Generalized Slotted MAC Protocol Exploiting LoRa Signal Collisions

Nancy El Rachkidy(1), Megumi Kaneko(2), Alexandre Guitton(1)
(1) Université Clermont Auvergne, CNRS, LIMOS, F-63000 Clermont-Ferrand, France
(2) National Institute of Informatics, Hitotsubashi, 2-1-2, Chiyoda-ku, 101-8430 Tokyo, Japan
Emails: nancy.el_rachkidy@uca.fr, alexandre.guitton@uca.fr, megkaneko@nii.ac.jp

***Abstract*—LoRa is becoming widely used in low-power wide area networks as it enables a communication range of several kilometers with low energy consumption, but with a low bitrate. Collisions in LoRa further reduce the overall performance of the network, and more specifically the throughput. In this paper, we propose a slotted MAC protocol that enables the decoding of colliding LoRa signals. It is based on a new decoding technique at the physical layer that is able to decode the symbols of many frames in collision. Simulation results show that our MAC protocol significantly increases the achievable performance of LoRa networks. For instance, for 25 nodes having a duty-cycle of 10% and with SF7, the throughput with our protocol is 11% larger than the existing protocols.**

***Index Terms*—LoRa, LoRaWAN, LPWAN, Collision resolution.**

I. Introduction

Low-Power Wide Area Networks (LPWANs) [1] use long-range communication technologies (such as LoRa [2]) to cover large zones with a low energy consumption.

The LoRa (for Long-Range) physical layer [2] uses a Chirp Spread Spectrum (CSS) modulation in order to extend communication range at the cost of throughput. CSS enables simultaneous reception on different channels or with different spreading factors (SFs). However, collisions occur when several end-devices transmit frames simultaneously on the same channel and SF, and when these frames are received with a similar power. Thus, retransmissions are needed, which reduces the overall throughput.

Recently, many works have studied the impact of SF on collisions. For instance, several researchers have shown that the SFs of LoRa are not completely orthogonal in practice [3], [4], [5]. Others have shown that LoRa exhibits relatively good capture effect capabilities [6], [7]. Authors in [7], [8] proposed new protocols to decode fully synchronized colliding signals. In [7], the authors have described a flooding algorithm called concurrent transmissions (CT) where frames are broadcast according to a tree structure. They have shown that in practice, synchronized collisions that occur at each level of the tree can generally be captured by LoRa, and thus CT is more efficient than other algorithms which attempt to avoid collisions. The main drawback of this work is that the decoding of these fully synchronized colliding signals depends on the capture effect, and is therefore not controlled by the protocol. In [8], the authors have shown that it is possible to use the tiny time offsets of low-cost LoRa transmitters in order to distinguish the signals of colliding users. They have shown with real-world experiments that collisions that are fully synchronized in theory can be cancelled. However, their contributions are limited to synchronized signals only.

In our previous works [9], [10], we focused on decoding slightly desynchronized LoRa signals. In [9], we proposed a new decoding algorithm operating at the physical layer, which enables two slightly desynchronized LoRa signals to be decoded, independently of the tiny time offsets of the hardware. In [10], we adapted the decoding algorithm to the case of several slightly desynchronized signals. We designed a MAC protocol called CR-MAC that enables the usage of our decoding algorithm. CR-MAC divides time into slots and each slot into subslots. CR-MAC is able to cancel most collisions within a slot if each transmitter uses a different subslot. It exhibits good performance compared to the LoRaWAN (Long Range Wide Area Network) protocol. However, it has the following main weaknesses: (1) when several nodes choose the same subslot, CR-MAC is unable to decode any signal for this whole slot, and (2) partial knowledge acquired in previous failed decoding attempts is not reused to improve the decoding of retransmissions.

Figure 1 shows an example of frames transmitted during several slots. In slot S_1, there is only one frame transmitted. Thus, there is no collision and the frame can be decoded by LoRa. In slot S_2, two frames are transmitted simultaneously (that is, in the same subslot). The algorithm of [8] is likely to decode them. In slot S_3, two frames are slightly desynchronized. Both the algorithm of [9] and CR-MAC can decode them, but the algorithm of [8] does not apply. In slot S_4, three frames are slightly synchronized. Here, CR-MAC can decode them, but the algorithm of [9] fails. Finally, in slot S_5, some frames are synchronized and some frames are slightly desynchronized. No existing algorithm (including CR-MAC) is able to decode them.

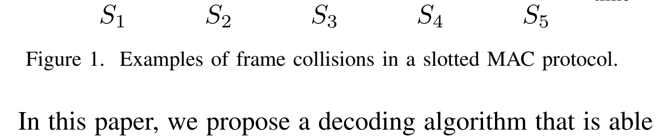


Figure 1. Examples of frame collisions in a slotted MAC protocol.

In this paper, we propose a decoding algorithm that is able to handle all cases of collisions shown on Figure 1, as well

978-1-7281-4490-0/20/$31.00 ©2020 IEEE

as a generalized slotted MAC protocol called GS-MAC. GS-MAC tackles the main weaknesses of CR-MAC: it can deal simultaneously with synchronized signals as well as slightly desynchronized signals, by comparing the possible symbols at each subslot (rather than only at each symbol frontier), which yields better decoding capabilities. GS-MAC also stores uncertainties to help decoding further retransmissions.

The main contributions that distinguish this work from our previous work are the following.

1) We propose an algorithm that decodes most collisions occurring within a slot, even if several transmitters use the same subslot. This algorithm uses the frequency information at each subslot in order to deduce the symbols of each frame. Moreover, the proposed algorithm is able to decode more frames than CR-MAC, even when colliding frames are sent on distinct subslots.
2) GS-MAC incorporates the management of retransmissions. Indeed, GS-MAC stores symbol uncertainties from previous collisions in order to help the decoding of retransmitted frames.
3) The simulation results show the significant performance improvements achieved by our new decoding algorithm and GS-MAC compared to CR-MAC, jointly in terms of network throughput, energy efficiency and delay.

II. System Model and Baseline Methods

The system model consists of a network of several end-devices, acting as transmitters, and a single gateway, acting as a receiver. The gateway can decode several signals received simultaneously, as long as they are sent on different channels or SFs.

LoRaWAN [11] is a MAC layer designed for LoRa. It allows end-devices to communicate to a network server through gateways. LoRaWAN uses an ALOHA mechanism: when an end-device has a frame to transmit, it transmits it without channel sensing. After the transmission, it waits for a short period in order to receive an acknowledgment. Then, it switches its radio off for a long duration (generally 99 times the time on air), in order to respect the duty-cycle on ISM bands of 1%. LoRaWAN defines very low throughputs: from 250 bps (for SF12) to 11000 bps (for SF7).

Slotted LoRaWAN is a protocol based on LoRaWAN, in which transmissions can only occur at the beginning of a slot.

CR-MAC [10] is based on an algorithm identifying frequency changes at the frontier of each symbol: when the set of received frequencies changes before and after the frontier of a symbol, the changed frequency is for the new symbol.

Figure 2 shows the time division of CR-MAC. Each gateway sends periodic beacons on each SF. Upon receiving a beacon, each end-device starts S consecutive slots ($S = 3$ slots on the figure), whose duration is equal to the maximum frame transmission plus one symbol. To transmit a frame, an end-device has to wait for the beginning of a slot. It then draws a random number r between 0 and the number of subslots M ($M = 4$ subslots on the figure), and transmits the frame after r/M-th of the symbol duration.

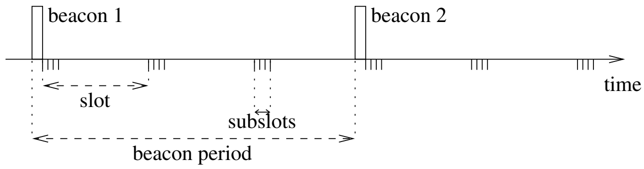


Figure 2. Time division of the CR-MAC protocol.

III. Proposed GS-MAC protocol

Our proposed protocol consists of a decoding algorithm applied to a slotted MAC protocol. The decoding algorithm is in charge of deducing possible values for the symbols of overlapping frames, from the set of superimposed frequencies. The MAC protocol is in charge of ensuring that resolvable collisions occur, and deals with retransmissions when the frames are not fully decoded.

A. Decoding algorithm

We assume here that several nodes have transmitted their frame during the same slot, each starting at a random subslot in $\{0, 1, \ldots, M-1\}$. The algorithm listens to the superimposed frequencies at each subslot and for each symbol in order to determine the possible symbols for all frames. As the frequencies are time-dependent, the frequencies have to be shifted depending on the current subslot. The shift is always negative, as we have to go back to the time where the symbol was started. The actual value of the shift depends on the symbol start time and on the current time, which is dependent on the current subslot m.

The algorithm revolves around Property 1 (concrete examples will follow for ease of understanding).

Property 1. *Let us denote by $F(s,m)$ the set of frequencies received at subslot m of the s-th symbol, and by M the total number of subslots. All frequencies are computed modulo 2^{SF}.*

- *The $(s-1)$-th symbol of a frame sent in a subslot $m' > m$ belongs to $\{f - \delta | f \in F(s,m)\}$, where $\delta = (M + m - m') \times 2^{SF}/M$.*
- *The s-th symbol of a frame sent in a subslot $m' \leq m$ belongs to $\{f - \delta | f \in F(s,m)\}$, where $\delta = (m - m') \times 2^{SF}/M$.*

Proof. The shift δ in the frequencies comes from the fact that in LoRa chirps, the frequency of symbols increases as time passes. If a frame is sent in a previous subslot $m' \leq m$, δ is proportional to $m - m'$. Otherwise, it is proportional to $M + m - m'$. □

For each symbol, one set of possible values is obtained at each subslot. The transmitted symbol is in the intersection of all these sets.

Algorithm 1 uses this property in order to decode colliding frames. The main loop considers all symbol positions and all subslots sequentially. For each pair (s,m), the set of frequencies $F(s,m)$ is obtained, and the values of the possible symbols are computed for all subslots m'. These values are intersected with the previous values obtained. At the end of the algorithm, the variable $symb[s, m']$ contains all possible values (and ideally, a single value) for the s-th symbol of frames starting at subslot $m' \in [0; M-1]$.

Algorithm 1: Decoding colliding LoRa signals.

Require: P the set of subslots where the start of a frame has been detected from the overlapping preambles
$s \leftarrow 0$
$m \leftarrow$ the last subslot of P
$F(s, m) \leftarrow$ detect frequencies
while $F(s, m) \neq \emptyset$ **do**
  if $m \in P$ **then**
    for $m' \in [0; M-1]$ **do**
      if $m' > m$ **then**
        $\delta = (M + m - m') \times 2^{SF}/M$
        $F' \leftarrow \{f - \delta | f \in F(s, m)\}$
        $symb[s-1, m'] \leftarrow symb[s-1, m'] \cap F'$
      else
        $\delta = (m - m') \times 2^{SF}/M$
        $F' \leftarrow \{f - \delta | f \in F(s, m)\}$
        $symb[s, m'] \leftarrow symb[s, m'] \cap F'$
      end if
    end for
  end if
  wait until next subslot
  if $m < M$ **then**
    $m \leftarrow m + 1$
  else
    $m \leftarrow 0$
    $s \leftarrow s + 1$
  end if
  $F(s, m) \leftarrow$ detect frequencies
end while

This algorithm introduces a computational overhead when colliding signals occur, but since the computation is performed by the gateway, we believe that it is acceptable. The time complexity of the algorithm is only $\mathcal{O}(Mn)$, where n is the length of the largest frame in symbols.

1) Case of distinct subslots: Let us consider a first example for three frames sent on three different subslots, with $SF = 7$. Frame 1 consists of symbols $(10, 11, 12, 13)$ and is sent on subslot 0, frame 2 is $(20, 21, 22, 23)$ and is sent on subslot 1, and frame 3 is $(30, 31, 32, 33)$ and is sent on subslot 3, as shown on figure 3. In the following, we highlight the correct symbol in the set of possible symbols to help the reader.

Table I shows the steps of our algorithm, including the frequencies detected by the gateway for each symbol number and at each subslot. Until subslot 3 of symbol 0, no frequency is decoded due to the overlapping of the upchirps of frame 1 with the downchirps of the preamble of frame 3.

Subslot 3 of symbol 0: The detected frequencies are $\{30, 84, 106\}$ (see Fig. 3). This means that symbol 0 of frames

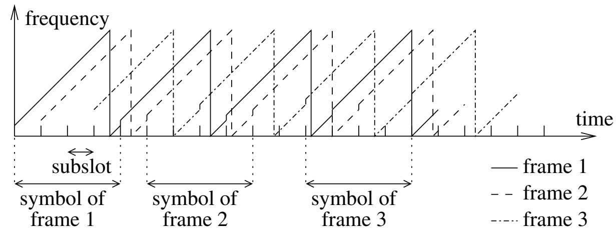


Figure 3. The frequency changes for each received frame at the receiver.

starting at subslot 0 (such as frame 1) is in $\{\mathbf{10}, 62, 116\}$. Indeed, $30 - 96 = -66 = 62$ modulo 128, $84 - 96 = -12 = 116$ modulo 128, and $106 - 96 = 10$ modulo 128. Symbol 0 of frames starting at subslot 1 (such as frame 2) is in $\{\mathbf{20}, 42, 94\}$ (the shift is now 64 because the current subslot is 2). There is no frame starting at subslot 2, as no starting upchirp was detected during the preamble. Finally, the symbol 0 of frames starting at subslot 3 (such as frame 3) is in $\{\mathbf{30}, 84, 106\}$ (the shift is now 0).

Subslot 0 of symbol 1: The detected frequencies are $\{11, 62, 116\}$ (see Fig. 3). This means that symbol 1 of frames starting at subslot 0 is in $\{\mathbf{11}, 62, 116\}$ (the shift is 0). Symbol 0 of frames starting at subslot 1 is in $\{\mathbf{20}, 43, 94\}$ (the shift is 96). Since it was previously identified that this symbol was in $\{\mathbf{20}, 42, 94\}$, only the set $\{\mathbf{20}, 94\}$ remains. Symbol 0 of frames starting at subslot 3 is in $\{\mathbf{30}, 84, 107\}$ (the shift is 32). Since it was previously identified that this symbol was in $\{\mathbf{30}, 84, 106\}$, only the set $\{\mathbf{30}, 84\}$ remains.

Subslot 1 of symbol 1: The detected frequencies are $\{21, 43, 94\}$ (see Fig. 3). This means that symbol 1 of frames starting at subslot 0 is in $\{\mathbf{11}, 62, 117\}$. Only set $\{\mathbf{11}, 62\}$ remains. Symbol 1 of frames starting at subslot 1 is in $\{\mathbf{21}, 43, 94\}$. Symbol 0 of frames starting at subslot 3 is in $\{\mathbf{30}, 85, 107\}$. Only value $\mathbf{30}$ remains.

Subslot 2 of symbol 1: No new information is obtained, since no frame was transmitted at subslot 2 (this is known from the preamble).

Subslot 3 of symbol 1: The detected frequencies are $\{31, 85, 107\}$. This means that symbol 1 of frames starting at subslot 0 is in $\{\mathbf{11}, 63, 117\}$. Only value $\mathbf{11}$ remains. Symbol 1 of frames starting at subslot 1 is in $\{\mathbf{21}, 43, 95\}$. Only set $\{\mathbf{21}, 43\}$ remains. Symbol 1 of frames starting at subslot 3 is in $\{\mathbf{31}, 85, 107\}$.

Until subslot 3 of symbol 4: The algorithm continues until the time where no frequency is detected, which means that all transmissions have stopped.

2) Case of colliding subslots: This algorithm can be applied even if several nodes choose the same subslot to transmit their frame. Some colliding frames will not be decoded, but the algorithm will be able to compute the possible superimposed symbols. This will be reused later by the MAC protocol to help decoding retransmissions (see Subsection III-B).

Let us consider a second example for three frames sent on two subslots, with $SF = 7$. Frame 1 is $(10, 11, 12, 13)$ and is sent on subslot 0, frame 2 is $(20, 21, 22, 23)$ and is sent on

Table I
STEP-BY-STEP EXECUTION OF ALGORITHM 1 - CASE OF DISTINCT SUBSLOTS.

symbol position s	subslot m	frequencies $F(s,m)$	frame 1	frame 2	frame 3
0	3	$\{30,84,106\}$	$s_0^1 \in \{\mathbf{10},62,116\}$	$s_0^2 \in \{\mathbf{20},42,94\}$	$s_0^3 \in \{\mathbf{30},84,106\}$
1	0	$\{11,62,116\}$	$s_1^1 \in \{\mathbf{11},62,116\}$	$s_0^2 \in \{\mathbf{20},94\}$	$s_0^3 \in \{\mathbf{30},84\}$
1	1	$\{21,43,94\}$	$s_1^1 \in \{\mathbf{11},62\}$	$s_1^2 \in \{\mathbf{21},43,94\}$	$s_0^3 = \mathbf{30}$
1	3	$\{31,85,107\}$	$s_1^1 = \mathbf{11}$	$s_1^2 \in \{\mathbf{21},43\}$	$s_1^3 \in \{\mathbf{31},85,107\}$
2	0	$\{12,63,117\}$	$s_2^1 \in \{\mathbf{12},63,117\}$	$s_1^2 = \mathbf{21}$	$s_1^3 \in \{\mathbf{31},85\}$
2	1	$\{22,44,95\}$	$s_2^1 \in \{\mathbf{12},63\}$	$s_2^2 \in \{\mathbf{22},44,95\}$	$s_1^3 = \mathbf{31}$
2	3	$\{32,86,108\}$	$s_2^1 = \mathbf{12}$	$s_2^2 \in \{\mathbf{22},44\}$	$s_2^3 \in \{\mathbf{32},86,108\}$
3	0	$\{13,64,118\}$	$s_3^1 \in \{\mathbf{13},64,118\}$	$s_2^2 = \mathbf{22}$	$s_2^3 \in \{\mathbf{32},86\}$
3	1	$\{23,45,96\}$	$s_3^1 \in \{\mathbf{13},64\}$	$s_3^2 \in \{\mathbf{23},45,96\}$	$s_2^3 = \mathbf{32}$
3	3	$\{33,87,109\}$	$s_3^1 = \mathbf{13}$	$s_3^2 \in \{\mathbf{23},45\}$	$s_3^3 \in \{\mathbf{33},87,109\}$
4	0	$\{65,119\}$	$s_4^1 \in \{65,119\}$	$s_3^2 = \mathbf{23}$	$s_3^3 \in \{\mathbf{33},87\}$
4	1	$\{97\}$	$s_4^1 = 65$	$s_4^2 = 97$	$s_3^3 = \mathbf{33}$
4	3	$\{\}$	no symbol	no symbol	no symbol

subslot 0 too, and frame 3 is $(30, 31, 32, 33)$ and is sent on subslot 3. Table II shows the steps of our algorithm. Note that at the end, frame 3 is completely decoded, and the algorithm has obtained all superimposed symbols for frames 1 and 2.

3) Decoding the first symbols: The algorithm is unable to decode the first symbol of the frames sent on any subslot except the last. Indeed, during the first subslots of the first symbol, the data upchirps of these frames overlap with preamble downchirps of the last transmitted frame. Thus, no frequency can be extracted, and the algorithm misses important information for these frames. To avoid this issue, we propose to introduce an arbitrary symbol in frames, which can be ignored upon decoding. This symbol only has to be different from the first data symbol, in order to provide meaningful information through the first frequency change.

In addition, MAC protocols built on top of our algorithm might need to identify the nodes that are transmitting frames, to acknowledge their transmissions. This is typically required by our MAC protocol described further (see Subsection III-B). To do this, we make the following assumption: the node IDs are in the range $[0; 2^{SF}-1]$ (for SF7, this means that there are at most 128 node IDs, and for SF12, this means that there are at most 4096 node IDs). This assumption is discussed later, in Section V. Then, we propose the following modification to frames: the node identification n is shifted by δ_m and is appended to each frame, and it is followed by another different symbol, with $\delta_m = -m \times 2^{SF}/M$, m being the subslot used by the transmission. The shift of the first symbol allows the gateway to know the ID of all nodes transmitting, which is exactly the set $F(1, 0)$ (thanks to δ_m). It also serves as the first arbitrary symbol which can be ignored on the decoding. The second symbol is always different from the first, and can be used to deduce the node identification too.

For instance, let us consider SF7, 4 subslots, node 10 transmitting at subslot 0, and node 20 transmitting at subslot 1. For node 10, the node number is shifted by $\delta_0 = 0$ and appended to the frame. Thus, the frame starts with 10. For node 20, the node number is shifted by δ_1. The resulting value is $20 - 128/4 = -12 = 116$. We can observe that $F(1, 0) = \{10, 20\}$, which is the set of all node IDs.

B. MAC protocol

The GS-MAC protocol divides time into slots. All transmissions of data frames start at the beginning of a slot, with a random backoff of less than one symbol corresponding to the subslot. After the transmission, there is a short receive window for acknowledgment, whose length is equal to the maximum frame duration, which can also be used to listen to synchronization beacons. Thus, the slot duration is set to twice the maximum time on air of a frame plus one symbol.

When a frame contains no uncertainties, it is completely decoded. Then, it can be acknowledged by the gateway at the end of the same slot. If a frame contains uncertainties, it has to be retransmitted. In this case, the gateway stores the current uncertainties for the corresponding node ID. This mechanism reduces the number of retransmissions needed to decode collided frames, as the receiver has a previous partial knowledge of retransmitted frames.

Let us consider again the example of Table II. At the end of the decoding, frame 3 is completely decoded, and thus can be acknowledged. Frames 1 and 2 both have uncertainties: the gateway does not know which were the actual frames, although it knows that the symbols are within $(\{10, 20, 62\}, \{11, 21\}, \{12, 22\}, \{13, 23\})$. The first symbol uncertainty can be reduced to $\{10, 20\}$ (see Subsection III-A3, thanks to the relationship between the first two symbols). Moreover, the gateway knows the ID of the two nodes that were transmitting from the first symbol: they are nodes 10 and 20 in the example. Thus, the gateway stores $(10, \{\mathbf{11}, 21\}, \{\mathbf{12}, 22\}, \{\mathbf{13}, 23\})$ for node 10, and $(20, \{11, \mathbf{21}\}, \{12, \mathbf{22}\}, \{13, \mathbf{23}\})$ for node 20. This information is kept until the node retransmits the frame, and is used to initialize the possible symbols for the retransmitted frame.

It is interesting to notice that each collision in the same subslot produces uncertainties in frames. After a retransmission, even if the retransmitted frame is again in collision, it is likely that the uncertainties will cancel out, and that the frame might be decoded.

IV. SIMULATION RESULTS

In this section, we highlight the performance of our proposed GS-MAC protocol, denoted Prop. GS-MAC in the fig-

Table II
STEP-BY-STEP EXECUTION OF ALGORITHM 1 - CASE OF NON-DISTINCT SUBSLOTS.

symbol position s	subslot m	frequencies $F(s,m)$	frame 1	frame 2	frame 3
0	3	$\{30,106,116\}$	$s_0^1 \in \{\mathbf{10},20,62\}$	$s_0^2 \in \{10,\mathbf{20},62\}$	$s_0^3 \in \{\mathbf{30},106,116\}$
1	0	$\{11,21,62\}$	$s_1^1 \in \{\mathbf{11},21,62\}$	$s_1^2 \in \{11,\mathbf{21},62\}$	$s_0^3 = \mathbf{30}$
1	3	$\{31,107,117\}$	$s_1^1 \in \{\mathbf{11},21\}$	$s_1^2 \in \{11,\mathbf{21}\}$	$s_1^3 \in \{\mathbf{31},107,117\}$
2	0	$\{12,22,63\}$	$s_2^1 \in \{\mathbf{12},22,63\}$	$s_2^2 \in \{12,\mathbf{22},63\}$	$s_1^3 = \mathbf{31}$
2	3	$\{32,108,118\}$	$s_2^1 \in \{\mathbf{12},22\}$	$s_2^2 \in \{12,\mathbf{22}\}$	$s_2^3 \in \{\mathbf{32},108,118\}$
3	0	$\{13,23,64\}$	$s_3^1 \in \{\mathbf{13},23,64\}$	$s_3^2 \in \{13,\mathbf{23},64\}$	$s_2^3 = \mathbf{32}$
3	3	$\{33,109,119\}$	$s_3^1 \in \{\mathbf{13},23\}$	$s_3^2 \in \{13,\mathbf{23}\}$	$s_3^3 \in \{\mathbf{33},109,119\}$
4	0	$\{65\}$	$s_4^1 = 65$	$s_4^2 = 65$	$s_3^3 = \mathbf{33}$
4	3	$\{\}$	no symbol	no symbol	no symbol

ures, compared to slotted LoRaWAN and CR-MAC protocols.

A. Parameter settings

Simulations were carried out using our own simulator developed in Java. We did not use an existing simulator as collision resolution requires significant changes in the decoding of frames. Our results are averaged over 200 repetitions (due to the variability of the results). We consider a network with a single gateway acting as a receiver, and we vary the number of end-devices. We set the size of the sent frames to 10 symbols: the identifier of each transmitting end-device is coded on 2 symbols[1] and the payload is 8 symbols (that is, from 7 bytes to 12 bytes, depending on SF). We also consider that all end-devices are operating with a duty cycle of 10% (this allows us to simulate fewer nodes than with a duty-cycle of 1%, while generating a large traffic). We consider that transmissions are on the same channel, with the same SF, that no capture effect occurs as in [9], [10][2], and we consider a high signal-to-noise ratio regime. We set the bandwidth to 125 kHz in order to have a fair comparison of the delay. We set the number of slots in a beacon period to 1000 slots, and for CR-MAC and GS-MAC, we consider 4 subslots per slot (which is a pessimistic value, in order to deal with noisy channels). Slotted LoRaWAN uses no backoff. The transmission power is set to 13 dBm (power consumption of 28 mW [12]). Finally, in case of collisions, we do not limit the number of retransmissions for any of the three evaluated MAC protocols.

B. Throughput

Figure 4 shows the average total throughput in terms of the number of end-devices in the network for slotted LoRaWAN, CR-MAC, and our GS-MAC protocol, for SF7 (on the left) and SF12 (on the right). Using slotted LoRaWAN, the throughput decreases while increasing the number of end-devices as more collisions arise. Using CR-MAC, the throughput is larger than the one computed with slotted LoRaWAN as CR-MAC is able to decode frames that are sent in the same slot but with different subslots. GS-MAC outperforms both slotted LoRaWAN and CR-MAC as it can decode frames even if they are sent on the same subslot of the same slot, as well as decoding more frames than CR-MAC when they are all sent on distinct subslots. Moreover, when frames collide, retransmissions for each frame are scheduled following a simple retransmission mechanism. Although these retransmitted frames might collide with other frames, GS-MAC can use the stored information about previous collisions in order to help the decoding algorithm for the current collision. Finally, we can notice on the figure that SF7 performs better than SF12 in terms of throughput. Indeed, with SF7, the frame duration is smaller than the frame duration with SF12. Thus, the time-off duration is smaller with SF7 than with SF12, and end-devices are able to send more frames with SF7 than with SF12. For 25 nodes and SF7, the throughput of GS-MAC is 2.3 times larger than with slotted LoRaWAN, and is 11% larger than with CR-MAC. For SF12, the throughput of GS-MAC is 9 times larger than with slotted LoRaWAN, and twice larger than with CR-MAC.

C. Energy Efficiency

Figure 5 shows the energy efficiency, defined as the total number of successfully decoded bits divided by the total consumed energy based on the total number of transmissions (including retransmissions) for the three protocols. For all evaluated protocols, the energy efficiency decreases with the number of end-devices in the network as collisions become more frequent in denser networks. Moreover, GS-MAC shows better performance (the gain reaches up to 53% compared to slotted LoRaWAN and up to 30% compared to CR-MAC) for dense networks. This is due to the fact that slotted LoRaWAN does not handle collisions and thus the retransmissions are frequent in dense networks. CR-MAC is more robust than slotted LoRaWAN in terms of collisions, but it cannot handle collisions occuring in the same subslot of the same slot, unlike GS-MAC. Finally, we notice that with SF7, the energy efficiency is higher for all the protocols than with SF12, due to shorter transmission durations.

D. Delay

Figure 6 shows the average system delay in terms of the number of end-devices in the network, for the three protocols. The delay increases with the number of end-devices. Indeed,

[1]LoRaWAN specifies that the ID of nodes is stored on 3 to 5 symbols (for SF12 and SF7 respectively). We used a shorter length as we introduce additional constraints on the ID (see Subsection III-A3 and Section V).

[2]Note that, combined with capture effects, the proposed method can further improve system performance.

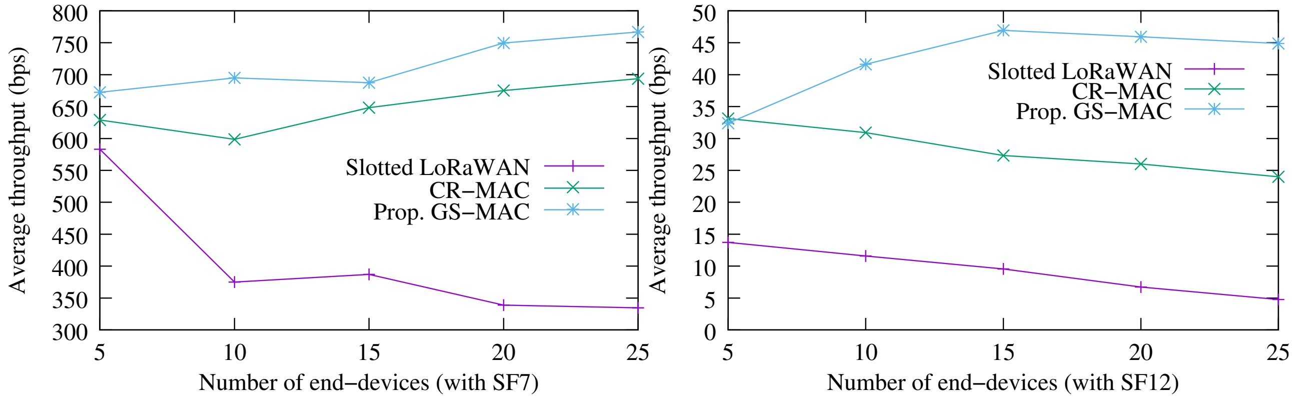


Figure 4. GS-MAC considerably improves the system throughput compared to CR-MAC and slotted LoRaWAN.

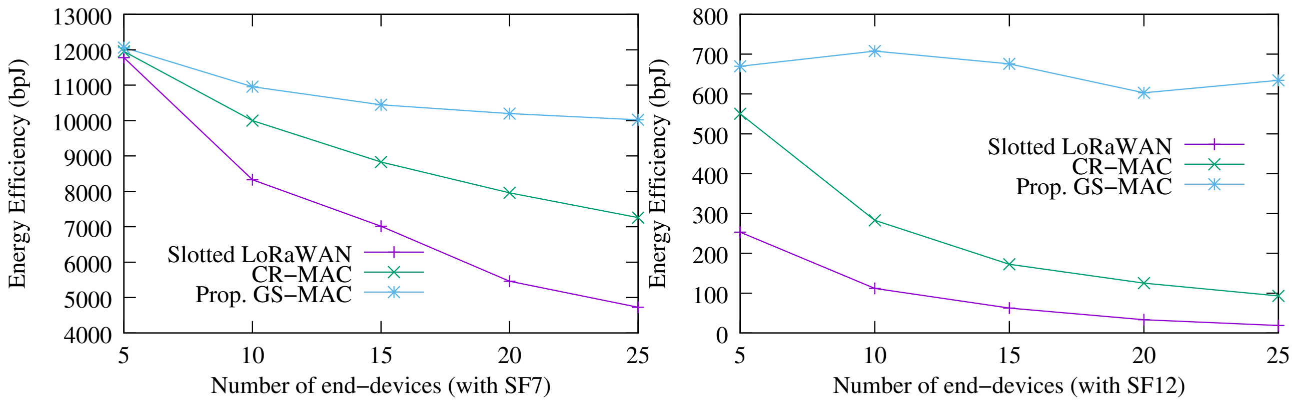


Figure 5. The energy efficiency is reduced by increasing the number of end-devices in the network.

as slotted LoRaWAN does not handle collisions, retransmissions become frequent in dense networks, which leads to large delay. CR-MAC outperforms slotted LoRaWAN as it can decode frames in the same slot. However, in a large network, the probability that end-devices choose different subslots within the same slot decreases. GS-MAC protocol outperforms slotted LoRaWAN and CR-MAC because it is able to handle these collisions. Moreover, GS-MAC can decode a frame even if it is always in collision, as it reduces the uncertainties on each symbol at each collision. That is why we observe a significant reduction of the delay of our MAC protocol compared to slotted LoRaWAN and CR-MAC (respectively up to 5 and 3 times for SF7, and up to 63 and 16 times for SF12). We also notice that the largest delay is achieved with SF12 as the transmission time of the frames and their time-off are much larger than with SF7.

V. Discussion

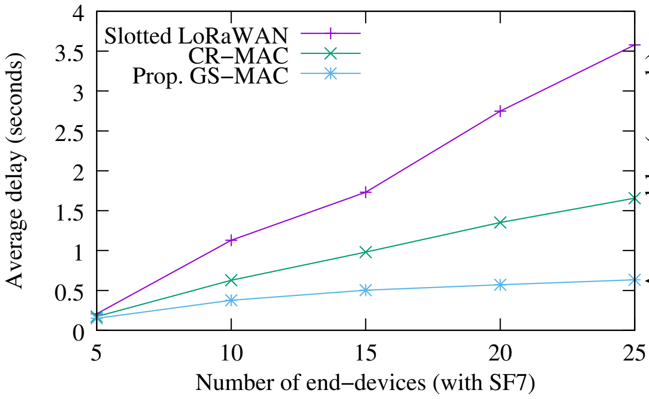
We briefly discuss here three aspects of our protocol.

A. *Differences with CR-MAC [10]*

The decoding algorithm of GS-MAC has three major differences with the decoding algorithm of CR-MAC. First, it uses the frequencies at each subslot, rather than only at the frontier of each symbol. This increases the decoding capabilities compared to CR-MAC. Indeed, at each line of Table I or Table II, GS-MAC performs three computations overall for three different symbols, compared to CR-MAC which performs only one computation for one symbol. Second, GS-MAC can be applied even if several nodes choose the same subslot. This greatly increases the conditions of usage compared to CR-MAC. Table III shows the average number of decodable frames for both GS-MAC and CR-MAC, as a function of the number of nodes (averaged over one million repetitions), for eight subslots. For CR-MAC, this is four times the average number of slots where all four frames are on different subslots. For GS-MAC, this is the average number of subslots where there is a single frame. For instance, if there are 8 subslots and 4 nodes, CR-MAC can decode on average about 1.64 frames and GS-MAC 2.68 frames. Third, although GS-MAC is unable to decode frames sent during the same subslot, it still computes the possible superimposed symbols, which greatly helps decoding the retransmitted frames.

Table III
Average number of decodable frames for 8 subslots

Number of nodes	1	2	4	8	12	16
Decod. frames for CR-MAC	1	1.75	1.64	0.02	0	0
Decod. frames for GS-MAC	1	1.75	2.68	3.14	2.76	2.16



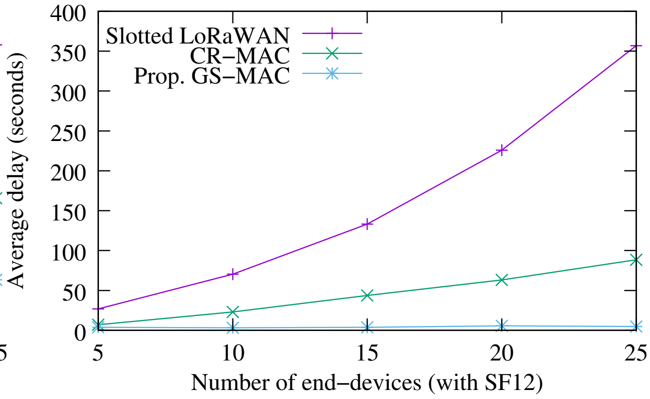


Figure 6. The delay for GS-MAC outperforms the delay obtained by slotted LoRaWAN and CR-MAC.

B. Incorrect guessing of symbols

Some specific cases of collisions cannot be decoded using our assumptions, even if the transmissions are sent on distinct subslots [9]. This means that in some cases, Algorithm 1 and the algorithm of [10] might fail. When this happens, the algorithms either obtain no possible value for a given symbol of a frame, or might acknowledge an incorrect frame. The first case can be corrected by removing all prior information for this node and retransmitting the frame. The second case can be detected by an incorrect CRC. Both cases are rare, in practice. Note that in our simulations, we did not detect any error frames for both SF7 and SF12, for any repetition.

C. Dealing with large node IDs

We assumed that nodes are numbered from 0 to $2^{SF}-1$, in order for the node ID to be stored on one symbol. To weaken this constraint and be able to deal with large node IDs, we propose that large IDs[3] are randomly allocated to nodes, and that the gateway knows the IDs of all nodes within its range. When the gateway receives frames in collision, it can then determine the valid sets of nodes transmitting simultaneously. The probability of incorrectly attributing a frame to a node becomes very small: for instance, for SF7, if the number of nodes in the network is 10000, the probability that a pair of colliding nodes can be mistaken with another pair of colliding nodes is smaller than 0.04%.

VI. Conclusion

Collisions in LoRa negatively impact the network performance in terms of throughput, energy consumption and delay. In this paper, we proposed an efficient decoding algorithm for colliding LoRa signals with broader conditions of operation than existing algorithms, as it encompasses both synchronized and slightly desynchronized collisions. Then, we proposed a new MAC protocol based on this decoding algorithm, with a specific support for retransmissions. This GS-MAC protocol is even capable of decoding signals that are always in collisions with other signals, as the uncertainties on the symbols reduce at each retransmission. Simulation results showed that the GS-MAC protocol outperforms the existing protocols in terms of the overall network performance. For instance, in our simulations, the throughput gain of GS-MAC is 11% for SF7 and 100% for SF12, compared to the existing protocols. The reduction of delay is also very high: it ranges from 62% (for SF7) to 95% (for SF12).

[3]The value of the ID in our proposition is between 0 and $2^{SF}-1$. However, in LoRaWAN, the value of ID is between 0 and $2^{32}-1$.

References

[1] U. Raza and P. Kulkarni and R. Xu and M. Sooriyabandara, “Low Power Wide Area Networks: an Overview,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 855–873, January 2017.

[2] Semtech Corporation, “AN1200.22 LoRa Modulation Basics,” Semtech, Application note Revision 2, 2015, accessed 2018-01-29. [Online]. Available: http://www.semtech.com/uploads/documents/an1200.22.pdf

[3] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, “LoRa throughput analysis with imperfect spreading factor orthogonality,” *IEEE Wireless Communications Letters*, 2018.

[4] D. Croce, M. Gucciardo, I. Tinnirello, D. Garlisi, and S. Mangione, “Impact of spreading factor imperfect orthogonality in LoRa communications,” in *International Tyrrhenian Workshop on Digital Communication (TIWDC)*, ser. Communications in Computer and Information Science (CCIS), vol. 766, 2017, pp. 165–179.

[5] G. Zhu, C.-H. Liao, M. Suzuki, Y. Narusue, and H. Morikawa, “Evaluation of LoRa receiver performance under co-technology interference,” in *IEEE Consumer Comm. and Networking Conf. (CCNC)*, 2018.

[6] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, “Impact of LoRa imperfect orthogonality: Analysis of link-level performance,” 2018.

[7] C. Liao, G. Zhu, D. Kawabara, and M. Suzuki, “Multi-hop LoRa Network enabled by concurrent transmission,” *IEEE Access*, 2017.

[8] R. Eletreby, D. Zhang, S. Kumar, and O. Yagan, “Empowering low-power wide area networks in urban settings,” in *ACM Sigcomm*, 2017.

[9] N. El Rachkidy, A. Guitton, and M. Kaneko, “Decoding superposed LoRa signals,” in *IEEE LCN*, 2018, pp. 184–190.

[10] N. El Rachkidy, A. Guitton, and M. Kaneko, “Collision Resolution Protocol for Delay and Energy Efficient LoRa networks,” *IEEE Trans. Green Comm. and Networking*, vol. 3, no. 2, pp. 535–551, 2019.

[11] LoRa Alliance Technical Committee, “LoRaWAN 1.1 Specification,” LoRa Alliance, Standard V1.1, 2017.

[12] Semtech Corporation, “SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver,” Semtech Corporation, Tech. Rep., March 2017, rev. 3.1.