M}$ using (3.15)

Realign receiver by $\tilde{\lambda}_{\text{STO}}$ samples

Final Synchronization Stage:

$Y^{l+4}[i] = \text{DFT}\left(y^{l+4}[n] \odot x_0^*[n] \odot e^{-j2\pi\widehat{\lambda}_{\text{CFO}}\frac{n}{N}}\right)$

$Y^{l+8}[i] = \text{DFT}\left(y^{l+8}[n] \odot x_0[n] \odot e^{-j2\pi\widehat{\lambda}_{\text{CFO}}\frac{n}{N}}\right)$

$\hat{s}_{\text{up}} = \arg\max_i \left|Y^{l+4}[i]\right|$, $\hat{s}_{\text{down}} = \arg\max_i \left|Y^{l+8}[i]\right|$

Compute $\widehat{L}_{\text{CFO}}$, $\widehat{L}_{\text{STO}}$ with $\hat{s}_{\text{up}}$, $\hat{s}_{\text{down}}$ using (3.12) and (3.13)

Compute $\widehat{\lambda}_{\text{STO}}$ with $\{Y^{\text{avg}}_{\hat{s}_{\text{up}}-1}, Y^{\text{avg}}_{\hat{s}_{\text{up}}}, Y^{\text{avg}}_{\hat{s}_{\text{up}}+1}\}$ and $\widehat{M} = \left(N - \widehat{L}_{\text{STO}}\right) \bmod N$ using (3.15)

Correct carrier frequency by $B/N\left(\widehat{L}_{\text{CFO}} + \widehat{\lambda}_{\text{CFO}}\right)$ Hz

Realign receiver by $\widehat{L}_{\text{STO}} + \widehat{\lambda}_{\text{STO}} - \tilde{\lambda}_{\text{STO}} + N/4$ samples

---

Processing of the Payload:

for $m = l + 5$ to $l + N_P + 4$ **do**

  $Y^m[i] = \text{DFT}\left(y^m[n] \odot x_0^*[n]\right)$, $\hat{s}_m = \arg\max_i |Y^m[i]|$

end for

of L_{STO}, i.e., $\widetilde{M} = N - \tilde{s}_{\text{up}}$. The receiver subsequently realigns itself in time to correct the fractional STO using $\tilde{\lambda}_{\text{STO}}$. This first correction of the fractional STO increases the probability of correctly demodulating the symbols s_{up} and s_{down} in the final stage of the algorithm. In a practical receiver, if this correction cannot be applied instantly, only two upchirps should be averaged in $Y^{\text{avg}}[i]$ such that the correction can be performed during the reception of the upchirp with index $l+3$.

Since the estimate $\widetilde{M}$ used in the computation of $\tilde{\lambda}_{\text{STO}}$ is likely to be close to $L_{\text{STO}} - L_{\text{CFO}}$ instead of $M = \lceil L_{\text{STO}} + \lambda_{\text{STO}} \rceil$, the accuracy of the estimate $\tilde{\lambda}_{\text{STO}}$ depends on L_{CFO}. We show in Section 3.5 that large offset values L_{CFO} only slightly increase the probability of packet synchronization failure, due to the precision loss of the estimate $\tilde{\lambda}_{\text{STO}}$.

Finally, as the definitive estimation of $\widehat{\lambda}_{\text{STO}}$ using (3.15) can only take place once L_{STO} is estimated, the averaged DFT bins are stored in memory until the reception of a downchirp.

Final Synchronization Stage

The final entire upchirp and the single entire downchirp are used to estimate $\hat{s}_{\text{up}}$ and $\hat{s}_{\text{down}}$. With these values, the receiver then uses (3.12) and (3.13) to estimate L_{CFO} and L_{STO}, respectively. A definitive estimation of λ_{STO} is afterwards computed using (3.15), with $\widehat{M} = N - \widehat{L}_{\text{STO}}$ and the DFT bins stored in memory. The receiver performs the final correction of the CFO and STO using the above estimates and proceeds to the demodulation of the payload.

Processing of the Payload

After the final synchronization stage, the receiver proceeds to the demodulation of the N_P payload symbols as explained in Section 2.1.

3.4.2 Synchronization Algorithm from *Bernier et al.*

We now briefly introduce the synchronization algorithm presented in *Bernier et al.* [50], whose pseudocode is given in Algorithm 2 to ease the comparison with our algorithm. In [50], the preamble detection and the estimation of the fractional CFO are performed using all the upchirps of the preamble, except the last one. The fractional CFO is then corrected only for the remainder of the preamble. The final upchirp and one single entire downchirp are used to obtain the symbols $\hat{s}_{\text{up}}$ and $\hat{s}_{\text{down}}$, respectively. The receiver subsequently estimates L_{CFO} and L_{STO} using $\hat{s}_{\text{up}}$ and

Algorithm 2 Synchronization algorithm from *Bernier et al.*

$l = 0$

Preamble Detection and Fract. CFO Estimation:

while preamble not detected **do**

  $l = l + 1$

  $Y^l[i] = \mathrm{DFT}\left(y^l[n] \odot x_0^*[n]\right)$, $\hat{s}^l = \arg\max_i \left|Y^l[i]\right|$

  $z^l = \sum_{p=-2}^{2} Y^l[\hat{s}^l + p] \cdot \overline{Y}^{l-1}[\hat{s}^l + p]$

  $\mathrm{PreambleDetection}\left(\hat{s}^l, \hat{s}^{l-1}, \ldots, \hat{s}^{l-5}\right)$

end while

$\widehat{\lambda}_{\mathrm{CFO}} = \frac{1}{2\pi} \mathrm{angle}\left(\sum_{p=0}^{4} z^{l-p}\right)$

Coarse Synchronization Stage:

$Y^{l+1}[i] = \mathrm{DFT}\left(y^{l+1}[n] \odot x_0^*[n] \odot e^{-j2\pi\widehat{\lambda}_{\mathrm{CFO}}\frac{n}{N}}\right)$

$Y^{l+5}[i] = \mathrm{DFT}\left(y^{l+5}[n] \odot x_0[n] \odot e^{-j2\pi\widehat{\lambda}_{\mathrm{CFO}}\frac{n}{N}}\right)$

$\hat{s}_{\mathrm{up}} = \arg\max_i \left|Y^{l+1}[i]\right|$, $\hat{s}_{\mathrm{down}} = \arg\max_i \left|Y^{l+5}[i]\right|$

Compute $\widehat{L}_{\mathrm{CFO}}$, $\widehat{L}_{\mathrm{STO}}$ with $\hat{s}_{\mathrm{up}}$, $\hat{s}_{\mathrm{down}}$ using (3.12) and (3.13)

Correct carrier frequency by $B/N\left(\widehat{L}_{\mathrm{CFO}} + \widehat{\lambda}_{\mathrm{CFO}}\right)$ Hz

Realign receiver by $\widehat{L}_{\mathrm{STO}} + N/4$ samples

Compute $\widehat{\lambda}_{\overline{\mathrm{STO}}}^{l+5}$ with $\{Y^{l+5}[\hat{s}_{\mathrm{down}} + 1], Y^{l+5}[\hat{s}_{\mathrm{down}}], Y^{l+5}[\hat{s}_{\mathrm{down}} - 1]\}$ using (3.16)

Processing of the Payload:

for $m = l + 6$ to $l + N_P + 5$ **do**

  $\Lambda_{\mathrm{STO}}^m = \frac{1}{m-l-5} \sum_{k=l+5}^{m-1} \Lambda_{\mathrm{STO}}^k + \widehat{\lambda}_{\overline{\mathrm{STO}}}^k$

  Realign receiver by $\Lambda_{\mathrm{STO}}^m - \Lambda_{\mathrm{STO}}^{m-1}$ samples

  $Y^m[i] = \mathrm{DFT}\left(y^m[n] \odot x_0^*[n]\right)$, $\hat{s}_m = \arg\max_i |Y^m[i]|$

  Compute $\widehat{\lambda}_{\overline{\mathrm{STO}}}^m$ with $\{Y^m[\hat{s}_m - 1], Y^m[\hat{s}_m], Y^m[\hat{s}_m + 1]\}$ using (3.16)

end for

$\hat{s}_{\text{down}}$, and performs a coarse synchronization which corrects the carrier frequency with $\widehat{L}_{\text{CFO}} + \widehat{\lambda}_{\text{CFO}}$ and the integer STO using $\widehat{L}_{\text{STO}}$, but not the fractional STO.

The fractional STO is estimated using a tracking loop that relies on the estimator $\widehat{\lambda}_{\overline{\text{STO}}}$ of (3.16). This tracking loop is enabled after the demodulation of the downchirp, i.e., during the payload, and performs a symbol-by-symbol estimation of λ_{STO}. Since the STO is considered constant, its correction for the m-th symbol can be improved by averaging all the previous estimates of the fractional STO. This averaging is not discussed in [50], however we include it in the algorithm for a fair comparison, since it improves the performance of their receiver.

Unlike the proposed algorithm, the algorithm from [50] does not leverage the upchirps of the preamble to estimate and correct the fractional STO before the estimation of the integer offsets. As a consequence, the presence of the uncorrected λ_{STO} increases the probability of a wrong demodulation of the symbols $\hat{s}_{\text{up}}$ and $\hat{s}_{\text{down}}$. Since these symbols are used to compute L_{STO} and L_{CFO}, erroneous estimations of the integer offsets frequently arise and strongly deteriorate the performance of the receiver, as shown in the following section. We also show next that the usage of the high-variance estimator from (3.16) in a symbol-by-symbol tracking loop further degrades the performance of their receiver.

3.5 Performance Evaluation

In this section, we assess the performance of our proposed synchronization algorithm and we compare it to the performance of the only other Nyquist-rate low-complexity algorithm in the literature, i.e., the algorithm from *Bernier et al*. Simulation results under realistic parameters show that our algorithm results in a significant improvement of the packet error rate compared to *Bernier et al*. In addition, we analyze the impact of large CFO values on the proposed algorithm. Finally, the error rates of both receivers for coded LoRa packets are presented and compared to the performance of a theoretical perfectly synchronized receiver.

f_c \ B	125kHz	250kHz	500kHz
433 MHz	$\pm 0.07N$	$\pm 0.03N$	$\pm 0.02N$
868 MHz	$\pm 0.14N$	$\pm 0.07N$	$\pm 0.03N$

Table 3.1 Maximum L_{CFO} offset observed by a LoRa receiver with a crystal precision of 20 ppm for the different standardized bandwidths and ISM carrier frequencies.

3.5.1 Simulation Parameters

For both our synchronization algorithm and the algorithm in [50], the receiver samples a signal of bandwidth $B = 125$ kHz with AWGN at a carrier frequency of $f_c = 868$ MHz. The bandwidth of the signal does not impact the performance of the receiver, except for the range of the CFO, as explained below. All error rates are evaluated through Monte-Carlo simulations with 10^5 trials per SNR level. Each packet starts with a preamble, as described in Section 2.1. The payload consists of $N_P = 28$ modulated LoRa symbols, either coded or uncoded. The packets reach the receiver through the channel defined in (3.1). The preamble detection schemes are considered ideal for both algorithms, i.e., the preamble of a packet is always detected, and do not impact the simulation results. For the simulations with coding (CR $= 4/7$), the BICM scheme described in Section 2.1 is simulated and the receiver implements a deinterleaver and a hard Hamming decoder after the demodulation stage.

To compare the two algorithms in a realistic simulation setup, we allow the sampling time offset to take values uniformly in the range $\tau \in \left[0, \frac{N}{B}\right[$. The receivers sample the signal at an oversampled frequency $f_s' = RB$, with $R = 10$, to allow a correction of the fractional time offset. For both algorithms, the oversampled signal is decimated by R before the dechirping stage, i.e., synchronization and demodulation are done at the Nyquist frequency $f_s = B$. When a receiver realigns itself in time, it selects the decimation index among the R available that is the closest to the fractional STO estimated by the synchronization algorithm.

As explained in Section 3.3, and similarly to the SX1276 transceiver [38], the LoRa receiver studied in this work can only estimate a CFO $\Delta f_c \in \left[-\frac{B}{4}, \frac{B}{4}\right]$. Considering a typical carrier frequency $f_c = 868$ MHz and the smallest bandwidth $B = 125$ kHz, only CFOs Δf_c up to 36 ppm can be es-

timated. Low-power IoT end nodes are usually designed with a low-cost bill of materials, and the crystal oscillators used for the carrier frequency synthesis often have limited accuracy. For instance, the SX1276 receiver embeds a crystal with a 20 ppm precision [38], whereas gateways use more accurate temperature-controlled oscillators. Table 3.1 presents the maximum L_{CFO} observable among all possible carrier frequencies and bandwidths for a crystal precision of 20 ppm. The chosen simulation parameters $B = 125$ kHz and $f_c = 868$ MHz are hence the ones inducing the greatest integer CFO values. In our simulations, the CFO is always uniformly distributed in the range $\Delta f_c \in [-20, 20]$ ppm, unless specified otherwise. The estimated CFO is corrected by shifting the sampled signal in frequency, as explained for each studied algorithm in Section 3.4.

3.5.2 Analysis of the Packet Error Rates for SF = 8

Fig. 3.7 shows the overall packet error rate (PER) for our synchronization scheme and the algorithm from *Bernier et al.* [50]. The transmitted packets use SF $= 8$ and are uncoded. The overall PER (solid lines) is obtained by applying the synchronization algorithm (either the proposed or the one in [50]) to the received packets, estimating and correcting all the time and frequency offsets as each algorithm determines, and finally deciding on the survival of the packet by checking if all symbols of the payload were demodulated correctly.

We observe that the algorithm of [50] results in a degraded overall PER performance (solid orange curve) at high SNRs. On the contrary, we observe that the PER performance of the proposed algorithm (solid blue line) is very good and close to the performance of a perfectly synchronized receiver (thick gray line). For example, to attain an overall PER of 10^{-2}, we observe that our receiver requires at most 0.5 dB higher SNR than a perfectly synchronized receiver.

To provide an in-depth understanding of the performance of both algorithms, we deconstruct the overall packet error rate into two error events. The first error event corresponds to a synchronization failure which results in a complete packet loss. This happens when, after estimation and correction, the residual offsets prevent the receiver from correctly demodulating any symbol in the payload, even in the absence of noise [31]. The second error event, i.e., a payload error, arises when the receiver manages to correctly synchronize to the packet, but experiences demodulation errors

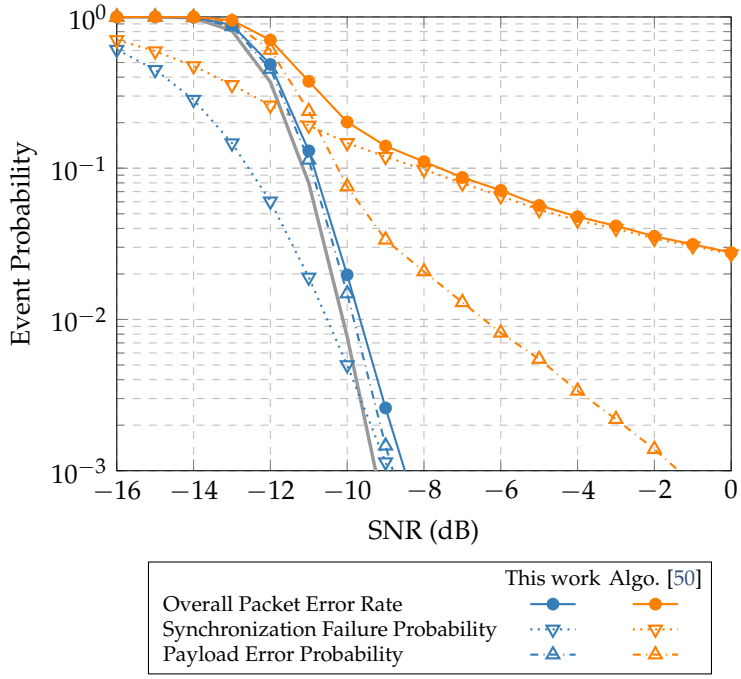


Fig. 3.7 PERs of the proposed algorithm and the algorithm from [50] for uncoded LoRa with SF $= 8$ and $N_P = 28$ payload symbols. The thick gray line shows the PER for a perfectly synchronized receiver.

in the payload due to small residual offsets and AWGN. We note that the overall PER shown in Fig. 3.7 for both algorithms is agnostic to the definition of these two error events.

Regarding the first error event, we consider that the algorithm is effectively unable to synchronize to a packet if, after the estimation and correction of all four integer and fractional offsets, the demodulated symbols obtained using (2.7) are all wrong even in the absence of AWGN. This happens when $|\widehat{L}_{\text{CFO}} + \widehat{\lambda}_{\text{CFO}} - \frac{N\Delta f_c}{B} - \widehat{L}_{\text{STO}} - \widehat{\lambda}_{\text{STO}} + B\tau| > 0.5$, since a residual total offset greater than 0.5, i.e., the middle point between two samples, shifts the position of a symbol s in frequency to a DFT bin index other than s. A shift greater than 0.5 thus prevents the receiver from correctly demodulating any symbol in the payload, even in very high SNR regions or in the total absence of noise [31]. We note that the above definition, which treats time and frequency offsets in a unified manner, inherently

takes into account the case where a residual integer CFO compensates for a residual integer STO. Such a case results, in the absence of AWGN, in a correct demodulation of the payload and is hence not considered as a synchronization failure. It however introduces a small inter-symbol interference (ISI) which degrades the performance during the demodulation of the payload [31].

The second error event concerns demodulation errors in the payload that are caused by AWGN. As explained in Section 3.2, residual fractional offsets left after the synchronization stage increase the probability of demodulation errors. Let $\lambda_{\text{res}} = \widehat{L}_{\text{CFO}} + \widehat{\lambda}_{\text{CFO}} - \frac{N\Delta f_c}{B} - \widehat{L}_{\text{STO}} - \widehat{\lambda}_{\text{STO}} + B\tau$ be the total residual fractional offset that remains after the synchronization stage, where $\lambda_{\text{res}} < 0.5$. When both integer offsets are successfully corrected, and following (3.5), the DFT of a modulated symbol s becomes $Y[i] = N \cdot \Pi(i - s, \lambda_{\text{res}}) + Z[i]$, i.e., the impact on the demodulation of both residual fractional offsets becomes indeed equivalent and equal to λ_{res}. At a given SNR and spreading factor, the payload error probability is strongly influenced by the distribution of the residual values λ_{res} left by the synchronization stage.

In Fig. 3.7, we observe that the synchronization failure probability is limiting the performance of the algorithm of [50], especially in high SNR regions. This is because *Bernier et al.* does not correct the fractional STO during the preamble, but only during the payload. The uncorrected fractional STO increases the probability of wrongly demodulating the preamble symbols, and thus of incorrectly estimating the integer offsets. On the contrary, our proposed algorithm, which estimates and corrects the fractional STO during the preamble, accurately estimates both integer offsets, even for low SNR values, resulting in a synchronization failure probability that does not limit the overall performance.

Moreover, the receiver from [50] also exhibits higher payload error probabilities than our receiver. This stems from the fact that the function used by the estimator of the fractional STO has a horizontal inflection point at the origin [26]. As such, the inverse function has a vertical tangent at the origin (see also Fig. 9 of [50]) that increases the variance when estimating very small offsets, leading therefore to instabilities in the estimation of the fractional STO in the tracking loop [26]. On the other hand, thanks to the low-variance estimator $\widehat{\lambda}_{\text{STO}}$ of (3.15) that is using three upchirps in the preamble, our receiver is capable of accurately correcting the fractional STO, a fact that results in much improved payload error probability.

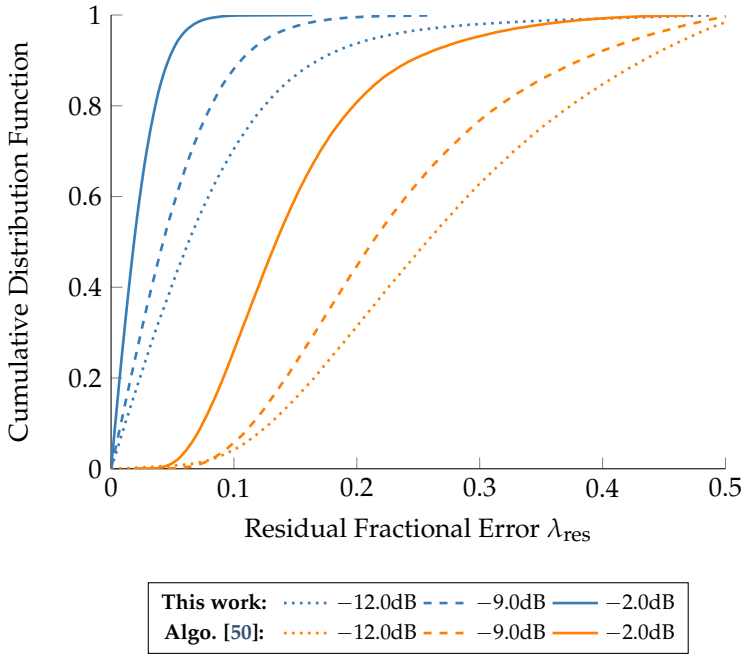


Fig. 3.8 Cumulative Distribution Function of the residual fractional error λ_{res} for the proposed algorithm versus the algorithm from [50] and different SNR values, with SF $= 8$ and $N_P = 28$. Only correctly synchronized frames are included in the metric.

To better assess the impact of the residual fractional offsets on the payload error probability, in Fig. 3.8 we plot the cumulative distribution function (CDF) of the worst-case λ_{res} value for both algorithms. We note that the estimator used in the proposed algorithm results in the same value (which can also be considered worst) for all 28 symbols, since the estimation is performed during the preamble. For the algorithm of [50], we do not consider the values λ_{res} after the first demodulation error, to avoid affecting the CDF by the error propagation in the tracking loop (the estimation in the tracking loop may become completely unstable when the DFT bin of a symbol is not correctly identified). Since both algorithms use the same estimator for the fractional CFO, the differences between the CDFs mainly illustrate the differences of residual errors on the fractional STO.

For the receiver of [50] and an SNR of -9 dB, 60% of the packets contain at least one payload symbol that experiences a residual offset λ_{res} greater

than 0.2. Such high values of λ_{res} significantly increase the probability of a wrong demodulation of a symbol. Regarding our proposed receiver, we observe that for the same SNR more than 95% of the synchronized packets experience a very low residual fractional error λ_{res} that is below 0.1 during the demodulation of the payload. As illustrated in Fig. 3.2, such very low fractional offsets have a small impact on the demodulation performance.

All the above results strongly illustrate the need for the receiver to correct the fractional offsets before estimating the integer offsets, as well as the need to use an estimator for the fractional STO with a small variance. Overall, the performance evaluation indicates that both the early correction of the fractional CFO and STO and the use of the more accurate estimator $\widehat{\lambda}_{\text{STO}}$ instead of $\widehat{\lambda}_{\overline{\text{STO}}}$ enable our proposed receiver to attain PERs of 10^{-3} in low SNR regions that are relevant for LoRa. On the contrary, the receiver from [50] levels off at an error floor for the overall PER that is between 10^{-1} and 10^{-2}.

3.5.3 Impact of the CFO Range on the Performance

As explained in Section 3.4, the proposed algorithm iteratively estimates the fractional STO in two steps. The first estimate $\tilde{\lambda}_{\text{STO}}$ is computed with the approximation $\widetilde{M} = N - \tilde{s}_{\text{up}}$ instead of $\widehat{M} = \widehat{L}_{\text{STO}}$. In practical SNR regions, we have $\tilde{s}_{\text{up}} \approx L_{\text{CFO}} - L_{\text{STO}}$ and the accuracy of the approximation $\widetilde{M}$ thus depends on the value of L_{CFO}. Large values of L_{CFO} decrease the precision of $\tilde{\lambda}_{\text{STO}}$, which in turn increases the probability of wrongly estimating the integer offsets L_{CFO} and L_{STO} and of experiencing synchronization failures. Unlike our receiver, the receiver from [50] does not use a similar approximation, its synchronization failure probabilities hence do not depend on the value of L_{CFO} when $\Delta f_c \in \left[-\frac{B}{4}, \frac{B}{4}\right]$.

In Fig. 3.9 we show the overall PER of the proposed synchronization algorithm for SF $= 8$ and different ranges of CFO. We observe that CFO values below 24 ppm only induce a negligible deterioration of the PER. However, obtaining a PER of 10^{-2} with CFO in the range $[24, 35]$ ppm requires 1 dB stronger SNR than for $\Delta f_c = 0$. This loss increases to 2.5 dB for a PER target of 10^{-3}. Nevertheless, such large CFO values are unlikely to arise in practice as 20 ppm is the expected maximum offset for oscillators in commercial receivers [26].

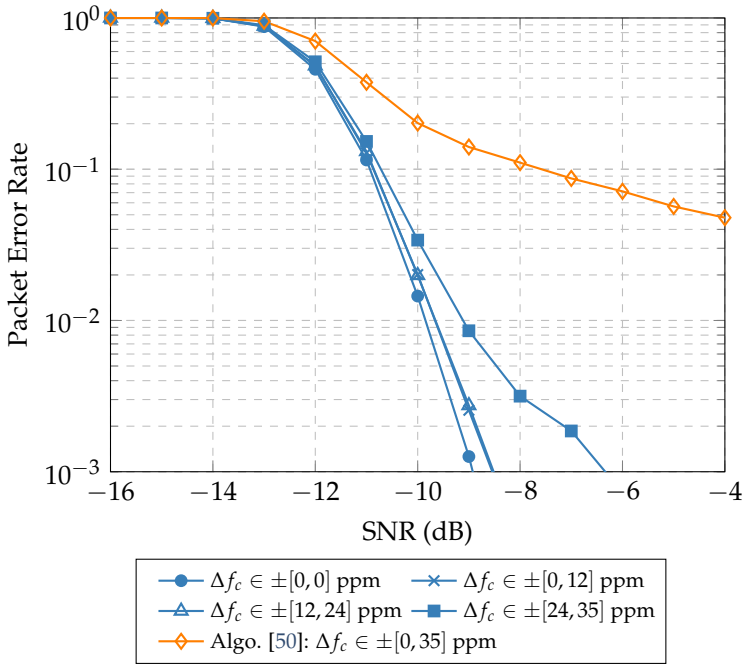


Fig. 3.9 PER of the proposed synchronization algorithm for uncoded LoRa with different CFO ranges, $B = 125$ kHz, $f_c = 868$ MHz, SF $= 8$ and $N_P = 28$. The PER of the receiver from [50] does not depend on the CFO range.

3.5.4 Packet Error Rates for Coded LoRa

Fig. 3.10 presents the overall PER of the proposed synchronization algorithm and the algorithm from [50] for three different SFs when a code rate CR $= 4/7$ is used. Regarding first the receiver of [50], due to the predominance of synchronization failures as described previously, its error rates level off for all SFs at error floors between 10^{-1} and 10^{-2}. We do not observe any gain due to coding gain for the overall PER of [50], since the performance in that case is limited by the synchronization failures in the preamble, which is uncoded.

On the contrary, our proposed receiver does benefit from the Hamming code, especially in low SNR regions. The hard-decoding stage indeed decreases the probability of experiencing payload errors [60], which are the dominating error events for our receiver at low SNR. For a target PER of

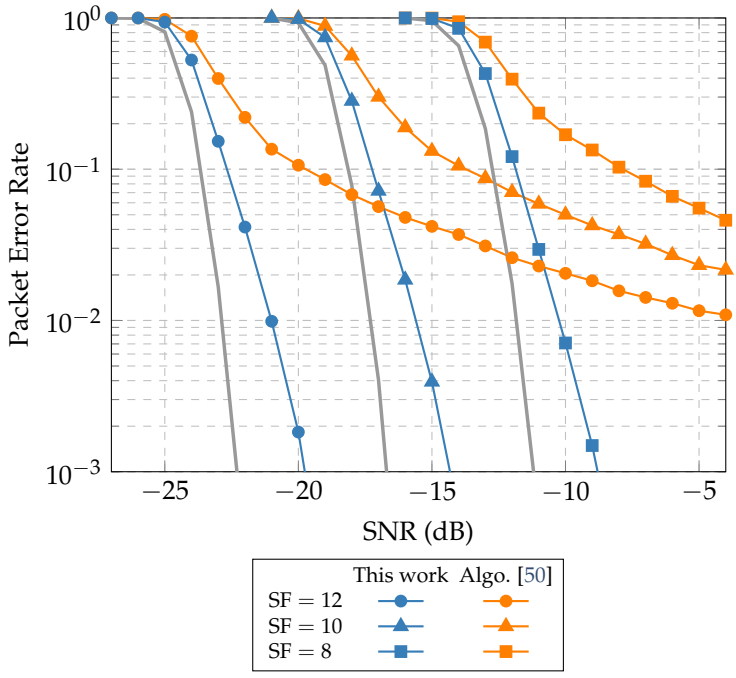


Fig. 3.10 PER of the proposed algorithm and the algorithm from [50] for coded LoRa with different SFs, $N_P = 28$ and a Hamming code $(4,7)$. The thick gray lines show the PER for a perfectly synchronized receiver.

10^{-1} and SF $= 8$, we observe 1 dB gain between the coded scenario from Fig. 3.10 and the uncoded scenario showed in Fig. 3.7. This gain however decreases at higher SNRs as synchronization failures tend to become the dominating error events. Whereas our receiver requires only 1 dB higher SNR than a theoretical perfectly synchronized receiver to reach a 10^{-3} PER in the absence of coding, this difference increases to 2 dB with coding. In the latter scenario, only synchronization failures significantly cause packet errors, whereas in the uncoded case, the probabilities of experiencing a synchronization failure or a payload error are almost identical, as shown in Fig. 3.7.

3.6 Complexity Analysis

We finally discuss the complexity of our proposed synchronization algorithm and the algorithm of [50]. Both algorithms rely on the conventional demodulation stage of a LoRa receiver, with some additional computations and memory requirements. All synchronization computations are performed at the Nyquist-rate $f_S = B$, as explained in Section 3.5. We show that the overhead of the synchronization is minor compared to the complexity of the demodulation stage itself.

The proposed algorithm and the algorithm from [50] use similar estimators for the fractional CFO and for the integer offsets. Estimating the integer offsets L_{CFO} and L_{STO} involves only three simple arithmetic operations. The estimator $\widehat{\lambda}_{\text{CFO}}$ requires a single angle computation and a constant, but small, number of complex Multiply and Accumulate (MAC) operations, as listed in Table 3.2. Regarding the fractional STO, both our estimator $\widehat{\lambda}_{\text{STO}}$ from (3.15) and the estimator $\widehat{\lambda}_{\overline{\text{STO}}}$ from (3.16) carry out a fixed number of arithmetic operations on three DFT outputs Y_i, including one complex division. It is worth underlining that the latter estimator requires an inversion of the non-linear function $T(x)$, whereas the former does not and is hence simpler to implement.

Offset	Operations
L_{CFO}, L_{STO}	Two integer additions, one integer division
λ_{CFO}	11 complex MACs, 1 angle
λ_{STO}	6 complex additions, 10 complex MACs 2 complex divisions

Table 3.2 Operations carried out by the proposed algorithm per packet to estimate the integer and fractional parts of the STO and CFO.

The algorithmic complexity of the demodulation stage given in (2.7) is $\mathcal{O}(N \log N)$ when the DFT is implemented using a Fast Fourier Transform (FFT), and hence depends on the spreading factor. This stage also requires the storage of N samples in memory, since the dechirping operation and FFT can be performed in place. As the studied synchronization algorithms involve only a constant and limited number of operations, their computational cost is negligible compared to the complexity of the demodulation stage, especially for large spreading factors.

In summary, our proposed synchronization algorithm exhibits a very

low computational overhead, similar the algorithm of *Bernier et al.* but with much better performance results. This low complexity allows a straightforward implementation of the algorithm on the ULP SDR platform that will be introduced in Chapter 4.

3.7 Discussion and Summary

In this chapter, we designed and evaluated a synchronization algorithm for Nyquist-rate LoRa receivers that is robust to both carrier frequency and sampling time offsets. We derived a new low-variance estimator for the fractional STO, since an accurate correction of this offset is crucial to reach the low sensitivity levels provided by the CSS modulation. We also showed that for a sampling frequency $f_s = B$, the estimation of this fractional component is intertwined with the estimation of the integer parts of the STO and CFO.

To avoid a complex joint estimation of these offsets, we proposed instead a low-complexity synchronization algorithm that iteratively estimates the fractional STO using the new estimator. For a target packet error rate of 10^{-3}, performance evaluations show that a LoRa receiver using the proposed synchronization algorithm and a hard Hamming decoder requires only 1 or 2 dB higher SNR compared to a perfectly synchronized receiver, depending on the code rate. Finally, beside its good performance, the proposed synchronization algorithm also requires only very few additional computations compared to the demodulation stage of a Nyquist-rate LoRa receiver, which thus makes it a good candidate for the LoRa SDR presented in the next chapter.

4

A sub-mW Cortex-M4 MCU for IoT Software-Defined Radios

The traditional architecture of LPWAN radios, implemented in hardware and often tied to a single protocol, limits the reconfigurability of IoT end nodes and thus their effective lifetimes. We discussed in Chapter 1 that this absence of interoperability with other standards worsens the environmental burden of IoT applications. In particular, the arrival in the future of new LPWAN technologies may trigger the replacement of the already deployed end nodes, since these devices cannot be upgraded to newer standards and otherwise require the expensive maintenance of legacy networks.

Multiple radio access technologies (multi-RAT), and SDRs in particular, have the potential to improve the interoperability of IoT end nodes. In this chapter, we study the software implementation of LPWAN PHY layers on a general-purpose MCU processor. This approach allows to change the communication protocol by means of a software update, even after the device has been deployed. Yet, since IoT devices are often energy-constrained, the underlying challenge is to implement the digital baseband processing in software while maintaining a very low power consumption. To overcome this problem, we propose a complete MCU architecture that follows the approach from [61] by offloading the generic signal processing to a con-

figurable hardware accelerator. A software implementation running on a general-purpose low-power processor is then responsible for the protocol-specific digital baseband computations.

In this chapter, we describe and evaluate the implementation of two commercial LPWAN protocols, namely Sigfox and LoRa, on our custom MCU architecture that has specifically been designed for LWPAN SDRs. We first analyze in Section 4.1 the literature on multi-RAT for IoT devices, and we explain why the LPWAN SDR architecture of [61] is the best trade-off between implementation flexibility and performance. In Section 4.2, we conduct a hardware/software co-design of the MCU architecture, including the selection of the processor (CPU). We then describe in Section 4.3 the software implementation of the LoRa and Sigfox PHY layers on this particular MCU architecture. Finally, thanks to a prototype fabricated in 28 nm FDSOI, we perform in Section 4.4 an experimental evaluation of both SDR implementations in a testbed and we discuss their performance.

4.1 Motivations and Related Works

Several approaches have been proposed in the literature, or implemented in commercial devices, to provide multi-RAT to IoT end nodes. These solutions can be classified into three categories: end nodes stacking several hardware radios, integrated hardware radios implementing multiple PHY layers, and SDRs whose PHY layer can be updated. In this section, we conduct a brief survey on multi-RAT implementations that follow the aforementioned approaches, and we finally motivate why SDRs on ULP MCUs are a promising solution to enhance the lifetime of IoT devices.

4.1.1 Devices Stacking Multiple Hardware Radios

The authors of [62] and [63] suggest combining several hardware LPWAN radios in a single IoT end node to benefit from the integration of multiple wireless technologies. The underlying idea is to design a hardware platform with an embedded MCU, which may already include an integrated LPWAN radio, and to add one or several external radio modules featuring different PHY layers.

A dual-RAT prototype built around an STM32F21x MCU (featuring an ARM Cortex-M3 CPU and 512 kB of flash memory) with two external LP-

WAN radios (LoraWAN and NB-IoT) is demonstrated in [63]. Each radio is responsible for a single protocol, but both transceivers can be used concurrently thanks to a multi-threaded software implementation. The authors demonstrate the feasibility of a multi-RAT device with two external radios, but very little analysis of the presented design is conducted (e.g., the power consumption and efficiency of the prototype is barely addressed).

An open-source multi-RAT platform supporting three LPWAN standards (LoRaWAN, Sigfox and NB-IoT) is presented in [62]. The platform consists of an MCU Murata module (featuring an ARM Cortex-M0+ CPU and an integrated SX1276 transceiver), a custom NB-IoT extension shield with a Quectel BG96 module, and a power management unit (PMU). The integrated SX1276 radio provides both LoRaWAN and Sigfox connectivity. The PMU contains two coulomb counters that allow to independently evaluate the power consumption of the Murata module and of the NB-IoT shield. For short packets (i.e., payloads between 1 and 12 bytes), the energy required by the platform to transmit one bit of data is 3.6, 20.5 and 27.2 mJ/bit for LoRaWAN, Sigfox and NB-IoT, respectively. In the case of LoRaWAN and NB-IoT, which allow longer payloads, the energy per bit however strongly decrease with the length of the packet (down to 0.46 mJ/bit for NB-IoT and payloads longer than 256 bytes). These measurements confirm that multi-RAT devices provide different operating modes that allow to trade energy for coverage, payload size or reliability simply by changing the communication protocol.

4.1.2 Multi-PHYs Hardware Radios

A second solution for IoT multi-RAT consists in the design of transceivers implementing several protocols. Infrequently, some commercial MCUs or external radios follow this approach for protocols that share the same frequency bands, since the same RF front-end can be re-used. For instance, the Semtech SX1276 transceiver implements the LoRa PHY layer [38], but also a generic (G)FSK modulation that can be configured to match the specifications of Sigfox [62].

Regarding MCUs with integrated radios, the STMicroelectronics MCUs of the STM32WL5x family embed a sub-GHz on-chip radio supporting the LoRaWAN, Sigfox and W-MBus protocols [64]. In the 2.4 GHz frequency bands, the Nordic nRF52833 MCU provides connectivity for Bluetooth (Low Energy and Mesh), NFC and IEEE 802.15.4 with Zigbee [65].

4.1.3 LPWAN Software-Defined Radios

Nevertheless, the two previously discussed approaches do not allow to support protocols that were not considered when designing the end node, or protocols that are yet to come. Upgrading existing nodes by stacking new external radios is also not always a possible solution due to design- and application-specific limitations (compatibility, size, cost, ...). A more flexible approach consists in the design and deployment of software-defined radios [61, 66, 67].

In an SDR, the RF front-end is designed to be parameterizable while most of the digital baseband (DBB) computations of the PHY layer are performed in software by a processor or a reconfigurable accelerator. Such an architecture enables updates of the communication protocol, and therefore allows a device, during its expected lifetime, to switch from one standard to another or simply to follow the latest revisions of a protocol. However, the main challenge of SDRs for IoT end nodes is to execute the DBB processing in software with a power consumption that fits in the 1–10 mW budget of LPWAN transceivers [16]. Three different architectures have been proposed in the literature to tackle this challenge.

In [66], the DBB computations are offloaded to a reconfigurable baseband accelerator specifically designed for LPWAN protocols. The accelerator datapath includes a register file, parallel arithmetic logic units (ALUs), a multi-level output adder tree to merge the outputs of the ALUs and a scalar computations unit. The ALUs and the adder tree can be reconfigured to implement DBB kernels (such as low-pass filters or matched filters) of a given PHY layer. The receiver datapaths of two protocols (Bluetooth Low Energy and IEEE 802.15.4 OQPSK) are implemented in the accelerator and evaluated. Nonetheless, the overall architecture is rather rigid and the set of configurable DBB kernels does not allow to implement all LPWAN protocols or algorithms (e.g., the FFT kernel is missing for the LoRa PHY).

In [67], an SDR platform for experimental IoT communications running on a generic MCU and a reprogrammable field-programmable gate array (FPGA) is presented. The large price ($55), bulky size and important power consumption (around 100 mW in Rx) of the platform are however incompatible with the specifications of cheap, small and low-power IoT end nodes.

In [61], the authors observe that most LPWAN standards share very similar characteristics (i.e., the use of small bandwidths in sub-GHz bands). They therefore propose an MCU architecture (shown in Fig. 4.1) with a

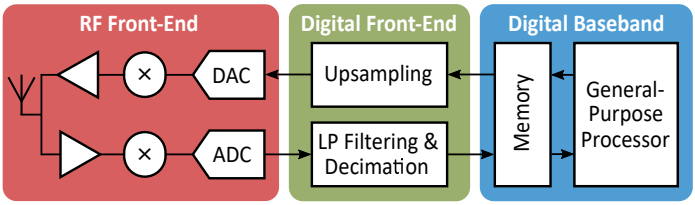


Fig. 4.1 Architecture of an IoT software-defined radio with a common sub-GHz RF front-end, a configurable digital front-end (DFE) and a general-purpose processor performing the protocol-specific digital baseband (DBB) computations.

single common RF front-end, a configurable digital front-end (DFE) that executes the generic baseband operations (e.g., low-pass filtering), and a general-purpose low-power CPU that carries out the protocol-specific DBB computations. The authors simulate a GFSK demodulation algorithm for Bluetooth Low Energy on a MCU design in 28 nm FDSOI with a low-power RISC-V processor. Simulation results indicate that for a 1-MHz symbol rate, the MCU requires a CPU frequency of 114 MHz and attains an estimated peak power consumption of 1.6 mW.

Out of the three studied architectures, only the architecture of [61] shows enough flexibility to implement PHY layers with different DBB algorithms, while maintaining a sufficiently low area overhead and power consumption. Yet, the work of [61] only conducts a preliminary evaluation of the proposed SDR architecture through simulation. A proof-of-concept low-power design actually implementing several standards is so far still missing in the literature. The objective of this chapter is therefore to design, fabricate and evaluate the first prototype of an SDR platform for LPWAN protocols running on a general-purpose ULP MCU. [1]

4.2 Designing a MCU Architecture for IoT SDRs

In this section, we explore several key architectural issues when designing an ULP SDR-capable MCU. We first motivate the choice of the ARM Cortex-M4 as CPU for the digital signal processing (DSP) of LPWAN phys-

[1] The proposed prototype has been designed with other researchers of the ECS group. This chapter describes in details only the contributions of the author to the prototype.

ical layers. We then explain the hardware implementation of our DFE. The overall architecture of our custom ULP MCU with the different peripherals (memories, DFE, ...) is eventually described.

4.2.1 Low-Power CPU Exploration

In the general architecture presented in Fig. 4.1, one of the most critical design choices consists in the selection of the general-purpose processor. Indeed, more advanced CPUs require fewer cycles to carry out a given operation, which therefore lowers the minimal CPU frequency at which the DSP needs to run. On the other hand, such processors also require a greater silicon area and consume more energy per cycle compared to low-area and low-power CPUs. The CPU choice hence amounts to a trade-off between the minimum required frequency, the consumed energy per cycle and the occupied silicon area.

During the exploration phase of the microcontroller design, we considered three different popular low-power CPUs for IoT MCUs: the ARM Cortex-M0, Cortex-M3 and Cortex-M4. These 32-bit processors all share a RISC instruction set and feature a 32-bit three-stage pipeline. The Cortex-M0 features the lowest area by implementing a Von Neumann architecture with a unique memory for both program and data, whereas the Cortex-M3 and M4 occupy more area by relying on a Harvard architecture, with distinct program memory (PMEM) and data memory (DMEM). Moreover, contrary to the other cores, the Cortex-M4 also contains two 16-bit single-instruction multiple-data (SIMD) lanes to accelerate fixed-point DSP computations. These additional SIMD instructions allow the Cortex-M4 to efficiently carry out DSP computations on pairs of Q16 fixed-point numbers. For instance, the `SMUAD` instruction performs two multiplications on two pairs of Q16 numbers and returns their sum in a single cycle. This instruction allows an efficient implementation of the multiplication of two complex 16-bit numbers.

When performing the CPU selection, we conducted benchmarks with two DSP kernels implemented in the ARM CMSIS DSP library. These kernels are used in our software implementations of the LoRa and Sigfox PHYs: a 256-point complex FFT and a FIR filter with decimation (63 taps, decimation factor of 52), both in Q16 fixed point representation. Fig. 4.2 shows the number of CPU cycles required for each kernel and the three studied processors. Compared to the Cortex-M0, we observe that the Cortex-

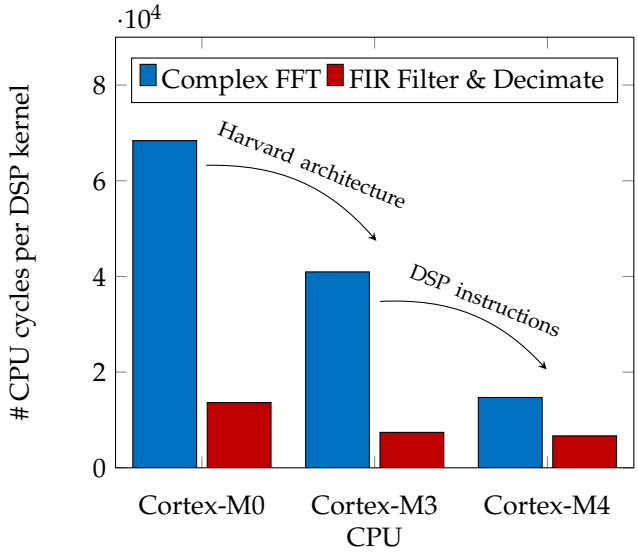


Fig. 4.2 CPU cycles required for a Q16 complex 256-point FFT and a FIR filtering and decimation of 260 Q16 complex samples (63 taps, decimation factor of 52) for three low-power ARM Cortex-M CPUs.

M3 requires 40% and 46% less CPU cycles to perform the FFT and the FIR filtering, respectively, thanks to its Harvard architecture. In the Von Neumann architecture, the Cortex-M0 needs to stall for one cycle to retrieve every data word (e.g., a FIR filter coefficient) since the retrievals of the next instructions and the data word cannot be parallelized. The Cortex-M3 and M4 do not suffer from this limitation. Furthermore, running the complex FFT on the Cortex-M4 instead of the M3 further reduces the number of cycles by 64%, thanks to SIMD instructions such as `SMUAD`.

This benchmarking illustrates the importance of selecting a sufficiently advanced CPU to attain both a real-time and energy-efficient software processing of LPWAN PHYs. To minimize their power consumption, ULP MCUs often rely on low supply voltages, which also imply low CPU frequencies. As such, ULP MCUs usually exhibit minimum energy points below 50 MHz [68]. Overall, the Harvard architecture and the SIMD DSP instructions of the Cortex-M4 offer a 4.65× speed-up for the complex FFT compared to the M0, at the cost of a CPU area only $\approx$ 3× larger in a 28 nm FDSOI technology [68, 69]. For the non-coherent ML LoRa detector of (2.7) and SF $= 8$, solely executing the complex 256-point FFT at the

Operation	# cycles / sample	Sampling freq.	Required CPU freq.
Q16 FIR Filter & Decimate (16 taps)‡	23.2	2 MHz	46.4 MHz
Q16 256-point FFT	57.4	125 kHz	7.2 MHz

‡ Offloaded to a hardware accelerator in this work.

Table 4.1 Required ARM Cortex-M4 CPU frequency for the two most intensive DSP operations of the LoRa baseband Rx chain.

Nyquist sampling frequency of 125 kHz already requires a minimum CPU frequency of 33.48 MHz (versus only 7.2 MHz for the M4). Since running the entire LoRa Rx chain on the Cortex-M0 would need high CPU frequencies (> 100 MHz), which would in turn require a higher supply voltage and therefore be energy inefficient, and since the Cortex-M3 and M4 have similar areas while the latter is computationally superior, we opted for the Cortex-M4 in our MCU architecture.

4.2.2 Hardware Offloading of Rx Low-Pass Filtering

As LPWAN protocols use different modulations with various data rates and signal bandwidths, both the sampling frequency of the baseband signal and its low-pass filtering in Rx have to be configurable. In the proposed architecture, we assume that the RF front-end has a single analog low-pass filter with a fixed cut-off frequency. The proper low-pass filtering to the actual bandwidth of the sampled signal must hence be done in the digital domain. A first solution would be to perform the low-pass filtering and decimation in software. Table 4.1 shows the required CPU frequency to carry out in software the two most intensive DSP operations of the Rx LoRa baseband chain implemented in this work: the low-pass FIR filtering with decimation and the complex FFT used to demodulate LoRa symbols. Since the low-pass filtering is performed at a higher frequency than the demodulation (oversampling rate $R = 16$ to allow subsequent timing synchronization), the FIR filter is one order of magnitude more computationally intensive than the FFT and would require a minimum CPU frequency of 46.4 MHz for its sole execution.

To reduce the energy consumption of the SDR, we offload instead the

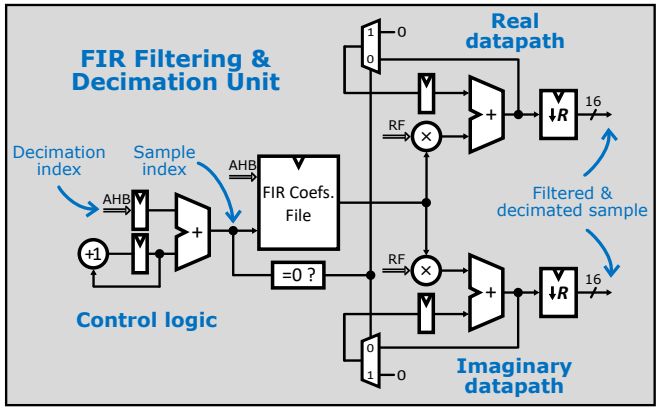


Fig. 4.3 Low-area digital implementation of the FIR filtering and decimation unit in the DFE. All registers are clocked with the same clock running at the frequency f_s.

generic low-pass filtering to a configurable hardware accelerator. This hardware accelerator is illustrated in Fig. 4.3. We implemented a low-area FIR filter with two multipliers and two accumulators, each of these operating on the in-phase and quadrature sample streams in parallel. The number of filter taps is equal to the decimation factor R, which allows carrying out a single multiply-and-accumulate operation per in-phase or quadrature input sample. The weights and the decimation factor of the FIR filter are configurable by the CPU through the AHB bus, as well as the decimation index, i.e., the relative input sample offset at which the computation of a new output sample begins. The programming of the decimation index enables a fine-grain time synchronization before the signal is decimated to a lower frequency and demodulated.

4.2.3 Microcontroller Architecture with Digital Front-End

Fig. 4.4 shows the architecture of the proposed SDR-capable MCU, codenamed SleepRider [68]. The design contains an ARM Cortex-M4 CPU, two 32-kB high-density (HD) SRAM memories, a Direct Memory Access (DMA) controller, a custom digital front-end (DFE), a unified frequency and back-bias regulation (UFBBR) unit and I/O peripherals. Except for the PMEM, the CPU and all other blocks share a single memory bus to exchange data. SleepRider MCU was designed for ULP consumption in

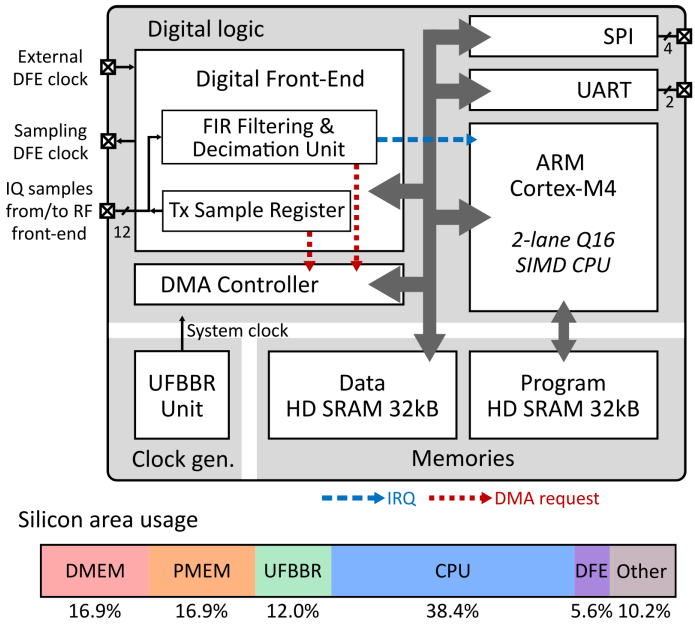


Fig. 4.4 Architecture of the proposed SDR-capable SleepRider microcontroller, with the silicon area usage of the underlying blocks.

28 nm FDSOI with various techniques, including ultra-low supply voltage (0.4V) for its logic [68]. The UFBBR unit jointly generates the main system clock and the back-bias voltages of the FDSOI transistors to compensate for PVT variations. The frequency of the system and CPU clock is programmable and spans from 12 to 80 MHz.

To minimize the power consumption, the protocol-independent baseband processing is offloaded to a DFE. Our custom DFE contains the FIR filtering and decimation unit previously described, and directly interfaces with the RF front-end and the data memory bus. The DFE features several configurable parameters to enable software implementations of IoT protocols with different characteristics (e.g., bandwidth, signaling rates). First, the DFE logic uses an external clock as reference sampling clock. The nominal frequency of the DFE clock is 2 MHz[2], but it can be adapted to match the signaling rate of the targeted PHY layer. Frequency dividers in

[2]The RF front-end used in this work has Rx low-pass filters with a minimum cut-off frequency of 0.75 MHz. The sampling frequency of the DFE must hence be above 1.5 MHz to avoid aliasing.

the DFE further allow to divide the external clock by 2, 4, 8 or 16 to generate the sampling frequency f_s at the RF front-end. Similarly, the weights and the decimation rate R (between 1 and 16) of the FIR filter can also be configured following the targeted waveform specifications.

The DFE contains separate datapaths and control logic for the Tx and Rx chains. In Rx, the block retrieves baseband samples from the RF chain at an oversampled frequency f_s. The oversampled samples are first low-pass filtered using the FIR filter and then decimated to the baseband frequency f_s/R. In Tx, the DFE directly transfers samples to the RF front-end without performing upsampling, i.e., the sampling rate is directly equal to f_s. In either Tx or Rx, the digital samples retrieved from or sent to the RF front-end are complex baseband samples coded on 2x12-bit fixed point numbers. Since the CPU contains two 16-bit SIMD lanes, these 12-bit words are extended to (resp. truncated from) 16 bits before entering (resp. leaving) the memory bus.

To alleviate the CPU load by avoiding costly context switches, the DMA controller is responsible for transferring the 32-bit complex baseband samples between the DMEM and the DFE. The memory transfers are organized as follows. The CPU first defines windows of L complex samples in the DMEM and indicates the length of these windows to the DFE. In Tx, when the CPU has modulated L samples, it configures the DMA to transfer the L 32-bit words to the DFE. The rate of the memory transfers is determined by the baseband frequency f_s/R, i.e., the DFE processes one complex sample at a time and only requests a new sample to the DMA controller when it has transmitted the previous one to the RF front-end. Similarly, in Rx, the CPU first configures the DMA controller for L transfers, and the DFE then issues for each new decimated sample a 32-bit transfer request to the controller. When L samples have been transferred, the DFE raises an interrupt request (IRQ) to the CPU, indicating that the window has been transmitted (in Tx) or is ready to be demodulated (in Rx).

4.3 Software Implementation of LPWAN PHYs

Since LoRa and Sigfox use the same sub-GHz ISM spectrum, whereas NB-IoT employs separate licensed frequency bands, the first two standards are a perfect match to demonstrate the multi-RAT capabilities of the proposed MCU architecture. In this section, we explain how the LoRa and Sigfox

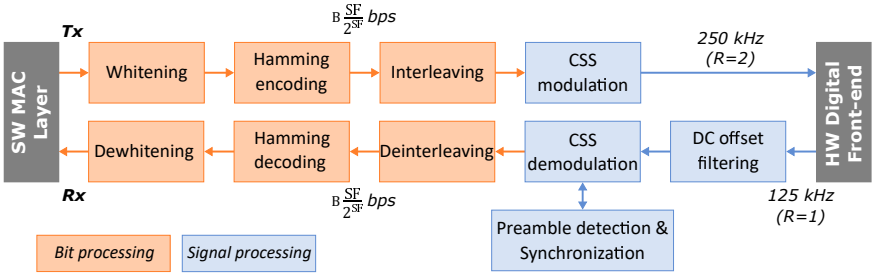


Fig. 4.5 Block diagram of the LoRa DBB processing for $B = 125$ kHz.

PHY layers have been implemented on our SDR-capable MCU. The Sigfox PHY layer is also briefly described.

Both DBB implementations are coded in C and rely on the ARM CMSIS DSP library, which provides complex DSP routines optimized for the Cortex-M4. All DSP software operations are executed in Q16 fixed-point to benefit from the SIMD instructions of the CPU.

4.3.1 LoRa: a Spread-Spectrum Symmetric PHY Layer

We first describe our software implementation of the LoRa PHY layer, which is illustrated in Fig. 4.5. In the Tx chain, the whitening, Hamming coding, interleaving operations are straightforwardly implemented in C using bitwise operations. LoRa symbols are generated using (2.1) with an oversampling rate $R = 2$ to obtain a spectral representation close to the signal generated by the reference transceiver SX1276 [38]. To transmit symbols in real time, the software relies on two ping-pong buffers. The CPU computes the modulated samples and stores them in one buffer, while the DMA controller transfers the samples from the other buffer to the DFE. After the transmission of an entire buffer, the DFE notifies the CPU with an IRQ and the roles of the ping-pong buffers are exchanged. If the CPU fills one buffer before the transmission of the other buffer, it waits for an IRQ of the DFE. The DFE frequency is set $f_s = 2B$ and the ping-pong buffers hence contain a single LoRa symbol of $2N$ complex samples each, as shown in Fig. 4.6.

In Rx mode, the DFE frequency is set to $f_s = 16B$ ($R = 16$). The DFE is configured to filter the received signal from the RF front-end according to the spectral representation of LoRa symbols and to decimate it to the Nyquist frequency B. The DMA transfers the decimated samples from the DFE to the DMEM. The first stage of the Rx chain is a first-order IIR filter

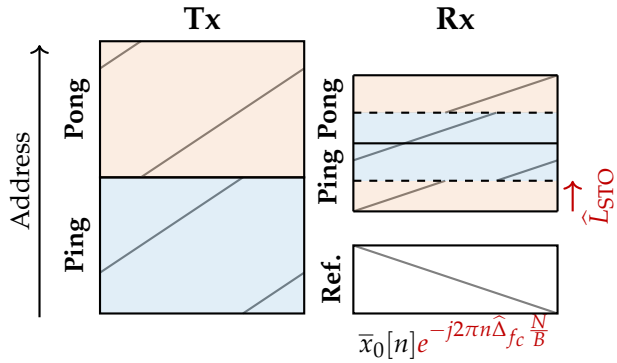


Fig. 4.6 Organization of the buffers in DMEM for LoRa Tx ($f_s = 2B$) and Rx (Nyquist rate, after decimation by the DFE) chains. The gray lines in the buffers show the instantaneous frequency of the modulated chirps.

$y[n] = \alpha y[n-1] + \frac{1+\alpha}{2}(x[n] - x[n-1])$ ($\alpha = 0.9$) that removes a residual DC offset from the RF front-end. To perform a real-time demodulation of the received samples, the Rx chain also uses two adjacent ping-pong buffers and a buffer storing $\overline{x}_0[n]$, the conjugate of the base waveform, as shown in Fig. 4.6. When a ping-pong buffer is full, the DFE notifies the CPU with an IRQ and the processor demodulates it by dechirping the sampled signal with the sequence $\overline{x}_0[n]$ stored in memory and by computing the FFT.

Before demodulating payload symbols, the receiver first needs to detect the preamble of an incoming packet and to synchronize to the transmitter. We showed in Chapter 3 that both the preamble detection and synchronization stages of a LoRa receiver can be implemented using the ML non-coherent demodulator of (2.7). A preamble is detected when the same symbol is repetitively demodulated. The receiver synchronization requires the correction of a CFO Δf_c and an STO τ. A receiver that is not synchronized in time retrieves windows of N samples containing two consecutive partial chirps instead of a single entire chirp.

The preamble detection and the estimation of the STO and the CFO are implemented following the synchronization algorithm derived and described in Chapter 3. The estimators of Algorithm 1 are straightforwardly implemented in C using the CMSIS DSP library, which leverages the SIMD DSP instructions of the CPU. The remaining of this description hence focuses on the correction of both offsets using the estimates $\hat{\tau}$ and $\widehat{\Delta}f_c$ of the

STO and CFO, respectively. The CFO is corrected by multiplying the reference waveform $\overline{x}_0[n]$ in memory by $e^{-j2\pi n\widehat{\Delta}_{f_c}\frac{N}{B}}$. The estimated STO is split into integer $\widehat{L}_{\text{STO}}$ and fractional $\widehat{\lambda}_{\text{STO}}$ parts. The integer STO is corrected by treating the two adjacent ping-pong buffers as a single circular buffer of two symbols, as illustrated in Fig. 4.6. The DSP routines of the CMSIS DSP library required by the demodulation (complex multiplication, FFT and magnitude computation) have been rewritten to properly handle this circular buffer. The processor can hence start the demodulation of a symbol anywhere in the circular buffer, depending on $\widehat{L}_{\text{STO}}$. The fractional STO λ_{STO}, i.e., the intra-sample time offset, is corrected by updating the decimation index in the DFE among the R available offsets. Once the entire packet is demodulated, the received bits are deinterleaved and decoded using a hard-decision Hamming decoder. The decoded payload bits are finally dewhitened and sent to the MAC layer.

4.3.2 Sigfox: an Asymmetric Ultra-Narrowband PHY Layer

Sigfox is an IoT wireless technology that runs on a single global network licensed by the homonymous company. The protocol achieves low receiver sensitivities thanks to ultra narrowband (UNB) communications, which minimize the noise power at the receiver. Contrary to LoRa, the Sigfox PHY is an asymmetric layer that shifts most of the signal processing complexity (synchronization, decoding, ...) to the gateway. We first briefly introduce the Sigfox PHY layer, and we subsequently explain how we implemented this protocol on our MCU architecture.

The Sigfox PHY Layer

To alleviate the load on the sensor, Sigfox relies on different modulation and coding schemes for the uplink and downlink channels. In uplink, the sensor encodes the payload with a convolutional code and transmits data using differential binary phase shift keying (DBPSK) at a fixed bit rate of 100 bps. Up to three repetitions of the same message, with different code polynomials, can be transmitted to bring time diversity at the gateway.

In downlink, the gateway uses a Gaussian frequency shift keying (GFSK) modulation with a bit rate of 600 bps, a frequency deviation of 800 Hz and a bandwidth-time product $BT = 0.5$. A BCH $(15, 11)$ forward error correcting code and a whitening sequence are applied on the payload bits before the modulation stage. Downlink messages are only sent when an answer

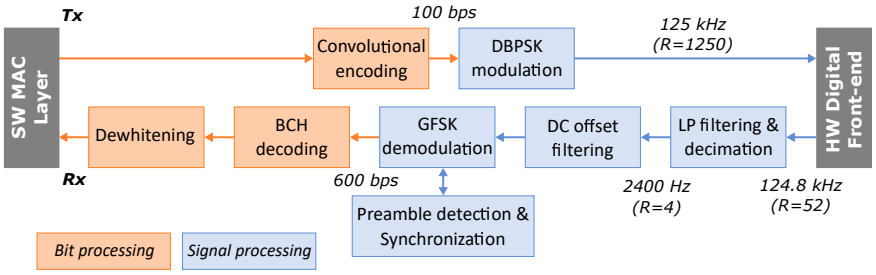


Fig. 4.7 Block diagram of the Sigfox DBB processing.

is requested by an uplink message. Since the gateway already estimated the CFO of the sensor during the reception of the uplink packet, it also performs a precorrection of this CFO when transmitting the downlink packet. This technique simplifies the synchronization procedure at the sensor, as no CFO correction is needed [70]. GFSK symbols can be straightforwardly demodulated using a frequency discriminator, i.e., by determining if the phase of the signal increases or decreases over the symbol period. Let $y[n]$ and $y[n-1]$ be two consecutive samples of a GFSK symbol. A conventional frequency discriminator computes the phase derivative $\phi[n]$ between the samples $y[n]$ and $y[n-1]$

$$\phi[n] = \Re\left(y[n]\right)\Im\left(y[n-1]\right) - \Im\left(y[n]\right)\Re\left(y[n-1]\right), \tag{4.1}$$

where $\Re\left(x[n]\right)$ and $\Im\left(x[n]\right)$ represent the real and imaginary parts of the sample $x[n]$, respectively.

Sigfox Software DBB Implementation

Whereas the CSS modulation of LoRa uses large bandwidths, Sigfox relies on UNB signaling with very low data rates. However, due to the limited filtering capabilities of our parameterizable RF front-end, a sampling frequency f_s above 1.5 MHz is required to avoid aliasing. The proposed DFE has hence been designed to operate with a nominal external clock frequency of 2 MHz. In Tx, this clock can further be divided down to $f_s = 125$ kHz. The low signaling rates of the Sigfox PHY compared to the high sampling frequencies f_s of the DFE imply that the Sigfox DBB needs to process samples with a high oversampling rate. Fig. 4.7 shows our software implementation of this PHY. The Tx chain only comprises two stages: the convolutional encoder and a DBPSK modulator. Using the

minimum sampling frequency $f_s = 125$ kHz, the modulation stage generates the DBPSK symbols ($+1$ or -1) with an oversampling rate of 1250 to achieve a baud rate of 100 symbols per second. The modulated samples are transferred to the DFE using the DMA controller, similarly to the LoRa DBB.

In Rx, the baud rate is 600 Hz and the GFSK waveform occupies a narrow bandwidth of approximately 2 kHz. To attain a sampling frequency in the DBB that is a multiple of the 600 Hz symbol rate, the external clock frequency is decreased from the 2 MHz nominal frequency to 1.9968 MHz. Moreover, since the FIR filter in the DFE (with 16 taps) is unable to implement a low-pass filter with a cut-off frequency at 2 kHz, a second low-pass filtering stage in software is needed. The DFE is configured to filter and decimate the baseband signal from the RF front-end to a sampling frequency of 124.8 kHz. The DMA controller transfers chunks of 1664 complex samples to the DMEM in ping-pong buffers, which are then again filtered and decimated by the CPU (63 taps, decimation factor of 52).

The baseband signal after the second filtering stage has a sampling frequency of 2400 Hz, with four samples per symbol. We therefore adapt the detection rule of (4.1) to use three consecutive samples per symbol from the four available: $\overline{\phi}[n] = \phi[n+1] + \phi[n]$, which improves the resistance to noise. When a preamble is detected, the STO is estimated by selecting the sample offset $\hat{\tau} \in \{0,1,2,3\}$ that maximizes the decision variable $\left|\overline{\phi}[4n+\hat{\tau}]\right|$ over successive preamble symbols. The k-th symbol is demodulated by retrieving the sign of the time-aligned frequency discriminator: $\hat{b}_k = \operatorname{sgn}\left\{\overline{\phi}[4k+\hat{\tau}]\right\}$. Upon the demodulation of all payload symbols, the demodulated bits are finally decoded with a hard-decision BCH decoder and dewhitened.

4.4 Experimental Performance Evaluation

We finally demonstrate and evaluate the LoRa and Sigfox SDRs in an experimental testbed. SleepRider MCU [68] is used in the testbed as a prototype of the microcontroller architecture introduced in Section 4.2. In this section, we first describe our testbed. We then present and discuss the energy consumption of the two SDR implementations on SleepRider, as well as the sensitivities in Rx of both SDRs.

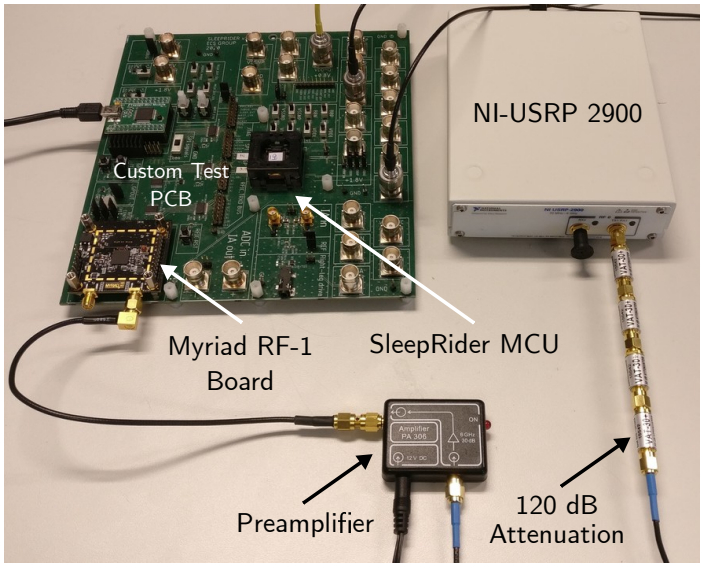


Fig. 4.8 Testbed with the SleepRider MCU, the Myriad-RF1 configurable transceiver, a low-noise preamplifier and a NI-USRP 2900.

4.4.1 Experimental Testbed with SleepRider MCU

The testbed is shown in Fig. 4.8. Beside the MCU, the testbed also contains a LimeSDR Myriad-RF1 LMS6002D configurable transceiver board, a low-noise preamplifier, passive attenuators and a NI-USRP 2900. The Myriad-RF1 is used as RF front-end and transfers complex baseband samples from or to the MCU on a 12-bit bus with a sampling clock driven by the DFE [71]. Because the Myriad-RF1 is originally designed for cellular communications with higher signal strengths than those of LoRa and Sigfox, a 30 dB low-noise preamplifier with a 3 dB noise figure is inserted to enhance the receiver gain of the RF front-end. Passive attenuators, with an overall attenuation of 120 dB, are used in Rx to obtain input signals from the USRP with power levels between -130 and -120 dBm.

When testing the LoRa SDR, the USRP is driven by a computer running the LoRa GNU Radio SDR implementation of [40], which is compatible with commercial LoRa radios. The Sigfox SDR has been validated in the testbed by replacing the USRP with the official Sigfox USB test dongle [72]. All experiments use a carrier frequency of 868 MHz.

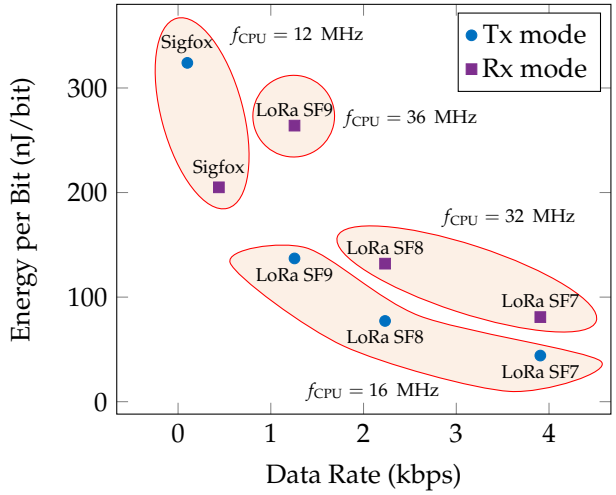


Fig. 4.9 Minimum CPU frequency and average energy required by the LoRa and Sigfox SDRs for processing an information bit in the DFE and the DBB.

4.4.2 Minimum CPU Frequency and Energy Consumption

We validated the functionality of the Tx and Rx chains for both LoRa and Sigfox in the testbed. For LoRa, we used the bandwidth value $B = 125$ kHz and the coding rate CR $= 4/7$ with the SFs 7, 8 and 9. We are unable to run the Rx chain for SFs higher than 9 as the total size of the program and FFT tables exceeds the available space in the PMEM (32 kB). This PMEM size is limited by the die area of our custom chip. Commercial MCUs however usually embed much larger PMEMs (up to 1 MB).

Fig. 4.9 shows the minimum CPU frequency required to run the SDRs in both Tx and Rx, as well as the average energy per bit transmitted/received by the DFE and DBB. To minimize the energy consumption, we scale down the CPU frequency using the UFBBR (by steps of 4 MHz) for each mode (Tx or Rx), PHY layer and SF in the case of LoRa. Contrary to commercial transceivers with integrated RF circuits, the RF front-end and the low-noise preamplifier in our setup are off-the-shelf components that are not optimized for low-power operation. Since our goal is to demonstrate that a software processing of LPWAN PHY layers can be realized within the 1–10 mW power budget of LPWAN transceivers [16], we exclude the non-optimized low-noise preamplifier and Myriad RF front-end from the power

measurements and focus only on the DFE and DBB.

We first discuss the LoRa SDR. The most intensive part of the Tx LoRa DBB is the CSS modulation, which has a linear complexity $\mathcal{O}(N)$ with respect to the symbol period $T = \frac{2^{\mathrm{SF}}}{B} = \frac{N}{B}$. The MCU hence needs for all SFs a minimum CPU frequency of 16 MHz with a constant power consumption of 172 µW. However, since the data rate of the CSS modulation decreases with the SF, the energy required to encode and modulate one bit of information data increases exponentially with the SF, from 44 nJ/bit for SF 7 to 137 nJ/bit for SF 9. Similarly, the Rx chain is dominated by the demodulation stage, which has an algorithmic complexity of $\mathcal{O}(N \log N)$ because of the FFT. Incrementing the SF from 7 to 9 hence slightly increases both the CPU frequency (from 32 to 36 MHz) and the power consumption (from 316 to 332 µW). This $\mathcal{O}(N \log N)$ complexity explains why the consumed energy per bit increases with the SF faster in Rx than in Tx.

Regarding the Sigfox SDR, we observe in Fig. 4.9 that the Tx DBB has a lower energy efficiency with 324 nJ required per information bit, despite the simple signaling DBPSK scheme. In this mode, the MCU has an average power consumption of 32 µW, which is only amortized over the low data rate of 100 bps. This 32 µW power consumption corresponds to the average power of the MCU at 12 MHz when spending 99.5% of its time idle. It could be reduced if the specific MCU design was capable of either running at a lower frequency without having its power dominated by static leakage, or by going in deep-sleep mode between two symbols. Both solutions are unfortunately not implemented in SleepRider MCU due to practical design limitations. The same issue applies to the Rx chain to a lesser extent, with 61% the time waiting on the next symbol and a power consumption of 90 µW.

Nevertheless, the power consumption of the DSP in the MCU prototype, between 32 and 332 µW, lie well below the total power usage of LPWAN transceivers. As a comparison, the commercial LoRa SX1276 transceiver consumes 34 mW in Rx [38], and the Sigfox-compatible GFSK receiver from [70] draws 14.5 mW. The power budgets of both transceivers are dominated by the consumption of the RF front-end. These results indicate that the power needed to run the DSP of our SDRs remains negligible compared to the overall power budget of a LPWAN transceiver.

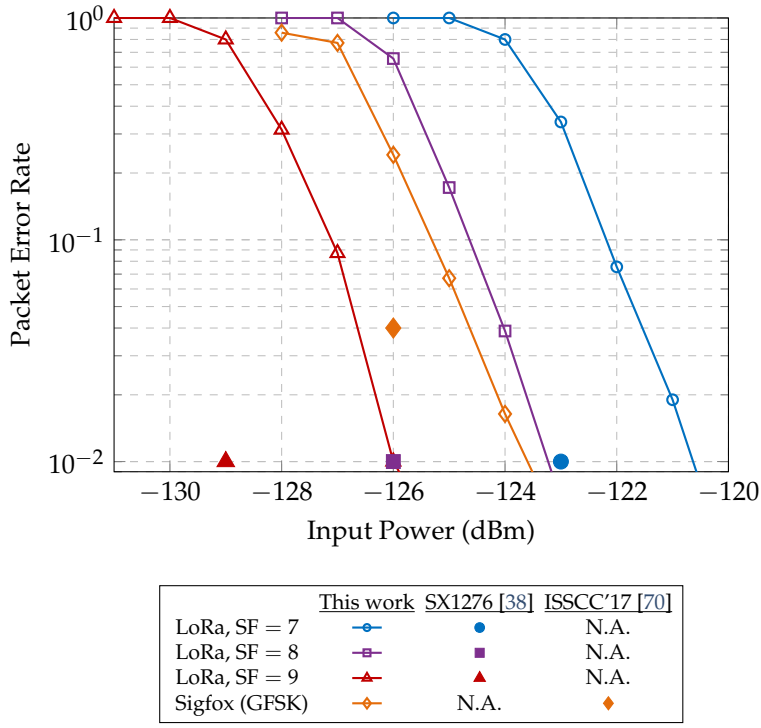


Fig. 4.10 Experimental PERs when receiving LoRa (64 payload bytes) and Sigfox (8 payload bytes) packets for different input powers.

4.4.3 Evaluation of the SDR Rx Sensitivities

Finally, we assess the performance of our SDR Rx chains by measuring the packet error rate (PER) versus the signal power at the input of the preamplifier. In this experiment, we generate different input power values by sweeping the Tx gain of the USRP. To evaluate the sensitivity of our Sigfox radio in Rx, we recorded Sigfox downlink packets from the USB dongle with the USRP, and then replayed these packets with the USRP. 1000 LoRa or downlink Sigfox packets are transmitted per power level. Fig. 4.10 shows the experimental PERs of both SDRs, as well as the advertised sensitivity levels of the LoRa SX1276 transceiver and the GFSK receiver from [70]. We first observe that our LoRa SDR receiver benefits from a 3 dB spreading gain when the SF is incremented, similarly to the SX1276. Yet, for a target PER of 10^{-2}, our receiver requires about 3 dB

higher input power than the SX1276. This 3 dB gap is mainly due to our implementation of the low-pass FIR filter in the DFE which only has 16 taps and hence features non-negligible side lobes outside the signal bandwidth. A hardware implementation of the FIR filter in the DFE with several multipliers could however improve the Rx sensitivities of our SDRs at the cost of a larger area. For the Sigfox SDR, a 4% PER is attained at an input power level of -124.5 dBm, compared to -126 dBm for the GFSK receiver from [70]. Overall, the results presented in Fig. 4.10 indicate that our MCU prototype is capable of receiving packets of two different protocols with near state-of-the-art sensitivity levels.

4.5 Discussion and Summary

Long-term maintenance of end nodes will be an important challenge to keep the operational expenses of IoT applications low. In particular, the prevailing hardware architecture of IoT radios, often tied to a single protocol, strongly weakens the interoperability of the devices. To mitigate this issue, we designed in this chapter an MCU architecture suitable for LPWAN SDRs with a very low power consumption for the DBB computations.

The proposed architecture features (a) a general-purpose low-power processor for the protocol-specific computations (b) a hardware DFE for the generic signal processing and (c) an ultra-low power digital implementation with low supply voltage and frequency scaling. We implemented in software the physical layers of the Sigfox and LoRa standards, and we validated both implementations in a testbed with an MCU prototype in 28nm CMOS FDSOI. Experimental measurements show that both SDRs reach near-state-of-the-art sensitivities below -120 dBm in Rx. Even more so, our prototype performs the digital signal processing of the two protocols with a power consumption between 32 and 332 µW, thus demonstrating that an SDR can fit in the 1–10 mW power budget of a LPWAN transceiver.

To the best of our knowledge, this work is the first to demonstrate an actual prototype of an ULP SDR with implementations of two commercial IoT protocols. Our design however exhibits some limitations that could be addressed in future works. As mentioned, our hardware FIR low-pass filter implementation has a very low silicon area, but at the cost of a degraded input SNR to the DBB. A more complex FIR filter implementation could improve the sensitivities of the SDRs by several dBs. In addition, we ex-

perimentally validated the LoRa PHY only for a bandwidth $B = 125$ kHz. Although our SDR implementation could theoretically receive LoRa packets for $B = 250$ kHz by increasing the CPU frequency to 64–68 MHz, SleepRider MCU is not capable of running at frequencies higher than 80 MHz, which would be required to process packets at $B = 500$ kHz.

This second limitation could be pushed back by using more powerful CPUs, e.g., the recent ARMv8.1 CPUs with the Helium M-Profile Vector Extension (MVE) for DSP computations [73]. In this regard, a recent work has demonstrated a RISC-V instruction set architecture (ISA) extension specifically designed for LPWAN DBB processing [74]. Simulation results show that their ISA extension decreases the number of cycles of a LoRa synchronization algorithm by more than 50%. The MCU architecture described in this chapter should hence not be considered as a final design, but rather as a first proof-of-concept whose performance can still greatly be improved. Further works associating hardware/software co-design of ULP MCUs and the conception of general-purpose low-power processors with advanced DSP capabilities are thus still needed to truly pave the way for commercial IoT SDRs.

PART II
Multi-User Receivers for LoRa Gateways

5

A Maximum-Likelihood-based Two-User Receiver

LoRaWAN uses non-slotted ALOHA as multiple access scheme to avoid an energy-intensive synchronization mechanism between users. Since the end nodes are not synchronized and share non-orthogonal waveforms, collisions between frames may arise. It has become well-known over the years that the throughput of a LoRaWAN network rapidly decreases for an increasing number of users, as a result of the collisions [17, 18, 22, 75, 76]. This shortcoming poses a threat to the longevity of LoRaWAN gateways and end nodes, since operators may upgrade their infrastructure to another protocol when their existing network becomes interference-limited.

To improve the resilience of LoRaWAN networks, the second part of this thesis focuses on the design of multi-user receivers for LoRa gateways. Such a two-user receiver theoretically enables up to a tripling of the throughput in a non-slotted ALOHA network. In this chapter, we derive from the maximum-likelihood criterion a practical LoRa receiver capable of demodulating two colliding LoRa packets using the same SF. Although the CSS modulation is not initially designed for the joint demodulation of concurrent users, we show that the proposed ML-derived two-user detector inherently leverages the difference of time, carrier frequency and

received power between users to demodulate the colliding symbols. To further demonstrate the practicality of the proposed detector for actual LoRaWAN gateways, we also design a multi-user synchronization algorithm robust to interference. Simulation and experimental results indicate that a LoRa receiver combining the proposed synchronization algorithm and two-user detector is capable of detecting and demodulating two interfering users with satisfactorily error rates, even at low SNR.

This chapter is organized as follows. In Section 5.1, we enumerate the different approaches studied in the literature to improve the performance of LoRaWAN networks, including a discussion on previously proposed multi-user LoRa receivers. In Section 5.2, we describe the analytical model of superimposed signals from two LoRa users with the same SF. In Section 5.3, we derive a practical two-user detector from the maximum likelihood sequence estimator using different complexity reduction techniques. In Section 5.4, we evaluate the performance of the proposed detector with simulation results. Different parameters influencing the error rates of the receiver are discussed. In Section 5.5, we present a multi-user synchronization algorithm that is robust to same-SF interference, and we jointly evaluate its performance together with the proposed two-user detector in simulation and in an experimental testbed.

5.1 Motivations and Related Works

We mentioned in Chapter 2 that LoRa waveforms using different SFs are quasi-orthogonal to each other. As a consequence, interference between colliding LoRa users can be one of two very different types: *inter-SF interference* and *same-SF interference*. Due to the spreading, the impact of inter-SF interference is different from the impact of same-SF interference. Notably, LoRa transmissions using different SFs only suffer from inter-SF interference at very low signal-to-interference ratios (SIR) (starting from around -16 dB on average) thanks to near-orthogonal signaling [44]. Since concurrent transmissions with different SFs can be demodulated independently in parallel, commercial LoRa gateways embed several demodulators that are dynamically assigned to different SFs. On the other hand, simultaneous transmissions with the same SF are far from orthogonal. For conventional single-user LoRa receivers, colliding packets with the same SF exhibit a capture effect, such that the strongest signal can often be demodulated if

the SIR and the SNR are sufficient, but at the expense of the weaker signal [60]. Experimental measurements have shown that for a 3-dB difference in power between two users, there is a 97% chance of demodulating the packet of the strongest user [77], but the packet of the weaker user is generally lost.

As of today, two main approaches have been followed in the literature to overcome same-SF collisions, which are the dominating error events in interference-limited LoRaWAN networks. Below, we briefly summarize these attempts, and we further explain why multi-user LoRa receivers are a promising, but also challenging, solution to improve the reliability and scalability of LoRaWAN networks.

5.1.1 Approaches to Improve LoRaWAN at the MAC Layer

Several improvements at the MAC layer have been proposed over the last years to overcome the same-SF interference issue. These solutions either extend the underlying ALOHA multiple access of LoRaWAN [78, 79, 80, 81, 82] or design new MAC layer synchronization mechanisms [83, 84, 85].

In the first category, carrier sense multiple access (CSMA) has been investigated in [78, 79, 80], with LoRa devices implementing a listening before talk mechanism. Although CSMA helps to prevent collisions for users close to the gateway, it offers little gain over pure ALOHA in realistic networks because of the hidden terminal problem [79]. In [81, 82], time synchronization algorithms are designed and implemented on top of the traditional LoRaWAN specifications. For each SF, the channel is divided into time slots and the transmissions of the devices are regulated to fit into these slots, which allows the network to benefit from the throughput increase of slotted ALOHA, as shown in Fig. 5.1.

To completely avoid the issue of collisions, other works in the literature seek to implement time-division multiple access (TDMA). In such schemes, the gateways periodically send beacons that enable the devices to obtain an accurate time reference for their transmissions. In [83], a static distribution of the time slots based on the addresses of the devices is performed when they connect to the network. In [84], the gateway performs a fine-grained scheduling of the time slots and regularly broadcasts the time-resource repartition to all devices. These TDMA schemes however oblige the end nodes to listen to beacons from the gateway to maintain their synchronization in time, which greatly increases their power consumption.

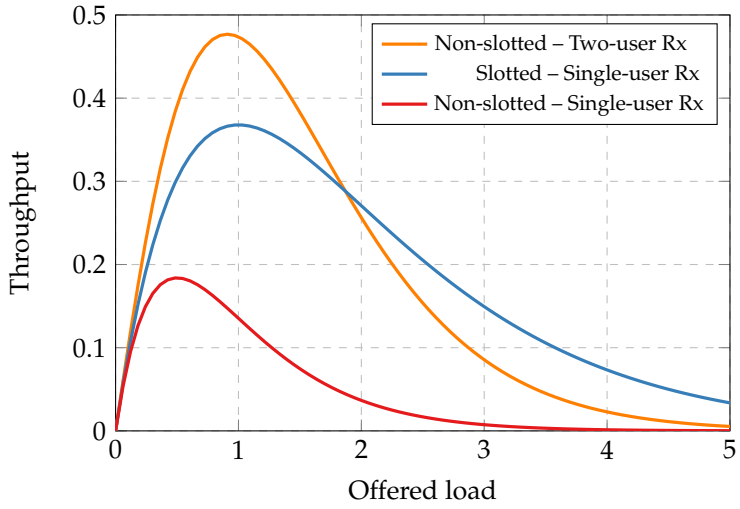


Fig. 5.1 Theoretical throughputs of non-slotted ALOHA and slotted ALOHA networks. Two-user receivers strongly improve the throughput of non-slotted ALOHA networks as they are capable of decoding two colliding packets.

5.1.2 Multi-User Receivers for Dense LoRaWAN Networks

In this chapter, we aim to improve the reliability and scalability of LoRaWAN networks without negatively impacting the energy consumption of the nodes, i.e., by keeping pure ALOHA as multiple access scheme. Instead of modifying the MAC layer, we propose a two-user LoRa receiver capable of demodulating two colliding users with the same SF. This receiver is located at the mains-powered gateway, which does not suffer from energy constraints. To anticipate the possible gains of such a receiver, the theoretical throughput of a pure ALOHA network with an ideal two-user receiver is shown in Fig. 5.1 [86, 87]. At the optimal offered load, the deployment of a two-user receiver can triple the network throughput compared to a conventional single-user receiver, and can even surpass the performance of slotted ALOHA without requiring any time synchronization between the devices. In a similar spirit, but by means of stochastic geometry, the authors of [88] evaluate that the usage of a two-user SIC receiver allows to increase the number of nodes in a network by 2.59, without deteriorating its reliability.

Yet, the design of an efficient two-user receiver has only recently received the appropriate attention. Several LoRa multi-user receivers have so far been proposed [89, 90, 91, 92, 93, 94, 95, 96]. The receivers presented in [89, 90, 91, 93] demodulate all the symbols received during a single symbol period, and subsequently try to attribute the demodulated signals to their corresponding user. In [89], the different frequency offsets between the users are used to distinguish their symbols. A similar technique is used in [92], which leverages the time offsets between colliding packets instead. In [90], the received power of each user is also used for the decision, whereas in [91], both the time and frequency offsets between users are taken into account. The authors of [93] propose a scaled despreading operation that allows a more robust estimation of the time offset between the colliding users and hence of their corresponding symbols. All of these receivers, however, rely on a joint demodulation of the colliding symbols, which is difficult to perform at the low SNR levels (-25 dB to -5 dB) at which LoRa commonly operates.

On the contrary, the receivers of [94, 95, 96] implement interference cancellation mechanisms that enable separate detections of the colliding symbols. A parallel interference cancellation LoRa receiver relying on the differences in carrier frequencies and received powers between users is introduced in [94]. Similarly, two successive interference cancellation (SIC) receivers are presented in [95] and [96]. Most notably, the authors of [95] show that their SIC receiver is capable of efficiently demodulating the user with the strongest received power, but the bit error rates of the weaker users all floor between 10^{-4} and 10^{-3}.

The aforementioned works designed multi-user receivers that heuristically leverage one or two transmitter-specific features (e.g., carrier offset, time offset, received powers) to distinguish the contributions of the respective users. These receivers however fail to attain useful error rates at the low SNRs that are characteristic of LoRa communications. The first objective of this chapter is therefore to derive the expression of a maximum-likelihood two-user LoRa receiver that optimally uses all properties of a signal with two colliding same-SF packets. However, such a pure ML receiver is inherently computationally too complex for a realistic digital baseband implementation of a LoRa gateway. We hence subsequently introduce complexity-reduction techniques to obtain a more practical two-user receiver that still succeeds to demodulate colliding users at low SNR.

5.2 Signal Model for Two Interfering Users

In this section, we describe the Nyquist-rate ($f_s = B$) model of superimposed signals from two users with the same SF, namely user A and user B. The baseband signal at the gateway is recovered with a single receiving antenna and radio-frequency front-end. Since LoRa uses a non-slotted ALOHA multiple access scheme, the users are neither synchronized among themselves nor to the gateway. Let us define $s_{\mathrm{A}}^{(k)}$ and $s_{\mathrm{B}}^{(k)}$ as the k-th colliding symbols sent by users A and B, respectively. The symbols $s_{\mathrm{A}}^{(1)}$ and $s_{\mathrm{B}}^{(1)}$ are hence the two first information symbols of both users that collide with each other. To simplify the explanation and the mathematical formulation of the multi-user detector, we assume that the gateway is initially synchronized in frequency and time to user A, whose packet arrives first. This can be achieved with a standard single-user synchronization procedure, such as the algorithm described in Chapter 3. We define $\tau \in [0, N)$ as the relative sample-level time offset between the first sample of the k-th symbol transmitted by user A and the first sample of the k-th symbol of user B, as illustrated in Fig. 5.2. This sample offset is split into an integer part $L_{\mathrm{STO}} = \lfloor \tau \rfloor$ and a non-integer part $\lambda_{\mathrm{STO}} = \tau - \lfloor \tau \rfloor$, with a slightly different definition than the one of Chapter 3.

Since the transmission of user B experiences an STO $\tau \geq 0$ with respect to user A, the first $\lfloor \tau \rfloor$ samples of symbol $s_{\mathrm{A}}^{(k)}$ overlap with symbol $s_{\mathrm{B}}^{(k-1)}$ and the last $N - \lfloor \tau \rfloor$ samples of symbol $s_{\mathrm{A}}^{(k)}$ overlap with symbol $s_{\mathrm{B}}^{(k)}$. The contribution of user B to the k-th window of N samples $y^{(k)}[n]$ can therefore be split into two parts, namely $y_{\mathrm{B},1}^{(k)}[n]$ for $n \in \mathcal{N}_1 = \{0, \ldots, \lceil \tau \rceil - 1\}$ and $y_{\mathrm{B},2}^{(k)}[n]$ for $n \in \mathcal{N}_2 = \{\lceil \tau \rceil, \ldots, N-1\}$, with

$$y_{\mathrm{B},1}^{(k)}[n] = e^{j2\pi \left(\frac{(n+N-\tau)^2}{2N} + (n+N-\tau) \left(\frac{s_{\mathrm{B}}^{(k-1)}}{N} - \frac{1}{2} - u\left[n - n_{f,1}^{(k)} \right] \right) \right)}, \tag{5.1}$$

$$y_{\mathrm{B},2}^{(k)}[n] = e^{j2\pi \left(\frac{(n-\tau)^2}{2N} + (n-\tau) \left(\frac{s_{\mathrm{B}}^{(k)}}{N} - \frac{1}{2} - u\left[n - n_{f,2}^{(k)} \right] \right) \right)}, \tag{5.2}$$

and $n_{f,1}^{(k)} = \lceil \tau \rceil - s_{\mathrm{B}}^{(k-1)}$ and $n_{f,2}^{(k)} = N + \lceil \tau \rceil - s_{\mathrm{B}}^{(k)}$.

Prior to synchronization, both users are affected by distinct carrier fre-

quency offsets relative to the receiver, namely $\Delta f_{c,\mathrm{A}}$ and $\Delta f_{c,\mathrm{B}}$. However, since we assume the receiver to be synchronized to user A, there is only a single effective CFO $\Delta f_c = \Delta f_{c,\mathrm{B}} - \Delta f_{c,\mathrm{A}}$ that affects the signal from user B. The CFO can also be split into integer and fractional parts using the same definition as in Chapter 3, i.e., $L_{\mathrm{CFO}} = \lfloor N\frac{\Delta f_c}{f_s} \rceil$ and $\lambda_{\mathrm{CFO}} = N\frac{\Delta f_c}{f_s} - L_{\mathrm{CFO}}$, where $\lfloor \cdot \rceil$ is the rounding operator.

Finally, the users have different transmit powers and experience independent single-tap channels h_{A} and h_{B}. Let $P_{\mathrm{A}} = |h_{\mathrm{A}}|^2$ and $P_{\mathrm{B}} = |h_{\mathrm{B}}|^2$ be the received powers of users A and B, respectively, which we assume to be constant. However, since LoRa is intentionally designed for non-coherent detection, we do not make any assumption on the phase coherence of successive symbols. We thus define $\theta_{\mathrm{A}}^{(k)}$ and $\theta_{\mathrm{B}}^{(k)}$ as the initial phases of the k-th symbol of user A and B, respectively. The discrete-time baseband-equivalent model of the received signal sampled at $f_s = B$ that represents the k-th window of N samples is therefore

$$\begin{aligned} y^{(k)}[n] &= \sqrt{P_{\mathrm{A}}} e^{j\theta_{\mathrm{A}}^{(k)}} x_{s_{\mathrm{A}}^{(k)}}[n] + z^{(k)}[n] \\ &+ \begin{cases} \sqrt{P_{\mathrm{B}}} e^{j\theta_{\mathrm{B}}^{(k-1)}} c[n] y_{\mathrm{B},1}^{(k)}[n], & n \in \mathcal{N}_1, \\ \sqrt{P_{\mathrm{B}}} e^{j\theta_{\mathrm{B}}^{(k)}} c[n] y_{\mathrm{B},2}^{(k)}[n], & n \in \mathcal{N}_2, \end{cases} \end{aligned} \tag{5.3}$$

where $c[n] = e^{j2\pi n \frac{\Delta f_c}{f_s}}$ is the effective CFO affecting user B and $z^{(k)}[n] \sim \mathcal{CN}(0, \sigma^2)$.

5.3 A Practical ML-derived Two-User Detector

In this section, we design a detector that is capable of jointly demodulating the symbols from two superimposed LoRa users. This detector is derived from the two-user maximum likelihood sequence estimator (MLSE), whose general structure is explained in [97]. We first derive the expression of the MLSE for the signal model defined in (5.3) and argue that its complexity is prohibitive in practice. To enable a practical implementation of the multi-user detector, we subsequently propose three simplifications to the two-user MLSE that strongly reduce its computational complexity.

5.3.1 Two-user LoRa Maximum Likelihood Sequence Estimator

As previously explained, the two-user detector is only needed when the frames of two LoRa users overlap in time. The non-colliding parts of the frames can be demodulated with the single-user detector of (2.7), while only the colliding parts need to be processed with the two-user detector derived in this section. We note here that in case of an overlap, the colliding part generally dominates the overall error rate. The proposed two-user detector uses as input the baseband signal $y^{(k)}[n]$ from (5.3), which is recovered with a single receiving antenna and sampled at the Nyquist frequency $f_s = B$. Let M be the number of symbols of user A colliding with symbols from user B. Following (5.3), the colliding parts of the frame can be decomposed into M successive windows of N samples. Similar to the single-user detector in (2.7), we propose to dechirp every window $y^{(k)}[n]$ with a reference downchirp $x_0^*[n]$ since the dechirped windows $\tilde{y}^{(k)}[n]$ possess a simpler analytical representation than $y^{(k)}[n]$ which facilitates the derivation of the ML detector.

In an AWGN channel, the MLSE seeks the sequence of symbols that minimizes the sum of the distances between the dechirped samples and the presumed contributions of each user [97]. Given the baseband model defined in (5.3), followed by the dechirping operation, the two-user MLSE seeks the set of candidate symbols $\bar{\boldsymbol{s}}^{(k)} = \{\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}, \bar{s}_{\mathrm{B}}^{(k)}\} \in \{0, \dots, N-1\}^3$ for $k \in \{0, \dots, M-1\}$ maximizing the probability

$$p = C \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{i=0}^{M-1} \sum_{n=0}^{N-1} \left\| \tilde{y}^{(i)}[n] - \tilde{x}^{(i)}\left[n \middle| \bar{\boldsymbol{s}}^{(i)}\right] \right\|^2 \right) \right) \tag{5.4}$$

where $\tilde{x}^{(k)}\left[n \middle| \boldsymbol{s}^{(k)}\right]$ represents the dechirped contributions of both users in the k-th window and $C = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{MN}$. The expression of $\tilde{x}^{(k)}\left[n \middle| \bar{\boldsymbol{s}}^{(k)}\right]$ is

$$\begin{aligned}\tilde{x}^{(k)}\left[n \middle| \boldsymbol{s}^{(k)}\right] &= \sqrt{P_{\mathrm{A}}} e^{j\theta_{\mathrm{A}}^{(k)}} e^{j2\pi \frac{n}{N} s_{\mathrm{A}}^{(k)}} \\ &+ \begin{cases} \sqrt{P_{\mathrm{B}}} e^{j\theta_{\mathrm{B}}^{(k-1)} + 2\pi\lambda_{\mathrm{STO}} u\left[n - n_{f,1}^{(k)}\right]} \cdot e^{j2\pi \frac{(n+N-\tau)}{N}\left(s_{\mathrm{B}}^{(k-1)} - \tau + N\frac{\Delta f_c}{f_s}\right)}, & n \in \mathcal{N}_1, \\ \sqrt{P_{\mathrm{B}}} e^{j\theta_{\mathrm{B}}^{(k)} + 2\pi\lambda_{\mathrm{STO}} u\left[n - n_{f,2}^{(k)}\right]} \cdot e^{j2\pi \frac{(n-\tau)}{N}\left(s_{\mathrm{B}}^{(k)} - \tau + N\frac{\Delta f_c}{f_s}\right)}, & n \in \mathcal{N}_2, \end{cases} \end{aligned} \tag{5.5}$$

The dechirped signal $\tilde{x}^{(k)}\left[n \middle| \boldsymbol{s}^{(k)}\right]$ depends on the offsets Δf_c and τ, the

phases $\{\theta_{\mathrm{A}}^{(k)}, \theta_{\mathrm{B}}^{(k-1)}, \theta_{\mathrm{B}}^{(k)}\}$, and the received powers P_{A} and P_{B} of each user.

The maximum likelihood sequence can be computed with reduced complexity using the Viterbi algorithm [97]. However, since there are N^2 possible states per window and N^3 possible state transitions between two windows, the complexity of this maximum likelihood sequence detector is prohibitive, especially for large SFs (i.e., large N). In the remainder of this section, we propose three modifications that simplify the maximum likelihood sequence detector of (5.4) to a more practical detector.

5.3.2 From Joint to Individual Maximum Likelihood Decisions

To avoid such a complex receiver, we suggest to move from a joint demodulation of all colliding symbols to individual decisions that are tied to windows of N samples. For each window k and the corresponding dechirped signal $\tilde{y}^{(k)}$, an individual decision amounts to detecting the overlapping symbols $\boldsymbol{s}^{(k)} = \{s_{\mathrm{A}}^{(k)}, s_{\mathrm{B}}^{(k-1)}, s_{\mathrm{B}}^{(k)}\}$. Let $\mathcal{L}(\bar{\boldsymbol{s}}^{(k)})$ denote the likelihood that the symbols $\bar{\boldsymbol{s}}^{(k)} = \{\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}, \bar{s}_{\mathrm{B}}^{(k)}\}$ have been transmitted in the k-th window

$$\mathcal{L}(\bar{\boldsymbol{s}}^{(k)}) = C' \exp\left(-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}\left\|\tilde{y}^{(k)}[n] - \tilde{x}^{(k)}\left[n|\bar{\boldsymbol{s}}^{(k)}\right]\right\|^2\right), \tag{5.6}$$

where $C' = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N$. An individual maximum likelihood decision for the k-th window is obtained by selecting the symbols $\bar{\boldsymbol{s}}^{(k)}$ that maximize the likelihood $\mathcal{L}(\bar{\boldsymbol{s}}^{(k)})$. This likelihood can be expressed more intuitively after some simple algebraic transformations and after the removal of all constant factors. We seek to maximize

$$\begin{aligned}\mathcal{L}'\left(\bar{\boldsymbol{s}}^{(k)}\right) &= \log\frac{\mathcal{L}\left(\bar{\boldsymbol{s}}^{(k)}\right)}{C'} = A - \sum_{n=0}^{N-1}\left\|\tilde{y}^{(k)}[n] - \tilde{x}^{(k)}\left[n|\bar{\boldsymbol{s}}^{(k)}\right]\right\|^2 \\ &= \sum_{n=0}^{N-1}\left[-\left\|\tilde{y}^{(k)}[n]\right\|^2 + 2\Re\left\{\tilde{y}^{(k)}[n]\tilde{x}^{(k)}\left[n|\bar{\boldsymbol{s}}^{(k)}\right]^*\right\}\right. \\ &\quad \left. - \left\|\tilde{x}^{(k)}\left[n|\bar{\boldsymbol{s}}^{(k)}\right]\right\|^2\right] + A,\end{aligned} \tag{5.7}$$

where A is a constant independent of $\bar{\boldsymbol{s}}^{(k)}$. We ignore in the following developments all terms that are independent of the candidate symbols $\bar{\boldsymbol{s}}^{(k)}$.

By developing $\tilde{x}^{(k)}\left[n|\bar{\boldsymbol{s}}^{(k)}\right]$ following (5.5), distributing all terms and using the relation $z_1 + z_1^* = 2\Re\{z_1\}$, we obtain

$$\begin{aligned}\mathcal{L}'\left(\bar{\boldsymbol{s}}^{(k)}\right) = &\sum_{n=0}^{N-1} 2\Re\left(\tilde{y}^{(k)}[n]\cdot\sqrt{P_\mathrm{A}}e^{-j\theta_\mathrm{A}^{(k)}}e^{-j2\pi\frac{n}{N}\bar{s}_\mathrm{A}^{(k)}}\right)\\ &+\sum_{n=0}^{\lceil\tau\rceil-1} 2\Re\left(\tilde{y}^{(k)}[n]\cdot\sqrt{P_\mathrm{B}}e^{-j\theta_\mathrm{B}^{(k-1)}}\tilde{y}_{\mathrm{B},1}^{*(k)}[n]\right)\\ &+\sum_{n=\lceil\tau\rceil}^{N-1} 2\Re\left(\tilde{y}^{(k)}[n]\cdot\sqrt{P_\mathrm{B}}e^{-j\theta_\mathrm{B}^{(k)}}\tilde{y}_{\mathrm{B},2}^{*(k)}[n]\right)\\ &-\sum_{n=0}^{\lceil\tau\rceil-1} 2\Re\left(\sqrt{P_\mathrm{A}P_\mathrm{B}}e^{j\left(\theta_\mathrm{A}^{(k)}-\theta_\mathrm{B}^{(k-1)}\right)}e^{j2\pi\frac{n}{N}\bar{s}_\mathrm{A}^{(k)}}\tilde{y}_{\mathrm{B},1}^{*(k)}[n]\right)\\ &-\sum_{n=\lceil\tau\rceil}^{N-1} 2\Re\left(\sqrt{P_\mathrm{A}P_\mathrm{B}}e^{j\left(\theta_\mathrm{A}^{(k)}-\theta_\mathrm{B}^{(k)}\right)}e^{j2\pi\frac{n}{N}\bar{s}_\mathrm{A}^{(k)}}\tilde{y}_{\mathrm{B},2}^{*(k)}[n]\right)\\ &-N\left(P_\mathrm{A}+P_\mathrm{B}\right),\end{aligned} \tag{5.8}$$

where $\tilde{y}_{\mathrm{B},1}^{(k)}[n]$ and $\tilde{y}_{\mathrm{B},2}^{(k)}[n]$ represent the dechirped signals of the candidate symbols of user B in the k-th window

$$\tilde{y}_{\mathrm{B},1}^{(k)}[n] = e^{j2\pi\frac{(n+N-\tau)}{N}\left(\bar{s}_\mathrm{B}^{(k-1)}-\tau+N\frac{\Delta f_c}{f_s}\right)}e^{j2\pi\lambda_\mathrm{STO}u\left[n-n_{f,1}^{(k)}\right]}, \tag{5.9}$$

$$\tilde{y}_{\mathrm{B},2}^{(k)}[n] = e^{j2\pi\frac{(n-\tau)}{N}\quad\left(\bar{s}_\mathrm{B}^{(k)}\quad-\tau+N\frac{\Delta f_c}{f_s}\right)}e^{j2\pi\lambda_\mathrm{STO}u\left[n-n_{f,2}^{(k)}\right]}. \tag{5.10}$$

After dividing $\mathcal{L}'(\bar{\boldsymbol{s}}^{(k)})$ by two, removing the constant term $N\left(P_\mathrm{A}+P_\mathrm{B}\right)$ and by grouping the terms containing $\tilde{y}_{\mathrm{B},1}^{(k)}[n]$ or $\tilde{y}_{\mathrm{B},2}^{(k)}[n]$ together, the log-

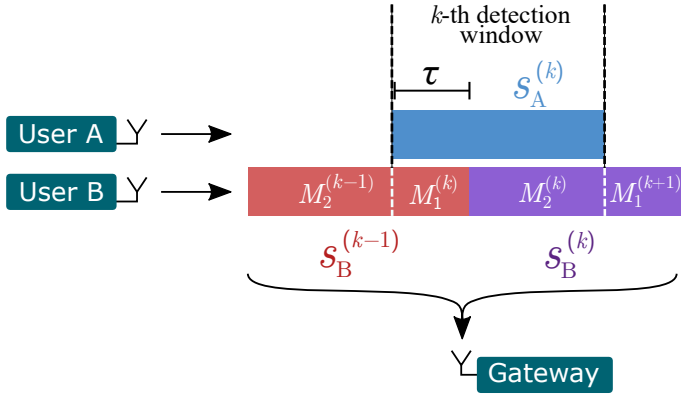


Fig. 5.2 Illustration of the matched filters outputs used by the detection of the overlapping symbols $s_A^{(k)}$, $s_B^{(k-1)}$ and $s_B^{(k)}$.

likelihood becomes

$$\begin{aligned}\mathcal{L}'\left(\bar{\boldsymbol{s}}^{(k)}\right) = &\sum_{n=0}^{N-1} \Re\left(\sqrt{P_A} e^{-j\theta_A^{(k)}} \tilde{y}^{(k)}[n] e^{-j2\pi \frac{n}{N} \bar{s}_A^{(k)}}\right) \\ &+ \sum_{n=0}^{\lceil\tau\rceil-1} \Re\left[\tilde{y}_{IC}^{(k)}[n] \sqrt{P_B} e^{-j\theta_B^{(k-1)}} \tilde{y}_{B,1}^{*(k)}[n]\right] \\ &+ \sum_{n=\lceil\tau\rceil}^{N-1} \Re\left[\tilde{y}_{IC}^{(k)}[n] \sqrt{P_B} e^{-j\theta_B^{(k)}} \tilde{y}_{B,2}^{*(k)}[n]\right],\end{aligned} \tag{5.11}$$

where $\tilde{y}_{IC}^{(k)}[n] = \tilde{y}^{(k)}[n] - \sqrt{P_A} e^{j\theta_A^{(k)}} e^{j2\pi \frac{n}{N} \bar{s}_A^{(k)}}$ is the interference-cancelled version of $\tilde{y}^{(k)}[n]$, i.e., the dechirped received signal with the presumed contribution of user A removed.

Each term of (5.11) corresponds to a matched filtering of one of the symbols $\bar{s}_A^{(k)}, \bar{s}_B^{(k-1)}, \bar{s}_B^{(k)}$ in the k-th window. To ease the writing of $\mathcal{L}'(\bar{\boldsymbol{s}}^{(k)})$, we define three matched filters $Y^{(k)}$, $M_1^{(k)}$ and $M_2^{(k)}$ that are connected to the symbols $s_A^{(k)}$, $s_B^{(k-1)}$, and $s_B^{(k)}$, respectively, as shown in Fig. 5.2. Since the receiver is synchronized to user A, the symbol $s_A^{(k)}$ translates to a Kronecker delta in the frequency domain. Accordingly, the first term of (5.11) corresponds to the DFT of the entire dechirped signal $\tilde{y}^{(k)}[n]$. The matched filter

of the symbol $\bar{s}_{\mathrm{A}}^{(k)}$ can hence be expressed as

$$Y^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}\right] = \sum_{n=0}^{N-1} \tilde{y}^{(k)}[n] \cdot e^{-j2\pi \frac{n}{N} \bar{s}_{\mathrm{A}}^{(k)}}. \tag{5.12}$$

The two remaining terms in (5.11) evaluate the contribution from user B, which is not synchronized. As the waveforms carrying the symbols $\bar{s}_{\mathrm{B}}^{(k-1)}$ and $\bar{s}_{\mathrm{B}}^{(k)}$ are modified from their original shape (i.e., a Kronecker delta) by the CFO and STO of user B, the individual ML detector uses two specific matched filters depending on the offsets Δf_c and τ, which we denote as $M_1^{(k)}$ and $M_2^{(k)}$

$$M_1^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right] = \sum_{n=0}^{\lceil\tau\rceil-1} \left(\tilde{y}^{(k)}[n] - \sqrt{P_{\mathrm{A}}} e^{j\theta_{\mathrm{A}}^{(k)}} e^{j2\pi \frac{n}{N} \bar{s}_{\mathrm{A}}^{(k)}} \right) \tag{5.13}$$

$$\cdot\, e^{-j2\pi \frac{(n+N-\tau)}{N} \left(\bar{s}_{\mathrm{B}}^{(k-1)} - \tau + N\frac{\Delta f_c}{f_s} \right)} e^{-j2\pi\lambda_{\mathrm{STO}} u\left[n - n_{f,1}^{(k)}\right]}, \tag{5.14}$$

$$M_2^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k)}\right] = \sum_{n=\lceil\tau\rceil}^{N-1} \left(\tilde{y}^{(k)}[n] - \sqrt{P_{\mathrm{A}}} e^{j\theta_{\mathrm{A}}^{(k)}} e^{j2\pi \frac{n}{N} \bar{s}_{\mathrm{A}}^{(k)}} \right) \tag{5.15}$$

$$\cdot\, e^{-j2\pi \frac{(n-\tau)}{N} \left(\bar{s}_{\mathrm{B}}^{(k)} - \tau + N\frac{\Delta f_c}{f_s} \right)} e^{-j2\pi\lambda_{\mathrm{STO}} u\left[n - n_{f,2}^{(k)}\right]}. \tag{5.16}$$

The matched filters $M_1^{(k)}$ and $M_2^{(k)}$ for the symbols $\bar{s}_{\mathrm{B}}^{(k-1)}$ and $\bar{s}_{\mathrm{B}}^{(k)}$ can be interpreted as follows. To detect both symbols, the individual ML detector eliminates the presumed contribution $\sqrt{P_{\mathrm{A}}} e^{j\theta_{\mathrm{A}}^{(k)}} e^{j2\pi \frac{n}{N} \bar{s}_{\mathrm{A}}^{(k)}}$ of user A, corrects the CFO Δf_c and phase jump $2\pi\lambda_{\mathrm{STO}}$, and then computes a partial DFT over the first $\lceil\tau\rceil$ and the remaining $N - \lceil\tau\rceil$ samples in the window, respectively.

By substituting in (5.11) the expressions of the matched filters $Y^{(k)}$, $M_1^{(k)}$ and $M_2^{(k)}$ and by adding back the exponential function, we obtain the final metric $\Lambda_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)})$ subject to maximization

$$\begin{aligned} \Lambda_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)}) = \exp\Big[& \sqrt{P_{\mathrm{A}}} \Re\Big(e^{-j\theta_{\mathrm{A}}^{(k)}} Y^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}\right] \Big) \\ & + \sqrt{P_{\mathrm{B}}} \Re\Big(e^{-j\theta_{\mathrm{B}}^{(k-1)}} M_1^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right] \Big) \\ & + \sqrt{P_{\mathrm{B}}} \Re\Big(e^{-j\theta_{\mathrm{B}}^{(k)}} M_2^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k)}\right] \Big) \Big], \end{aligned} \tag{5.17}$$

Each term of $\Lambda_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)})$ can be interpreted as a (partial) DFT matched to each one of the symbols contained in the window $y^{(k)}[n]$.

Yet, the matched filters $M_1^{(k)}$ and $M_2^{(k)}$ provide only partial information on the symbols $s_{\mathrm{B}}^{(k-1)}$ and $s_{\mathrm{B}}^{(k)}$. As illustrated in Fig. 5.2, the symbol $s_{\mathrm{B}}^{(k-1)}$, in red, is connected to the matched filters $M_2^{(k-1)}$ and $M_1^{(k)}$, whereas the symbol $s_{\mathrm{B}}^{(k)}$, in purple, is connected to the matched filters $M_2^{(k)}$ and $M_1^{(k+1)}$. Detecting these symbols using only the information from the k-th window is clearly suboptimal. To efficiently demodulate the symbols $s_{\mathrm{B}}^{(k-1)}$ and $s_{\mathrm{B}}^{(k)}$, the detector must also use the partial information provided by the matched filters $M_2^{(k-1)}$ and $M_1^{(k+1)}$ of the preceding and the following windows, respectively. We hence suggest to first estimate $s_{\mathrm{A}}^{(k)}$ by considering for every candidate symbol $\bar{s}_{\mathrm{A}}^{(k)}$ the most likely symbols $\bar{s}_{\mathrm{B}}^{(k-1)}$ and $\bar{s}_{\mathrm{B}}^{(k)}$ independently of the previous and next window as

$$\hat{s}_{\mathrm{A}}^{(k)} = \arg\max_{\bar{s}_{\mathrm{A}}^{(k)}} \max_{\bar{s}_{\mathrm{B}}^{(k-1)}, \bar{s}_{\mathrm{B}}^{(k)}} \Lambda_{\mathrm{ML}}\left(\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}, \bar{s}_{\mathrm{B}}^{(k)}\right). \tag{5.18}$$

We then decide on $s_{\mathrm{B}}^{(k-1)}$ based on the prior decisions on $s_{\mathrm{A}}^{(k-1)}$ and on the latest decision on $s_{\mathrm{A}}^{(k)}$ as follows

$$\hat{s}_{\mathrm{B}}^{(k-1)} = \arg\max_{\bar{s}_{\mathrm{B}}^{(k-1)}} \left| M_2^{(k-1)}\left[\hat{s}_{\mathrm{A}}^{(k-1)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right] + M_1^{(k)}\left[\hat{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right] \right|. \tag{5.19}$$

The decision on $s_{\mathrm{B}}^{(k)}$ is deferred to the next time step. This rule requires that the receiver keeps in memory the vector containing the N matched filter outputs $M_2^{(k-1)}\left[\hat{s}_{\mathrm{A}}^{(k-1)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right]$ of the previous window.

5.3.3 Removing the Dependency on the Phases $\theta_{\mathrm{A}}^{(k)}$ and $\theta_{\mathrm{B}}^{(k)}$

The individual ML detection rules given in (5.18) and (5.19) require the knowledge of the phases $\theta_{\mathrm{A}}^{(k)}$, $\theta_{\mathrm{B}}^{(k-1)}$ and $\theta_{\mathrm{B}}^{(k)}$. As previously mentioned, we do not assume that consecutive symbols of a user have identical phase offsets, i.e., we allow $\theta_{\mathrm{A}}^{(k-1)} \neq \theta_{\mathrm{A}}^{(k)}$. Although the phase of each user can theoretically be tracked over successive symbols, such phase-tracking scheme strongly increases the complexity of the receiver and its sensitivity

to distortions. In this work, we propose instead to avoid this tracking and to remove the dependency on the phases by marginalizing over them in $\Lambda_{\mathrm{ML}}\left(\bar{\boldsymbol{s}}^{(k)}\right)$.

First, the phases $\theta_{\mathrm{B}}^{(k-1)}$ and $\theta_{\mathrm{B}}^{(k)}$ can be easily marginalized over $[-\pi, \pi]$, leading to a metric that relies on the magnitudes of $M_1^{(k)}$ and $M_2^{(k)}$

$$\begin{aligned}\Lambda'_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)}) &= \iint_{-\pi}^{\pi} \Lambda_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)})\, d\theta_{\mathrm{B}}^{(k-1)}\, d\theta_{\mathrm{B}}^{(k)} \\ &= \exp\left(\sqrt{P_{\mathrm{A}}}\Re\left(e^{-j\theta_{\mathrm{A}}^{(k)}} Y^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}\right]\right)\right) \\ &\cdot I_0\left(\sqrt{P_{\mathrm{B}}}\left|M_1^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right]\right|\right) \\ &\cdot I_0\left(\sqrt{P_{\mathrm{B}}}\left|M_2^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k)}\right]\right|\right),\end{aligned} \tag{5.20}$$

where $I_0(x)$ is the first order modified Bessel function of the first kind [98]. This function is akin to e^x and has a practical closed-form expression. The only phase term remaining is $\theta_{\mathrm{A}}^{(k)}$. Proper consideration of this phase is however more challenging, as it is used in all three terms of $\Lambda'_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)})$.

To lower the complexity of the integral over $\theta_{\mathrm{A}}^{(k)}$, we propose to use an estimate of this phase in the matched filters $M_1^{(k)}$ and $M_2^{(k)}$. In a single-user scenario, the phase θ of a demodulated symbol $\hat{s}$ can be estimated by using the phase of the DFT bin $\hat{s}$, i.e., $\hat{\theta} = \arctan\left(Y\left[\hat{s}\right]\right)$ [60]. We propose to use the same estimator with the DFT $Y^{(k)}$ to obtain an estimate $\hat{\theta}_{\mathrm{A}}^{(k)}$ of the phase of the candidate symbol $\bar{s}_{\mathrm{A}}^{(k)}$. Let $\widetilde{M}_1^{(k)}$ and $\widetilde{M}_2^{(k)}$ be modified versions of the filters $M_1^{(k)}$ and $M_2^{(k)}$ where the variable $\theta_{\mathrm{A}}^{(k)}$ is replaced by the estimate $\hat{\theta}_{\mathrm{A}}^{(k)}$ in the expression of the matched filters $M_1^{(k)}$ and $M_2^{(k)}$. Also, let $\widetilde{\Lambda}(\bar{\boldsymbol{s}}^{(k)})$ be a modified version of the function $\Lambda'_{\mathrm{ML}}(\bar{\boldsymbol{s}}^{(k)})$ where the filters $\widetilde{M}_1^{(k)}$ and $\widetilde{M}_2^{(k)}$ are used instead of $M_1^{(k)}$ and $M_2^{(k)}$, respectively. A single occurrence of $\theta_{\mathrm{A}}^{(k)}$ remains in $\widetilde{\Lambda}(\bar{\boldsymbol{s}}^{(k)})$, which is marginalized similarly to $\theta_{\mathrm{B}}^{(k-1)}$ and

$\theta_{\mathrm{B}}^{(k)}$ to yield the final detection metric

$$\begin{aligned}\Lambda(\bar{\boldsymbol{s}}^{(k)}) &= \int_{-\pi}^{\pi} \widetilde{\Lambda}(\bar{\boldsymbol{s}}^{(k)})\, d\theta_{\mathrm{A}}^{(k)} \\ &= I_0\left(\sqrt{P_{\mathrm{A}}}\left|Y^{(k)}\left(\bar{s}_{\mathrm{A}}^{(k)}\right)\right|\right) \\ &\cdot I_0\left(\sqrt{P_{\mathrm{B}}}\left|\widetilde{M}_1^{(k)}(\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)})\right|\right) \\ &\cdot I_0\left(\sqrt{P_{\mathrm{B}}}\left|\widetilde{M}_2^{(k)}(\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k)})\right|\right).\end{aligned} \tag{5.21}$$

Due to the use of the non-ideal estimate $\hat{\theta}_{\mathrm{A}}^{(k)}$, the detector of (5.21) no longer follows the maximum-likelihood criterion. The accuracy of $\hat{\theta}_{\mathrm{A}}^{(k)}$ depends not only on the noise level σ^2, but also on the contribution of user B that might overlap in the frequency domain with the symbol of user A. The accuracy of this approximation is therefore better when $P_{\mathrm{A}} > P_{\mathrm{B}}$, i.e., when the synchronized user is the strongest user. To maintain a decent interference cancellation in the matched filters $\widetilde{M}_1^{(k)}$ and $\widetilde{M}_2^{(k)}$, whose effectiveness strongly depends on the quality of the estimate $\hat{\theta}_{\mathrm{A}}^{(k)}$, we assume in the remainder of this thesis that the receiver is always synchronized to the strongest user. In Section 5.5, we show that the synchronization stage of a practical receiver can always take care of this requirement by allowing the receiver to re-synchronize to an incoming user if that user has a higher received power than an active first user.

5.3.4 Pruning the Set of Candidate Symbols $\bar{s}_{\mathrm{A}}^{(k)}$

Computing $\hat{s}_{\mathrm{A}}^{(k)}$ with (5.18) requires to evaluate a significant number of metrics $\Lambda(\bar{\boldsymbol{s}}^{(k)})$. Per candidate symbol $\bar{s}_{\mathrm{A}}^{(k)}$, N outputs for each of the matched filters $M_1^{(k)}$ and $M_2^{(k)}$ need to be evaluated. Since there are also N candidates for $\bar{s}_{\mathrm{A}}^{(k)}$, the complexity for demodulating a window thus grows quadratically with N and becomes quickly prohibitive for large spreading factors. We previously explained that synchronizing to the strongest user improves the estimate $\hat{\theta}_{\mathrm{A}}^{(k)}$. We now show that the synchronization to the strongest user can also be used to drastically reduce the complexity of the two-user detector.

According to (5.5), the contribution of user A to the dechirped signal

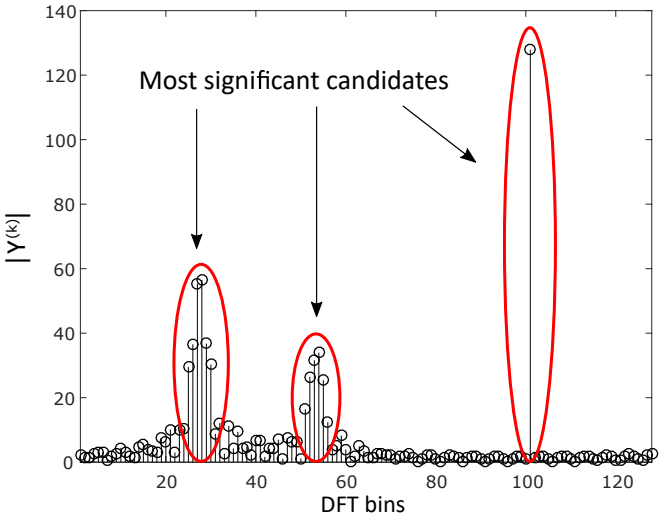


Fig. 5.3 Magnitude of the DFT $Y^{(k)}$ for $s_{\mathrm{A}}^{(k)} = 100$, $s_{\mathrm{B}}^{(k-1)} = 85$, $s_{\mathrm{B}}^{(k)} = 59$, SF $= 7$, $\tau = 32.5$, $\Delta f_c = 0$ and $P_{\mathrm{A}} = P_{\mathrm{B}}$. Due to the STO τ, the symbols of user B are shifted in frequency and scattered on several DFT bins.

$\tilde{y}^{(k)}[n]$ is a single complex tone of frequency $\frac{s_{\mathrm{A}}^{(k)}}{N}$. On the contrary, the contribution of user B is no longer a single complex tone due to the effective carrier frequency offset Δf_c and the sampling time offset τ. For the non-synchronized user, the offset components L_{STO}, λ_{STO} and λ_{CFO} scatter the energy of the symbols $s_{\mathrm{B}}^{(k-1)}$ and $s_{\mathrm{B}}^{(k)}$ over several frequency bins [42]. As illustrated in Fig. 5.3, the DFT of $\tilde{y}^{(k)}[n]$ contains a Kronecker delta corresponding to the symbol $s_{\mathrm{A}}^{(k)}$, along with two bell-shaped functions centered around the frequencies $\frac{s_{\mathrm{B}}^{(k-1)} - \tau + N\frac{\Delta f_c}{f_s}}{N}$ and $\frac{s_{\mathrm{B}}^{(k)} - \tau + N\frac{\Delta f_c}{f_s}}{N}$. Both time and frequency offsets thus modify the spectral representation of the symbols from user B. We show in Section 5.4 that the presence of an STO and a CFO between the users is actually beneficial to the performance of the two-user detector.

Since our receiver synchronizes to the strongest user, it is very likely at reasonable SNR levels that the DFT bin containing the symbol $s_{\mathrm{A}}^{(k)}$ belongs to the limited set of bins with significant magnitudes. We therefore propose

to reduce the complexity of (5.18) by selecting as candidates $\bar{s}_{\mathrm{A}}^{(k)}$ only the K bins with the largest magnitudes $\left|Y^{(k)}\left(\bar{s}_{\mathrm{A}}^{(k)}\right)\right|$ among the N bins of the DFT $Y^{(k)}$. This rule reduces the total amount of matched filters $M_1^{(k)}$ and $M_2^{(k)}$ to evaluate per window from N^2 to KN.

The pruning of the set of candidate symbols $\bar{s}_{\mathrm{A}}^{(k)}$ significantly reduces the complexity of the receiver. Per candidate symbol $\bar{s}_{\mathrm{A}}^{(k)}$, N matched filter outputs $M_1^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k-1)}\right]$ and $M_2^{(k)}\left[\bar{s}_{\mathrm{A}}^{(k)}, \bar{s}_{\mathrm{B}}^{(k)}\right]$ are evaluated, each requiring $\mathcal{O}(N)$ complex operations. Since the N outputs of the matched filter $Y^{(k)}$ can be computed in $\mathcal{O}(N \log N)$ operations using an FFT, the computational complexity of the receiver is dominated by the evaluation of the matched filters $M_1^{(k)}$ and $M_2^{(k)}$. The demodulation of a window of N samples hence implies only $\mathcal{O}(KN^2)$ complex operations, compared to a complexity of $\mathcal{O}(N^3)$ without the pruning. We observed in simulation trials significant performance gains for $K = 2$ compared to $K = 1$, independently of the spreading factor. Selecting $K > 5$ however does not further improve the performance for either user. In practical receivers, values of K between 2 and 5 should hence be used, depending on the available computational resources.

5.4 Performance Evaluation of the Two-User Detector

In this section, we provide simulation results for the symbol error rate (SER) of both the stronger and the weaker colliding users using the detector derived in the previous section. The principal parameters that impact the SER are the STO τ between the users, the fractional part λ_{CFO} of the effective CFO, the difference $\Delta P = P_{\mathrm{A}} - P_{\mathrm{B}}$ of the received powers, and the spreading factor SF. We analyze and discuss the impact of these parameters through Monte-Carlo simulations.

The simulations are conducted in MATLAB as follows. For every trial, 10′000 collisions with overlapping payloads of $N_P = 32$ random LoRa symbols each are transmitted following the baseband model defined in (5.3). Since the length of LoRa payloads dominates over the preamble length, we only consider in this section collisions between two payloads. As Nyquist-rate sampling is simulated, the signal bandwidth has no influence on the SERs. The transmitted symbols are demodulated using

the detection rules (5.18) and (5.19) with the metric $\Lambda(\bar{\boldsymbol{s}}^{(k)})$ from (5.21). The simulations in this section also assume perfect synchronization, i.e., the received power, CFO, and STO of each user are known. Results obtained without these idealized conditions and using a practical synchronization algorithm are presented in Section 5.5. Furthermore, we use the simplification described in Section 5.3 and we choose the symbol candidates $\bar{s}_{\mathrm{A}}^{(k)}$ for user A only from a limited set of the $K = 5$ most significant DFT bins. Unless specified otherwise, the following results are obtained for SF $= 7$. However, we show in Section 5.4.4 that our detector benefits from the spreading gain of the LoRa modulation (about 2.5 dB per SF) in the same way as a single-user receiver. The results obtained for SF $= 7$ can hence serve as approximations for other SFs by appropriately subtracting the corresponding SNR offsets.

5.4.1 Impact of Sampling Time Offset

In Fig. 5.4, we compare four scenarios for different STO values and SF $= 7$, i.e., $N = 128$. In particular, we compare the worst-case scenario with $\tau = 0.0$ (i.e., two users fully synchronized), the best-case scenario with $\tau = 64.5$ (i.e., $\tau = N/2 + 0.5$), and two intermediate cases with $\tau = 16.0$ (i.e., $\tau = N/8$) and $\tau = 16.25$. The power difference between the two users is $\Delta P = 3$ dB. The SERs in the different scenarios are shown for both the strong and the weak user. The thick gray lines show the SER for the typical non-coherent single-user receiver (2.7) in the absence of interference. As expected, the performance with interference, both for the two-user and the multi-user receiver, is always worse than the no-interference case depicted by the gray lines. We note here that the two-user receiver demodulates the strongest user (solid curves) with error rates that are similar to the single-user detector in (2.7), but in the presence of interference. More interestingly, we observe that our two-user receiver allows even the weaker user (dashed curves) to be demodulated with useful SER values, while the typical single-user receiver would not be able to demodulate the weaker user at all.

We now discuss in detail the influence of the STO $\tau = L_{\mathrm{STO}} + \lambda_{\mathrm{STO}}$ on the demodulation. Considering first the impact of the integer offset L_{STO}, we clearly observe in Fig. 5.4 that a larger integer STO (up to $N/2$) reduces the SERs of both users. For the strongest user, the required SNR to attain a 10^{-3} SER, is 2.5 dB lower for $\tau = 16.0$ (solid orange curve) compared

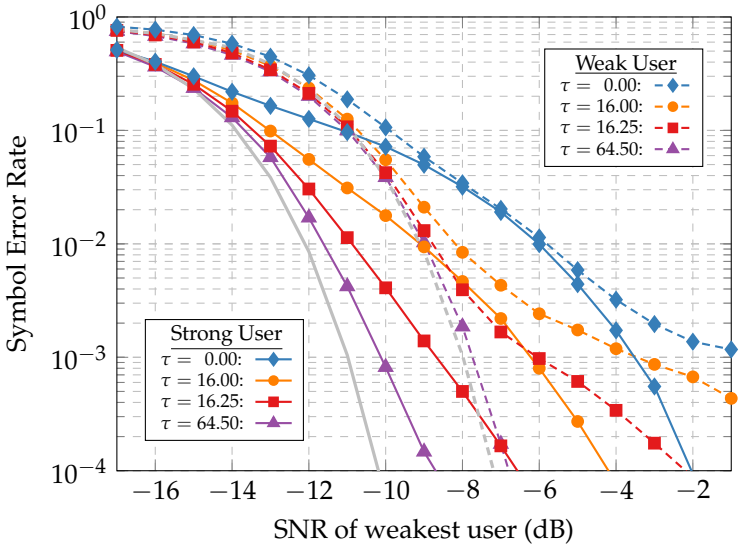


Fig. 5.4 SERs of both users for different values of τ with SF $= 7$, $\Delta f_c = 0$ and $\Delta P = 3$ dB. The thick gray lines show the SER for the non-coherent single-user receiver (2.7) in the absence of interference ($P_{\mathrm{B}} = -\infty$ dB) [43].

to $\tau = 0.0$ (solid purple curve). A similar behavior can be observed for the fractional part of the STO. We observe that a fractional STO (up to 0.5) can improve the error rate at ranges that are comparable to large integer values of STO and very close to the no-interference case. For example, the weakest user for $\tau = 16.25$ (i.e., $\lambda_{\mathrm{STO}} = 0.25$, dashed red curve) reaches a SER of 10^{-3} at -6 dB SNR, whereas for $\tau = 16.0$ ($\lambda_{\mathrm{STO}} = 0$, dashed orange curve) the required SNR is -4 dB.

The aforementioned observations indicate that the more the interfering users are desynchronized in time, the easier it is for the two-user receiver to separate and demodulate them. This effect can be explained by the contribution of each user to the DFT of the dechirped signal. While the contribution of the strongest and synchronized user is always a Kronecker delta, the spectral representation of the contribution of the weakest user depends on the STO τ. For $\tau = 0$, the symbols of both users are Kronecker deltas in the frequency domain and the matched filter $M_1^{(k)}$ is identical to the DFT $Y^{(k)}$. In such a time aligned case, it is difficult to distinguish the symbols of the two users as they are entirely correlated to each other. In the presence

of an STO $\tau \neq 0$, the contribution of the weakest user is no longer a single peak, but two bell-shaped functions scattered across several DFT bins, as introduced in Section 5.3 and observed in Fig. 5.3. Since the symbols of both users become less correlated in the presence of an STO, it is easier for the maximum likelihood detector to separate the contribution of each user. Therefore, integer or fractional STOs close to $N/2$ or 0.5, respectively, significantly improve the performance of the proposed two-user detector. The extent of this behavior is also shown in Fig. 5.4, where, to demodulate the weakest user at a 10^{-3} SER, the best-case scenario with $\tau = 64.5$ (dashed blue curve) requires a 7 dB lower SNR than the worst-case scenario with $\tau = 0.0$ (dashed purple curve).

5.4.2 Impact of Carrier Frequency Offset

Similar to the STO, the effective CFO Δf_c between the users also modifies the spectral representation of the symbols sent by user B [60]. Yet, contrary to the STO, only the fractional part λ_{CFO} modifies the spectral representation of the symbols of user B. An integer CFO L_{STO} only shifts the symbols of user B by an integer number of DFT bins, and therefore does not improve the detection performance.

The SERs for scenarios with SF $= 7$ and with different values of λ_{CFO} and λ_{STO} are presented in Fig. 5.5. Compared to the baseline $\lambda_{\text{CFO}} = 0.0$ with $\tau = 16.0$, an offset $\lambda_{\text{CFO}} = -0.5$ decreases the required SNR to attain a 10^{-3} SER by 3.5 dB for the strongest user. For the weakest user, the required SNR for the same SER is reduced by 3 dB. We have shown in Chapter 3 that the contributions of the fractional STO and CFO, although similar, do not modify the spectral representation of LoRa symbols in an identical way. Compared to the CFO, the fractional STO additionally induces a phase jump of $2\pi\lambda_{\text{STO}}$ when the LoRa symbol folds in frequency, as modeled in (5.5). As a consequence, the performance of the detector depends on both values of λ_{STO} and λ_{CFO}, and is not only defined by the difference of the two $-\lambda_{\text{STO}} + \lambda_{\text{CFO}}$. This behavior is illustrated in Fig. 5.5 with the scenario $\lambda_{\text{CFO}} = 0.5$ and $\lambda_{\text{STO}} = 0.5$. Although the fractional offsets cancel each other in the term $s_{\text{B}}^{(k)} - \tau + N\frac{f_c}{f_s}$ of (5.5), the phase jump of $2\pi\lambda_{\text{STO}}$ still enables important performance gains for the two-user detector compared to the worst-case fully time and frequency aligned scenario.

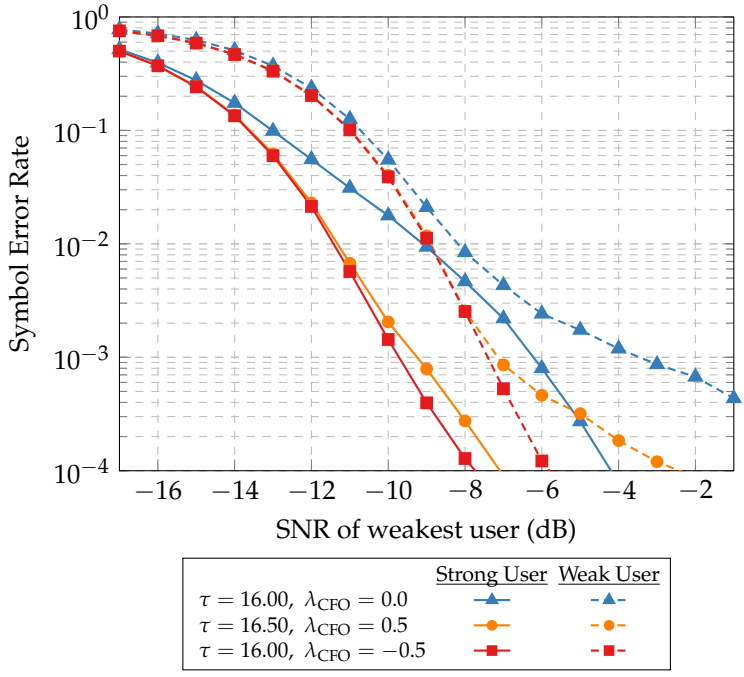


Fig. 5.5 SERs of both users for different values of τ and λ_{CFO} with SF $= 7$ and $\Delta P = 3$ dB.

5.4.3 Impact of the Received Powers Difference

We subsequently analyze the impact of the difference $\Delta P = P_A - P_B$ in the received powers of the colliding users. Fig. 5.6 shows the SERs of both the strong and the weak user for $\Delta P = 6, 3, 1.5$ and 0 dB, with SF $= 7$. We first discuss the effect of ΔP on the demodulation of the strongest user, and use the non-coherent single-user receiver in the absence of interference (2.7) as a reference. The results are presented for a fixed P_B, i.e., increasing ΔP amounts to increasing P_A. Hence, for the four values of ΔP studied, the baseline SER is inherently shifted horizontally by the corresponding increase of power P_A. For identical received powers ($\Delta P = 0$ dB), the SERs of both users become equivalent and level off near a SER of 10^{-3}. With $\Delta P = 1.5$ dB, the two-user detector reaches a SER value of 10^{-3} with an SNR of -6.5 dB, whereas the detector of (2.7) in a single-user scenario requires a 3-dB lower SNR for the same target. For $\Delta P = 3$ dB and $\Delta P = 6$ dB, the SNR difference between the two scenarios reduces to 2.5 dB

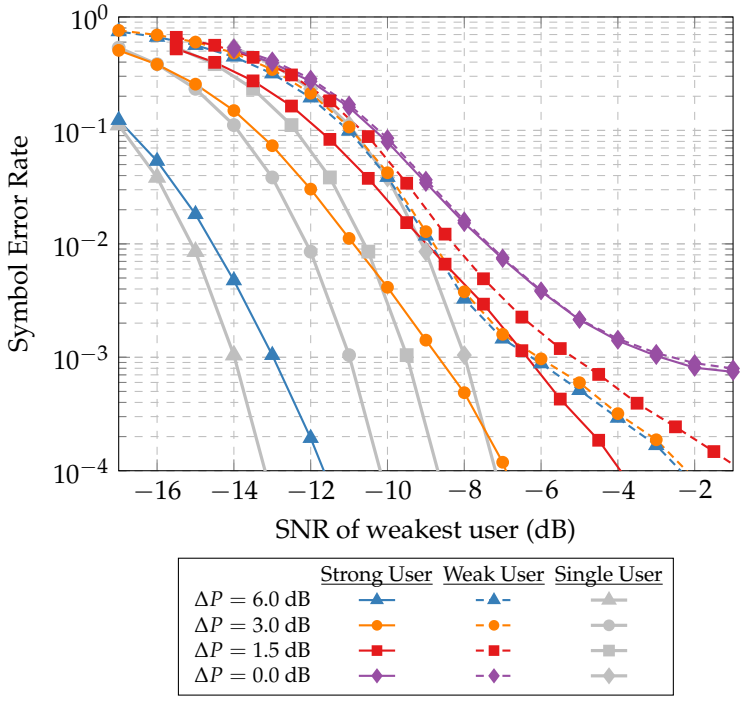


Fig. 5.6 SERs of both users for different values $\Delta P = P_\mathrm{A} - P_\mathrm{B}$ with SF $= 7$, $\tau = 16.25$ and $\Delta f_c = 0$. The thick gray lines show the SER of the single-user receiver (2.7) in the absence of interference ($P_\mathrm{B} = -\infty$ dB) for the SNRs of the strong user.

and 1 dB, respectively. These results illustrate that, for an increasing ΔP, the SER of the strongest user converges to the SER of the no-interference case.

Interestingly, provided a small difference of received powers between the users, the value of ΔP itself does not significantly affect the error rates for the weakest user. We observe for example that the performance of the weak user for $\Delta P = 3$ dB and $\Delta P = 6$ dB is almost identical and similar to the case with $\Delta P = 1.5$ dB. The error rates however increase for $\Delta P = 0$ dB. The two-user detection rule (5.21) notably relies on the energy of the symbols to identify their corresponding user. When both users exhibit close received powers, the two-user detector is more likely to confuse their respective demodulated symbols. In this situation, the uncoded receiver is unable to correctly assign some demodulated symbol sequences to their

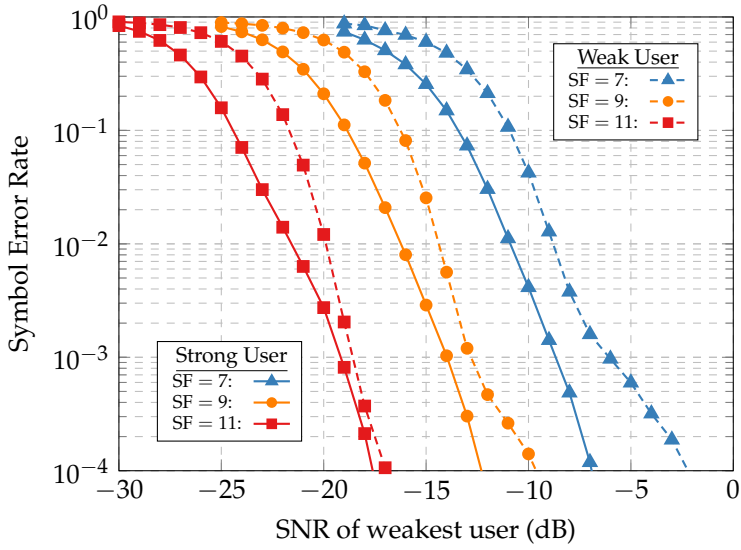


Fig. 5.7 SERs of both users for different spreading factors with $\Delta P = 3$ dB, $L_{\mathrm{STO}} = N/8$, $\lambda_{\mathrm{STO}} = 0.25$ and $\Delta f_c = 0$.

respective users, even in high SNR regions.

Overall, the proposed two-user receiver yields useful SER values for the weak user when at least a small difference of received power ΔP exists. This observed behavior is a very positive result, since in typical single-user LoRa receivers the weak user is always lost.

5.4.4 Impact of Different Spreading Factors

We finally discuss the impact of the spreading factor on the performance of the two-user detector. Fig. 5.7 shows the SER of both the strong and the weak user for three spreading factors SF $= 7, 9$ and 11. For the non-coherent single-user receiver (2.7), every increment of the SF by one enhances the spreading gain of the modulation by approximately 2.5 dB [60]. A similar behavior can be observed for the detection of the strongest user, as increasing the spreading factor by two reduces the SNR requirements by approximately 5 dB for all SERs. Regarding the weakest user, the required SNR to attain a 10^{-3} SER when increasing the spreading factor from 7 to 9 decreases by 7 dB. This 7 dB improvement not only stems from the in-

creased spreading gain, but also from more accurate estimates $\hat{\theta}_{\mathrm{A}}^{(k)}$.

Since the number of different LoRa symbols $N = 2^{\mathrm{SF}}$ increases with the spreading factor, the probability that, during window k, symbol $s_{\mathrm{A}}^{(k)}$ of the strong user overlaps in frequency with symbols $s_{\mathrm{B}}^{(k-1)}$ or $s_{\mathrm{B}}^{(k)}$ of the weak user also decreases. As a consequence, the phase estimator for $\hat{\theta}_{\mathrm{A}}^{(k)}$ is less prone to suffer from interference coming from the weakest user. For $\mathrm{SF} = 7$, an inflexion point in the error rate can be observed at a SER of $0.3 \cdot 10^{-2}$, i.e., for a -7 dB SNR. At lower SNR levels, the SER is dominated by errors due to noise, whereas for SNRs greater than -7 dB, the inaccuracies of the estimates $\hat{\theta}_{\mathrm{A}}^{(k)}$ prevent the receiver from correctly removing the contribution of user A before detecting the symbols of user B. By increasing the spreading factor to 9 and 11, this inflexion point drops to a SER of $0.9 \cdot 10^{-3}$ and $0.4 \cdot 10^{-3}$, respectively, and inaccurate estimates of $\hat{\theta}_{\mathrm{A}}^{(k)}$ become less relevant.

5.5 Software-Defined Radio Implementation

To demonstrate the practicality of the proposed two-user detector in a complete receiver chain, we implemented it in the GNU Radio software-defined radio environment, along with an adaptive multi-user synchronization algorithm that is robust to same-SF interference. The two-user detector derived in Section 5.3 requires estimates of the received power, STO, and CFO of each user. Conventional single-user synchronization algorithms, such as the algorithm described in Chapter 3, are not capable of accurately estimating these parameters in the presence of interference from another user. We hence designed a specific algorithm to this end.

Both the two-user detector and the synchronization algorithm are implemented in an out-of-tree GNU Radio module. This C++ implementation is open-source and available online [99]. In the following, we first give an overview of the overall receiver architecture, and then explain the multi-user synchronization algorithm in detail. We then perform Monte-Carlo simulations with the GNU Radio implementation to evaluate the impact of the synchronization algorithm on the performance of the two-user detector. The GNU Radio multi-user implementation is subsequently experimentally validated in a testbed with two interfering LoRa transmitters. Finally, we briefly discuss the complexity of the proposed receiver.

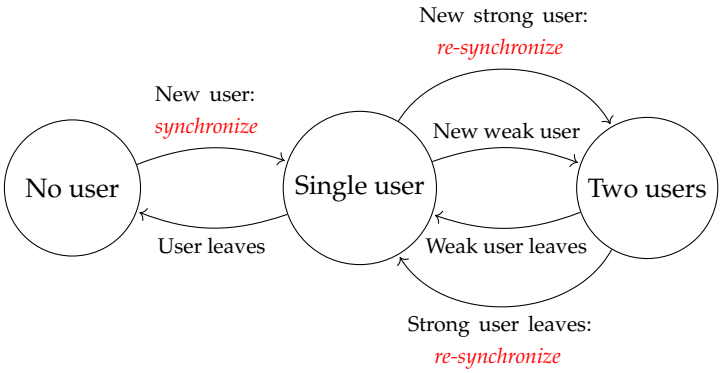


Fig. 5.8 Finite state machine representation of the receiver. The receiver always synchronizes to the strongest user.

5.5.1 Architecture of the Implementation

As previously explained, a LoRa user can start or stop transmitting at any time during the transmission of another user. Due to the nature of our two-user detector which demodulates simultaneously the symbols of two users, a gateway embedding this detector must be capable of tracking in real time the arrival of a second user during an ongoing transmission. To this end, the implementation of the two-user receiver follows a three-state finite state machine (FSM) which is illustrated in Fig. 5.8. Each state corresponds to the current number of colliding users, up to two.

The first stage of the receiver is a multi-user synchronization algorithm. Compared to the synchronization procedures of the multi-user receivers from [90, 91, 92, 93, 94, 95], we propose in this section an algorithm that exploits the quasi-orthogonality between upchirps and downchirps in the LoRa preamble to estimate the parameters of an arriving user in spite of another ongoing, colliding transmission. The input of the synchronization algorithm is an oversampled version $\bar{y}[n]$ of the received signal, sampled at the frequency $f_s' = RB$, where R is the oversampling factor. In the no-user or single-user states, the algorithm is constantly running to detect the preamble of a potential arriving user. When a new user is detected, the receiver estimates its power, CFO and STO using the oversampled received signal. In the presence of a single first user, the receiver synchronizes in time and frequency to this user. The synchronization comprises finding the first sample of the payload in the oversampled signal $\bar{y}[n]$, decimating

this signal by a factor R to the Nyquist frequency $f_s = B$, and correcting the estimated CFO, which finally yields the received signal $y[n]$. The typical non-coherent detector of (2.7) is then used to demodulate the payload symbols in $y[n]$.

Upon the arrival of a second user, the receiver switches to the two-user detection rules (5.18) and (5.19) that maximize the metric $\Lambda(\bar{\boldsymbol{s}}^{(k)})$ with $K = 5$. If the new user has a higher received power than the active first user, the receiver re-synchronizes in time and frequency to this new user. Maintaining the synchronization to the strongest user is an important condition to achieve decent interference cancellation in the two-user detector, as explained in Section 5.3. The synchronized signal is split into windows of N samples, and each window is sent to the demodulation stage. Since the matched filters $\widetilde{M}_1^{(k)}$ and $\widetilde{M}_2^{(k)}$ require the knowledge of the effective CFO Δf_c and the STO τ, the synchronization algorithm is always ahead of the demodulation by $N_{\text{pr}} + 4.25$ symbols, i.e., the length of the preamble. This buffering enables the receiver to detect the new user and estimate its parameters before demodulating the interfering symbols of the already present user.

5.5.2 Estimation of the Parameters and Preamble Detection

Contrary to synchronization algorithms for single-user receivers, the synchronization algorithm of our two-user receiver has to detect and estimate the parameters of a new user even in the presence of a colliding user. We already explained in Chapter 3 that, for a single-user receiver, demodulating the upchirps and downchirps in the preamble (see Fig. 2.4) allows an unsynchronized receiver to estimate the integer parts of the CFO and STO. In a noiseless environment, the symbols s_{up} and s_{down} of an upchirp and a downchirp demodulated with the single-user detector from (2.7) yield

$$\begin{aligned} s_{\text{up}} &= (L_{\text{CFO}} - L_{\text{STO}}) \bmod N, \\ s_{\text{down}} &= (L_{\text{CFO}} + L_{\text{STO}}) \bmod N. \end{aligned} \tag{5.22}$$

The integer offsets L_{CFO} and L_{STO} are estimated by summing and subtracting s_{up} and s_{down}, respectively, and dividing the result by two. However, the conventional single-user detector is typically not able to correctly demodulate these symbols in the presence of strong interference. Hence, to perform a robust estimation of the CFO and STO, we propose instead to leverage the repetition of the upchirps as well as the almost orthogonal re-

lationship between the 2.25 downchirps in the preamble of the second user and the modulated symbols of the first user [100].

The multi-user synchronization algorithm is divided into three stages. The first stage processes the upchirps of the preamble, the second processes the downchirps, and a final stage performs the preamble detection and the parameter estimation.

Estimation of s_{up} and λ_{CFO} using the Upchirps

Due to the potential presence of interferers, it is not reliable to demodulate s_{up} with the single-user detector of (2.7). To estimate s_{up}, the proposed algorithm downsamples $\bar{y}[n]$ by a factor R, and distributes the decimated preamble signal into successive windows of N samples. Each window is then dechirped with the downchirp $x_0^*[n]$. Let $\bar{Y}^{(k)}$ be the DFT of the dechirped signal in the k-th window. In the absence of interference, $\bar{Y}^{(k)}$ for an upchirp contains a single bell-shaped function centered around $s_{\text{up}} = (L_{\text{CFO}} - L_{\text{STO}}) \bmod N$, according to (3.5). While a new user sends a preamble containing N_{pr} identical upchirps $x_0[n]$, it is highly likely that the payload symbols sent by a colliding same-SF user change over consecutive windows. We suggest to use the temporal repetition of the same upchirp in the preamble to filter out the time-varying symbols from an interfering user. In particular, we multiply the magnitudes of $\bar{Y}^{(k)}$ over $N_{\text{pr}} - 1$ consecutive windows and we perform the symbol decision on this product

$$\hat{s}_{\text{up}}^{(k)} = \arg\max_{s} \prod_{i=k}^{k+N_{\text{pr}}-1} \left| \bar{Y}^{(i)}\,[s] \right|. \tag{5.23}$$

The multiplication increases the magnitude of the frequency bins containing the identical symbols s_{up}, but it attenuates the energy of the payload symbols from the interfering users, which are located in different bins.

The preamble upchirps are then also used to estimate the fractional carrier frequency offset λ_{CFO}. We here re-use the estimator of (3.9), i.e., this offset can be estimated by multiplying the DFT bin $\bar{Y}^{(k)}\left[\hat{s}_{\text{up}}^{(k)}\right]$ with the complex conjugate of the same bin from the previous upchirp, and looking at the phase of the product. In a two-user scenario however, there is a non-negligible probability that a symbol from the interferer overlaps in frequency with an upchirp of the new user, hence corrupting the estimate. Thus, to minimize the estimation errors due to noise or interference, we compute the estimate $\widehat{\lambda}_{\text{CFO}}^{(k)}$ over the same $N_{\text{pr}} - 1$ upchirps used to evalu-

ate $\hat{s}_{\text{up}}^{(k)}$ with the preamble upchirps

$$\widehat{\lambda}_{\text{CFO}}^{(k)} = \frac{1}{2\pi} \arg \left(\sum_{i=k+1}^{k+N_{\text{pr}}-2} \bar{Y}^{(i)} \left[\hat{s}_{\text{up}}^{(k)} \right] \bar{Y}^{*(i-1)} \left[\hat{s}_{\text{up}}^{(k)} \right] \right). \tag{5.24}$$

Estimation of s_{down} and λ_{STO} using the Downchirps

To estimate $s_{\text{down}} = (L_{\text{CFO}} + L_{\text{STO}}) \bmod N$ using the preamble downchirps, we suggest to perform a matched filtering on the whole received preamble. When demodulating a LoRa symbol, a matched filter with N modulated downchirps is equivalent to the combination of the dechirping stage and the DFT stage. An oversampled matched filter however allows to simultaneously estimate the fractional time offset λ_{STO}.

The proposed filter $H^{(k)}$ looks for two consecutive unmodulated downchirps in the time windows $k-1$ and k-th of RN samples each from the oversampled signal $\bar{y}[n]$. This filter also corrects for the fractional CFO, using the estimate $\widehat{\lambda}_{\text{CFO}}^{(k-N_{pr}-2)}$ obtained $N_{\text{pr}}-2$ windows before

$$\begin{aligned} H^{(k)}[n,p] = \frac{1}{N} \sum_{m=0}^{2N-1} & \bar{y}[R(Nk+m-n)-p] \\ & \cdot \exp\left(j2\pi \left(\frac{m^2}{2N} - \frac{m}{2} - \frac{m}{N} \widehat{\lambda}_{\text{CFO}}^{(k-N_{pr}-2)} \right) \right), \end{aligned} \tag{5.25}$$

where $p \in \{0, \ldots, R-1\}$.

Let P be the received power of the arriving user. In the absence of interferers and noise, the filter reaches a maximum output magnitude of $2\sqrt{P}$ when it is aligned to the two downchirps. In this case, the peak at $2\sqrt{P}$ is attained for $n_{\text{max}}^{(k)} = s_{\text{down}}$ and $p_{\text{max}}^{(k)} = \lfloor R\lambda_{\text{STO}} \rceil$. Similarly, when only one downchirp is aligned, i.e., one window before or after the maximum at $2\sqrt{P}$, the magnitude of the output attains $\sqrt{P}$, for the same values of n and p. As the filter $H^{(k)}[n,p]$ is matched to downchirps, symbols from interferers, which are modulated with upchirps, are almost orthogonal and can hence be seen as AWGN [100]. This property is illustrated in Fig. 5.9.

Preamble Detection

The preamble detection is performed using the outputs $H^{(k)}[n,p]$ of the matched filter. The indices $n_{\text{max}}^{(k)}$, $p_{\text{max}}^{(k)}$ of the largest absolute value of the

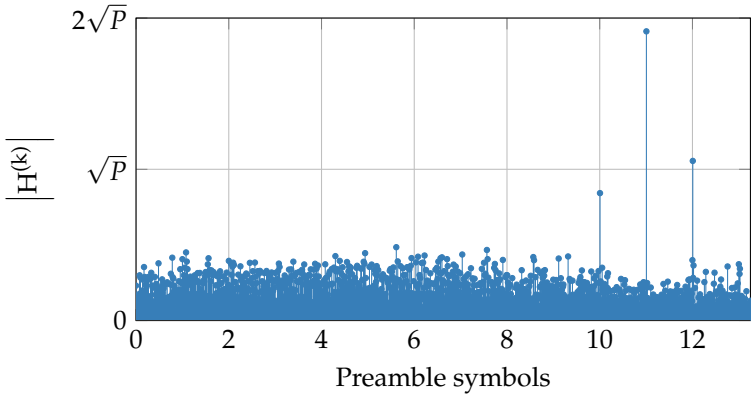


Fig. 5.9 Output of the matched filter $H^{(k)}$ on a received signal containing an entire preamble from a new user and random payload symbols from a 3-dB stronger interfering user with the same SF. The matched filter yields three distinct peaks of heights $\sqrt{P}$, $2\sqrt{P}$, $\sqrt{P}$ spaced by RN samples when the downchirps of the preamble are processed. Conversely, the energy of the preamble upchirps and interfering payload symbols is spread over the whole filter output.

matched filter output $\left|H^{(k)}[n,p]\right|$ are identified for each window. The algorithm detects a preamble when a succession of peaks of height $\sqrt{P}$, $2\sqrt{P}$, $\sqrt{P}$ spaced by RN samples is observed, such that the following conditions are met:

$$\left|H^{(k-1)}[n_{\max}^{(k)}, p_{\max}^{(k)}] - \sqrt{\widehat{P}^{(k)}}\right| < T^{(k)}, \tag{5.26}$$

$$\left|H^{(k+1)}[n_{\max}^{(k)}, p_{\max}^{(k)}] - \sqrt{\widehat{P}^{(k)}}\right| < T^{(k)}, \tag{5.27}$$

where $T^{(k)} = \sqrt{\widehat{P}^{(k)} + \frac{\sigma^2}{N} + \frac{2}{\pi}}$ is a threshold that depends on the noise level and the spreading factor, and $\widehat{P}^{(k)} = \left(0.5H^{(k)}\left[n_{\max}^{(k)}, p_{\max}^{(k)}\right]\right)^2$ is the estimated received power.

Upon detection of a preamble, the symbols $\hat{s}_{\text{up}}^{(k-N_{\text{pr}}-2)}$ and $\hat{s}_{\text{down}}^{(k)} = n_{\max}^{(k)}$ are used to estimate L_{CFO} and L_{STO} following (5.22). The values $\widehat{\lambda}_{\text{STO}}^{(k)} = \lfloor \frac{p_{\max}^{(k)}}{R} \rceil$ and $\widehat{\lambda}_{\text{CFO}}^{(k-N_{\text{pr}}-2)}$ are selected as estimates of the fractional STO and CFO, respectively. It is worth noting that the proposed pream-

ble detection rule is not guaranteed to work when the downchirps of two preambles collide, i.e., when the absolute time offset between the starts of the packets is smaller than 2.25 symbols. The probability that this scenario arises when two users collide is however negligible in practice, since the preamble is 12.25 symbols long and LoRaWAN frames carry at least 33 payload symbols for SF = 7 and 23 symbols for SF = 12 [33, 27].

5.5.3 Detector Performance with Synchronization Algorithm

In the following, we evaluate in simulation the performance of our two-user detector when the parameters of the colliding users are estimated using the proposed multi-user synchronization algorithm. To analyze the performance of the two-user detector under non-ideal, but realistic synchronization conditions, we first focus on the per-user SER when the receiver jointly demodulates two superimposed users after synchronizing with the proposed algorithm. The impact of the synchronization on the SERs is then further analyzed by studying the root mean square errors (RMSE) of the STO, the CFO and the received power estimates.

The following results are obtained by simulating collisions in an AWGN channel between two LoRa transmitters using the GNU Radio software-defined-radio environment. In each Monte-Carlo trial, both transmitters send a LoRa packet with a preamble of 12.25 symbols and 32 random data symbols to the receiver. $70'000$ trials are carried out per SNR level. To clearly illustrate the impact of the synchronization stage on the two-user detector, STO and CFO values that maximize the detection performance for both users have been chosen. The starting time of the transmission of the second user is delayed by 15 symbols and $\tau = 64.5$ samples with respect to the first user. The CFO between the two users is set such that $L_{\text{CFO}} + \lambda_{\text{CFO}} = 20.5$ and the transmit power of the second user is chosen 3-dB stronger than the power of the first user. AWGN is simulated at the receiver, and different SNR levels are obtained by sweeping the transmit powers of both users together. The receiver samples the received signal with an oversampling factor of $R = 8$. For the first user, only its payload symbols are subject to interference, whereas the colliding symbols of the second user comprise both preamble and payload symbols. Only the 15 overlapping payload symbols are used for the computation of the SERs. Trials in which the packet of the weakest user is not detected are not taken into account, as the two-user detector is not enabled in this scenario. For

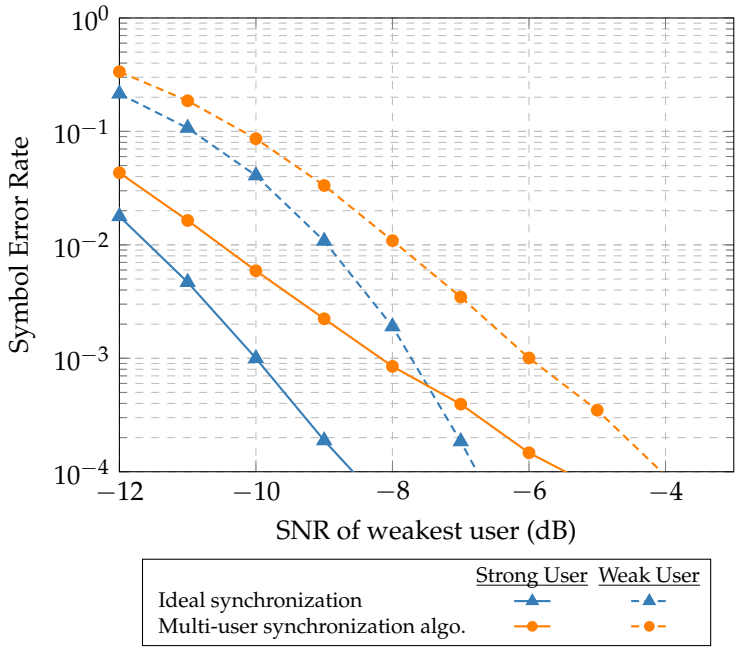


Fig. 5.10 SERs of both users for the proposed two-user detector with perfect synchronization and our multi-user synchronization algorithm for $\tau = 64.5$, $L_{\text{CFO}} = 20$, $\lambda_{\text{CFO}} = 0.5$, SF $= 7$ and $\Delta P = 3$dB.

all simulated SNR levels and for both users, the probability of preamble misdetection is on average at least one order of magnitude smaller than the frame error rate.

Fig. 5.10 shows the SERs of both users with ideal synchronization and when the user parameters are estimated with the proposed multi-user synchronization algorithm. These SERs do not include the demodulation of the symbols that do not collide with another frame (i.e., the single-user state in the FSM of Fig. 5.8), since the proposed two-user detector is not used for these symbols that have an overall much lower SER. For a target SER of 10^{-3}, we observe for both users a loss of about 2 dB between the simulations with perfect synchronization and with our practical multi-user synchronization algorithm. To understand why this loss applies to both users, we show the root mean square errors (RMSE) of the estimates of the fractional STO λ_{STO}, fractional CFO λ_{CFO} and received power P of each user in Fig. 5.11. For all three parameters, the RMSEs of the first and

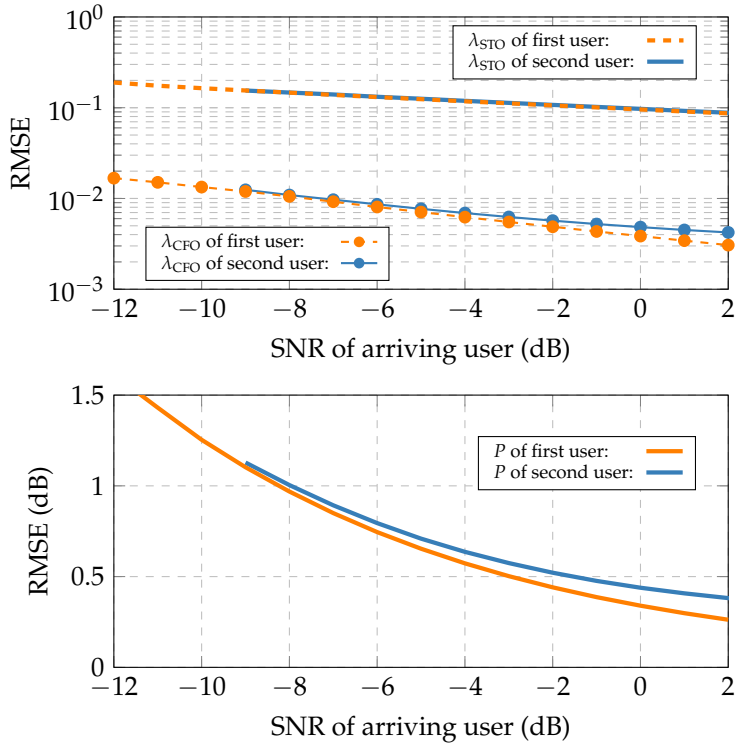


Fig. 5.11 Root mean square errors of estimates of the fractional STO, fractional CFO and received power of each user for $R = 8$.

second user are very similar, i.e., the estimates of the second user are not impaired by the presence of the other colliding user. In our simulations, the integer STO and CFO are always correctly estimated when the preamble is detected. Our synchronization algorithm is hence robust to same-SF interferers, but induces a performance loss for the two-user detector.

The 2-dB loss is mainly due to the fractional STO, whose estimates are one order of magnitude less accurate than those of λ_{CFO}. The inaccuracy on λ_{STO} stems from the fact that our estimator relies on a matched filter with an oversampling factor R followed by a decision for the best match. As such, the resolution of the estimates $\widehat{\lambda}_{\text{STO}}^{(k)}$ is limited by R, even in the absence of noise. The estimation of the fractional CFO does not suffer from this limitation. We finally note that the estimator of the received power is biased as its RMSE floors at high SNR. This bias however does not strongly

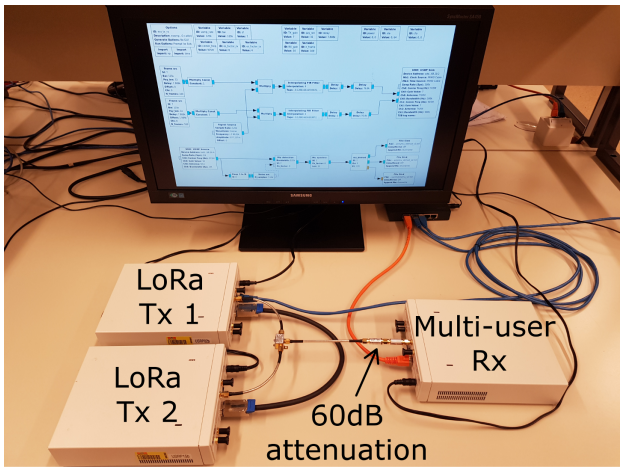


Fig. 5.12 Multi-user testbed with two USRPs as Tx and one USRP as Rx.

impact the two-user detector, since the SERs of both users decrease well below 10^{-4}.

5.5.4 Experimental Validation in a Testbed

We subsequently seek to validate our GNU Radio implementation in an experimental testbed, shown in Fig. 5.12. The testbed uses three National Instruments (NI) 2920 USRP transceivers, with two acting as transmitters and the third one implementing the two-user receiver. The transmission of the LoRa frames is achieved using the open-source GNU Radio LoRa prototype of *Tapparel et al.* described in [40], which has been tested to be compatible with commercial LoRa devices. The nominal carrier frequency used by both transmitters is 868 MHz. To avoid interference from other sources in the 868 MHz band, the transmitters are connected to the multi-user receiver with an RF combiner. Two -30 dB attenuators are inserted between the combiner and the receiver to attain the low SNR regions of interest. Both transmit USRPs share the same reference clock, which enables GNU Radio to define the STO τ and the effective CFO Δf_c between the transmitters. The CFO and STO between the transmitters and the receiver are not controlled in the testbed, but are corrected by the receiver when it synchronizes to the strongest transmitter.

The following results are obtained by creating collisions between the

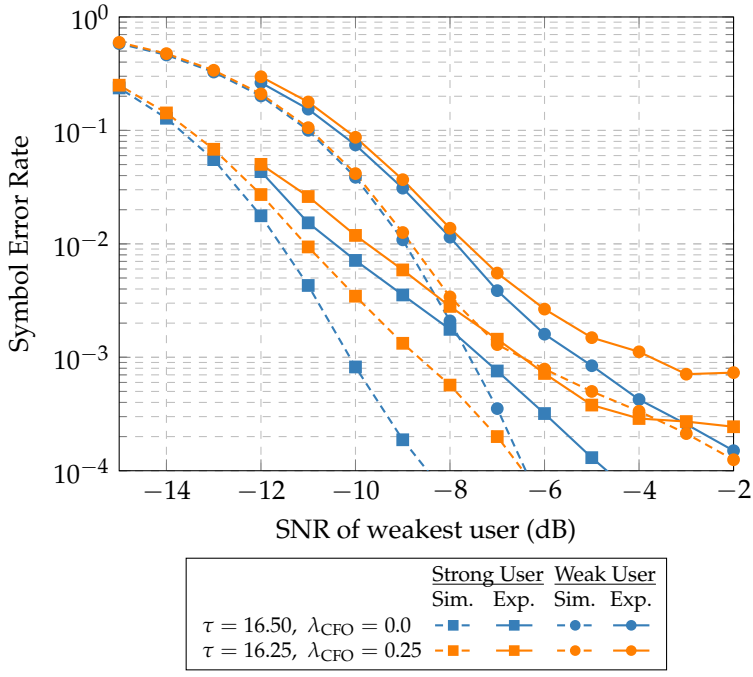


Fig. 5.13 Simulation with ideal synchronization and experimental SERs of the two transmitters for different values of τ and λ_{CFO}, with SF $= 7$, $L_{CFO} = 0$ and $P_A - P_B = 3$ dB.

two transmit USRPs. In each experiment, both transmitters send a LoRa packet with a preamble of 12.25 symbols and 32 random data symbols to the receiver. The starting time of the transmission of the second user is delayed by 15 symbols and τ samples with respect to the first user. The SER vs SNR curves are obtained by sweeping through the transmit gains of the transmit USRPs. A total of 20 000 experiments are performed for each SNR level. As for the simulation results, we focus on per-user SERs when the receiver jointly demodulates two superimposed users. The SNR is measured at the arrival of the first user, as the ratio between the power of the maximum DFT bin over the power contained in the rest of the bins after synchronization. The SER is computed following the same methodology used in the previous section.

Fig. 5.13 shows the experimental SERs of both users along with Monte-Carlo simulation results for two different combinations of STOs τ and frac-

tional CFOs λ_{CFO}, $P_{\text{A}} - P_{\text{B}} = 3$ dB and an oversampling factor $R = 8$. The simulation results assume perfect synchronization, i.e., the received power, CFO, and STO of each user are known. The SER differences between the simulation and experimental results mainly illustrate the impact of the synchronization stage and the non-idealities present in the testbed.

For a target SER of 10^{-3}, we observe that the SNR loss is at best of 2.5 dB (for $\tau = 16.50$ and $\lambda_{\text{CFO}} = 0$), and is thus similar to the previous results of Fig. (5.10). This SER target is also attained for both users in the scenario $\tau = 16.25$ and $\lambda_{\text{CFO}} = 0.25$. Overall, these results validate that our GNU Radio implementation of the proposed two-user receiver is capable of detecting and demodulating two colliding LoRa packets in an experimental setup. We however note some performance losses with respect to the Monte-Carlo simulations under ideal synchronization. Regarding the scenario with $\tau = 16.25$ and $\lambda_{\text{CFO}} = 0.25$, we observe that the experimental SERs of both users level off for SNRs higher than -4 dB. This flooring of the SERs at high SNRs, which does not arise in simulation, highlights the difficulty of the detector to demodulate colliding users in scenarios where both users have similar spectral representations and when residual synchronization errors remain.

5.5.5 Receiver Complexity Evaluation

We finally evaluate the complexity of the proposed two-user receiver, including the synchronization stage. As discussed in Chapter 2, conventional single-user LoRa gateways use a single N-point FFT per symbol of N samples to carry out both the synchronization and the demodulation. Assuming a Radix-2 FFT implementation, the processing of each input sample implies $\frac{N}{2} \log N$ complex multiplications. Our two-user detection rule requires, in addition to the FFT, the computation of KN outputs of the matched filters $\widetilde{M}_1^{(k)}$ and $\widetilde{M}_2^{(k)}$ per symbol, which amounts to KN complex MAC operations per sample. For SF $= 7$ and SF $= 12$, this overhead corresponds to 640 and 20480 MACs per sample ($K = 5$), respectively, whereas the FFT requires 3.5 and 6 complex multiplications per sample. Similarly, the synchronization stage uses R parallel matched filters of length $2N$ on top of the FFT, which necessitate $2N$ additional MAC operations per sample (256 and 8192 MACs for SF $= 7$ and SF $= 12$, respectively).

For the typical bandwidth and sampling frequency $B = f_s = 125$ kHz, the computational overhead of our receiver varies between $112 \cdot 10^6$ (SF $=$

7) and $3.5 \cdot 10^6$ (SF $=$ 12) MACs per second. As a comparison, 5G NR mobile terminals carry out 4096-point FFTs with sampling rates up to $f_s =$ 122 MHz [101], requiring $732 \cdot 10^6$ complex multiplications per second for a Radix-2 FFT. Hence, thanks to the small bandwidths of LoRa signals, the overall complexity of the proposed gateway receiver remains similar to those of modern cellular terminals.

5.6 Discussion and Summary

Multi-user receivers are required to overcome the scalability limitations of LoRa networks. In this chapter, we derived from the ML criterion a multi-user detector capable of demodulating symbols from two interfering LoRa users. The proposed detector inherently leverages the omnipresent differences in received power, time offsets, and frequency offsets to separate and demodulate the contribution of each user. For at least a 1.5 dB difference of received power between the users, the detector is able to demodulate both users with useful error rates. To illustrate how this two-user detector may operate in a practical receiver, we designed a synchronization algorithm that is robust to same-SF interference and implemented both blocks in GNU Radio. Experimental results have shown that our GNU Radio implementation demodulates two colliding users with SERs below 10^{-3}, hence demonstrating the practicality of such a receiver.

Yet, the different performance analyses conducted in this chapter indicate that the ML-derived two-user receiver shows lower performance when either (a) both users are closely synchronized in time and frequency and thus share similar spectral representations (b) the difference of received power between the users is small (c) residual synchronization errors remain after the synchronization stage. In such scenarios, the probability of a symbol error greatly increases when colliding symbols overlap in frequency. These overlaps in frequency, which are inherent to the two-user signal model of (5.3), are the dominating error events and may induce error rate floors at high SNR, as discussed in Sections 5.4.4 and 5.5.4. Fortunately, the BICM strategy implemented in the LoRa PHY provides redundancy and time-diversity to overcome this issue. Since we did not yet consider the interleaving and coding at the receiver, the objective of the next chapter is to further improve the proposed two-user detector by fully exploiting the BICM scheme of LoRa.

6

A Two-User Successive Interference Cancellation Soft-Detector

We derived in the previous chapter a ML-based two-user detector and we demonstrated its practicality by evaluating it in a testbed with an SDR implementation. However, we did not investigate so far the potential benefits of the interleaving and coding in the LoRa PHY for a multi-user receiver. A previous work has shown that the BICM scheme of LoRa provides important performance gains in Rayleigh fading channels, when the demodulation and decoding stages use soft decisions [19]. Since the dominating error events of the two-user detector from Chapter 5 are the infrequent overlaps in frequency of interfering symbols, an intuitive follow-up to the previous chapter is to assess whether the time-diversity provided by the BICM also helps to overcome these events in a two-user soft-receiver. The first objective of this chapter is therefore to design a successive interference cancellation (SIC) two-user detector that implements a soft-demodulation and soft-decoding of each user based on the matched filters derived in Chapter 5. Monte-Carlo simulations of this SIC receiver show that leveraging the coding with soft decisions greatly decreases the required SNRs to successfully demodulate two colliding users.

Moreover, Chapter 5 only focused on the PHY layer implementation of

a multi-user LoRaWAN gateway. The corresponding throughput improvements at the MAC layer have also not been analyzed. The second objective of this chapter is hence to evaluate the benefits of deploying our proposed SIC two-user soft-receiver in a LoRaWAN network. To this end, we extended the LoRaWAN module [22] of the ns-3 simulator with a model of the proposed two-user gateway. Network simulation results indicate that, for a target frame error rate of 1%, our proposed detector is capable of serving 4.7 times more LoRa end nodes than a conventional single-user gateway.

This chapter is organized as follows. In Section 6.1, we explain the implementation of a single-user LoRa soft-receiver. In Section 6.2, we design an iterative two-user SIC soft-detector and evaluate its performance for different parameters (coding rate, spreading factor, ...). Finally, we evaluate and discuss in Section 6.3 the benefits of our proposed receiver on the scalability and reliability of a LoRaWAN network.

6.1 Iterative Single-User BICM LoRa Soft-Receiver

Iterating between a soft-input soft-output (SISO) demodulator and a SISO decoder is an efficient approach to improve the error rate performance over hard-decisions receivers [102]. The architecture of such an iterative single-user LoRa receiver is shown in Fig. 6.1 [19]. To describe its functioning, we now consider only one user transmitting a sequence of $k_c\text{SF}$ payload bits over an AWGN channel. Following the notations introduced in Chapter 2, we denote the transmitted payload bits as $b^{(m,p)}$ with $m \in \{0,\dots,\text{SF}-1\}$ and $p \in \{0,\dots,k_c-1\}$ denoting the row and column, respectively, of the bit matrix before encoding and interleaving. Further, let $\bar{b}^{(m,k)}$ and $\bar{b}'^{(k,m)}$ with $k \in \{0,\dots,n_c-1\}$ be the corresponding coded bits before and after interleaving, respectively. The n_c rows of bits $\bar{b}'^{(k,m)}$ are mapped to the symbols $x^{(k)}[n]$ using (2.2).

The corresponding n_c received symbols are represented by $y^{(k)}[n] = h^{(k)}x^{(k)}[n] + z^{(k)}[n]$, where $h^{(k)} \in \mathbb{C}$ is the complex-valued channel gain during the transmission of the k-th symbol. As in Chapter 5, we assume that the magnitudes of the gains $h^{(k)}$ are constant and equal to $\sqrt{P}$ during the whole frame reception, whereas the phase of $h^{(k)}$ may change over time.

For a non-coherent detection, the likelihood that the transmitter sent

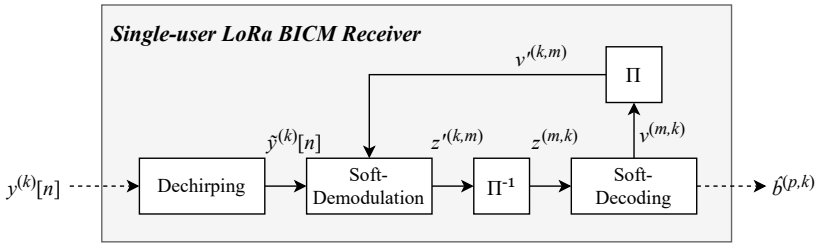


Fig. 6.1 Illustration of an iterative BICM LoRa soft-receiver.

the candidate symbol $\bar{s}$ during the k-th window is given by

$$\Lambda(\bar{s}|y^{(k)}) = K \cdot I_0\left(\frac{\sqrt{P}}{\sigma^2}\left|Y^{(k)}[\bar{s}]\right|\right), \tag{6.1}$$

where $Y^{(k)}[\bar{s}]$ is the N-point DFT of $\tilde{y}^{(k)}[n] = y^{(k)}[n] \cdot x_0^*[n]$ and K is a constant common to all symbols [19].

To compute soft information at the output of the demodulator, the likelihoods $\Lambda(\bar{s}|y^{(k)})$ are mapped to log-likelihood ratios (LLRs) [102]

$$z'^{(k,m)} = \log \frac{\sum\limits_{\bar{s}:g_m(\bar{s})=1} \Lambda(\bar{s}|y^{(k)}) \prod\limits_{i=0,i\neq m}^{\mathrm{SF}-1} e^{g_i(\bar{s})v'^{(k,i)}}}{\sum\limits_{\bar{s}:g_m(\bar{s})=0} \Lambda(\bar{s}|y^{(k)}) \prod\limits_{i=0,i\neq m}^{\mathrm{SF}-1} e^{g_i(\bar{s})v'^{(k,i)}}}, \tag{6.2}$$

where $g_i(\bar{s})$ returns the i-th bit of the symbol $\bar{s}$ in the Gray labeling, and $v'^{(k,m)}$ are the a priori LLRs computed by the soft Hamming decoder after re-interleaving. During the first iteration, we have $v'^{(k,m)} = 0$. In practical receivers, the complexity of computing the LLRs $z'^{(k,m)}$ is reduced by using the max-log approximation [103], which here yields

$$\begin{aligned} z'^{(k,m)} = &\max_{\bar{s}:g_m(\bar{s})=1}\left[\log I_0\left(\frac{\sqrt{P}}{\sigma^2}|Y[\bar{s}]|\right) + \sum_{i=0,i\neq m}^{\mathrm{SF}-1} g_i(\bar{s})v'^{(k,i)}\right] \\ &- \max_{\bar{s}:g_m(\bar{s})=0}\left[\log I_0\left(\frac{\sqrt{P}}{\sigma^2}|Y[\bar{s}]|\right) + \sum_{i=0,i\neq m}^{\mathrm{SF}-1} g_i(\bar{s})v'^{(k,i)}\right]. \end{aligned} \tag{6.3}$$

The LLRs $z'^{(k,m)}$ are subsequently de-interleaved and fed to a Hamming soft-decoder (e.g., [104]). This soft-decoder outputs new LLR values

$v^{(m,k)}$, from which an estimation $\hat{b}^{(m,p)}$ of the transmitted bits is obtained. To further improve the receiver performance, several iterations between the soft-demodulator and the soft-detector can be computed. To this end, the LLRs $v^{(m,k)}$ are re-interleaved and fed as a priori information $v'^{(k,m)}$ to the soft-demodulator.

6.2 Designing a Two-User SIC LoRa Soft-Detector

Motivated by the resilience of the BICM scheme against time-varying channels, we present in this section a two-user SIC soft-detector derived from the previously described iterative single-user detector. This two-user soft-detector re-uses key elements of the ML-based two-user detector from Chapter 5, including the same signal model. The strongest user is also decoded first, and its contribution to the received signal is removed from the received signal before the weakest user is decoded. Yet, contrary to the ML-based detector, the proposed two-user soft-detector leverages the soft-output of the Hamming decoding to achieve a more accurate interference cancellation of the strongest user.

As for the two-user receiver described in Chapter 5, we assume for the remainder of this chapter that the proposed two-user detector is always synchronized to the strongest user and that it has perfect knowledge of the users received powers P_{A} and P_{B}, and of the STO τ and the residual CFO Δf_c between them.

The architecture of the proposed two-user soft-detector is described first. We then explain in more detail its interference-cancellation algorithm and the asynchronous soft-demodulator used to detect the symbols of the non-synchronized weak user. An analysis of the performance of the two-user detector concludes this section.

6.2.1 Architecture of the Proposed Two-User SIC Soft-Detector

Fig. 6.2 shows the architecture of the proposed successive interference cancellation two-user soft-detector. The detector contains two branches, each detecting a different user. The upper branch performs first the demodulation and detection of the user with the strongest received power. In the lower branch, the presumed signal contribution of the strongest user is subtracted from the received signal and the message of the weakest user

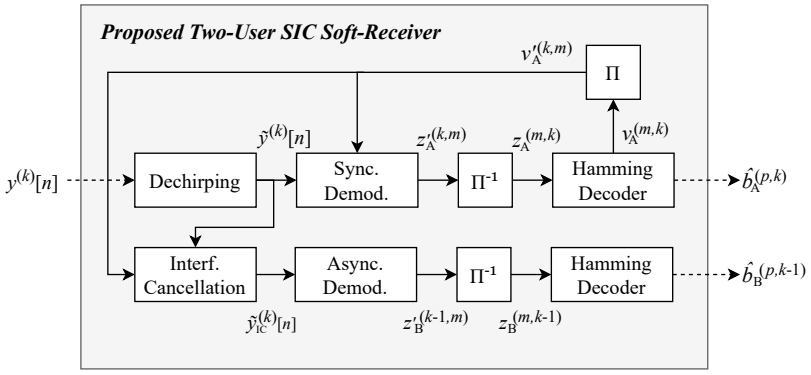


Fig. 6.2 Illustration of the proposed two-user SIC soft-receiver.

is decoded. If a single user is detected by the synchronization stage, only the upper branch can be used, and the proposed detector falls back to the single-user soft-detector presented in Section 6.1.

In the following derivations, we assume without loss of generality that $P_{\mathrm{A}} > P_{\mathrm{B}}$. The gateway is hence synchronized to user A. The received samples after synchronization are split into windows $y^{(k)}[n]$ of N samples, following the signal model from (5.3). The detector first dechirps each window of N samples, yielding the dechirped signal $\tilde{y}^{(k)}[n]$ whose expression without AWGN is provided in (5.5). The operations used to detect the strongest user are strictly identical to those of the single-user detector presented in Section 6.1. The soft-demodulator of (6.3) processes the dechirped signal $\tilde{y}^{(k)}[n]$ with $\sqrt{P} = \sqrt{P_{\mathrm{A}}}$ and by considering the contribution of user B only as additional white noise. The LLRs $z'^{(k,m)}_{\mathrm{A}}$ from the demodulator are then de-interleaved and fed to the SISO Hamming decoder from [104]. The output LLRs $v^{(m,k)}_{\mathrm{A}}$ of the soft-decoder are subsequently re-interleaved and sent back to the soft-demodulator as a priori information. This iterative decoding of user A leverages the channel code to further overcome the interference of user B. Let N_I be the number of iterations carried out by the detector in the branch of the strongest user. We later show that two iterations are sufficient in practice, and that selecting $N_I > 2$ does not significantly improve the decoding performance.

Upon the execution of the N_I iterations in the upper branch, the interleaved LLRs $v'^{(k,m)}_{\mathrm{A}}$ after decoding are used by an interference-cancellation algorithm to estimate and subtract the contribution of user A from the

dechirped signal $\tilde{y}^{(k)}[n]$. Let $\tilde{y}_{\mathrm{IC}}^{(k)}[n]$ be the interference-cancelled signal. Due to the STO τ, each window $\tilde{y}_{\mathrm{IC}}^{(k)}[n]$ contains two symbols of user B. The demodulation of the k-th symbol of user B hence requires the specific matched filters from Chapter 5 that account for the STO τ on the windows $\tilde{y}_{\mathrm{IC}}^{(k)}[n]$ and $\tilde{y}_{\mathrm{IC}}^{(k+1)}[n]$. This asynchronous soft-demodulator outputs LLR values $z_{\mathrm{B}}^{\prime(k,m)}$ which are finally interleaved and used in a SISO Hamming decoder to produce estimates $\hat{b}_{\mathrm{B}}^{(p,k)}$ of the payload bits of user B. The decoding of user B is always done in a single iteration since it is not impaired by the interference of user A.

6.2.2 Cancellation of the Strongest User

The proposed detector performs a hard cancellation of the symbols of user A. To this end, the interleaved LLRs $v_{\mathrm{A}}^{\prime(k,m)}$ are mapped back to probabilities. The probability that user A transmitted the candidate symbol $\bar{s}$ in the k-th window is

$$p\left(s_{\mathrm{A}}^{(k)} = \bar{s} \middle| v_{\mathrm{A}}^{\prime(k,m)}\right) = \prod_{i=0}^{\mathrm{SF}-1} \frac{e^{g_i(\bar{s}) v_{\mathrm{A}}^{\prime(k,i)}}}{1 + e^{v_{\mathrm{A}}^{\prime(k,i)}}}. \tag{6.4}$$

The detector then selects the symbol $\hat{s}_{\mathrm{A}}^{(k)}$, which is the symbol that was the most likely to have been sent by user A

$$\hat{s}_{\mathrm{A}}^{(k)} = \arg\max_{\bar{s}} \; p\left(s_{\mathrm{A}}^{(k)} = \bar{s} \middle| v_{\mathrm{A}}^{\prime(k,m)}\right). \tag{6.5}$$

To subtract the estimated symbol $\hat{s}_{\mathrm{A}}^{(k)}$ from $\tilde{y}^{(k)}[n]$, the detector requires an estimate of its initial phase $\theta_{\mathrm{A}}^{(k)}$. We here re-use the phase estimator introduced in Chapter 5, where $\theta_{\mathrm{A}}^{(k)}$ is estimated by retrieving the phase of the bin $\hat{s}_{\mathrm{A}}^{(k)}$ in the DFT $Y^{(k)}$, i.e., $\hat{\theta}_{\mathrm{A}}^{(k)} = \arctan\left(Y^{(k)}\left[\hat{s}_{\mathrm{A}}^{(k)}\right]\right)$. The presumed contribution of user A is then cancelled from the dechirped signal $\tilde{y}^{(k)}[n]$

$$\tilde{y}_{\mathrm{IC}}^{(k)}[n] = \tilde{y}^{(k)}[n] - \sqrt{P_{\mathrm{A}}} e^{j\hat{\theta}_{\mathrm{A}}^{(k)}} e^{j2\pi \hat{s}_{\mathrm{A}}^{(k)} \frac{n}{N}}. \tag{6.6}$$

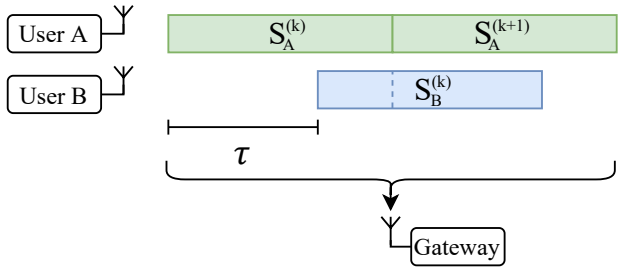


Fig. 6.3 Illustration of the asynchronicity in time between the users A and B, with the sampling time offset τ.

6.2.3 Demodulation of the Weakest User

Since the gateway is not synchronized to user B, the demodulation of this user must take into account both the STO τ and the residual CFO Δf_c. Following the dechirped signal model of (5.5) and as illustrated in Fig. 6.3, the symbol $\bar{s}_B^{(k-1)}$ of user B is split over the two windows $\tilde{y}^{(k-1)}[n]$ and $\tilde{y}^{(k)}[n]$. For the demodulation of this user, we propose to re-use the matched filter expressions $M_1^{(k)}$ and $M_2^{(k)}$ of (5.13) and (5.15), respectively, obtained in Chapter 5. Therefore, to detect $\bar{s}_B^{(k-1)}$, the asynchronous soft-demodulator computes the matched filters $M_2^{(k-1)}$ and $M_1^{(k)}$ over the last $N - \lceil \tau \rceil$ samples of $\tilde{y}_{IC}^{(k-1)}[n]$ and the first $\lceil \tau \rceil$ samples of $\tilde{y}_{IC}^{(k)}[n]$, respectively.

If the symbols $s_A^{(k-1)}$ and $s_A^{(k)}$ have correctly been cancelled in $\tilde{y}_{IC}^{(k-1)}[n]$ and $\tilde{y}_{IC}^{(k)}[n]$, respectively, it can be shown that the likelihood that user B sent the candidate symbol $\bar{s}_B^{(k-1)}$ is

$$\Lambda\left(\bar{s}_B^{(k-1)} \middle| \tilde{y}_{IC}^{(k)}[n]\right) = K' I_0\left(\frac{\sqrt{P_B}}{\sigma^2}\left|Y_B^{(k-1)}\left[\bar{s}_B^{(k-1)}\right]\right|\right), \tag{6.7}$$

where $Y_B^{(k-1)}[\bar{s}] = M_2^{(k-1)}[\bar{s}] + M_1^{(k)}[\bar{s}]$ is the summation of both matched filters, and K' is a constant common to all symbols of user B. Since the proposed detector does not iterate for the weakest user, the LLRs at the

output of the soft-demodulator are evaluated as

$$\begin{aligned} z_{\mathrm{B}}'^{(k-1,m)} = & \max_{\bar{s}:g_m(\bar{s})=1} \left[\log I_0 \left(\frac{\sqrt{P_{\mathrm{B}}}}{\sigma^2} \left| Y_{\mathrm{B}}^{(k-1)}[\bar{s}] \right| \right) \right] \\ & - \max_{\bar{s}:g_m(\bar{s})=0} \left[\log I_0 \left(\frac{\sqrt{P_{\mathrm{B}}}}{\sigma^2} \left| Y_{\mathrm{B}}^{(k-1)}[\bar{s}] \right| \right) \right]. \end{aligned} \tag{6.8}$$

6.2.4 Performance Analysis

The performance of our two-user SIC detector is now analyzed for different parameters. In the following results, we always simulate a scenario with two interfering users using the same SF. The users have a received power difference $\Delta P = P_{\mathrm{A}} - P_{\mathrm{B}}$ of 1.5 dB. This choice of ΔP is motivated by the fact the spreading gain of the modulation increases by approximately 3 dB when the SF is incremented [105]. Therefore, in a network where the SFs of all users are optimally allocated, all nodes using the same SF should have at the gateway a difference of received powers ΔP of at most 3 dB.

Coding Rate

We first study the impact of the coding rate on the decoding performance in a typical scenario with SF $= 7$. Fig. 6.4 shows the bit error rate (BER) of the strongest and weakest user, which are separated in time by a small STO $\tau = \frac{N}{8} = 16.0$. There is no residual CFO, i.e., $\Delta f_c = 0$. The strongest user is demodulated using $N_I = 2$ iterations.

In the absence of coding (CR $= 4/4$) and for a target BER of 10^{-4}, we observe that the strong and weak user need an SNR of 3.5 dB and 2 dB to be correctly demodulated, respectively. In a single-user scenario, this BER is attained for a much lower SNR of -7 dB [19]. When enabling the coding with CR $= 4/5$, the required SNRs to achieve the same BER are -1.5 dB and -3 dB for the strongest and weakest user, respectively, whereas in a single-user scenario, the corresponding SNR is at -8.5 dB. With CR $= 4/7$, the required SNRs even further decrease to -8 dB for both users. The required single-user SNR at this coding rate is -10 dB [19]. When comparing the required SNRs for the strongest user in the presence and the absence of a second user, we see that the gap reduces from 10.5 dB for CR $= 4/4$ to 2 dB for CR $= 4/7$. This important reduction indicates that the combination of the interleaving and the Hamming coding in the BICM scheme strongly helps a SIC soft-detector to demodulate the strongest user in the presence of same-SF interference.

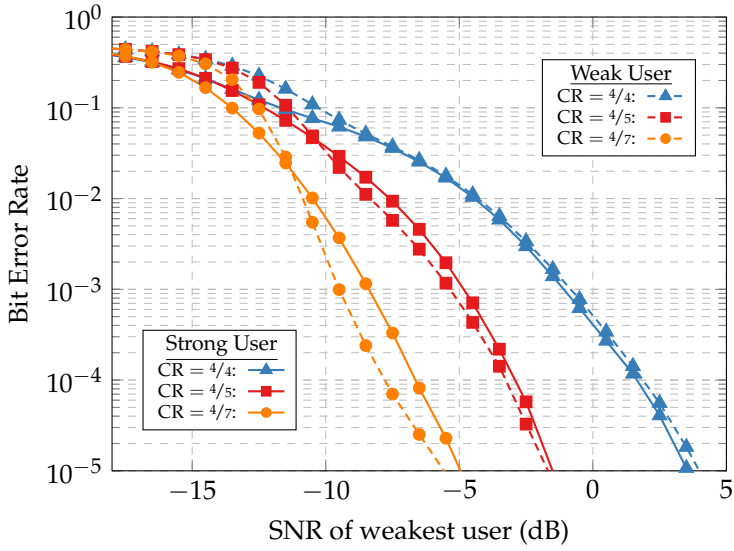


Fig. 6.4 BERs of both users for different coding rates CR, with SF $= 7$, $\tau = 16.0$, $\Delta f_c = 0$ and $\Delta P = 1.5$ dB. For CR $= 4/5$ and $4/7$, the strongest user is decoded with $N_I = 2$ iterations.

Number of Iterations

We now analyze the benefits of performing several iterations when demodulating the strongest user. For single-user BICM communications, it is well-known that Gray labeling offers the best first-pass performance, but yields no significant extra gain with iterative decoding [106]. These results have been confirmed in [19] for an iterative BICM LoRa single-user receiver. The authors show that for SF $= 7$, CR $= 4/7$ and a 10^{-4} BER, increasing the number of iterations from one to four only brings an SNR gain of 0.5 dB. We here show that in the case of our two-user SIC receiver, more significant SNR gains can be obtained.

Fig. 6.5 presents the BERs of both users for one, two and four iterations, with the same parameters as before and a coding rate of $4/7$. In the present example, increasing N_I to two decreases the required SNR for the strongest user by 2 dB, and by 1.25 dB for the weakest user. Improving the demodulation of the strongest user also benefits the BER of the weakest user, as the contribution of the former is better cancelled before demodulating the latter. However, no further significant gains are observed for $N_I = 4$. We

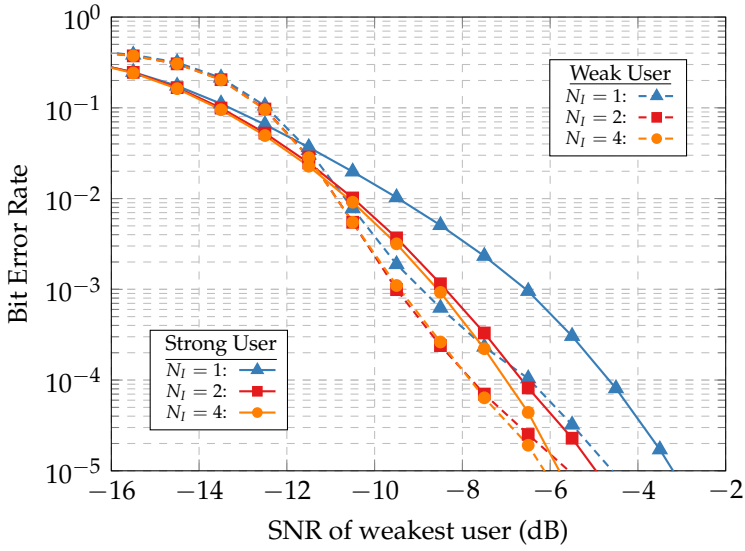


Fig. 6.5 BERs of both users for different numbers of iterations in the decoding of the strongest user, with SF $= 7$, CR $= 4/7$, $\tau = 16.0$, $\Delta f_c = 0$ and $\Delta P = 1.5$ dB.

hence suggest limiting the number of iterations to two, in order to keep the complexity of the receiver as low as possible.

Averaging over the STO and CFO

The previous results have been obtained for specific values of STO τ and CFO Δf_c. However, we showed in Chapter 5 that the performance of a multi-user LoRa receiver strongly depends on the synchronization in time and frequency of the colliding users. While the contribution of the strongest and synchronized user to the signal after dechirping $\tilde{y}^{(k)}[n]$ is always a Kronecker delta in the frequency domain, the spectral representation of the asynchronous user symbols is, depending on the STO and CFO, modified to a bell-shaped function that spreads across several DFT bins. The more the signal of the asynchronous user is spread out, the higher is the probability of correctly decoding the synchronized user.

To evaluate the performance of our proposed two-user detector in practical scenarios, Fig. 6.6 shows the averaged probability of correctly receiving the overlapping frames of two users for random values of τ and Δf_c, with $\Delta P = 1.5$ dB and CR $= 4/7$. The values of the STO τ and CFO Δf_c are

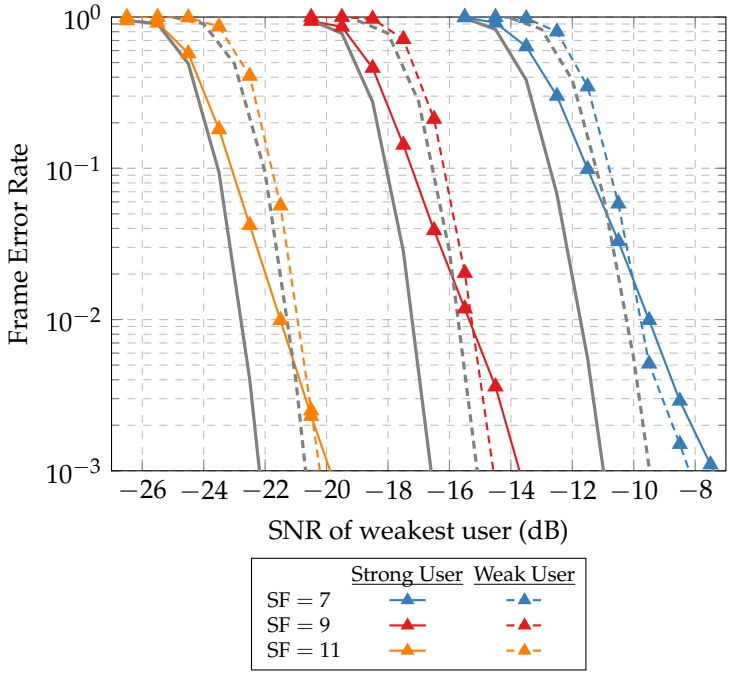


Fig. 6.6 Averaged FERs of both users over random values of τ and Δf_c for different spreading factors, with $\Delta P = 1.5$ dB, CR $= 4/7$ and $N_I = 2$ iterations. For all spreading factors, the colliding frames have a length of 10 information bytes. The thick gray lines show the FER of the proposed receiver in the absence of a second user.

uniformly distributed in the intervals $[0, N)$ and $[-\frac{f_s}{2N}, \frac{f_s}{2N}]$, respectively. The frame error rates (FERs) of the same detector in the presence of a single user at the same SNR levels (shown in gray) are also presented as baseline performance. For SF $= 7$ and a target FER of 10^{-2}, we observe that our SIC soft-receiver requires only 2 dB and 0.5 dB higher SNRs to demodulate the strongest and weakest user, respectively, compared to the corresponding single-user scenarios. At higher spreading factors, we also observe that our proposed detector fully benefits from the spreading gain of the modulation. These averaged FERs show that our proposed two-user receiver is capable of satisfactorily decoding the interfering users for all SFs, with only a small SNR penalty compared to the corresponding single-user scenarios.

6.3 Network-Level Performance Evaluation

To better assess the benefits of the proposed two-user soft-detector, this section presents a network-level performance evaluation using the discrete-event network simulator ns-3 [107]. To this end, we significantly extend the ns-3 LoRaWAN module from [22] to include models of (a) a single-user gateway with capture effect as modeled in [22] (b) a more realistic parametric single-user soft-receiver PHY model and (c) a detailed PHY model of our proposed two-user soft-detector. We first briefly explain how the network simulations are conducted within ns-3. We then describe the three studied LoRa gateway models, and we also explain the simulation parameters. On this basis, we finally analyze the performance of a LoRaWAN network using the different implemented gateway models to compare the single-user with the proposed two-user detector.

6.3.1 Simulation Methodology

We simulate a single LoRa cell, composed of one central gateway surrounded by end nodes. Both the gateway and the end nodes have models specifying their behaviour at the physical and MAC layers. We reuse all the models from [22], except the physical layer models that we substitute with more detailed and more accurate stochastic models. To evaluate the performance of the single- and two-user receivers, we perform Monte-Carlo (MC) simulations by computing for each frame the probability $p_{\text{err,frame}}$ that it is lost (i.e., the decoding led to one or more bit errors). The outcome of each frame (received or lost) is then decided accordingly at random using $p_{\text{err,frame}}$. To compute this probability, the frame under consideration is divided into N_S subframes $S^{(n)}$, with $n \in \{0, N_S - 1\}$. Let $I^{(n)}$ be the set of same-SF users that transmit a frame during the transmission of $S^{(n)}$. The division of the current frame into subframes is done such that each subframe has a set of interferers $I^{(n)}$ that is distinct from its neighbours $I^{(n-1)}$ and $I^{(n+1)}$, as illustrated in Fig. 6.7. The error probability $p_{\text{err}}^{(n)}$ of each subframe $S^{(n)}$ is evaluated using the PHY model of the gateway. Since the N_S error events of the subframes are independent, we

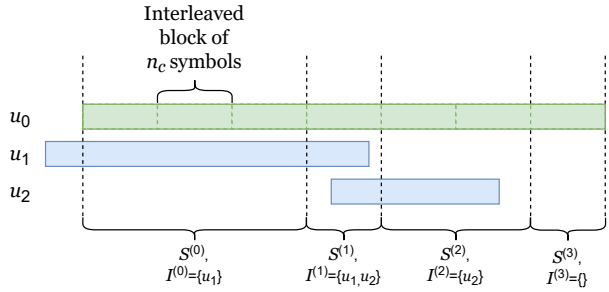


Fig. 6.7 Illustration of subframe division for a frame of user u_0, colliding with the frames of two other same-SF interferers u_1 and u_2.

can multiply them to obtain the overall frame error probability

$$p_{\text{err,frame}} = 1 - \prod_{n=0}^{N_S-1} \left(1 - p_{\text{err}}^{(n)}\right). \tag{6.9}$$

In our PHY models, we use look-up tables of FERs to rapidly access the probabilities $p_{\text{err}}^{(n)}$ which are obtained offline in PHY layer MATLAB Monte-Carlo simulations. To ensure the independence of the subframes, despite the coding and interleaver, these FERs can only be obtained for subframe lengths that are multiples of n_c symbols, i.e., the size of an interleaved block. We hence approximate the discretization of a frame into subframes $S^{(n)}$ to the closest interleaver boundaries.

Furthermore, due to the difference in nature between same-SF and inter-SF interference, these two kinds of collisions need to be modeled differently. Same-SF interference is directly taken into account in all three considered PHY gateway models when we obtain the probabilities $p_{\text{err}}^{(n)}$ offline. To reduce the number of offline Monte-Carlo simulations, the inter-SF interference is modeled online within the network simulator by its impact on the signal-to-interference-plus-noise ratio (SINR). Since the dechirping sequence $x_0^*[n]$ used in the receivers is specific to each SF, the dechirped symbols of an interfering user using a different SF remain spread over the signal bandwidth. We therefore suggest to evaluate the SINR of each frame received at the gateway, and to use this ratio as SINR parameter in the error probabilities $p_{\text{err}}^{(n)}$. Let u be the user of interest, and $\overline{I}^{(n)}$ be the set of inter-SF interferers active during the transmission of the subframe $S^{(n)}$. We denote

the received power of a user i at the gateway as P_i. To keep tractable models, we compute a single SINR value for each frame by selecting the highest interference level that is observed during its reception

$$\text{SINR} = \min_{n} \frac{P_u}{\sum\limits_{i \in \overline{I}^{(n)}} P_i + \sigma^2}. \tag{6.10}$$

The SINRs hence account for the inter-SF interference and the noise, but not the same-SF interference.

6.3.2 Description of the Gateway Models

We now describe the PHY layer models of the three evaluated receivers in more detail.

Threshold-based Single-User Receiver

This model implements the behavior of a single-user LoRa gateway similarly to [22]. Two conditions on the SINR and on the level of same-SF interference need to be met for a subframe to be lost. The subframe is considered lost if the observed SINR is below a threshold $\mathcal{S}_{\text{SF}}$, which is defined as the required SINR to attain an uncoded FER of 0.1% for the SF used in the absence of both same-SF and inter-SF interference. In addition, a same-SF interferer is considered sufficiently strong to prevent the correct decoding of the user of interest if the received power of the interferer is less than 3 dB below that of the considered user [108]

$$p_{\text{err}}^{(n)} = \begin{cases} 1 \text{ if } \begin{cases} 10 \log \frac{P_u}{P_i} < 3 \text{ dB for any } i \in I^{(n)} \\ \text{or} \\ \text{SINR} < \mathcal{S}_{\text{SF}}, \end{cases} \\ 0 \text{ else.} \end{cases} \tag{6.11}$$

Single-User Receiver with Soft-Decoding

Assuming a hard threshold for the capture effect as described above or in other publications [22, 108] is however pessimistic. In fact, in some cases, a frame can still be demodulated for lower differences in received power ΔP between two same-SF colliding nodes (same-SF SIR) [77]. To allow for a fair comparison between conventional single-user LoRa receivers and our proposed two-user receiver, we here seek to better model the performance of

a single-user soft-detector in the presence of a single same-SF interferer. To this end, we evaluate in MATLAB the FERs of the single-user soft-detector presented in Section 6.1, but with the two-user signal model of (5.3). Similarly to our proposed receiver, we assume that this single-user receiver is always capable of synchronizing itself to the strongest user, and that it performs two iterations of soft-demodulation and soft-decoding, but only for the strongest user. We compute, using MC simulations, a look-up table $\mathcal{L}_{1\mathrm{U}}\left(\mathrm{SF}, \mathrm{CR}, \Delta P, \mathrm{SINR}, L\right)$ of the FERs for given values of the SF, coding rate CR, same-SF SIR ΔP, SINR level and subframe length L in symbols. To reduce the dimensionality of this table, these FERs are computed by averaging over all possible values of the STO τ and residual CFO Δf_c, as explained in Section 6.2.4.

Let $L^{(n)}$ be the length of the subframe under evaluation. We consider this subframe to be lost if any same-SF colliding frame has a greater received power, or if at least two other weaker same-SF colliding frames are present with same-SF SIRs that are lower than 3 dB. Otherwise, the probability of loss $p_{\mathrm{err}}^{(n)}$ for this subframe is retrieved from the look-up table

$$p_{\mathrm{err}}^{(n)} = \begin{cases} 1 \text{ if } \begin{cases} 10\log\frac{P_u}{P_i} < 0 \text{ dB for any } i \in I^{(n)} \\ \text{or} \\ 10\log\frac{P_u}{P_{i_1}} < 3 \text{ dB and } 10\log\frac{P_u}{P_{i_2}} < 3 \text{ dB}, \\ \quad \text{for any } i_1 \in I^{(n)}, i_2 \in I^{(n)} \text{ s.t. } i_1 \neq i_2, \end{cases} \\ \mathcal{L}_{1\mathrm{U}}(\mathrm{SF}, \mathrm{CR}, \frac{P_u}{P_i}, \mathrm{SINR}, L^{(n)}) \text{ else.} \end{cases} \tag{6.12}$$

Two-User SIC Receiver with Soft-Decoding

Similarly to the previous model, we also compute a look-up table $\mathcal{L}_{2\mathrm{U}}$ of the FERs of our proposed two-user receiver using MATLAB MC simulations. Contrary to the single-user receiver, the table $\mathcal{L}_{2\mathrm{U}}$ also contains FERs for values $10\log\frac{P_u}{P_i} < 0$ dB, since the two-user receiver also allows the demodulation of a weaker user. Nevertheless, in the presence of more than two concurrent same-SF users with sufficiently strong received powers, we assume that the subframe is lost. The corresponding subframe error rate lookup table for ns-3 corresponds to

$$p_{\mathrm{err}}^{(n)} = \begin{cases} 1 \text{ if } \begin{cases} 10\log\frac{P_u}{P_{i_1}} < 3 \text{ dB and } 10\log\frac{P_u}{P_{i_2}} < 3 \text{ dB}, \\ \quad \text{for any } i_1 \in I^{(n)}, i_2 \in I^{(n)} \text{ s.t. } i_1 \neq i_2, \end{cases} \\ \mathcal{L}_{2\mathrm{U}}(\mathrm{SF}, \mathrm{CR}, \frac{P_u}{P_i}, \mathrm{SINR}, L^{(n)}) \text{ else.} \end{cases} \tag{6.13}$$

6.3.3 Simulation Parameters

We now discuss the system parameters used in our network-level simulations.

Spatial Distribution of the Nodes

The simulated network consists in a circular cell with a central gateway. The end nodes surround the gateway following a random uniform distribution. This distribution leads to a homogeneous node density in the cell.

Frame Length

All frames are 25-bytes long, with 13 bytes corresponding to the LoRaWAN MAC header and 12 bytes for the application payload. This payload length respects the fair access policy of The Things Network [109].

Frequency Bands

To keep our simulations as focused as possible, we simulate in ns-3 only one frequency band. The throughput results obtained below can simply be multiplied by the number of frequency bands used in practice to obtain the overall network throughput.

Channel Propagation

A log-distance path loss model is used to evaluate the received power of each node at the gateway. According to [110], we fix the path loss exponent to 3.76 and the reference loss as 7.7dB at one meter. Those value are representative for an urban area, with a gateway located at a height of 15 meters. On top of the path loss model, we also add an SNR penalty of 3 dB to account for the RF noise figure and synchronization non-idealities.

Spreading Factor Allocation

We assume that all nodes follow a greedy behavior and select the lowest possible spreading factor to decrease their transmission time, and hence their energy consumption. The range of each SF is calculated based on the previously explained channel propagation model such that all nodes using this SF have an uncoded FER of at most 0.1% in the absence of any interference. This policy results in a central disc containing all nodes with SF 7, and non-overlapping annuli whose distances increase with the SFs. The range of each SF and the repartition of nodes per SF are listed in Table 6.1.

Spreading Factor	7	8	9	10	11	12
Radius [km]	3.32	3.99	4.80	5.76	6.93	8.32
Percentage of Nodes [%]	16	7	10	15	21	31

Table 6.1 Outcome of the spreading factor allocation across the cell.

Power Allocation

Since the spreading gain of LoRa increases by 3 dB for each increment of the SF, the nodes inside the annuli have a received power difference among them of at most 3 dB. On the contrary, the nodes with SF 7, all located in the central disc, may have more significant received power differences between them depending on their location in the disk. To reduce the level of inter-SF interference and maintain decent SINRs for the most distant nodes in the cell, we reduce the power of SF 7 nodes in steps of 3 dB as they approach closer to the gateway while keeping the 0.1% FER constraint. Since LoRaWAN nodes can only vary their output power programmatically in the range 0 to 14 dBm, the nodes with the SFs from 8 to 12 always use the highest output power of 14 dBm, and the lowest transmit power for SF 7 nodes is 0 dBm. Fig. 6.8 summarizes the outcome of both the SF and power allocation in the simulated cell.

6.3.4 Results

We now evaluate and compare the system performance parameters with the three previously introduced models. We first study independently for each SF the gain in channel throughput of our proposed two-user detector compared to the two models of single-user receivers. We then perform an evaluation of the overall FER within the cell for an increasing node density, by taking into account all spreading factors.

Throughputs for a Single Spreading Factor

Fig. 6.9 presents the throughput attained for our proposed two-user detector and the two single-user models as a function of the offered load for the SFs 7 and 8, alongside the performance of single-user pure and slotted ALOHA, and two-user pure ALOHA schemes. To enable a better interpretation, and comparison of the results to the well studied baseline ALOHA performance, these first simulations use a single SF and ignore inter-SF interference.

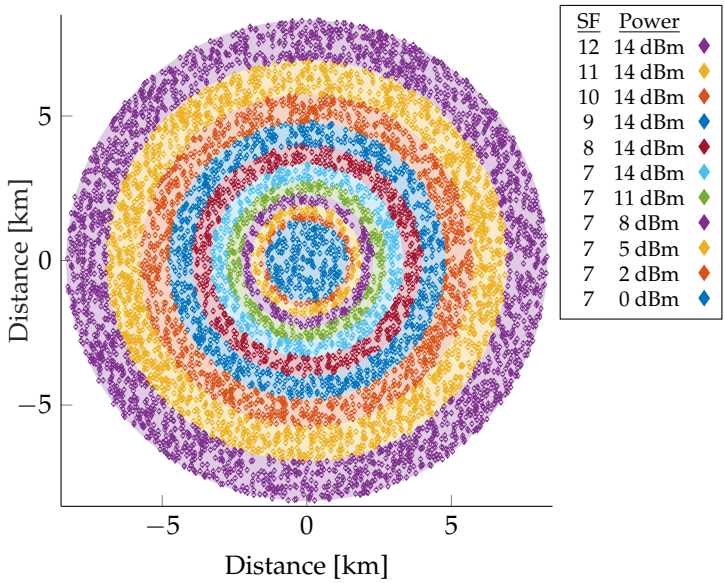


Fig. 6.8 Power and spreading factor allocation within the cell.

We first study the throughput of the nodes with SF 8. We observe that the performance of the threshold-based single-user receiver model closely matches the throughput expectation of a pure ALOHA network. This result is expected as the only cause of frame losses are same-SF collisions, which are always destructive in the SF 8 annulus because all nodes have received powers in the same 3 dB range. On the contrary, thanks to the more precise modeling using look-up tables, we see that the real throughput of a single-user soft detector is significantly better compared to the pessimistic pure ALOHA model. Indeed, the single-user receiver is capable of sometimes recovering from same-SF collisions and successfully decode the frames of the strongest user, even when the power difference to the interferer is smaller than 3 dB. Its network performance, however, remains below that of a pessimistic slotted ALOHA network. Regarding our proposed two-user soft-detector, we observe that its network throughput is only slightly below that of a the two-users pure ALOHA model. Since the two-user ALOHA model assumes an ideal detection of two colliding users, this result confirms that our detector is capable of decoding two colliding frames with a high reliability and that collisions between more than two users are rare.

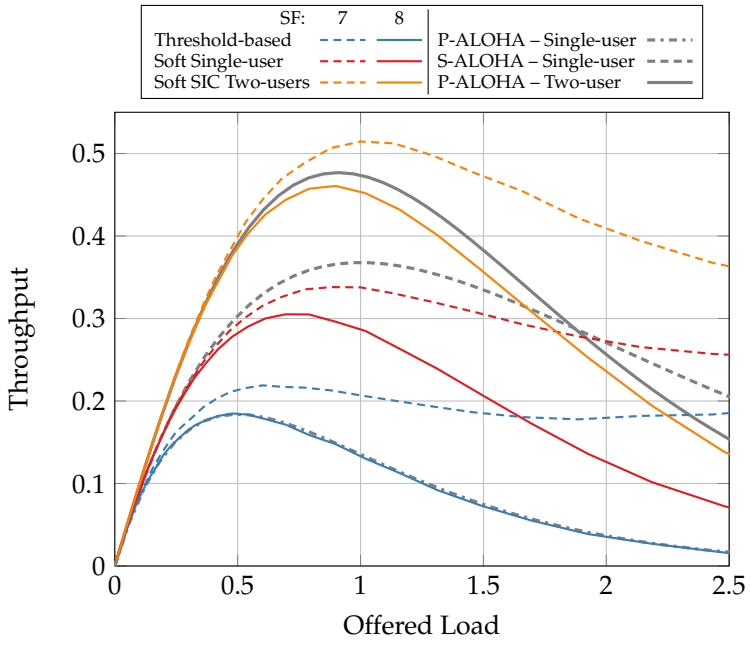


Fig. 6.9 Throughput vs. offered traffic for different receiver models and SFs with CR = 4/7, alongside the theoretical performance of pure ALOHA (P-ALOHA), slotted ALOHA (S-ALOHA), and pure two-user ALOHA networks.

The results observed in Fig. 6.9 hold for all the other SFs greater than 8. However, the throughput of SF 7 is a special case since the nodes using this SF are located in a disc, and not an annulus. Despite the power reduction scheme previously explained, these nodes benefit from a greater diversity of received powers. This wider power range increases the probability to benefit from the capture effect, therefore resolving some collisions without requiring any special treatment from the receiver. In our topology, this improved throughput concerns 16% of all nodes. The percentage of SF 7 nodes however increases when the cell size becomes smaller.

Overall Throughputs with all Spreading Factors

We now analyze the overall frame error rate performance of a LoRaWAN network by also fully accounting for the inter-SF interference. Fig. 6.10 presents the global FER of a LoRaWAN network in which all nodes have a fixed duty cycle of 0.1% for an increasing density of users. We however

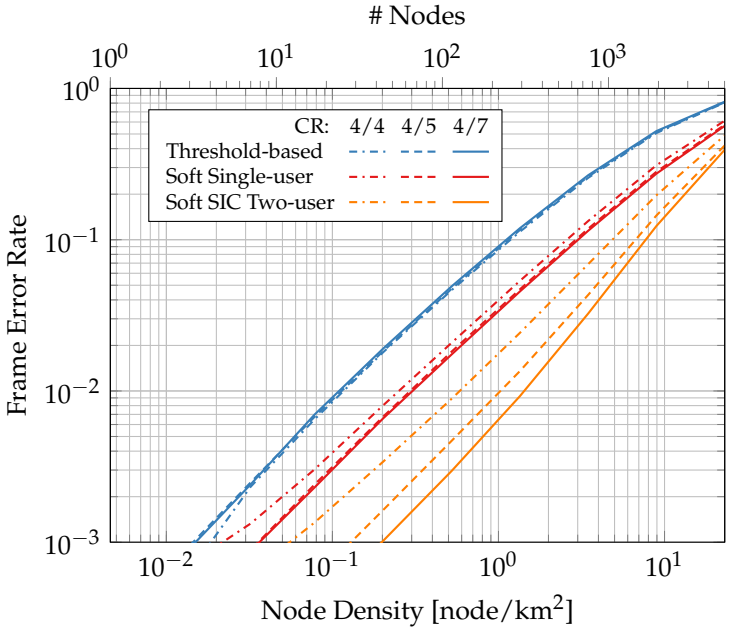


Fig. 6.10 Frame error rate of a LoRaWAN network for different coding rates, with an uplink duty cycle of 0.1%.

note that modifying the transmission duty cycle parameter only results in a horizontal translation of the curves in Fig. 6.10. Furthermore, including for all users variable payload lengths between 1 and 25 bytes, instead of fixed payload lengths, leads to very similar results. The following analysis hence holds for all duty cycles and payload lengths.

In the following, we always target a FER of 1%. We first compare the two single-user models. We observe that the threshold-based model with capture is severely pessimistic, as the single-user soft-detector is actually capable of serving two times ($2\times$) more users thanks to its ability to sometimes recover the strongest one of two colliding users, even without coding. Also, the impact of the coding rate is not visible at all in the threshold-based model, as it solely uses the SINR of a frame to determine if it is correctly received. Conversely, the single-user soft-detector benefits from a decrease of the coding rate. However, at the network-level, the dominating error events remain the losses of the weakest users frames, and the benefits of the soft-decoding of the strongest user are hence marginal.

We finally evaluate the benefits of our proposed two-user SIC soft-detector at the network-level, compared to the single-user soft-detector. For a coding rate of $4/5$, we observe an increase of the node density for the 1% target FER by a factor 3.3. This gain even increases to a factor 4.7 when switching to a coding rate $4/7$. Contrary to the single-user detector, the coding rate plays a more important role for the two user detector as the dominating error events are not collisions among three users, which are very rare, but rather the incapacity of recovering the weakest user, which has a lower SINR. Specifically, in the simulated network, for a duty cycle of 0.1%, a target FER of 1% and CR $= 4/7$, our two-user receiver enables a LoRaWAN gateway to serve 310 nodes, whereas a realistic model of a conventional single-user LoRaWAN gateway is only capable of serving 65 nodes.

6.4 Discussion and Summary

Collisions between frames using the same spreading factor are the most important source of errors in dense LoRaWAN networks. In this chapter, we address this scalability limitation by designing a successive interference cancellation LoRa receiver capable of decoding two concurrent users with the same spreading factor. Our proposed iterative two-user detector is the first to explicitly leverage the bit-interleaved coded modulation scheme of the LoRa PHY layer with both a soft-demodulator and a soft-decoder. In particular, the usage of a low code-rate (i.e., CR $= 4/7$) is paramount to reach useful error rates in the low SNR regime that is characteristic of LoRa communications. Using PHY layer Monte-Carlo simulations, we then built accurate models of the frame error rates under same-SF interference of both a single-user LoRa soft-receiver and of our two-user soft-receiver. Network-level simulations in the network simulator ns-3 indicate that our proposed two-user detector is capable of serving 4.7 times more end nodes than a conventional single-user LoRa gateway, without requiring any modifications to the protocol.

To conclude this chapter, we finally point out that we conducted only a single network-level performance analysis for a simple network topology. Some aspects of the ns-3 simulation should be improved for a more accurate analysis. In particular, our two-user gateway model in ns-3 does not take into account a potential performance loss due to the multi-user

synchronization. Yet, we observed in the previous chapter that such synchronization is difficult to achieve and that its outcome may degrade the performance of the proposed detector. Also, many parameters of the LoRaWAN MAC layer influence the network performance, e.g., the requests for acknowledgment, the allocation of the transmit powers and spreading factors, or the presence of several gateways. Further research is hence needed to assess whether gateways with the proposed two-user detector are also beneficial in more complex and realistic network scenarios.

7

Conclusion

The overwhelming success of IoT applications has sustained an exponential deployment of connected objects. Yet, current IoT wireless technologies, the majority of whom have been designed about ten years ago, cannot handle massive networks with tens of thousands of devices. As first-generation IoT networks progressively reach their limits, much effort is spent today innovating and designing more robust and scalable networking protocols. Since the conventional architecture of LPWAN radios does not allow switching or upgrading the communication protocol, the arrival of next-generation LPWAN technologies will inevitably cause the replacement of the already deployed end nodes. This likely disposal of billions of functional electronic devices however amounts to a disgraceful misuse of energy, materials and greenhouse gases emissions.

To overcome this dilemma between innovation and environment, we studied in this thesis two complementary approaches to help preventing the technological obsolescence of LPWAN radios. The first approach consists in the design of an ULP MCU architecture in which the protocol-specific digital baseband processing of the radio is implemented in software and is hence reprogrammable. The second approach involves the design of multi-user LoRa receivers that push back the current scalability limits of LoRaWAN networks. We conclude this thesis by summarizing the outcomes of this work for each approach separately, and by discussing some connected open research problems.

Towards ULP IoT Software-Defined Radios

Although SDRs can improve the interoperability of IoT end nodes, the principal obstacle remains the high power consumption of such radios. We have seen that the implementation of ULP SDRs for LPWAN protocols requires both low-complexity digital baseband algorithms and a low-power circuit architecture with an efficient hardware/software partitioning. With the goal of implementing a LoRa SDR in mind, we designed in Chapter 3 a low-complexity synchronization algorithm for Nyquist-rate LoRa receivers. This algorithm estimates and corrects the carrier frequency and sampling time offsets with a negligible computational overhead, and induces only a small performance loss compared to an ideally synchronized receiver. In Chapter 4, we designed an ULP MCU architecture that carries out most of the digital baseband processing of LPWAN protocols in software. The general-purpose processor of the MCU performs the protocol-specific computations, while a reconfigurable hardware digital front-end executes the generic signal processing.

Leveraging the low-complexity LoRa synchronization algorithm and the ULP MCU architecture, we developed SDR implementations of the LoRa and Sigfox protocols for our MCU prototype fabricated in 28 nm CMOS FDSOI. To the best of our knowledge, this prototype is the first to experimentally showcase a software execution of the PHY layers of two LPWAN protocols with a sub-mW power consumption for the digital signal processing.

If this first prototype validates the concept of LPWAN SDRs running on ULP MCUs, it also raises new research questions and opportunities. We implemented both the Sigfox and LoRa standards, but we could only demonstrate the LoRa SDR for the smallest signal bandwidth due to a lack of computational power. Furthermore, we did not investigate the implementation of a NB-IoT transceiver, whose standard is much more complex. Another candidate of interest would be the recent DECT-2020 NR mesh standard, which operates at 1.9 GHz [111]. To fully implement all of these protocols on a single ULP SDR platform, further research must study possible improvements of the proposed hardware/software architecture with more advanced processors and more efficient designs of the digital front-end. The design of RF front-ends that can efficiently adapt to several frequency bands and bandwidths, while maintaining a small area and a low power consumption, is also of great interest.

Multi-User Receivers for LoRa Gateways

Even though LPWAN SDRs are a promising solution to improve the interoperability of IoT end nodes, this approach cannot apply to the devices that have already been deployed. End nodes operating in interference-limited LoRaWAN networks are particulary at risk of being replaced, and even more so with the upcoming deployment of the new LR-FHSS PHY layer.

To prevent the technological obsolescence of LoRaWAN devices, we investigated in the second part of this thesis the design of multi-user receivers for LoRa gateways. In Chapter 5, we derived from the maximum-likelihood criterion a two-user LoRa detector that is capable of decoding colliding frames from two interfering end nodes. We implemented this two-user detector in an SDR testbed, along with a multi-user synchronization algorithm robust to interference. However, since the performance of this detector is inherently limited by the nature of the CSS modulation, we then designed in Chapter 6 a SIC two-user LoRa soft-receiver that explicitly leverages the BICM scheme of LoRa. This second receiver implements an iterative demodulation and decoding strategy with soft decisions to improve the detection and the cancellation of the strongest user. Finally, we evaluated the network-level performance improvements provided by the proposed soft-receiver in a simple LoRaWAN network topology.

Thanks to the SDR implementation, we demonstrated that multi-user receivers are a practical solution to improve the scalability of LoRaWAN networks. In addition, network-level simulation results show that, for a low code rate, a gateway embedding the proposed two-user SIC receiver allows to serve 4.7 times more end nodes than a conventional gateway.

However, important questions still need to be addressed before deploying multi-user receivers in actual LoRaWAN networks. First, we evaluated the proposed multi-user synchronization algorithm of Chapter 5 in a limited number of scenarios. Since the LoRa PHY has not been designed for multi-user detection, the synchronization to multiple users is a challenging problem and causes a non-negligible performance degradation in our SDR testbed. Further research is needed to assess whether more efficient synchronization algorithms can reach satisfactorily performance in all scenarios. Similarly, more detailed network simulations are required to precisely evaluate the benefits of the proposed two-user detectors in realistic LoRaWAN networks. In particular, a thorough network-level study should investigate the scalability of a network with multi-user LoRa receivers compared to a network relying on the new LR-FHSS PHY layer.

Bibliography

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of Things applications: A systematic review," *Computer Networks*, vol. 148, pp. 241–261, Jan. 2019.

[3] G. Callebaut, G. Leenders, J. Van Mulders, G. Ottoy, L. De Strycker, and L. Van der Perre, "The art of designing remote IoT devices—technologies and strategies for a long battery life," *Sensors*, vol. 21, no. 3, p. 913, Jan. 2021.

[4] T. Pirson and D. Bol, "Assessing the embodied carbon footprint of IoT edge devices with a bottom-up life-cycle approach," *Journal of Cleaner Production*, vol. 322, p. 128966, Nov. 2021.

[5] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2nd quarter 2017.

[6] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, Mar. 2019.

[7] G. Leenders, G. Callebaut, G. Ottoy, L. Van der Perre, and L. De Strycker, "Multi-RAT for IoT: The potential in combining LoRaWAN and NB-IoT," *IEEE Communications Magazine*, vol. 59, no. 12, pp. 98–104, Dec. 2021.

[8] L. M. Hilty and B. Aebischer, *ICT innovations for sustainability*, ser. Advances in Intelligent Systems and Computing. Springer, 2015, vol. 310.

[9] S. B. Boyd, *Life-cycle assessment of semiconductors*. Springer Science & Business Media, 2012.

[10] A. Plepys, "The grey side of ICT," *Environmental impact assessment review*, vol. 22, no. 5, pp. 509–523, Nov. 2002.

[11] P.-V. C. Louis-Philippe, Q. E. Jacquemotte, and L. M. Hilty, "Sources of variation in life cycle assessments of smartphones and tablet computers," *Environmental Impact Assessment Review*, vol. 84, p. 106416, Sep. 2020.

[12] M. Proske, D. Sánchez, C. Clemm, and S.-J. Baur. (2020) Life cycle assessment of the fairphone 3. Accessed July 2022. [Online]. Available: https://www.fairphone.com/wp-content/uploads/2020/07/Fairphone_3_LCA.pdf

[13] M. Stead, P. Coulton, J. Lindley, and C. Coulton, *The Little Book of Sustainability for the Internet of Things*. Imagination Lancaster, 2019, https://eprints.lancs.ac.uk/id/eprint/131084, accessed July 2022.

[14] R. Rai and J. Terpenny, "Principles for managing technological product obsolescence," *IEEE Transactions on components and packaging technologies*, vol. 31, no. 4, pp. 880–889, Dec. 2008.

[15] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, "A survey on technologies, standards and open challenges in satellite IoT," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1693–1720, 3rd quarter 2021.

[16] D. D. Wentzloff, A. Alghaihab, and J. Im, "Ultra-low power receivers for IoT applications: A review," in *2020 IEEE Custom Integrated Circuits Conference (CICC)*, 2020, pp. 1–8.

[17] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2016, pp. 59–67.

[18] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, Apr. 2017.

[19] T. Elshabrawy and J. Robert, "Evaluation of the BER performance of LoRa communication using BICM decoding," in *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, 2019, pp. 162–167.

[20] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Investigating theoretical performance and demodulation techniques for LoRa," in *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, 2019, pp. 1–6.

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

[22] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of LoRaWAN performance under different parameter settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, Jan. 2019.

[23] G. Boquet, P. Tuset-Peiró, F. Adelantado, T. Watteyne, and X. Vilajosana, "LR-FHSS: Overview and performance analysis," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 30–36, 2021.

[24] S. Verdu. Cambridge university press, 1998.

[25] O. Seller and N. Sornin, "Low power long range transmitter," Feb. 2 2016, US Patent 9,252,834.

[26] ——, "Low complexity, low power and long range radio receiver," Jan. 4 2018, US Patent Appl. 15/620,364.

[27] LoRaWAN Alliance. (2017) LoRaWAN specification v1.1. Accessed July 2022. [Online]. Available: https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/

[28] M. Knight and B. Seeber, "Decoding LoRa: Realizing a modern LPWAN with SDR," in *Proceedings of the GNU Radio Conference*, 2016.

[29] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, Sep. 2016.

[30] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017.

[31] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "LoRa digital receiver analysis and implementation," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 1498–1502.

[32] M. Chiani and A. Elzanaty, "On the LoRa modulation for IoT: Waveform properties and spectral analysis," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8463–8470, Oct. 2019.

[33] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018.

[34] 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, Sep. 2017.

[35] G. Callebaut and L. Van der Perre, "Characterization of LoRa Point-to-Point path loss: Measurement campaigns and modeling considering censored data," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1910–1918, Mar. 2019.

[36] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology," in *2015 14th International Conference on ITS Telecommunications (ITST)*, 2015, pp. 55–59.

[37] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu, "Measurement, characterization, and modeling of LoRa technology in multifloor buildings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 298–310, Jan. 2020.

[38] *SX1276/77/78/79: 137 MHz to 1020 MHz Low Power Long Range Transceiver*, Semtech, Aug. 2016, rev. 5.

[39] G. Pasolini, "On the lora chirp spread spectrum modulation: Signal properties and their impact on transmitter and receiver architectures," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 357–369, Jan 2022.

[40] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, "An open-source LoRa physical layer prototype on GNU radio," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[41] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Investigating theoretical performance and demodulation techniques for LoRa," in *1st International Workshop on Data Distribution in Industrial and Pervasive Internet (DIPI 2019)*, Jun. 2019.

[42] O. Afisiadis, S. Li, J. Tapparel, A. Burg, and A. Balatsoukas-Stimming, "On the advantage of coherent LoRa detection in the presence of interference," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 581–11 593, Jul. 2021.

[43] T. Elshabrawy and J. Robert, "Closed-form approximation of LoRa modulation BER performance," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1778–1781, Sep. 2018.

[44] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa imperfect orthogonality: Analysis of link-level performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, Apr. 2018.

[45] O. Afisiadis, A. Burg, and A. Balatsoukas-Stimming, "Coded LoRa frame error rate analysis," in *2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[46] T. Elshabrawy and J. Robert, "Interleaved chirp spreading LoRa-based modulation," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3855–3863, Apr. 2019.

[47] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[48] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Towards an SDR implementation of LoRa: Reverse-engineering, demodulation strategies and assessment over Rayleigh channel," *Computer Communications*, vol. 153, pp. 595–605, Mar. 2020.

[49] G. Baruffa, L. Rugini, L. Germani, and F. Frescura, "Error probability performance of chirp modulation in uncoded and coded LoRa systems," *Elsevier Digital Signal Processing*, vol. 106, Nov. 2020.

[50] C. Bernier, F. Dehmas, and N. Deparis, "Low complexity LoRa frame synchronization for ultra-low power software-defined radios," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3140–3152, May 2020.

[51] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, "A multi-channel software decoder for the LoRa modulation scheme," in *IoTBDS*, 2018, pp. 41–51.

[52] Z. Xu, P. Xie, S. Tong, and J. Wang, "From demodulation to decoding: Towards complete LoRa PHY understanding and implementation," *ACM Transactions on Sensor Networks (TOSN)*, Jul. 2022.

[53] *SX1301 Datasheet*, Semtech, Jun. 2017, v2.4.

[54] S. Li, U. Raza, and A. Khan, "How agile is the adaptive data rate mechanism of LoRaWAN?" in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206–212.

[55] L. Vangelista and A. Cattapan, "Start of packet detection and synchronization for LoraWAN modulated signals," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4608–4621, Jun. 2022.

[56] V. Savaux, C. Delacourt, and P. Savelli, "On time-frequency synchronization in LoRa system: From analysis to near-optimal algorithm," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 10 200–10 211, Jun. 2022.

[57] B. G. Quinn, "Estimation of frequency, amplitude, and phase from the DFT of a time series," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 814–817, Mar. 1997.

[58] E. Jacobsen and P. Kootsookos, "Fast, accurate frequency estimators [DSP Tips & Tricks]," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 123–125, May 2007.

[59] C. Yang and G. Wei, "A noniterative frequency estimator with rational combination of three spectrum lines," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 5065–5070, Jun. 2011.

[60] O. Afisiadis, M. Cotting, A. Burg, and A. Balatsoukas-Stimming, "On the error rate of the LoRa modulation with interference," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1292–1304, Feb. 2019.

[61] H. B. Amor and C. Bernier, "Software-hardware co-design of multi-standard digital baseband processor for IoT," in *IEEE DATE Conference 2019*, 2019, pp. 646–649.

[62] G. Leenders, G. Callebaut, L. Van der Perre, and L. De Strycker, "Multi-RAT IoT–what's to gain? An energy-monitoring platform," *arXiv preprint arXiv:2207.07371*, 2022.

[63] K. Mikhaylov, M. Stusek, P. Masek, V. Petrov, J. Petajajarvi, S. Andreev, J. Pokorny, J. Hosek, A. Pouttu, and Y. Koucheryavy, "Multi-RAT LPWAN in smart cities: trial of LoraWAN and NB-IoT integration," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[64] *Multiprotocol LPWAN dual core 32-bit Arm Cortex-M4/M0+ LoRa, (G)FSK, (G)MSK, BPSK, up to 256KB Flash, 64KB SRAM*, STMicroelectronics, Mar. 2022, rev. 3.

[65] *nRF52833 – Objective Product Specification*, Nordic Semiconductors, Sep. 2019, version 0.7.

[66] Y. Chen, S. Lu, H.-S. Kim, D. Blaauw, R. G. Dreslinski, and T. Mudge, "A low power software-defined-radio baseband processor for the Internet of Things," in *2016 IEEE international symposium on High Performance Computer Architecture*, 2016, pp. 40–51.

[67] M. Hessar, A. Najafi, V. Iyer, and S. Gollakota, "TinySDR: Low-power SDR platform for over-the-air programmable IoT testbeds," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 1031–1046.

[68] R. Dekimpe, M. Schramme, M. Lefebvre, and D. Bol, "SleepRider: a 5.5μW/MHz Cortex-M4 MCU in 28nm FD-SOI with ULP SRAM, biomedical AFE and fully-integrated power, clock and back-bias management," in *2021 IEEE Symposium on VLSI Circuits*, 2021, pp. 1–2.

[69] D. Bol, M. Schramme, L. Moreau, P. Xu, R. Dekimpe, R. Saeidi, T. Haine, C. Frenkel, and D. Flandre, "SleepRunner: A 28-nm FDSOI ULP Cortex-M0 MCU with ULL SRAM and UFBR PVT compensation for 2.6-3.6-μw/DMIPS 40-80-MHz active mode and 131-nW/kB fully retentive deep-sleep mode," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 7, pp. 2256–2269, Jul. 2021.

[70] D. Lachartre, F. Dehmas, C. Bernier, C. Fourtet, L. Ouvry, F. Lepin, E. Mercier, S. Hamard, L. Zirphile, S. Thuries *et al.*, "7.5 a TCXO-less 100Hz-minimum-bandwidth transceiver for ultra-narrow-band sub-GHz IoT cellular networks," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 134–135.

[71] *LMS6002D: Multi-band Multi-standard Transceiver with Integrated Dual DACs and ADCs*, Lime Microsystems, Mar. 2012, rev. 1.1.0.

[72] *Sigfox SDR-USB100M SDR Dongle*, Sigfox, Jun. 2018.

[73] A. Skillman and T. Edsö, "A technical overview of Cortex-M55 and Ethos-U55: Arm's most capable processors for endpoint AI," in *2020 IEEE Hot Chips 32 Symposium (HCS)*, 2020, pp. 1–20.

[74] H. B. Amor, C. Bernier, and Z. Přikryl, "A RISC-V ISA extension for ultra-low power IoT wireless signal processing," *IEEE Transactions on Computers*, vol. 71, no. 4, pp. 766–778, Apr. 2022.

[75] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, p. 1193, May 2017.

[76] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan, and M. Gidlund, "Scalability analysis of a LoRa network under imperfect orthogonality," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1425–1436, Mar. 2019.

[77] R. Fernandes, R. Oliveira, M. Luís, and S. Sargento, "On the real capacity of LoRa networks: the impact of non-destructive communications," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2437–2441, Dec. 2019.

[78] T.-H. To and A. Duda, "Simulation of LoRa in ns-3: Improving LoRa performance with CSMA," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[79] L. Beltramelli, A. Mahmood, P. Österberg, and M. Gidlund, "LoRa beyond ALOHA: An investigation of alternative random access protocols," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3544–3554, May 2020.

[80] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for LoRa," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, Sep. 2020, pp. 1–13.

[81] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted ALOHA on LoRaWAN - design, analysis, and deployment," *Sensors*, vol. 19, no. 4, p. 838, Feb. 2019.

[82] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018.

[83] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the industrial internet of things," *Computer Communications*, vol. 153, pp. 1–10, Mar. 2020.

[84] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3097–3109, Oct. 2018.

[85] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-demand LoRa: Asynchronous TDMA for energy efficient and low latency communication in IoT," *Sensors*, vol. 18, no. 11, Nov. 2018.

[86] S. Nagaraj, D. Truhachev, and C. Schlegel, "Analysis of a random channel access scheme with multi-packet reception," in *2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–5.

[87] I. Arun and T. Venkatesh, "Order statistics based analysis of pure ALOHA in channels with multipacket reception," *IEEE Communications Letters*, vol. 17, no. 10, pp. 2012–2015, Oct. 2013.

[88] J. M. de Souza Sant'Ana, A. Hoeller, R. D. Souza, H. Alves, and S. Montejo-Sánchez, "LoRa performance analysis with superposed signal decoding," *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 1865–1868, Nov. 2020.

[89] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proceedings of the Conference of the ACM SIGCOMM 2017*, Aug. 2017, pp. 309–321.

[90] B. Hu, Z. Yin, S. Wang, Z. Xu, and T. He, "SCLoRa: Leveraging multi-dimensionality in decoding collided LoRa transmissions," in *IEEE 28th International Conference on Network Protocols*, 2020, pp. 1–11.

[91] X. Xia, Y. Zheng, and T. Gu, "FTrack: Parallel decoding for LoRa transmissions," *ACM Transactions on Networking*, Nov. 2020.

[92] S. Tong, Z. Xu, and J. Wang, "CoLora: Enabling multi-packet reception in LoRa," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, 2020, pp. 2303–2311.

[93] S. Tong, J. Wang, and Y. Liu, "Combating packet collisions using non-stationary signal scaling in LPWANs," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 234–246.

[94] M. O. Shahid, M. Philipose, K. Chintalapudi, S. Banerjee, and B. Krishnaswamy, "Concurrent interference cancellation: decoding multi-packet collisions in LoRa," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 503–515.

[95] M. A. B. Temim, G. Ferré, B. Laporte-Fauret, D. Dallet, B. Minger, and L. Fuché, "An enhanced receiver to decode superposed LoRa-like signals," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7419–7431, Aug. 2020.

[96] D. Garlisi, S. Mangione, F. Giuliano, D. Croce, G. Garbo, and I. Tinnirello, "Interference cancellation for LoRa gateways and impact on network capacity," *IEEE Access*, vol. 9, pp. 128 133–128 146, Jun. 2021.

[97] S. Verdu, "Optimum Multiuser Detection," in *Multiuser Detection*. Cambridge university press, 1998, ch. 4, pp. 154–233.

[98] J. Proakis and M. Salehi, "Optimum Receivers for AWGN Channels," in *Digital Communications*. McGraw-Hill, 2008, ch. 4, pp. 160–289.

[99] Multi-User LoRa repository. Accessed July 2022. [Online]. Available: https://www.epfl.ch/labs/tcl/resources-and-sw/lora-multi-user-receiver/

[100] Z. Xu, S. Tong, P. Xie, and J. Wang, "FlipLoRa: Resolving collisions with up-down quasi-orthogonality," in *17th Annual IEEE International Conference on Sensing, Communication, and Networking*, Jun. 2020, pp. 1–9.

[101] H. Kim, "Design principles for 5G communications and networks," in *Design and optimization for 5G wireless communications*. John Wiley & Sons, 2020, ch. 6, pp. 197–238.

[102] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo-coded M-ary noncoherent orthogonal modulation," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 9, pp. 1739–1747, Sep. 2005.

[103] M. Ivanov, C. Häger, F. Brännström, A. G. i Amat, A. Alvarado, and E. Agrell, "On the information loss of the max-log approximation in BICM systems," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3011–3025, Jun. 2016.

[104] B. Müller, M. Holters, and U. Zölzer, "Low complexity soft-input soft-output Hamming decoder," in *2011 50th FITCE Congress ICT: Bridging an Ever Shifting Digital Divide*, 2011, pp. 1–5.

[105] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017.

[106] X. Li, A. Chindapol, and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding and 8 PSK signaling," *IEEE Transactions on communications*, vol. 50, no. 8, pp. 1250–1257, Aug. 2002.

[107] nsnam. (2021) ns-3 network simulator. [Online]. Available: https://www.nsnam.org/

[108] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "LoRa technology demystified: From link behavior to cell-level performance," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 822–834, Feb. 2019.

[109] The Things Network. Frequency plans. Accessed July 2022. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/frequency-plans/index.html

[110] 3GPP, "Radio frequency (RF) system scenarios (release 16)," 3GPP, Tech. Rep. 36.942, Jan. 2016.

[111] R. Kovalchukov, D. Moltchanov, J. Pirskanen, J. Säe, J. Numminen, Y. Koucheryavy, and M. Valkama, "DECT-2020 New Radio: The next step toward 5G massive machine-type communications," *IEEE Communications Magazine*, vol. 60, no. 6, pp. 58–64, Jun. 2022.