

Using multiple Spreading Factors simultaneously in LoRa 2.4 GHz: some solutions

Augustin Bonnel 

Univ. Grenoble Alpes, Grenoble INP*, CNRS, LIG, GIPSA-Lab
38000 Grenoble, France
augustin.bonnel@univ-grenoble-alpes.fr

Martin Heusse 

Univ. Grenoble Alpes, Grenoble INP*, CNRS, LIG
38000 Grenoble, France
martin.heusse@univ-grenoble-alpes.fr

Jose-Ernesto Gomez-Balderas 

Univ. Grenoble Alpes, Grenoble INP*, CNRS, GIPSA-Lab
Grenoble, France
jose-ernesto.gomez-balderas@grenoble-inp.fr

Abstract—At 2.4GHz, all LoRa transceivers need to agree in advance on the spreading factor and bandwidth. However, a lot of applications could benefit from using e.g. SF5 and SF12 simultaneously, to communicate far with some nodes and fast with others. In this paper we propose several solutions that do not require global synchronization between nodes. The first family of solutions consists in adding more hardware, with various levels of coordination between the transceivers. Furthermore, we also demonstrate that, by mixing preamble sampling, repetitions and alternating sampling, it is possible to exchange SF5 and SF12 frames using only one transceiver. We validate the proposals using a discrete event simulator to assess the benefits of each solution. We show that we can realistically exploit two virtual transmission channels with complementary properties, without having to choose once and for all between range and data rate.

Index Terms—LoRa 2.4 GHz, Spreading Factors, Preamble Sampling, Repetitions, Discrete Events Simulation

I. INTRODUCTION

LoRa is receiving a lot of attention for a wide range of applications ranging from large scale meter data collection to real-time FPV drone control. The versatility of LoRa stems from the fact that its modulation parameters are very potent and accessible. Moreover, for all settings, the transceivers are low power and low cost. LoRa was designed to operate in ISM bands, with a strong initial focus on the sub-GHz band for LPWAN (Low Power Wide Area Network) type of usage. In this paper, we focus on the 2.4 GHz band in which channel access per node is not as limited as in the sub-GHz bands.

The LoRa modulation has been available in the 2.4 GHz band for a few years now; it operates in a worldwide available band and frees the applications of any duty cycle constraint. Some systems started to use it extensively such as *ExpressLRS*. LoRa 2.4 GHz operates similarly to its parent modulation in the sub-GHz band. One important difference is that in the sub-GHz band, some chips are intended to be at the heart of gateways: they can track many frequency channels and spreading factors simultaneously for parallel reception of up to tens of frames. In the 2.4 GHz band, there is no comparable hardware. To mimic a gateway, the existing proposals consist in having multiple statically configured chips on the same

node [1], [2]. The typical application of LoRa 2.4 GHz is that of point to point or point to multi-point communication on a previously agreed-on spreading factor and frequency channel. Depending on the (fixed) parameters, LoRa provides a large choice of communication ranges and data rates [3], [4].

Our objective though, is to tackle situations in which slow long range and fast short range communications would have to happen simultaneously, so that every nodes of the network could benefit from the use of different communication ranges and data rates. In this way, the communication between nodes would become more versatile, somewhat bringing to LoRa 2.4 GHz chips the capabilities only associated with sub-GHz LoRa gateways. Some systems have a true need for such diversity of communications.

For instance in multi robot systems, collaboration between the agents often implies direct and frequent communications between them. In addition, the range of such systems would be improved by having at the same time long range communications with other groups of robots or a monitoring station. As an example, Brinon-Arranz et al. [5], propose that the robots leverage a circular formation to achieve a collaborative source seeking. The solutions presented in this paper would be an enabler for their system: a group of robots could communicate with other groups of robots performing a similar task with the long range communication link while using short range communication for collaboration within the group. As another example, a swarm of autonomous UAVs (such as described by Chaumette et al. [6]) could leverage the short communications between the UAVs collaborating on the task, while maintaining a long range reliable communication link with the ground control station. This idea, under the concept of “bridge node”, was briefly introduced in [7]. In our setup, with the simultaneous use of the two modulations, bridge nodes would be transparent.

The difficulty is that having a radio for each type of traffic is obviously more costly and cumbersome than managing to make it work with a single radio. More importantly, even though transmissions on different SFs and frequency channels are well isolated from each other in general, the transmission from an antenna will jam the signal from a distant sender on

*Institute of Engineering Univ. Grenoble Alpes

all receivers located on the same equipment.

Our problematics are thus to coordinate the activities of the radios and make sure that one type of traffic does not monopolize the channel.

II. CONTEXT

LoRa transmission uses Chirp Spread Spectrum (CSS): LoRa symbols are chirps, simple frequency ramps within a given band that encode data in their frequency offset. Two parameters adjust the bitrate, knowing that a chirp at SF i carries i bits. On the one hand, the spreading factor (SF) defines the chirp duration, which doubles with every SF increment. On the other hand, the bandwidth specifies the frequency range that the chirp sweeps: doubling the bandwidth halves the duration of a chirp at same SF .

Adjustability and low transceiver complexity are not the only desirable characteristics of CSS. An important feature is that transmissions at different SF are quasi-orthogonal [8]; in other words, reception at SF i is mostly independent of concurrent traffic at SF j . Also, once a receiver is locked on a frame after preamble reception, it largely ignores other transmissions even at the same SF , provided that they are not markedly more powerful [9]–[12].

The Time-On-Air of a LoRa symbol is defined by:

$$T_{SF} = \frac{2^{SF}}{BW}, \quad (1)$$

with SF the spreading factor and BW the bandwidth of the symbol. Then, the frame time on air is roughly:

$$ToA \simeq (N_{PHY_preamble} + \frac{N_{bit_packet}}{SF} CR) T_{SF}, \quad (2)$$

where $N_{PHY_preamble}$ includes the preamble sequence of typically 8 chirps and the frequency correction part of 4.25 or 6.25 symbols; CR is the error correction code rate ($CR = \frac{4}{5}$ here) and N_{bit_packet} includes header, payload and error detecting code. For the simulations of this paper, we use the precise formulas provided by Semtech [13]. Furthermore, in the context of this paper we focus on SF5:

$$T_{SF5} = \frac{2^5}{1625} ms \quad (3)$$

or SF12 with half the bandwidth:

$$T_{SF12} = \frac{2^{12}}{812.5} ms, \quad (4)$$

so that the SF12 symbols are 256 times longer than the SF5 symbols.

With LoRa, the preamble changes with the data rate. In this way, a receiver will ignore frames on a spreading factor that it is not looking for.¹

In contrast with IEEE 802.11 for instance, a benefit of having different preambles is that the typical preamble of about

10 symbols is proportional to the symbol duration, so that the overhead is the same for all data rates. This is fortunate, since the symbol duration (and broadly, the data rate) can vary by 2 orders of magnitude depending on the settings.

The preamble size is configurable, though, and in the rest of this paper, we use this possibility to avoid losing frame by missing the preamble in specific circumstances.

In general, an active LoRa radio can either be in sending (Tx) or receiving (Rx) mode. In Rx mode, there are actually 3 substates: the radio may be in idle listening state, waiting for a preamble; after this, it can be locked on a frame reception; or it can finally be in Channel Activity Detection (CAD) mode, assessing if there is any kind of radio activity on the channel.

III. SOLUTIONS

As stated in the section I, symbols modulated by different spreading factors are nearly orthogonal, and that makes parallel transmissions between two pairs of radios possible in parallel. The flip side is that radios configured to operate on different spreading factors will ignore each other. To overcome this deafness problem and benefit from joint use of multiple spreading factors, we propose two families of solution: increasing hardware capabilities by using multiple transceivers per nodes, or changing the mechanisms of channel sampling and frame formats, in order to operate seamlessly on both spreading factors at the price of having to design a protocol to orchestrate these changes. The latter solution is more lightweight in terms of hardware, implying lower costs, weight and general complexity of the nodes.

As the minimum preamble size for a frame is 8 symbols [13] and knowing that the maximum preamble size has to be specified at the receiver side, we assume that in the worst case the SX1280 will take 8 symbols to detect the frame and lock on it (we guess that the SX1280 is able to do it in less, but we choose to be conservative). We assume that a radio locked in reception on an incoming frame waits until reception is complete to switch to sending mode, so the emission needs to be delayed (or dropped).

A. Using two radios

A node equipped with two radios is using one for SF5 and the other for SF12. Symbols modulated by different spreading factors are nearly orthogonal, but not enough that a receiver placed near a transmitter would not suffer from the interference. In layman's terms, a radio emitting a frame, on a two-radios node, will scramble any reception occurring on the other radio. Otherwise, a node is able to receive frames modulated by different spreading factors at the exact same time as well as sending frames on both spreading factors. In order to mitigate interference between the radios of the same node, we propose 3 different levels of coordination between the radios.

1) *Two radios without coordination:* In this case we do not have any coordination between the radios of the same node. The SF5 (resp. SF12) radio scrambles the SF12 (resp. SF5) radio channel samplings and receptions (even if the latter is

¹The situation is a bit more complex for the recent LR2021 transceivers, that are able to detect up to 4 spreading factors in parallel [14]. However, for $BW > 500$ kHz and $SF \geq 10$ the parallel detection is limited to 2 spreading factors, with a maximum difference of 4 between the spreading factors. This results in a parallel detection of SF8 to SF12 for the setup with best flexibility.

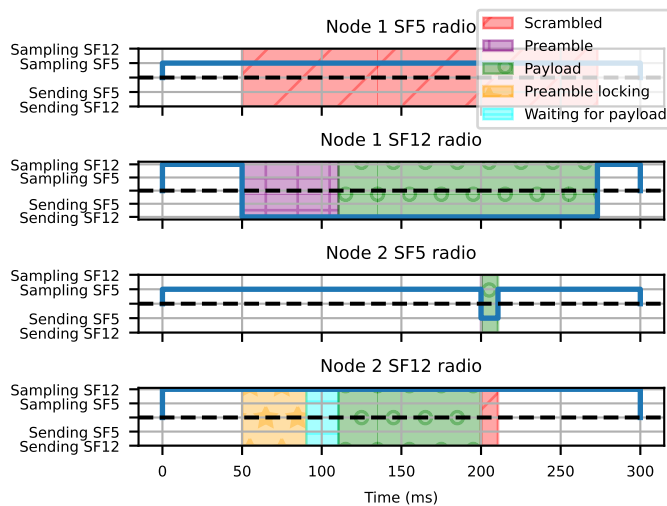


Fig. 1. Chronographs for the case of two-radios nodes without coordination. There is one subgraph per radio, so node 1 corresponds to the first and second chronographs etc. The blue line represents the state of the radio across time. Here, node 1 is sending a SF12 frame from $t = 50$ ms with its corresponding radio. During the transmission, node 1's SF5 radio is scrambled. Node 2 starts to detect the SF12 frame from node 1, locks on it, waits for the end of the preamble and then starts to receive the payload. Later on, node 2 generates an SF5 frame. It scrambles its own SF12 radio and aborts the reception of the SF12 frame. Node 1 does not receive the SF5 frame from node 2 as its SF5 radio is also scrambled. Node 2 is send is SF5 frame because there is no coordination between its two radios.

locked in reception on a frame!). We have chosen to explore this solution as a comparison against the other solutions including coordination between the two radios. Furthermore it is the obvious naive approach when thinking about a solution to communicate on two different spreading factors simultaneously.

The functioning of this solution appears in Figure 1: this example illustrates the case of an interruption of a SF12 frame reception from a node by a SF5 emission from the same node. As a result none of the transmissions succeed.

2) *Two radios with partial coordination:* Here, we propose that a radio from a node locked in reception makes the other radio on the same node reschedule or drop the frames it may want to send during the reception.

In Figure 2, the two nodes follow the same scenario as in Figure 1, but with more coordination. The second node, which is receiving the SF12 frame, drops the SF5 frame it intended to send because its other radio is locked in reception on the SF12 frame. As a result every frames that have been actually sent on the channels are received. However, if the SF5 frame arrived during the preamble locking period (yellow stars area in Figure 2), node 2 would send it and potentially miss the SF12 frame in the end.

3) *Two radios with complete coordination:* This solution is a small evolution of the previous one. The coordination between the radios is extended to the preamble locking phase, as soon as the radio detects activity on the channel and even though it still needs 8 preamble symbols to lock on the incoming frame (we made this assumption at the beginning

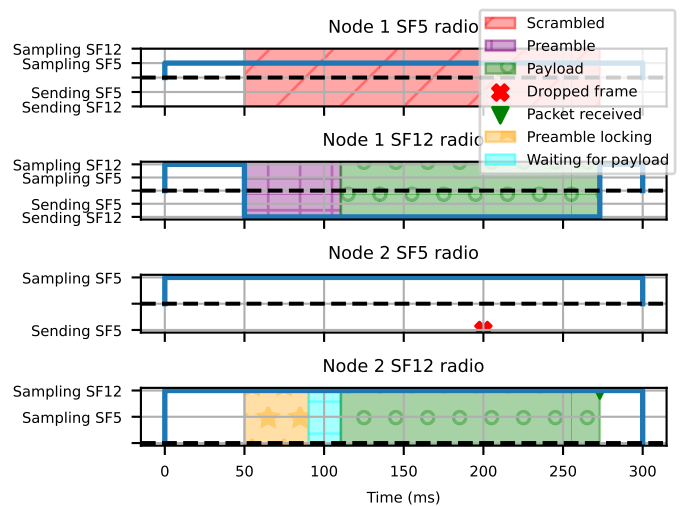


Fig. 2. Chronograph illustrating the two radios with partial coordination solution. There is one chronograph per radio. It is the same scenario as described in Figure 1, but here node 2 can not send the SF5 frame as it is already receiving the SF12 frame from node 1.

of this section III). This solution is not necessarily realizable (how to assert for activity before actual frame detection?), but the comparison with the other two radios solutions allows to gain some insights on the performances and plays the role of a baseline for comparisons purposes only. Indeed, as it is not achievable, the status of “solution” of this method is questionable. But as we do not know the capabilities of future hardware, this method will be treated as a theoretical solution.

B. Using one radio

In the case where multiplying hardware components is not desirable and even not feasible, changing the way to operate the radio is mandatory. Indeed, the lack of supplementary hardware forces to modify SF5 and SF12 channels sampling. This leads to a change in the format of the frames. As a consequence all nodes of the network must comply with this protocol.

On the one hand, the radio samples both channels alternatively to be able to detect frames on both spreading factors. On the other hand, to overcome the deafness brought by the discontinuous channel sampling, we extend the preamble of the SF12 frames similarly to preamble sampling techniques and we repeat SF5 frames so they fall into at least one SF5 sampling window.

1) *Alternate sampling:* We want to be able to receive SF5 and SF12 frames, so we need to have a way to sample frequently both SF5 (resp. SF12) channels for incoming frames. We chose an approach with no global scheduling among nodes. Thus, to virtually sample the two channels simultaneously, we need to visit alternatively both channels fast enough in order to detect incoming frames. As soon as activity is detected, the transceiver should stay on the channel in order to receive the incoming frame.

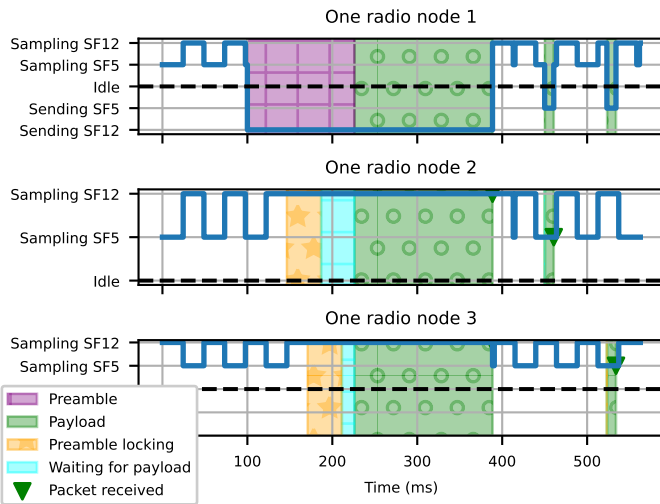


Fig. 3. Chronograph of the one radio solution. There are 3 nodes in this example, node 1 generates frames (at SF12, then SF5) whereas nodes 2 and 3 receive them. Node 1 sends a SF12 frame, node 2 detects it, first with a CAD, then locks on it and receives the payload; node 3 does the same by detecting the frame a bit later. Node 1 sends an SF5 frame, which is duplicated once: node 2 gets the first occurrence whereas node 3 gets the repetition. The preamble and locking phases are still present for the SF5 frames but not visible at this scale.

The SX1280 embeds a *Channel Activity Detect* (CAD) mechanism. This mechanism allows to detect activity (preamble symbols or data symbols) on a given spreading factor bandwidth pair. Keeping in mind that sampling duration is dependent of the spreading factor bandwidth pair, the time we spend trying to detect activity on the SF12 channel is much larger than the time required to detect the same “amount” of activity on SF5. Furthermore, all the time we are sampling the SF12 (resp. SF5) channel, is time where the radio is not detecting activity on the SF5 (resp. SF12) channel. For the sake of minimizing the SF12 sampling, we are going to use the CAD mechanism configured to a scanning period of 4 symbols (smaller than the assumption we made about the 8 preamble symbols required to lock on a frame, but reliable enough). Optimizing the sampling time of the SF5 channel is not relevant as the Time-On-Air of a SF5 frame is negligible compared to the Time-On-Air of a SF12 frame. Moreover the reduction of the SF5 sampling period would lead to less time dedicated for SF5, which is contradictory to our data rate objective. In such, we have chosen to have the same sampling period T_{sampling} for SF5 and SF12 [13]:

$$T_{\text{sampling}} = T_{\text{CAD}_n} = (n_{\text{symbols}} + \frac{2 \times SF + 3}{32}) \times T_{SF}, \quad (5)$$

where n_{symbols} is the number of symbols used for the scanning period. The rest of the equation corresponds to the analysis of what was measured during the scanning period.

Similarly to the number of preamble symbols required to lock on a frame, we will consider a conservative policy regarding the number of symbols (preamble or not) required to detect activity on a channel with a CAD: the CAD is

considered negative if we do not have 4 full symbols on the channel (in reality the chip might be able to assert activity with less symbols, but as a CAD is launched for a certain number of symbols and the analysis is made afterwards, its duration is fixed).

On the one hand, the receiver will use the CAD mechanism to detect activity during a SF12 sampling period. If the result is positive (activity detected), the transceiver will stay on the SF12 channel and launch a sampling of the channel. To receive the frame, the transceiver would still need to sample 8 full preamble symbols (as the CAD is a different mechanism from the reception mechanism, the SX1280 still needs to assert that the detected activity is an incoming frame). On the other hand, for SF5, at the start of a SF5 sampling period, the receiver will begin to sample the channel and wait for at least 8 preamble symbols of an incoming frame. The functioning of the alternating sampling is displayed in Figure 3: node 1 and 2 starts by sampling SF5 while node 3 starts by sampling SF12. Every SF12 sampling period is a CAD whereas every SF5 sampling period is a regular sampling/listening of the channel.

As the duration of a sampling period is equal to the time of a CAD of 4 symbols, this means that if a SF12 sampling period is shrunk (by an interruption: SF5 frame emission, or reception ending during a SF12 sampling period), the CAD will be launched with the guarantee of being negative (interrupted before the end). In the interest of not spending time sampling for nothing, we decided to implement a rule specifying that a CAD is launched until the end (if the sampling is not interrupted, frame emissions still have an higher priority than channel sampling). This leads to a reduction of the next SF5 sampling period. For instance: if a node sends a SF5 frame that ends at the middle of a SF12 sampling period, the latter will be resumed by a CAD, which will end in the middle of the next SF5 sampling period (provided that the CAD is not interrupted again). For the sake of not modifying the whole sampling schedule of the node, the SF5 sampling period is simply shrunk. Up to a point, as SF5 frames (and their detection) are short compared to the allocated SF5 sampling period, the node will be able to receive some frames during the remaining SF5 sampling time. This mechanism is visible in Figure 3, nodes 1 and 2 start a full CAD after emitting (resp. receiving) the SF12 frame.

To wind it up, we have a way to imitate the simultaneous sampling of SF5 and SF12, but even with this best effort solution, we still need to change the format of the frames. For instance, if a node sends a SF5 frame, every node, that is not sampling the SF5 channel at this time, will miss it. Similarly, if a node sends a SF12 frame, every node that started sampling SF12 just before the start of the frame would miss it (as it would not detect the frame early enough to lock on it with the default preamble length of 12 symbols). So, for this solution we need to change the way we are sending SF5 and SF12 frames.

2) *SF12 frames, Preamble sampling:* As sampling one channel is equivalent to have the radio turned off on the other,

we have a problem of sporadic listening. One existing solution to this problem is preamble sampling. This technique, widely used in wireless sensors networks [15], consists in extending the preamble of the frames in order that the recipient turns on its radio at least once during the preamble [16]. In our case, we can extend the preamble of the frames to ensure that radios sampling alternatively SF5 and SF12 will be able to listen to the preamble and lock on it.

This solution is well suited for SF12 because the duration of a sampling period is in the same order of magnitude as the duration of the default preamble, so the required extension will have a reasonable size. Moreover SF12 is not meant to provide a high data rate. Conversely, since SF5 symbols are very short compared to SF12 symbols, extending the preamble to cover a sampling cycle would result in a very long SF5 preamble, which would hinder the data rate advantage brought by SF5. As a consequence we use preamble sampling only for SF12, and the required duration of the preamble is computed as follows:

$$T_{\text{preamble}} = 3 \times T_{\text{sampling}} + T_{\text{RxMode}}, \quad (6)$$

with T_{preamble} the required duration of the SF12 preamble, T_{sampling} the duration of a sampling period (as defined in eq. 5) and T_{RxMode} the duration of the 8 symbols needed to lock on a SF12 frame. The preamble needs to be long enough to allow a radio to lock on the frame, that is why it needs to be at least longer than T_{RxMode} . Then, our preamble should be long enough to reach all radios regardless of their sampling schedule. As mentioned in section III-B1, we consider that a CAD that starts just before activity on the channel is considered negative, so the worst case scenario is a radio starting to sample SF12 just before the SF12 frame arrives. This results in a negative first CAD, then a SF5 sampling period and then another SF12 sampling period, which is why we need to add $3 \times T_{\text{sampling}}$ to the preamble.

We need to have $n \times T_{\text{SF12}} \geq T_{\text{preamble}}$. By increasing n one by one, we found that $n = 23$ leads to $n \times T_{\text{SF12}} \geq T_{\text{preamble}}$. However, we empirically discovered that $n = 25$ allows a node to receive a SF5 frame of 255 bytes starting at the end of a SF5 sampling period of the node (with the SF12 frame starting one symbol after the start of the SF12 sampling period prior to the SF5 sampling period). The retained value for the SF12 preamble length is thus $n = 25$ symbols.

The preamble sampling mechanism can be witnessed in Figure 3. We can see that node 1 sends a SF12 frame with an extended preamble (in Figure 3, the preamble is represented by the purple grid areas and the payload by the green dot areas). At the start of the frame, node 2 is sampling the SF5 channel, after that it starts a SF12 CAD. The CAD is positive so the node stays on SF12 and tries to lock on the frame (yellow stars in Figure 3), after 8 symbols the node asserts that a frame is present so it locks in reception and waits the end of the preamble (cyan grid in Figure 3). Then it receives the payload (green dots in Figure 3). Node 3 does the same but detects the frame a bit later as it started to sample the SF12 channel just before the SF12 frame, so it detects the activity

on the next SF12 sampling period only. On the same figure we can also see the CAD resuming mechanism: for nodes 1 and 2 the frame ends during a SF12 sampling period, so the CAD is launched for its entire duration and overlaps the next SF5 sampling period. This SF5 sampling period is shrunk, the next SF12 sampling period is starting in line with the original sampling cycle of the node.

3) *SF5 frames, Repetitions*: As mentioned in the previous section III-B2, using preamble sampling techniques for sending SF5 packets would hinder the data rate capabilities of the modulation, which is the advantage we are trying to leverage by using the SF5 modulation. Furthermore the duration of a SF5 sampling period is large compared to the time needed to lock on a SF5 frame. In addition, when the radios are in idle listening state (not receiving or sending a frame), they are either sampling the SF5 or the SF12 channel. So by sending the same frame twice with a delay equal to an odd multiple of the first occurrence, the idle listening radios would get either the first occurrence or the second occurrence of the frame. The date of the SF5 repetition can be computed as follows:

$$t_{\text{RepSF5}} = r \times T_{\text{sampling}} + t_{\text{SF5}}, \quad (7)$$

with t_{RepSF5} the date of the second occurrence of the SF5 frame that was first sent at t_{SF5} and $r > 1 \in \{2k + 1 : k \in \mathbb{N}\}$ the multiple of the sampling period used to define the repetition delay.

In order to cover the corner cases where the radio starts to detect a frame but is interrupted by the start of a SF12 sampling period, we extend the preamble to 16 symbols. So, the radio will be able to sample the last 8 preamble symbols from the repetition.

We have decided to set r to 3 in order to not increase too much the delay between the first occurrence of the packet and its second. In Figure 3 node 1 sends the first occurrence of a SF5 frame at 450 ms and its repetition at around 523 ms. The payloads of both frames (original and repetition) are strictly similar. Repeating the frame more than once would allow to reach nodes that were not listening or improve the reliability of our SF5 transmissions, but we do not evaluate this possibility herein. Overall, as SF5 frames are very short, re-sending the same payload a second time (a frame and its repetition) does not erase the benefits of the modulation speed.

IV. DISCRETE EVENTS SIMULATION

In order to compare the different solutions to use simultaneously two spreading factors in LoRa 2.4 GHz, we have developed our own discrete events simulator. Our goal is to assess the performance variations between the solutions and not to reproduce specific use cases. Here, we consider that all nodes are in the communication range of each other both at SF5 and SF12.

A. Simulator and simulation parameters

By reproducing the behavior of the SX1280 chip [13] and implementing our solutions in the simulator, we are able to simulate the timings and mechanisms related to the reception,

emission, channel activity detection and frame locking. On the one hand, this allows to validate the mechanisms involved in our solutions on simple scenarios and on the other hand, to implement random traffic generation. In addition to validating the solutions behavior in complex situations, random traffic provides statistics to compute the performances of the different solutions. The obtained performances are here only for relative comparisons between the different solutions as the simulator is not validated against real time measurements.

Each node generates packets for both spreading factors following a Poisson distribution with a sending rate dependent of the spreading factor. Similarly to the assumption we made in Section III, the nodes follow a conservative policy concerning the requirements to lock on a frame: they need to intersect 8 preamble symbols to start the reception. We also consider physical capture: we consider that time and sensitivity diversity across the receivers will allow them to receive one of two concurrent frames [12]. More than two frames on the channel result in a collision, and no reception succeeds. All frames with the same spreading factor have same fixed size. When not using preamble sampling, the nodes use a preamble of 12 symbols. The preamble size of the one radio solution is as described in the previous section III-B. A simulated time of 40 minutes, a SF5 (resp. SF12) payload size of 255 (resp. 20) bytes and a mean packets generation rate of 4 (resp. 0.2) packets per second for SF5 (resp. SF12), are the other parameters for the majority of the simulations.

B. Results metrics

To have a comparison between the different solutions, we have decided to look at the average data extraction rate (payload data rate). We have chosen this metric because it is independent of the format and number of repetitions of the frames, which vary between our solutions. The obtained average data extraction rate of a solution only has a meaning when comparing it with the average data extraction rates of other solutions obtained with the same simulator.

For this metric we need to know the average number of receptions by our nodes. Since we have multiple receivers, we have decided to weigh the amount of valid reception events (signaling that a node has correctly received a frame), by the total number of receivers:

$$R_{SF} = \frac{E_{\text{receptions}SF}}{N_{\text{receivers}}}, \quad (8)$$

with R_{SF} the average number receptions per node of frames modulated by SF , $E_{\text{receptions}SF}$ the total number of valid reception events of spreading factor SF and $N_{\text{receivers}}$ the total number of receivers.

The average data extraction rate is the average payload a node has received, divided by the simulation time:

$$D_{SF} = \frac{R_{SF} \times P_{SF}}{T_{\text{simulation}}}, \quad (9)$$

with D_{SF} the average payload data rate perceived at SF , P_{SF} the payload size of frames modulated by SF and $T_{\text{simulation}}$ the simulation duration.

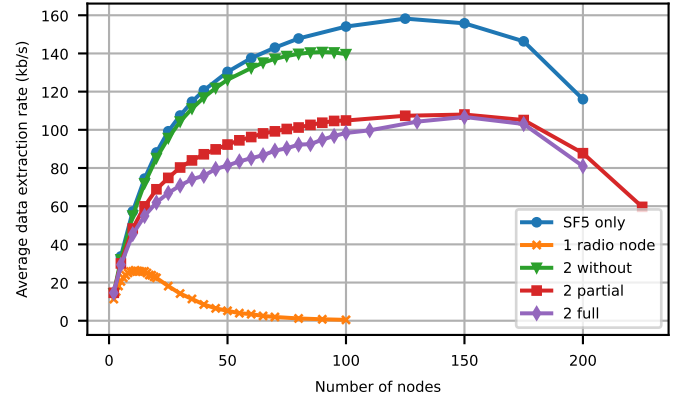


Fig. 4. Comparison of the SF5 average data extraction rates of the different solutions against the baseline: SF5 average payload data rate against a varying number of nodes with the same sending rates. “1 radio node” refers to the one radio node solution, “2 without” refers to the two radios node without coordination solution, “2 partial” refers to the two radios node with partial coordination solution and “2 full” refers to the two radios node with full coordination solution.

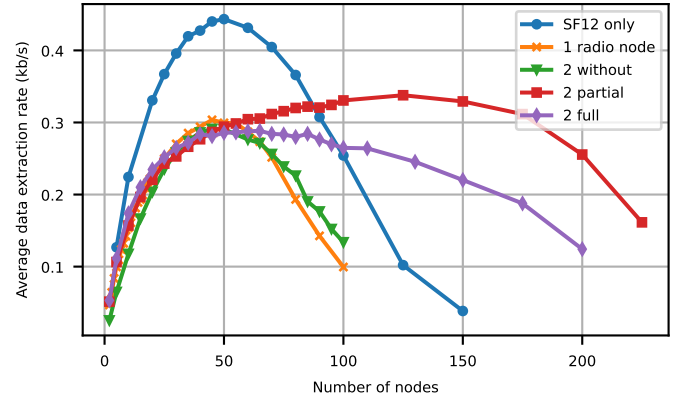


Fig. 5. Comparison of the SF12 average data extraction rates of the different solutions against the baseline: SF12 average payload data rate against various number of nodes with the same sending rates.

V. RESULTS AND ANALYSIS

A. General results

As expected, the average data extraction rate (Figures 4, 5) first increases, before reaching a maximum and then decreasing with the number of nodes in the network. With few radios, less frames are sent, so the channel is not occupied fully. With too many active radios the channel becomes saturated with a lot of collisions. Even if the SF5 data rate from the one radio solution (orange cross curve in Figure 4) is under-performing compared to the other, our solutions enable the joint use of SF5 and SF12 in the same network without global synchronization and scheduling.

Moreover we can see that, for every solutions, the achieved SF5 data extraction rate is way bigger than the achieved SF12 data extraction rate (Figure 6). This fulfills the objective of having an high SF5 data rate compared to the SF12 one, all

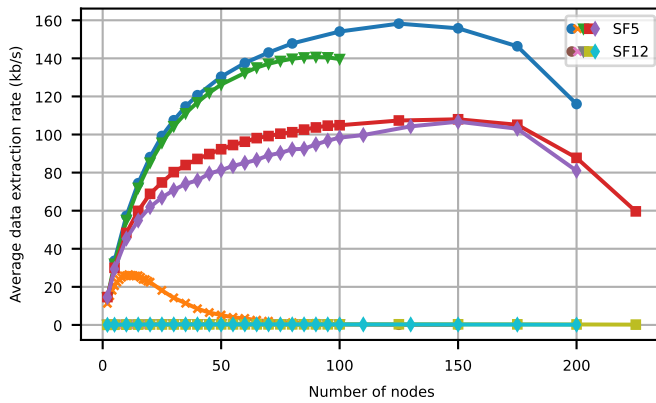


Fig. 6. Comparison of the SF5 and SF12 average data extraction rates between the different solutions: the SF12 data extraction rate is much lower than the achieved SF5 data extraction rate, as expected.

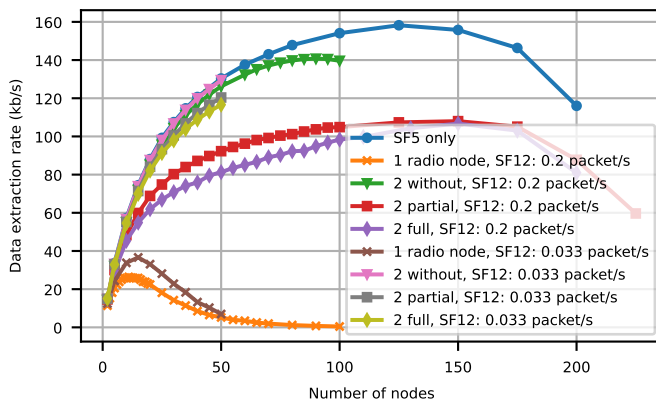


Fig. 7. Comparison of the SF5 average data extraction rates between a situation with high SF12 rate vs. with a low SF12 sending rate. As expected with less SF12 traffic, SF5 performance improves markedly.

of our solutions retain the data rate advantage of the SF5 modulation.

Furthermore, none of the solutions reach the same maximum of performances as the baseline solutions (blue dots in Figures 4, 5). This confirms the intuition, that, for all the presented approaches, the solution involves some performances losses in order to jointly use SF5 and SF12. However, we can clearly see in Figure 7 that decreasing the SF12 sending rate allows an increase of the SF5 performances. Apart from illustrating the coupling between SF5 and SF12, it demonstrates that our solutions are more suitable for applications with sporadic SF12 traffic, e.g. to carry important information over a longer distance. With rare bursts of SF12 traffic, nodes are still able to communicate with nodes further away while sending a lot of data to their close neighbors.

B. Limitations of the tested topology

Nevertheless, by looking at the SF12 data extraction rate of the two radios with partial coordination solution (red squares in Figure 5) and the two radios with full coordination solution

(purple diamonds in Figure 5), it is clear that these solutions outperform the baseline for 100 nodes and more. We guess that this is due to the presence of SF5: some nodes are busy with the other spreading factor either in reception or transmission. During this time the nodes are not sending SF12 frames, nor receiving them. This explains the performances losses compared to the baseline, but not so much why maximum data extraction is reached after all the other solutions already see a severe performance decrease. We believe that this is due to the specifics of the topology we are using. For these results the simulation assumes that all nodes are within communication range of each other at SF5 and SF12. As described in Section I, the goal of using SF5 and SF12 is to allow a group of nodes to communicate internally using SF5 (to get a high data rate), and then communicate with other groups of nodes that are part of the network using SF12 (leveraging the range and robustness of this modulation [3], [4]). Here the simulator only reproduces the behavior of a single group of nodes. As we have coordination between the radios of a node (a radio will not send a packet if the other radio is locked in reception), we have a strong coupling effect between SF5 and SF12. Under high load, this coupling leads to some sort of “synchronization” between the nodes: if some of them are locked on a SF5 frame they will not send a SF12 frame during that time. In other words, the group of nodes is virtually less loaded. This is equivalent to having some nodes turned off at SF12 for a fraction of the time.

Adding other groups of nodes to the simulation would allow them to send SF12 frames during remote SF5 activity. This traffic would disrupt the SF12 traffic and create collisions, resulting in a performance hit earlier in the simulation.

The synchronization also affects the load actually presented to the channel at SF5. The objective of the simulations in this paper is to validate the general principle of using different spreading factors while retaining their respective benefits. Interestingly, SF12 performance is less impacted relatively to the baseline than SF5, which is a good thing since the data rate at SF12 is somewhat limited from the start.

Along those lines, we remark that the two radios with full coordination solution tends to have better performances for SF12 under low load whereas it has better performances for SF5 under high loads (Figures 4, 5). This fulfills the objective of having reliable SF12 communications: as the SF12 preamble are long, locking on them is long, it blocks SF5 emissions from the other radio of the node and that helps to not interfere with the locking process.

Nonetheless, we notice performance differences between the two radios with full coordination and the two radios with partial coordination solutions for SF12 data extraction rates (Figure 5). It seems that having a stronger coordination policy between the radios of a node does not increase nodes performances. Under low loads, even if we see a SF5 decrease, SF12 performance does not increase compared to the two radios node solution with partial coordination. We explain this by SF12 frames having a higher priority so less SF5 frames are conveying on the channel, leading to a faster SF12 decrease

due to collisions between SF12 frames: the “synchronization” effect from the former paragraph is less present.

To summarize, even if the two radios with full coordination solution was designed as an idealized upper bound method, it does not outperform the other solutions in our simulation setup. So, further tests with different topology will help to assess if the two radios with full coordination solution actually makes sense.

C. Single radio nodes performances: trade-off and discussions

Another singular result is the SF5 performance of the one radio solution. Even with convenient SF12 performances (similar to the other solutions), until a certain number of radios, SF5 performances are still valuable. The one radio solution maximum SF5 average data extraction rate represents approximately 19% of the maximum achieved by the baseline (Fig. 4). So, for a certain number of nodes, with only one radio, it is possible to exchange transparently SF12 messages while also exchanging SF5 messages with an higher average data extraction rate (Fig. 7).

However, SF5 performances collapse quickly with the number of nodes in the network. We believe that this is partially due to the SF5 repetitions, but mainly to the non interrupting CAD mechanism at SF12, described in Subsection III-B. As a full CAD is launched every time we start it, nearby SF5 sampling periods are reduced. Broadly, the combined load of SF5 and SF12 reduces the sampling period allocated to SF5. This is why we are quickly losing a lot of the SF5 traffic. A tradeoff between SF5 and SF12 emerges in this solution. Due to our design choices, SF12 seems prioritized over SF5. This is serving our goal to have robust communications for long range important messages, but at the cost of decreasing SF5 performances. By removing the aforementioned non interrupting CAD mechanism, we may be able to reverse this tendency. Nevertheless, even if by optimizing the protocol we might obtain better SF5 performances, a form of tradeoff will remain. Playing with it would allow to adapt the protocol to different applications. In any case, we are still managing to convey simultaneously SF5 and SF12 traffic for various loads, and to the best of our knowledge, it is the first time this is achieved.

In conclusion, as far as we know, these simulations are the first proof that sharing a single LoRa chip on two SF works and has benefits. This opens a whole new spectrum of applications.

VI. CONCLUSION

In this paper, we explore the benefits and limitations of using concurrently two data rates in LoRa 2.4GHz. It turns out that even with a couple of radios, long and short range communications are not independent of each other, so that careful engineering is required. We have presented a new scheme for using a single radio shared between the two modulations. Interestingly, one of its properties is that long range communication has somewhat more priority than short range, and this may be a desirable feature for some applications. Obviously, with more hardware, the throughput is better, but

even with a single radio, the benefits of using a weaker modulations are very significant.

For future works, we plan to explore different configurations (sampling period, number of repetitions) for the single transceiver solution. Furthermore we plan to study a traffic made of bursts for SF12, which corresponds to the kind of traffic some applications would generate as SF12. Finally, we plan to implement our solutions on real hardware in order to evaluate its feasibility and real world performance.

ACKNOWLEDGMENT

This work was supported by the French National Research Agency (ANR) under the “France 2030” program (ANR-15-IDEX-02), within the LabEx PERSYVAL (ANR-11-LABX-0025-01)

REFERENCES

- [1] D. Zorbas, L. Di Puglia Pugliese, R. Saduakhas, and F. Guerriero, “Gateway configuration in 2.4 GHz LoRa networks,” *Internet of Things*, vol. 31, p. 101567, 2025.
- [2] P. Calvet, “LoRa® 2.4GHz 3 Channels Single SF Reference Design - Performance Report,” 2020.
- [3] G. H. Deréviackine, A. Guitton, O. Iova, B. Ning, and F. Valois, “Opportunities and Challenges of LoRa 2.4 GHz,” *IEEE Communications Magazine*, vol. 61, no. 10, pp. 164–170, Oct. 2023.
- [4] R. Falanji, M. Heusse, and A. Duda, “Range and Capacity of LoRa 2.4 GHz,” in *Mobile and Ubiquitous Systems: Computing, Networking and Services*, S. Longfei and P. Bodhi, Eds., Pittsburgh Pennsylvania, Nov. 2022.
- [5] L. Brinon-Arranz, L. Schenato, and A. Seuret, “Distributed Source Seeking via a Circular Formation of Agents Under Communication Constraints,” *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 104–115, 2015.
- [6] S. Chaumette, R. Laplace, C. Mazel, R. Mirault, A. Dunand, Y. Lecoutre, and J.-N. Perbet, “CARUS, an operational retasking application for a swarm of autonomous UAVs: First return on experience,” in *2011 - MILCOM 2011 Military Communications Conference*. Baltimore, MD, USA: IEEE, Nov. 2011, pp. 2003–2010.
- [7] Q. Cui, P. Liu, J. Wang, and J. Yu, “Brief analysis of drone swarms communication,” in *2017 IEEE International Conference on Unmanned Systems (ICUS)*. Beijing: IEEE, Oct. 2017, pp. 463–466.
- [8] C. Goursaud and J.-M. Gorce, “Dedicated networks for IoT : PHY / MAC state of the art and challenges,” *EAI endorsed transactions on Internet of Things*, 2015.
- [9] L. G. Roberts, “ALOHA Packet System with and without Slots and Capture,” *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975.
- [10] C. Lau and C. Leung, “Capture Models for Mobile Packet Radio Networks,” *IEEE Trans. Commun.*, vol. 40, pp. 917–925, 1992.
- [11] T. Attia, M. Heusse, and A. Duda, “Message in Message for Improved LoRaWAN Capacity,” in *IEEE International Conference on Computer Communications and Networks (ICCCN)*, Athens, Greece, Jul. 2021.
- [12] C. Pham and M. Ehsan, “Dense deployment of lora networks: Expectations and limits of channel activity detection and capture effect for radio channel access,” *Sensors*, vol. 21, no. 3, 2021.
- [13] Semtech, “SX1280/SX1281 datasheet,” Sept 2023.
- [14] J. Knapp, “LR2021 LoRa® Performance,” 2025.
- [15] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Networks,” *IEEE Communications Surveys & Tutorials*, vol. 12, pp. 222–248, 2010.
- [16] A. Bachir, M. Heusse, A. Duda, and K. K. Leung, “Preamble sampling MAC protocols with persistent receivers in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1091–1095, Mar. 2009.