



# SF-DS: A Slot-Free Decoding Scheme for Collided LoRa Transmissions

Weixuan Xiao, Nancy El Rachkidy, Alexandre Guitton

## ► To cite this version:

Weixuan Xiao, Nancy El Rachkidy, Alexandre Guitton. SF-DS: A Slot-Free Decoding Scheme for Collided LoRa Transmissions. IEEE Vehicular Technology Conference, Jun 2022, Helsinki, Finland. hal-03808037

**HAL Id: hal-03808037**

**<https://hal.science/hal-03808037>**

Submitted on 10 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SF-DS: A Slot-Free Decoding Scheme for Collided LoRa Transmissions

Weixuan Xiao, Nancy El Rachkidy, Alexandre Guitton

Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS,  
63000 Clermont-Ferrand, France.

{weixuan.xiao, nancy.el\_rachkidy, alexandre.guitton}@uca.fr

**Abstract**—Recent monitoring applications extensively rely on low-power wide-area networks, such as those provided by LoRa and LoRaWAN, in order to enable end-devices to communicate over long distances. However, in large-scale deployments, the small throughput of LoRa is further reduced due to collisions. In this paper, we propose a new scheme SF-DS that aims to decode colliding frames. SF-DS relies on frequency detection at each symbol frontier. It reduces detection errors when symbols of similar values are superposed. It is compatible with legacy LoRaWAN communications, and only requires modifications at the gateway. Our simulation results show that SF-DS is able to decode more collisions than the other protocols from the literature (about twice more for 16 nodes and SF12), therefore increasing the throughput and thus the scalability of the network.

## I. INTRODUCTION

In low-power wide area networks (LPWANs), end-devices can communicate to gateways located several kilometers away in one hop, with limited power. Thus, LPWANs are becoming increasingly used in most monitoring applications, including smart cities, smart agriculture, or smart health.

To enable a long communication range, the LPWAN technologies often drastically limit the bitrate. For instance, let us consider the case of the LoRaWAN MAC protocol [1] over the LoRa physical layer [2], which is the most frequent protocol combination in LPWANs. The nominal bitrate for the longest communication range in Europe is only 367 bits per second. This bitrate is further reduced by 25%-50% due to the error coding used in LoRa, by 99% due to the limited duty-cycle of 1% in the ISM band, and by the overhead from LoRaWAN. Thus, we believe that the resulting bitrate of a few bits per second is the hardest constraint in LPWAN applications.

Some LoRa receivers such as the SX1301 used in LoRaWAN gateways, are able to demodulate frames that are received simultaneously on different channels, or using different spreading factors (SFs). However, as the network density increases, collisions on the same channel and SF are likely to occur, which reduces the bitrate.

Several researchers have worked on collision resolution for LoRa signals, when the frames use the same channel and SF. They proposed protocols that leverage the features of LoRa signals. By solving some cases of collisions, the throughput is improved, the energy consumption is reduced, and the scalability of the network increases.

In this paper, we describe a new decoding scheme for LoRa called Slot Free Decoding Scheme (SF-DS). SF-DS uses the decoding technique from the GS-MAC protocol [3], with three new features: (i) it actually performs the frequency detection at the beginning of a symbol rather than relying on the oracle used by GS-MAC, (ii) it does not require slotted transmissions, unlike GS-MAC, which makes it compatible with legacy LoRaWAN transmitters, and (iii) it reduces the detection error when symbols of similar values are superposed. We validate SF-DS by an implementation in GNU Radio. Our simulation results show that SF-DS outperforms the other protocols from the literature.

The outline of the paper is as follows. Section II describes the LoRa physical layer, the demodulation of non-colliding frames, and the LoRaWAN MAC layer. Section III presents the main collision resolution protocols from the literature, with an emphasis on the GS-MAC protocol as it is the basis for SF-DS. Section IV introduces SF-DS, and details how we were able to remove the main design issues of GS-MAC. Section V presents our simulation results, by comparing the decoding capabilities of SF-DS with the best protocols from the literature. Finally, Section VI concludes our work.

## II. LORA AND LORAWAN

In this section, we first describe the LoRa physical layer, with a focus on how LoRa symbols are demodulated. Then, we present LoRaWAN, a widely used MAC protocol for the current LoRa networks.

### A. LoRa modulation

LoRa is a physical layer technology for LPWAN based on a chirp spread spectrum modulation. A LoRa frame is modulated as a sequence of chirps. A chirp consists of a linear frequency sweep over a given bandwidth (BW). The frequency of a normalized up-chirp encoding value 0 is as follows:

$$f_c(t) = \frac{BW}{SD}t, t \in [-\frac{SD}{2}, \frac{SD}{2}), \quad (1)$$

where  $SD = \frac{2^{SF}}{BW}$  is the symbol duration and SF is the spreading factor. Communications on different SFs are quasi-orthogonal. LoRa encodes the value  $m$  by applying an offset to the initial (normalized) frequency of the chirp:

$$f_c^m(t) = \begin{cases} \frac{BW}{SD}(t + \tau_m), & t \in [-\frac{SD}{2}, \frac{SD}{2} - \tau_m) \\ \frac{BW}{SD}(t + \tau_m) - BW, & t \in [\frac{SD}{2} - \tau_m, \frac{SD}{2}), \end{cases} \quad (2)$$

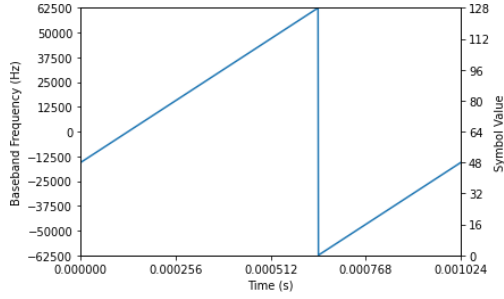


Fig. 1: An up-chirp encoding value 48 with SF7 and BW=125 kHz.

where  $\tau_m = \frac{m}{BW}$ . Note that the resulting (normalized) frequency is always within  $[-BW/2; BW/2]$ , thanks to the subtraction of  $BW$  in the second case.

Figure 1 shows an up-chirp encoding value 48 (as shown on the right-hand y-axis), with SF7 and BW=125 kHz. The resulting symbol duration  $SD$  is 1.024 ms. In the following, we will use the term *symbol* to denote either an up-chirp or the value of an up-chirp.

A LoRa frame consists of a preamble, an optional header and a payload. The preamble consists of a series of up-chirps (generally 8) encoding value 0, two up-chirps encoding the network identifier, and two and a quarter down-chirps. The header is composed of eight up-chirps encoding payload metadata and a CRC. The payload consists of several up-chirps encoding the data.

The LoRa coding scheme performs several operations in order to transform the bitstream generated by the application into the values encoded in symbols: it adds redundancy using a Hamming code with a code rate called CR; it whitens the data sequence, applies a diagonal interleaver to distribute the bits into several symbols, and performs a Gray decoding. The LoRa decoding scheme performs the inverse operations, in reverse order, in order to retrieve the original bitstream from the demodulated values of the received symbols.

### B. Focus on the demodulation of LoRa symbols

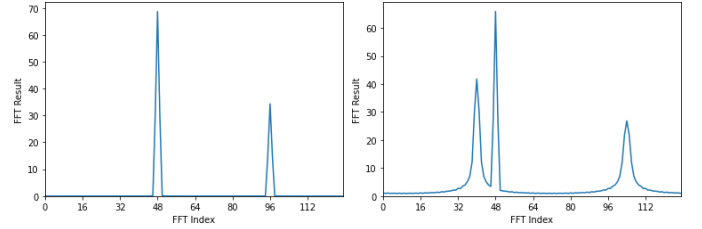
In order to demodulate the symbols of a frame, a LoRa receiver needs to be synchronized with the transmitter. This synchronization is performed thanks to the repeated up-chirps of the preamble. Then, the values from the header and the payload are computed by multiplying the received signal by a normalized down-chirp  $f_d(t)$  of value 0 and of duration  $SD$ . This multiplication of signals is equal to the addition of their frequencies. This addition removes the time-variant in the up-chirp as follows:

$$f_c^m(t) + f_d(t) = \begin{cases} \frac{BW}{SD}\tau_m, t \in [-\frac{SD}{2}, \frac{SD}{2} - \tau_m) \\ \frac{BW}{SD}\tau_m - BW, t \in [\frac{SD}{2} - \tau_m, \frac{SD}{2}). \end{cases} \quad (3)$$

Finally, the demodulator performs a Fast Fourier Transform (FFT) at each symbol duration, and outputs the highest FFT peak as the symbol value. Note that even though one of the two cases of Equation 3 is outside of  $[-BW/2; BW/2]$ , as the demodulator uses  $BW$  as sampling rate, the aliasing limits the

higher frequency component to a value modulo  $BW$ , which makes the two cases identical in value. The demodulated symbols during each period  $SD$  are then processed through the LoRa decoding scheme.

When several frames with the same SF arrive simultaneously at a receiver, a LoRa demodulator obtains several FFT peaks per  $SD$ , instead of a single peak. This is shown on Fig. 2, on two examples. Fig. 2(a) shows the two peaks produced during a single symbol duration of two synchronized frames: symbol 48 is received with a large power, while symbol 96 is received with a low power. The demodulator outputs the strongest peak with value 48. Due to the power difference in the two frames, it is likely that all the other symbols of the first frame will be captured too, which illustrates the capture effect of LoRa. Fig. 2(b) shows the three peaks produced during a single symbol duration of two desynchronized frames, with the receiver synchronized on the first frame: symbol 48 is the symbol of the first frame, symbol 32 is the end of a symbol of the second frame, and symbol 96 is the beginning of the next symbol of the second frame. Note that the values corresponding to the symbols of the second frames are slightly shifted, due to the delay of the second frame. Even though these two frames are received with similar power, the time offset between the frames make the peaks of the second frame appear weaker. Again, the demodulator outputs the strongest peak, that is value 48, and the correct symbol is decoded.



(a) Symbol 48 and symbol 96 from one synchronized two synchronized frames. The frame frame, and symbol 32 followed by with symbol 48 is received with twice symbol 96 from one desynchronized the power of the other frame. frame with the same power level.

Fig. 2: FFT peaks of two superposed frames with SF7.

However, LoRa receivers are often unable to capture frames received on the same SF, on the same frequency channel, and with similar receive power, due to the complexity of the received signal. Section III presents several algorithms from the literature aiming to resolve the collisions of LoRa.

### C. LoRaWAN protocol

LoRaWAN [4] is a simple MAC protocol operating on top of LoRa. In LoRaWAN, end-devices communicate with gateways using LoRa, and gateways communicate with a network server using an IP backhaul. LoRaWAN defines three classes of end-devices. Class A is for low-power uplink communications: end-devices can transmit at any time using an ALOHA mechanism. To do so, an end-device randomly chooses a channel, transmits the frame, and waits for a response by opening two short receive windows. Upon receiving a frame, the gateway

forwards it to the network server, waits for a potential answer from the network server, and sends the answer during either of these windows. Class B adds delay-guaranteed downlink communications, and is implemented using a beacon-based mechanism. Class C allows fast downlink communications by assuming that end-devices do not have energy constraints.

LoRaWAN provides different regional settings. European regional settings of LoRaWAN [5] define seven data rates (DRs) for LoRa communication, from DR0 to DR6. The bandwidth of channels is 125 kHz for all DRs, except for DR6 where it is 250 kHz. SF varies from 7 to 12 for DR5 to DR0, and is equal to 7 for DR6. The indicative physical bit rate varies between 11000 bps for DR6, down to 250 bps for DR0. It is further reduced by a 1% maximum duty-cycle. The data rate of LoRaWAN can be further reduced by collisions. Thus, the main target of collision resolution protocols is to improve the throughput by leveraging collisions.

### III. EXISTING COLLISION RESOLUTION PROTOCOLS

In this section, we briefly present collision resolution protocols from the literature. We give more details on the GS-MAC slotted protocol [3] as it is the basis of SF-DS.

#### A. Protocols based on the capture effect

In [6], the authors propose to leverage the capture effect using successive interference cancellation (SIC). The receiver recursively decodes the frame with the strongest signal, reproduces the corresponding signal, and removes it from the received superposed signals in order to decode weaker signals. The main drawback of SIC is that the capture effect cannot be applied when the received frames have similar power: indeed, in this case, the demodulator is unable to identify the correct FFT peak from several peaks of similar strengths. Note that the SIC requires a difference of power varying from 6 dB for SF7 to 20 dB for SF12.

#### B. Protocols that distinguish frames

The CHOIR protocol [7] applies the FFT on a large window (10 times the original window) in order to extract tiny frequency offsets in symbols. The authors show that these tiny frequency offsets come from hardware imperfections of the transmitters. They are stable during the whole frame duration, and can therefore be used to identify the symbols of each transmitter. While this approach distinguishes the transmitters and decodes collisions of a few frames, it does not scale with the number of concurrent transmitters.

The FTrack protocol [8] uses the time offset among colliding frames to decode them. When the demodulator uses the standard LoRa decoding scheme, each symbol becomes a constant frequency over  $SD$ , called a track. As the frequency of the symbols of a frame only changes at each symbol edge, FTrack uses these symbol edges to match the symbols to the correct frame. The main drawback of FTrack is that it requires a sufficient time offset between colliding frames, which has to be greater than one tenth of the symbol duration.

SCLoRa [9] leverages both time and power difference to attribute the demodulated symbols to the correct frame. SCLoRa uses sliding windows to identify the symbol edges of each frame from the superposed signal. The complexity of SCLoRa depends on the number and size of these sliding windows. To our knowledge, SCLoRa is currently the collision resolution protocol with the best performance. However, it still fails when the power of the frames are similar and when the relative time offsets are small.

#### C. GS-MAC protocol

GS-MAC has been proposed in [3]. It is a slotted MAC protocol: time is divided into sub-slots of duration  $SD/x$ , with  $x = 4$  typically. End-devices can only send frames at the beginning of a sub-slot. In GS-MAC, an oracle performs the detection of normalized frequencies at the frontier of each sub-slot. The frequencies from the  $x$  sub-slots of a given symbol are intersected together in order to remove the residual frequencies from symbols of frames using different sub-slots. If this intersection returns a set with a single frequency, the frequency corresponds to the symbol value in the frame sent during this sub-slot.

Figure 3 shows an example of GS-MAC with SF7 and  $x = 4$  sub-slots. The figure shows 1.25 symbols of two colliding frames. Frame 1 is sent in sub-slot 1 and is synchronized with the receiver, and frame 2 is sent in sub-slot 2. Frame 1 consists of symbols 12 (fully shown) and 34 (partially shown), and frame 2 consists of symbols 48 (partially shown) and 32 (fully shown). From the preamble (not shown here), GS-MAC knows that there is a frame starting in sub-slot 1 and another in sub-slot 2. For the frame starting in sub-slot 1, GS-MAC computes the set of all instantaneous frequencies for the first  $x$  slots, removes the drift caused by the linear frequency sweep of LoRa symbols, and intersects the sets. The obtained value is  $\{12, 16\} \cap \{32 - 32, 44 - 32\} \cap \{64 - 64, 76 - 64\} \cap \{96 - 96, 108 - 96\} = \{12, 16\} \cap \{0, 12\} \cap \{0, 12\} \cap \{0, 12\} = \{12\}$ , which is indeed the correct value for the first symbol of frame 1. The value of the second symbol of frame 1 cannot be computed as three sub-slots are missing, but it can be observed that 34 indeed belongs to the set of frequencies of the first sub-slot of the second symbol  $\{0, 34\}$ . Similarly, the second symbol of frame 2 is equal to  $\{32, 44\} \cap \{64 - 32, 76 - 32\} \cap \{96 - 64, 108 - 64\} \cap \{0 - 96, 34 - 96\} = \{32\}$ .

With the help of the instantaneous frequency detection oracle, GS-MAC is able to decode most frames that are sent alone in one sub-slot. However, GS-MAC is unable to decode frames sent in the same sub-slot, as the intersection of the  $x$  sets will produce as many values as there are frames colliding.

In [10], the authors proposed to leverage the LoRa coding, and especially the Hamming coding and the diagonal interleaver, in order to decode indistinguishable frames from GS-MAC<sup>1</sup>. They use the fact that not all sequences of symbols are valid, which enables to ignore some uncertainties caused by GS-MAC, and eventually leads to decoding more frames.

<sup>1</sup>The algorithm is more general though, and it can boost the decoding ratio of most collision resolution protocols.

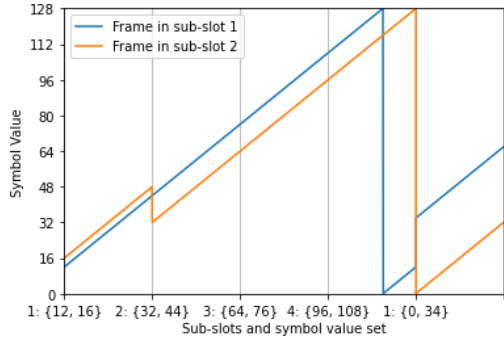


Fig. 3: Frequency changes of up-chirps from two superposed LoRa frames, transmitted in sub-slots 1 and 2, for the GS-MAC protocol.

However, it is not possible to perform instantaneous frequency detection, which is why LoRa detects frequencies over a long duration  $SD$ . Thus, GS-MAC cannot be implemented on a real hardware. Moreover, GS-MAC is a slotted protocol, and is thus not compatible with LoRaWAN end-devices.

#### IV. PROPOSED SF-DS

In this section, we present our main contribution SF-DS, which is a decoding scheme for LoRa collisions based on GS-MAC. First, we explain how to remove the oracle from GS-MAC, which was used to identify the instantaneous frequencies at each sub-slot. Then, we describe how to remove the sub-slots of GS-MAC, which enables legacy LoRaWAN transmissions to be decoded by our algorithm, even if they collide. Third, we discuss other improvements that make our algorithm work in many practical scenarios.

##### A. Adapting the instantaneous frequency detection of GS-MAC

The GS-MAC protocol [3] assumes that it is possible to instantaneously detect all frequencies at the frontier of each sub-slot. However, this is not possible in practice, especially when there are colliding LoRa frames. Thus, our aim in this subsection is to deduce the set of frequencies at the frontier of each sub-slot by sampling over the entire sub-slot.

Let the sample rate be equal to BW, and recall that each symbol is represented by  $2^{SF}$  samples. As  $SD$  is divided into  $x$  sub-slots in GS-MAC, each sub-slot has  $N_s = \frac{2^{SF}}{x}$  samples. Let us multiply the  $N_s$  samples by a normalized down-chirp, after being padded with 0, to remove the time-variant. Thus, we can obtain a constant frequency component from an FFT for each sub-slot. This frequency corresponds to the instantaneous frequency at the middle of a sub-slot, which can be translated to the instantaneous frequency at the beginning of the sub-slot by subtracting an offset corresponding to  $SD/(2x)$ . These  $x$  frequencies can then be combined.

Figure 4 shows this computation for sub-slot 1 (on the left) and sub-slot 2 (on the right), both for an up-chirp of value 48 sent during sub-slot 1 with SF7 and BW=125 kHz, and with  $x = 4$  sub-slots. The two top sub-figures show the part of the up-chirps received during the corresponding sub-slot. The two bottom sub-figures show the result of the multiplication by the down-chirp. It can be seen that the resulting frequency

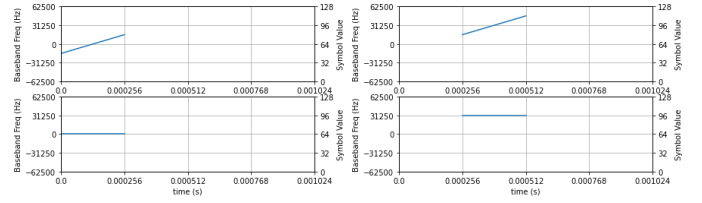


Fig. 4: The frequencies for sub-slot 1 (on the left) and sub-slot 2 (on the right), when encoding symbol 48 with SF7 and BW=125 kHz, with 4 sub-slots. The top sub-figures represent the partial up-chirp, while the bottom sub-figures represent the product with the down-chirp.

for sub-slot 1 is 64, which corresponds to the instantaneous frequency of the chirp 48 at the middle of the first sub-slot, that is after a duration of  $SD/8$ . The instantaneous frequency at the beginning of the first sub-slot is thus  $64 - 2^{SF}/8 = 64 - 16 = 48$ . Similarly, the instantaneous frequency of the chirp at the beginning of the second sub-slot is equal to  $96 - 2^{SF}/8 = 96 - 16 = 80$ . Note that 80 is the correct value, as  $48 + 32 = 80$ .

In GS-MAC, most frames that are sent alone in their own sub-slot can be decoded, even if there are colliding frames sent during other sub-slots. When several frames are sent in the same sub-slot, the receiver is unable to match the symbols with the frames. Figure 5 shows such an example. On the left, the FFT for the first sub-slot is shown. On the right, the FFT for the second sub-slot is shown. In both cases, two main peaks can be observed. Since the reception power of the two frames is the same, the height of each peak is the same, and the receiver cannot match the symbols to the frames. Note that the peaks 64 and 96 correspond to the symbol 64, and the peaks 112 and 16 correspond to the symbol 112. To partially cope with this, we propose to use the method proposed in [10]: in the presence of uncertainties, all the possible values are stored for each symbol, and all possible frames are constructed. Then, the frames whose Hamming bits are not valid after being deinterleaved are ignored. If a single valid frame remains after processing, then the decoding process is successful.

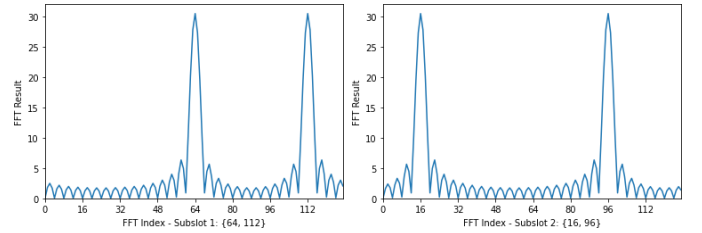


Fig. 5: The FFT result in sub-slot 1 (on the left) and sub-slot 2 (on the right) for two symbols that are synchronized. The values of the symbols are 48 and 96, SF is 7, BW=125 kHz, and there are four sub-slots.

##### B. Removing synchronization from GS-MAC

GS-MAC is a slotted protocol: it requires all end-devices to share a common time through synchronization, and end-devices can only start sending at the beginning of a sub-slot. Thus, GS-MAC is not compatible with LoRaWAN devices, which transmit using ALOHA. The synchronization of GS-MAC is difficult to achieve in practice, and the change in

LoRaWAN end-devices is a large drawback of this protocol. In this subsection, we explain how SF-DS removes this synchronization requirement. SF-DS uses virtual sub-slots to compute symbol values, but the transmissions do not need to be aligned with these virtual sub-slots, making SF-DS compatible with legacy LoRaWAN devices.

In SF-DS, the receiver constantly checks for preambles. The first detected preamble becomes the reference preamble, and enables the receiver to be synchronized with the frame. When new preambles are detected, the receiver stores the relative delay between the new preamble and the reference preamble. Then, SF-DS divides  $SD$  into  $2x$  virtual intervals which are synchronized with the reference frame, but not necessarily with the new frames. The relative delay of each frame is translated into one of these  $2x$  intervals. Intervals  $2x - 1$  and 0 virtually correspond to sub-slot 1 of GS-MAC, intervals 1 and 2 virtually correspond to sub-slot 2 of GS-MAC, and so on. When a symbol  $s_{dec}$  is obtained at the frontier of a virtual sub-slot, the original symbol  $s_{org}$  can be computed based on the relative time delay  $d$  of the frame and the frontier of a virtual sub-slot, using  $s_{org} = s_{dec} - (2^{SF} \cdot d / SD)$ .

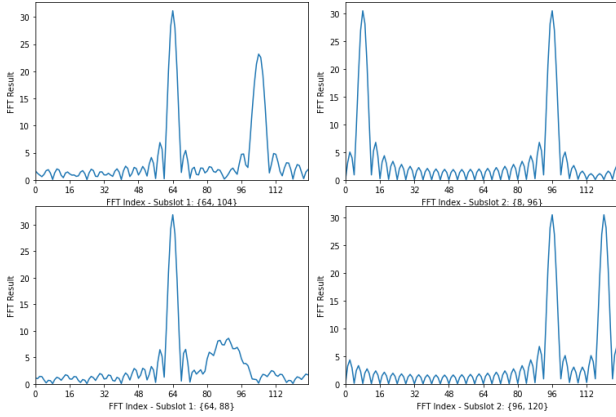


Fig. 6: The FFT results in sub-slot 1 (on the left) and 2 (on the right) for two symbols that are desynchronized. The values of the symbols are 48 and 96, SF is 7, BW=125 kHz, and there are eight intervals (corresponding to four sub-slots). The delay for the second symbol (having value 96) is one quarter of a sub-slot (on the top) or three quarters of a sub-slot (on the bottom).

Figure 6 shows two examples of delayed frames: on top, the delay is  $d = (1/4) \cdot (SD/x)$ , while on the bottom, the delay is  $d = (3/4) \cdot (SD/x)$ , both with  $x = 4$  virtual sub-slots. The sub-figures of the left show the FFTs for sub-slot 1, and the sub-figures of the right show the FFTs for sub-slot 2. For the top sub-figures, the delay is  $SD/16 \in [0; SD/8)$ , so the new frame is considered as being received in virtual interval 0. Thus, our algorithm performs the FFTs as if the two frames were completely synchronized. The FFT peaks in sub-slot 1 are 64 and 104, which correspond to an original symbol of the second frame equal to either  $64 - 8 = 56$  or  $104 - 8 = 96$ , due to the delay  $d$ . For the bottom sub-figures, the delay is  $3 \cdot SD/16 \in [SD/8; SD/4)$ , so the new frame is considered as being received in virtual interval 1. Thus, our

algorithm performs the FFTs as if the two frames were sent on a different sub-slot. The FFT peaks in sub-slot 2 are 96 and 120, which correspond to an original symbol of the second frame equal to either  $96 - 24 = 72$  or  $120 - 24 = 96$ . In both cases, the correct value of the symbol (which was 96) was successfully retrieved.

### C. Discussion on further improvements

To make SF-DS more practical and to further improve its performance, we implemented additional features that we briefly discuss here.

*Tolerating imperfect FFT results:* LoRa demodulators perform an FFT to detect the symbol frequency based on  $2^{SF}$  samples. The samples are usually multiplied by a window function to limit the spectral leakage of the FFT. In SF-DS, these samples are regrouped into virtual sub-slots. However, due to the padding of the partial up-chirp when performing the multiplication with the down-chirp, the incomplete samples yield imperfect FFT results. This can be observed by comparing Fig. 2 (perfect FFT results) and Fig. 5 (imperfect FFT results). With SF-DS, the main lobes of the FFT might have a width larger than one value. This width depends on the window function used by the FFT. Thus, we chose to use a Hamming window, which brings a resolution of about 5 values. In other words, the difference between the highest detected peak and the actual value can be up to 5 due to the close symbols from superposed frames. This behavior can be observed from the second peak of the top-left sub-figure of Fig. 6, which is wider than the first (however, the peak width is still smaller than one in this example though). Another example is when two close symbols are superposed, for instance 92 and 94: the detected value can be erroneous, with an error of 1 or 2. Consequently, SF-DS might not be able to decode the symbols by performing intersections over several virtual sub-slots. To correct this intrinsic demodulation error, we decided to add a tolerance of up to 5 symbols in SF-DS, which corresponds to the resolution of the window function. This tolerance is taken into account when performing the intersection of the frequency sets. While it brings occasional symbol errors of less than 5 values, it decreases the missing symbols due to the intersections in GS-MAC.

*Weaker peaks due to the time offset:* The FFT peaks of a delayed frame are generally weaker than the FFT peaks of the reference frame, due to the delay. This can be seen on the top-left sub-figure of Fig. 6. Indeed, even if the second frame was transmitted after the first frame, it is still considered being transmitted in the same virtual sub-slot (due to the delay rounding mechanism, introduced to remove the synchronization). This weaker FFT peak is easily influenced by the side lobes from the first frame, and is thus likely to be incorrectly decoded. In order to cope with this, we use the FFT results from the next virtual sub-slot (see the top-right sub-figure of Fig. 6), as they are less influenced by this delay, and are thus stronger. For instance, the demodulator might be unable to determine whether the peak of the delayed frame in sub-slot 1 is 104 or 105, due to the side lobe of the reference



symbol. However, the demodulator is able to identify that the peak is weaker than what is expected. Thus, it checks for sub-slot 2 to identify that the correct value is 104.

## V. SIMULATION RESULTS ON GNU RADIO

We implemented SF-DS on the GNU Radio SDR platform, in which the base-band in-phase/quadrature (I/Q) samples are transmitted and received in a channel. Each frame encodes 22 bytes that are randomly generated, with  $CR=4/5$ ,  $BW=125$  kHz and SF varying from 7 to 12. The low data rate mode is enabled for SF 11 and 12. This results into a payload of 43, 38, 38, 33, 33 and 33 symbols respectively, for SF 7-12. Each configuration is repeated 100 times.

### A. Simple scenario: two colliding frames with the same power

We first consider a simple scenario where there are only two colliding frames, with the same power. The second frame has a random delay between 0 and the duration of the first frame, which guarantees a collision. SF-DS uses 4 virtual sub-slots during the demodulation.

Figure 7 shows the Symbol Error Rate (SER) and the throughput, for LoRa, SF-DS, and a perfect decoding scheme. As the second frame arrives after the beginning of the first frame, LoRa can demodulate the beginning of the first frame. However, when the payload or preamble of the second frame collides with the first frame, the capture effect of LoRa cannot apply due to the similarity of the power of the two frames, and LoRa is unable to decode the symbols. This leads to a SER of more than 50% for LoRa, and a maximum throughput of about 900 bps for SF7. With SF-DS, symbol errors arise only when there are two superposed symbols with similar values (namely,  $\pm 5$ ). Indeed, in this case, it is hard to differentiate the peaks of the FFT. Since each symbol encodes SF bits, the probability of having two superposed symbols with similar values decreases as SF increases, which quickly reduces the SER. In terms of throughput, it can be seen that SF-DS is close to the maximum throughput assuming a perfect collision resolution scheme.

### B. Complex scenario: periodic transmissions of several nodes

We then focus on a more realistic scenario. We used the same setup as [9], in order to stay consistent in the comparison. There are between 2 and 20 nodes. All transmissions are received with a similar power. The SNR is set to 20 dB. Each node sends frames with an interval varying between 1000 ms and 2200 ms. The duration of each frame is about 103 ms for SF8, and 1480 ms for SF 12. Therefore, the risk of collisions is larger with SF12. We assumed that there is no duty-cycle limitations in order to simulate a heavy traffic. Again, SF-DS uses 4 virtual sub-slots during the demodulation.

Figure 8 shows the throughput as a function of the number of nodes, for SF8 and SF12, and for four protocols: LoRa, SCLoRa, FTrack and our proposed SF-DS. Note that the results for SCLoRa and FTrack are taken from [9], as we used the same setup. For SF8, the network is not yet saturated because there are fewer collisions. Thus, the performance of SF-DS is similar to the one of SCLoRa, which is currently

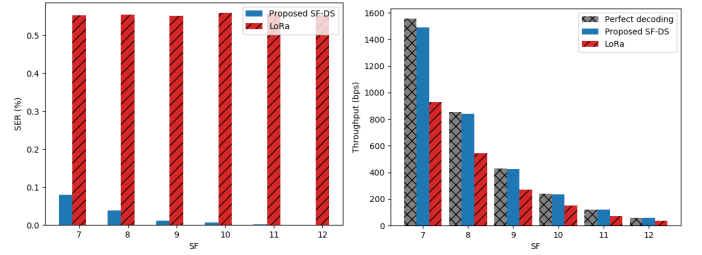


Fig. 7: When only two frames collide, SF-DS has a very low symbol error rate, which yields a throughput which is close to the maximum throughput achievable by a perfect decoding scheme.

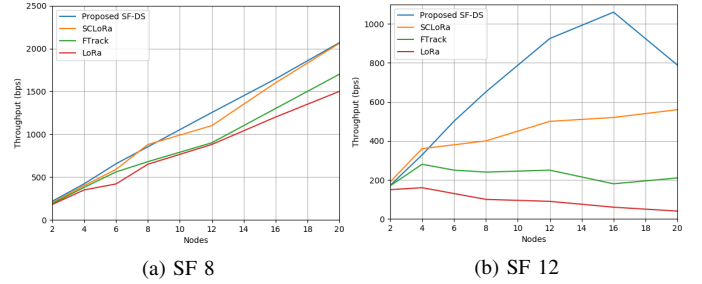


Fig. 8: Our proposed SF-DS has slightly better performance than SCLoRa for SF8, but has much better performance than all algorithms for SF12, especially when the number of nodes is large.

the best known collision resolution protocol for LoRa. For SF12, SF-DS is able to achieve a much higher throughput than SCLoRa: the gain is 103% at 16 nodes. This huge gain comes from three main factors: (i) the virtual sub-slots help identifying the individual frequencies when symbols are slightly desynchronized, especially as weaker FFT peaks can be tracked among virtual sub-slots, (ii) the use of a tolerance of  $\pm 5$  values when identifying imperfect FFT peaks, and (iii) the use of the redundancy bits from the Hamming code further improves the symbol decoding.

## VI. CONCLUSION

Collisions in LoRa negatively impact the network performance, and especially the throughput which is already very limited. Thus, collision resolution algorithms have been proposed in order to decode colliding frames. In this paper, we proposed a new decoding scheme called SF-DS, with the following features: (i) SF-DS does not require the transmitters to be synchronized, and is thus able to operate with legacy LoRaWAN devices, (ii) SF-DS uses virtual sub-slots in order to regroup and track signals sent at different times, (iii) SF-DS identifies ambiguous FFT peaks and takes into account the uncertainty caused by the Hamming window of the FFT and (iv) SF-DS leverages the spreading of bits produced by the LoRa interleaver in order to improve the frame decoding rate. We implemented SF-DS on GNU Radio. Our results show that SF-DS outperforms LoRa and the other collision resolution protocols. In our future work, we plan to improve our algorithm to correct additional errors caused by the noise and carrier frequency offset.

## REFERENCES

- [1] LoRa Alliance Technical Committee. LoRaWAN 1.1 Specification. Standard V1.1, LoRa Alliance, 2017.
- [2] Semtech Corporation. AN1200.22 LoRa Modulation Basics. Application note Revision 2, Semtech, 2015.
- [3] N. El Rachkidy, A. Guitton, and M. Kaneko. Generalized Slotted MAC Protocol Exploiting LoRa Signal Collisions. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, September 2020.
- [4] Semtech Corporation. LoRaWAN Specification v1.1. Technical Report Revision B, Semtech, 2017.
- [5] Semtech Corporation. LoRaWAN v1.1 Regional Parameters. Technical Report Revision A, Semtech, 2017.
- [6] B. Laporte-Fauret, M. A. Ben Temim, G. Ferre, D. Dallet, B. Minger, and L. Fuche. An Enhanced LoRa-Like Receiver for the Simultaneous Reception of Two Interfering Signals. In *Proceedings of Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, September 2019.
- [7] 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 Special Interest Group on Data Communication, SIGCOMM '17*, August 2017.
- [8] Xianjin Xia, Yuanqing Zheng, and Tao Gu. FTrack: Parallel Decoding for LoRa Transmissions. In *Proceedings of the ACM 17th Conference on Embedded Networked Sensor Systems*, New York, November 2019.
- [9] B. Hu, Z. Yin, S. Wang, Z. Xu, and T. He. SCLoRa: Leveraging Multi-Dimensionality in Decoding Collided LoRa Transmissions. In *Proceedings of the IEEE 28th International Conference on Network Protocols (ICNP)*, 2020.
- [10] W. Xiao, N. El Rachkidy, and A. Guitton. Recovering Colliding LoRa Frames from Uncertainties Using LoRa Coding. In *Proceedings of the IEEE 46th Conference on Local Computer Networks (LCN)*, 2021.