Received 11 August 2023; revised 19 September 2023; accepted 28 September 2023. Date of publication 12 October 2023;
date of current version 8 November 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3321875

A New Schedule-Based Scheme for Uplink Communications in LoRaWAN

CHÉKRA EL FEHRI [1], NOUHA BACCOUR [2], AND INÈS KAMMOUN [1] (Senior Member, IEEE)

[1] LETI Laboratory, National Engineering School, Sfax 3038, Tunisia

[2] REDCAD Laboratory, National Engineering School, Sfax 3038, Tunisia

CORRESPONDING AUTHOR: C. EL FEHRI (e-mail: chekraenisienne@gmail.com)

ABSTRACT Long Range Wide Area Network (LoRaWAN) is currently one of the leading communication technologies for the Internet of Things (IoT) connectivity. It offers long-range and wide-area communication at low-power, low cost and low data rate. However, several studies demonstrated that LoRaWAN exhibits excessive collisions and thus performance degradation especially at large scale. This is mainly due to the ALOHA-based channel access technique adopted for uplink communications in LoRaWAN. In this work, we present a new schedule-based schema which allows a deterministic allocation of time, channel and spreading factor, for a collision-free uplink communication in LoRaWAN. For the sake of interoperability, our schema does not involve any additional synchronization phase for end-devices, or major changes in LoRaWAN specification as suggested in most of the existing studies in the literature. The performance evaluation of our proposed scheme proved an outstanding improvement of the network performance in terms of latency and energy-consumption. For instance, for an inter-packet transmission interval equal to 1800s (one packet each 30 min) and a cell size equal to 1000 end-devices, results show that using the proposed schedule-based schema, the uplink communication latency and energy consumption are reduced respectively by 89% and 78%, compared to the original LoRaWAN legacy class A.

INDEX TERMS Class A, class B, energy efficiency, reliability, large scale, LoRaWAN, latency.

I. INTRODUCTION

LOW POWER Wide Area Networks (LPWANs) refer to a class of wireless networks that offer long-range and low-rate connectivity at low-power consumption. Currently, LPWANs [1] are widely used in Internet of Things (IoT) for a wide-area monitoring and supervisory control, using a large number of low-powered end-devices. Practical IoT applications for LPWAN technology include smart agriculture [2], environmental monitoring [3], smart cities [4], utilities management such as smart grids [5], etc.

In this study, we focus on Long Range Wide Area Network (LoRaWAN) [6], which is a leading LPWAN technology that gained widespread adoption in industry due to their unique features. For instance, LoRaWAN offers both public and private network deployments using free ISM bands and involves a solution for adaptive data rate contributing to reliable communications.

LoRaWAN technology encompasses the physical and the MAC layers. The MAC layer protocol, also referred as LoRaWAN is standardized by LoRa Alliance. The physical layer protocol is based on LoRa [7], which is a modulation technique patented by Semtech and allows Long Range coverage. As for the network architecture, LoRaWAN end-devices are connected via a star topology to a gateway LoRa that acts as a bridge to relay bi-directional communication traffic between end-devices and a central intelligence entity called network Server (netServer). Uplink communication in LoRaWAN is based on pure ALOHA (Advocates of Linux Open-source Hawaii Association) [8], where channel access is at the will of end-devices. However, the downlink communication is differentiated by three classes; class A, B and C. In class A, the downlink communication is initiated by an uplink packet transmission, i.e., performed after a certain delay following the end of uplink packet transmission. In class B, end-devices are synchronized via a beacon message periodically broadcasted by the gateway LoRa. Downlink communications in class B are scheduled in defined time ping slots. Finally, in class C end-devices are in a continuous listening to the channel to receive downlink packets, except when they are transmitting an uplink packet.

The use of ALOHA-based channel access technique for uplink communications in LoRaWAN can be explained by the fact that ALOHA is a widely recognized protocol for its low complexity, ease of implementation and low overhead. Further, it perfectly matches the sporadic and scarce data traffic commonly seen in IoT applications. However, ALOHA is also known to be unable to deliver QoS guarantee due to its random channel access nature, especially at high network traffic load typically found at large scale deployment of LoRaWAN. Channel capturing, retransmission procedure, data rate adaptation and the fine-tuning of physical layer parameters are a set of mechanisms provided by LoRaWAN specification [6] to meet reliable communications and mitigate collisions caused by the ALOHA-based channel access technique. However, several studies demonstrated that in large scale LoRaWAN, ALOHA remains the main source of excessive collisions leading to performance degradation [9], [10], [11]. Hence, several research works were devoted to the optimization of LoRaWAN channel access and then improve the reliability for uplink communications in large scale. These optimization efforts allow either to avoid or eliminate collisions which reduces packet loss, contributes to energy conservation and thus improve communication reliability.

In this work, we propose to optimize LoRaWAN channel access through a deterministic scheduling of time slot, channel and spreading factor for each uplink packet transmission, which eliminates collisions and improves communications reliability. In contrast to the existing literature, our proposed schedule-based mechanism does not involve an additional synchronization phase or major changes to the current LoRaWAN specification affecting protocol interoperability. End-devices join the network as class A by executing the Over-The-Air Activation procedure in accordance with LoRaWAN specification. After joining the network, they switch from class A to class B. Indeed, we propose to take advantage from the synchronization already provided by LoRaWAN class B for downlink communications. We exploit the beacon message used to synchronize class B end-devices for both downlink and uplink communications. Once synchronized, end-devices autonomously derive their time slot and channel for uplink transmissions.

The rest of the paper is structured as follows: In Section II, we present LoRaWAN key concepts including class A and class B operating modes, and the Over-The-Air Activation procedure. In Section III, we discuss related literature. Next, we describe our schedule-based scheme in Section IV. The analytic performance model of the proposed scheme is given in Section V. In Section VI, we evaluate the performance of the proposed scheme and we compare it to LoRaWAN legacy. Finally, we conclude the paper in Section VII.

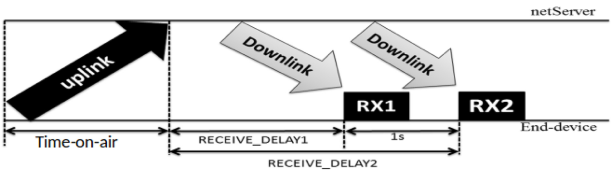


FIGURE 1. LoRaWAN-class A end-device timing.

II. BACKGROUND

LoRa is a modulation and coding scheme patented by Semtech for sub-1 GHz ISM bands. LoRa modulation is based on the Chirp Spread Spectrum (CSS) [12] technique. This technique allows (i.) more resilience and robustness to interference and Doppler effect and (ii.) long range coverage. LoRa offers six quasi-orthogonal spreading factors (SF). Using a larger SF extends the coverage range by decreasing the receiver sensitivity, but it comes at the cost of data rate. By default, there are three mandatory channels: 868.1MHz, 868.3MHz, and 868.5 MHz. When an end-device has a packet to send, it selects randomly one of the default channels.

LoRaWAN is an LPWAN communication technology standardized by LoRa Alliance [13]. LoRaWAN offers the MAC layer specification and runs on top of LoRa physical layer. The uplink traffic in LoRaWAN (traffic from the end-device to the netServer) may be either “confirmed” which needs to be acknowledged by the netServer, or “unconfirmed” which does not need any acknowledgment from the netServer. According to the required downlink communication (from netServer to end-device) latency and energy consumption, LoRaWAN defines three operation modes for end-devices: class A, class B and class C. Regardless of the end-device class, the uplink communication remains at the will of the node and uses an ALOHA-like channel access technique. Next, we give an overview of these operation modes followed by a description of the network join procedure, the so called Over-The-Air Activation procedure (OTAA).

A. LORAWAN CLASS A

In class A, the downlink communication is initiated by uplink packet transmissions: As illustrated in Fig. 1, the end-device opens a first receive window for the downlink communication, a RECEIVE_DELAY1 (set by default to 1s) after the end of the uplink packet transmission. A second receive window is opened one second later, that means a RECEIVE_DELAY2 (set by default to 2s) after the end of the uplink packet transmission. If the netServer has a downlink packet to send, it should send it at the beginning of a receive window.

The downlink packet sent in RX1 to the end-device uses the same channel as the uplink packet. On the other hand, the data rate of the downlink packet is fixed according to a mapping table [14] that gives the 'RX1DROffset' for a given uplink data rate. The 'RX1DROffset' default value is 0 which implies a downlink packet data rate equal to the uplink packet data rate. Downlink packets sent in RX2 use the default parameters 869.525 MHz / DR0 (SF12, 125 kHz).

B. LORAWAN CLASS B

Downlink communications in class B are scheduled in specific time-slots (called ping slots) derived by the end-devices once synchronized. These ping slots are allocated to ensure guaranteed delivery of MAC commands from netServer to end-devices. These last are synchronized by the reception of a beacon message which is broadcasted periodically by the LoRa gateway using the coding rate $CR = \frac{4}{5}$ and the data rate $DR3$ (SF9), over the channel frequency $869.525MHz$. As shown in Fig. 2, the beacon message is transmitted each (1) Beacon_period fixed to $128s$. The duration of a Beacon transmission corresponds to (2) a beacon_reserved time interval. Before each beacon transmission a (3) beacon_guard time interval is considered to avoid any interference between downlink packets and the beacon message. The time interval (4) Beacon_window separating the end of the Beacon_reserved and the beginning of the Beacon_guard is divided into (5) 4096 ping slots, each one with a time duration of $30ms$. Each end-device allocates (6)'PingNb' ping slots during which it wakes-up to listen to a downlink packet from the net-Server.

The ping slots allocation is based on a parameter called "PingOffset" [6], fixed pseudo-randomly by both the end-device and the netServer using the aes128_encrypt algorithm [15]. The PingOffset parameter is used to determine the time difference between the beginning of the beacon period and the beginning of the first ping slot. Two successive ping slots for a given end-device are separated by a time duration called a 'PingPeriod'. The two parameters 'PingNb' and 'PingPeriod' are expressed as follows:

$$PingNb = 2^{7-p} \tag{1}$$

$$PingPeriod = 2^{5+p}, \tag{2}$$

where p $(0 \leq p \leq 7)$ refers to the downlink periodicity.

Recall that the uplink communication in class B is performed exactly as in class A. For instance, the end-device opens the two receive windows RX1 and RX2 according to the same delays defined in class A. However, the uplink communication in class B is submitted to some constraints imposed by the schedule-based downlink communication, as shown next:

- When a receive window interferes with a ping slot or a Beacon_reserved, the end-device should cancel the receive window.
- The end-device can not send an uplink packet if the uplink transmission interferes with a ping slot or a Beacon_reserved.

C. OVER-THE-AIR ACTIVATION PROCEDURE (OTAA)

Before initiating any packet exchange, each end-device should be activated — joins the network, as a class A end-device (default class). LoRaWAN specification provides two activation methods: The first is performed by Personalization and suitable for small networks. The second method is called Over-The-Air Activation (OTAA) which is well suited for a large scale LoRaWAN. It consists in sending a join-request packet by the end-device and waiting for a join-accept packet from the netServer during a certain delay. The join-request packet includes the Join-Server identifier (JoinEUI), the end-device identifier (DevEUI) and the DevNonce — a counter incremented by the end-device each new join-request attempt. Once the join-request is received, the Join-Server returns a join-accept packet containing the following fields: JoinNonce which is a counter incremented every new join-accept sent to the end-device; the network identifier; the end-device address; DLSettings (Downlink Settings) field and an optional Channel List (CL) field.

III. RELATED WORK

The performance of LoRaWAN was subject of several research studies [16], [17], [18]. Performance metrics include packet delivery and collision rates [16], packet success ratio [17], retransmissions and energy consumption [18]. Whatever the measured metric, these studies confirm that LoRaWAN exhibits excessive collisions and thus performance degradation especially at large scale, due to the AlOHA-based channel access technique used in LoRaWAN. Hence, several research works were devoted to the optimization of LoRaWAN channel access. Existing optimization mechanisms can be classified into contention-based and schedule-based.

The idea behind contention-based mechanisms is to take the advantage of historical ALOHA optimization efforts to improve LoRaWAN channel access which is a variant of pure ALOHA. These mechanisms include Slotted ALOHA (S-ALOHA) [19], [20], [21], [22], the Carrier Sense Multiple Access (CSMA) [23] and its varieties x-CSMA [24], persistent CSMA (p-CSMA) [25], and the Listen-Before-Talk (LBT) [26].

Despite of the improvements provided by contention-based mechanisms, they are still enable to handle excessive collisions at large network scale and thus still suffer from network performance degradation. For instance, the S-ALOHA mechanisms introduced in [19], [20] guarantee good network performance for small scale networks only. Moreover, CSMA and Listen-Before-Talk schemes, require an additional time interval dedicated to sense the channel before sending an uplink packet leading to additional energy consumption.

As a result, recent research studies on LoRaWAN channel access optimization advocate the use of deterministic and contention-free channel access techniques. They are also referred as schedule-based techniques where channel access parameters such as time slot, channel, SF and Tx-power are scheduled for each transmission.

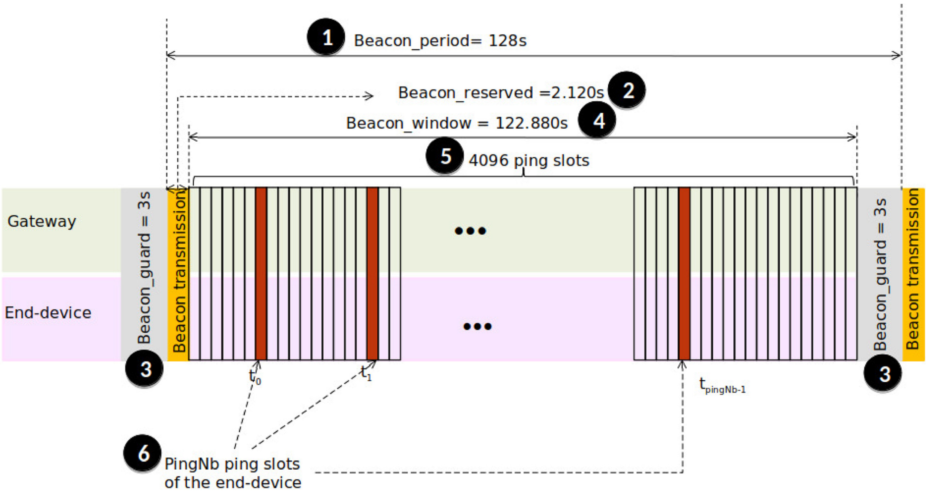


FIGURE 2. LoRaWAN-class B: Beacon period timing.

Except in [27], all existing schedule-based channel access optimization techniques require the modification of LoRaWAN specification: In [28], [29], [30], [31], [32], [33], a synchronization phase is introduced to synchronize class A end-devices, which are originally not synchronized entities. The downlink communication process of LoRaWAN is altered in [34] by allocating one time slot for downlink packets, and in [35] by grouping a set of acknowledgments into one message sent in a specific time slot. In [29] and [36], a new beacon structure is introduced, as a replacement of the original LoRaWAN beacon structure. The study carried in [37] involves the modification of the whole network and the hardware architectures. Finally, in [38] and [39], the authors suggest to replace the LoRaWAN MAC layer by the design of a new MAC protocol, "Lora+" and "Rt-lora" respectively.

In contrast to existing works, our work does not involve any of these modifications in LoRaWAN specification. The schedule-based mechanism that we propose is based on class B end-devices to take advantage of the synchronization phase and the beacon message structure described in LoRaWAN specification. The scheduling scheme presented in [27] also does not bring a major modification to LoRaWAN specification. However, it is not a collision free scheme and it provides a packet delivery ratio lower than 70%.

Next, we compare the schedule-based techniques in terms of (1) scheduled parameters (e.g., time slot, channel, SF, tx-power) and (2) the allocation technique (e.g., probabilistic, deterministic). These design choices significantly impact the performance of the schedule-based channel access technique. We also compare the schedule-based techniques in terms of delivered performance: presence of collisions, and collision-free. Collision free techniques are able to provide 100% of packet delivery ratio. The comparison between existing schedule-based channel access techniques for LoRaWAN is presented in Table 1.

In [27], the scheduling is performed based on SF parameter only. The SF allocation is viewed as an optimization problem where the total throughput of LoRaWAN network is maximized. The studies in [38], [40] and [28] considered the scheduling of channel frequencies in addition to the SF parameter. The allocation of SF and channel parameters is performed based on the RSSI estimation in [38] and channel utilization in [40]. In [29], the authors consider the scheduling of transmission power Tx-power in addition to SF and channel, in order to improve network performance.

In the above works, the scheduling of SF, channel frequency and Tx-power allowed to enhance the network performance by reducing yet not eliminating collisions. Consequently, packet error rates still increase with the increase of the number of end-devices or the traffic load in the network. To overcome this limitation, several studies consider the time slot allocation in the design of the schedule-based channel access scheme.

In [30], the introduced schedule-based scheme considers both SF and time slots allocation, where time slots are scheduled according to a deterministic scheme. Hence, the observed packet delivery ratio reaches 100%. The two schemes suggested in [39] and [36] consider the channel frequencies, SF and time slots, in a new MAC layer design, namely RT-LoRa and FCA-LoRa respectively.

In RT-LoRa protocol, confirmed traffic is scheduled based on a deterministic approach while the schedule of unconfirmed traffic is not deterministic as time, SF and channel are randomly allocated. FCA-LoRa protocol, defines a new beacon message broadcasted periodically by the gateway and contains a list of available SFs, channels and time slots to be used by LoRaWAN end-devices for each beacon period. Before each transmission, the end-device performs the CSMA-CA scheme to avoid collisions which provides more robustness to collisions, however it increases the end-device energy-consumption.

TABLE 1. Schedule-based channel access techniques for LoRaWAN.

Technique	Number of end-devices and class	Scheduled parameters	Evaluation metric	Collision free	Bringing major modification to LoRaWAN
[27]	500, class A	SF	throughput packet delivery ratio	X	no
[38]	500, class A/B	SF/channel	packet error rate packets rejected	X	yes
[40]	3000, class A	SF/channel	collisions DER energy fairness	X	yes
[28]	-, class A/B/C	SF/channel	throughput packet error rate	X	yes
[29]	class A 1 cell (up to 1000) multicell (up to 3500)	SF/tx_power/channel	packet error ratio throughput energy efficiency fairness	X	yes
[30]	5000, class A	SF and time	collision ratio packet delivery ratio energy consumption	✓	yes
[39]	100, class A/B/C	SF, time and channel	packet loss ratio end-to-end delay	X	yes
[36]	500, class A/B/C	SF/time/channel	colisions throughput channel utilization mean power consumption	✓	yes
[34]	1000, -	time slots	packet delivery ratio energy consumption retransmission lost packets non-ack packets	✓	yes
[31]	2000, class A	time slots	throughput network goodput packet delivery ratio battery capacity	✓	yes
[32]	300	time slots	packet delivery ratio collision ratio frame size execution time	✓	yes
[37]	9, class A/B/C	time slots	reliability latency lifetime	✓	yes
[33]	300, class A/B/C	time slots	collision probabilty throughput max.duty cycle	✓	yes
[35]	2000, class A	time slot channel tx-power SF	PDR collisions energy consumption device lifetime	✓	yes
our study	1000, class B	SF, time, and channel	latency energy consumption	✓	no

Hence, it can be inferred that including time slot allocation in the schedule-based channel access technique is not enough. Indeed, the time slot allocation should be deterministic to guarantee disjoint time slots for transmissions and thus exclude collisions. Existing studies based on a deterministic time slot allocation include [31], [32], [33], [34], [35], [37]. The scheme proposed in [34] derives time slots based on a hash algorithm executed using the device address. In [31], a Network Synchronization and Scheduling Entity (NSSE) is dedicated to synchronize the network entities and schedule time slots based on the uplink traffic requirements and data periodicity. The scheduling of time slots in [32], [37] and [33] is performed based on the end-device ID. Finally, 'Free' scheme [35] performs the scheduling of time slot, channel, SF and Tx-power which are communicated in a join-accept packet while modifying the DLSettings field. The schedule is based on the number of end-devices, the data they have in their buffer and the path loss estimation.

In our work, we introduce a schedule-based mechanism for the optimization of LoRaWAN channel access. Our technique does not bring any modification of the LoRaWAN specification and integrates a time slot scheduling based on a deterministic approach, which guarantee a collision-free channel access. We present in the next section a detailed description of our mechanism.

IV. PROPOSED SCHEDULED-BASED SCHEME FOR LORAWAN UPLINK COMMUNICATIONS

A. GENERAL DESCRIPTION

We consider a LoRaWAN cell, represented by a disc radius R. The cell is composed of one gateway and a number of end-devices ranging from 500 to 1000 end-devices. We recall that our objective is to design a scheduling scheme that enhances LoRaWAN performance by eliminating collisions, without bringing major modifications to the original LoRaWAN specification.

We consider the following assumptions:

- The network operates under perfect SFs orthogonality conditions.
- The end-device location is already known by the netServer.

Our designed scheme consists of two main phases: (1) The pre-configuration phase which is performed by the LoRaWAN netServer. This last attributes a group index (g_i) and a spreading factor (SF) to each end-device during the pre-configuration phase. Communicating these parameters to the end-device in question during this phase makes it switch to class B; and (2) The scheduling phase which is performed by the end-device to autonomously derive the channel and the time slot for uplink transmissions, based on the received parameters (g_i and SF). Hence, scheduled parameters include (1) SF ranging from SF7 to SF12, (2) time slot, which is selected among a set of available slots, and (3) channel, which can be either ch0 or ch1. In fact, uplink communications in our scheduling scheme are performed using 2 channels among the three available default channels of LoRaWAN. This design choice pertains to the 8 reception paths of LoRa gateways SX1301 [41] as detailed next.

B. THE PRE-CONFIGURATION PHASE

The netServer in LoRaWAN is an intelligent entity having a global view of the network. Its main responsibility is to configure end-devices and gateways in order to ensure secure and energy-efficient communication. In our scheme, the netServer is also responsible for the computation of two scheduling parameters with respect to each end-device, namely g_i and SF. These parameters are computed during the Over-The-Air Activation procedure (described in Section II-C), upon the reception of a join-request from the end-device, and then disseminated to the end-device over a join-accept. The join-accept packet includes 16-optional bytes for the "CFList" field. This field represents the list of channels that will be used by the network entities. We use the two first bytes of "CFList" field to communicate the scheduling parameters g_i and SF to the end-device. The exchange of join-request and join-accept packets between the netServer and the end-device is illustrated in Fig. 3.

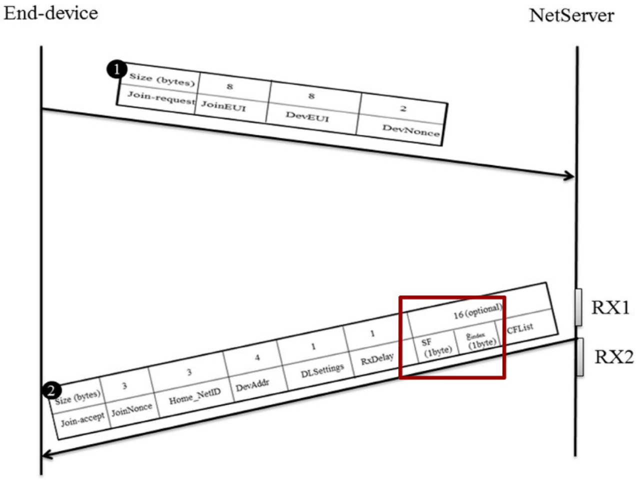


FIGURE 3. Dissemination of scheduling parameters g_i and SF to the end-device.

The end-device is activated when it receives the join-accept from the netServer. Then, it switches from class A to class B which represents the end of the pre-configuration phase and the beginning the scheduling phase.

Next, we introduce our algorithm (refer to Algorithm 1) for the determination of the couple (SF, g_i).

The netServer divides the network cell area into 6 annulus where each annulus is defined by two concentric circles with radius l_{j-1} and l_j, $1 \leq j \leq 6$. The 6 available spreading factors $SF7$ to $SF12$ are attributed to the 6 annulus l_1 to l_6 respectively. An annulus includes a number of end-devices depending on their distance to the gateway (refer to Fig. 4). SF is allocated to the end-device based on its membership to one of the 6 annulus. For instance, end-devices located in the first (nearest) annulus use SF7, and those located in the last (farthest) annulus use SF12. These settings as well as the concept of annulus are inspired from the study in [42], where they were demonstrated to guarantee a good signal reception. In fact, SF impacts the communication range (higher SF leads to longer range). Hence higher SFs are attributed to annulus/end-devices that are farther to the gateway. For instance, SF12 (the highest SF) provides longest range and it is attributed to the farthest annulus.

Our algorithm has as inputs the distance separating the end-device from the gateway d_{ed}, and the annulus l_1 to l_6. When the netServer receives a join-request packet from the end-device, it checks if it is for the first time. To do, it extracts the DevEUI of the end-device from the join-request packet to check if it exists in its database. If it exists then the join-request packet was previously received from the end-device. In this case, the netServer sends SF and g_i that were previously attributed to the end-device (Line 2 $\longrightarrow$ Line 5). Otherwise, the netServer first derives SF based on the end-device distance (Line 7 $\longrightarrow$ Line 18). Then it attributes the end-device to a group, defined by g_i, so that all the group

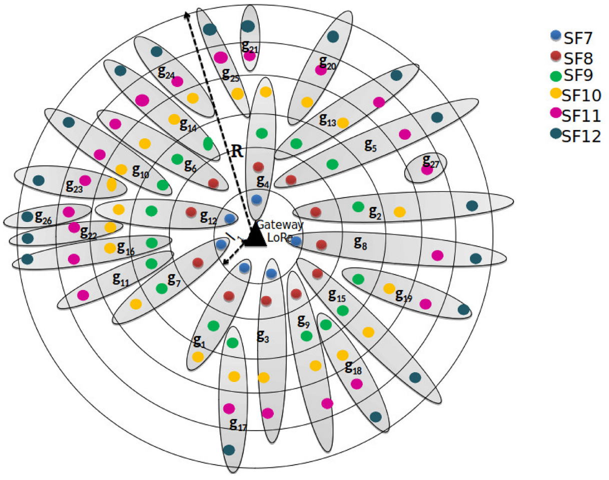


FIGURE 4. ***SF*, g_i allocation in the Pre-configuration phase.**

members have disjoint SFs. The maximum number of end-devices in a group is 4. This number is justified by the fact that the LoRa gateway SX1301 is equipped with 8 reception paths that make it able to demodulate simultaneously up to 8 packets. Hence, based on our approach two groups using different channels can transmit their respective packets simultaneously without a risk of collisions.

To derive the g_i for a given end-device, the netServer browses through available groups and looks for a group (1) having less than 4 end-devices and (2) does not involve an end-device having the same SF than that of the end-device in question. If the group is found, its index g_i is attributed to the end-device and the netServer sends parameters g_i and *SF* in the join-accept packet to the end-device (Line 19 ⟶ Line 26). If the netServer does not find a group for the end-device, it (i) creates a new group identified by g_i, (ii) attributes the end-device to the group and (iii) sends the parameters *SF* and g_i to the end-device (Line 27 ⟶ Line 30).

An example of group and spreading factor allocation within the whole network is given in Fig. 4, which also illustrates an example of the pre-configuration phase output.

Algorithm 1 *SF* and g_i Assignment

```
Input $d_{ed}$, $l_1$, $l_2$, $l_3$, $l_4$, $l_5$, $l_6$
Output $g_i$, $SF$
1:  DevEUI ← ExtractDevEUI()
2:  if exist(DevEUI)=True then
3:      SF ← GetSF(DevEUI)
4:      $g_i$ ← Getgroup(DevEUI)
5:      SendToNode(SF,$g_i$)
6:  else if exist(DevEUI)=False then
                                        ▷ Assigning SF to the end-device
7:      if $0 \leq d_{ed} \leq l_1$ then
8:          SF ← SF7
9:      else if $l_1 \leq d_{ed} \leq l_2$ then
10:         SF←SF8
11:     else if $l_3 \leq d_{ed} \leq l_4$ then
12:         SF←SF9
13:     else if $l_5 \leq d_{ed} \leq l_6$ then
14:         SF←SF10
15:     else if $l_7 \leq d_{ed} \leq l_8$ then
16:         SF←SF11
17:     else if $l_8 \leq d_{ed} \leq l_9$ then
18:         SF←SF12
                                        ▷ Assigning $g_i$ to the end-device
19:     GroupFound=False
20:     for $k$ ← 1 to $size(all\_groups)$ do
21:         if $size(group_k)$ ≤ 4 and contains($group_k$, GetSF(DevEUI))=False then
22:             $g_i \leftarrow k$
23:             AddNodeToGroup($g_i$, DevEUI)
24:             SendToNode(SF,$g_i$)
25:             GroupFound ←True
26:             break
27:         if k=size(all_groups) and GroupFound=False then
28:             $g_i \leftarrow k+1$
29:             BuildNewGroup($g_i$)
30:             SendToNode(SF,$g_i$)
```

C. THE SCHEDULING PHASE

At the end of the pre-configuration phase, all the end-devices are activated, have their scheduling parameters (*SF* and g_i), and switched to LoRaWAN class B. The objective of the scheduling phase is to allocate additional parameters to schedule the uplink transmission, namely the channel and the time slot. These parameters are attributed to each group and thus shared between end-devices belonging to the group. Importantly, each end-device is able to derive the channel and the time slot autonomously without any packet exchange, based on the knowledge of g_i parameter.

Recall that in class B, end-devices are synchronized using a beacon message broadcated periodically by the LoRa gateway [6]. Downlink traffic is scheduled over specific time slots and uplink communication is originally based on ALOHA (contention-based) and thus collisions are well frequent. In our approach, both uplink and downlink communications in LoRaWAN class B are scheduled to avoid collision. To achieve this goal, we consider another beacon message for uplink communication which is sent by the LoRa Gateway, following the downlink beacon message. Fig. 5 illustrates the time structure using our scheduling schema for LoRaWAN. It is a succession of an Uplink_beacon_period with a defined duration (1) followed by a Downlink_beacon period with a duration of 128s (6). Each beacon period is delimited by the transmission of two beacon messages (2) and includes a beacon_window for uplink (3) and downlink respectively, and a Beacon_guard time fixed to 3s. Note that the structure of Downlink_beacon_period is kept as in the specification (described in Section II). In what concerns Uplink_beacon_period, the difference is in the Uplink_Beacon_window, which is divided into uplink time slots (4). These slots are separated by a clock time drift which is counted as a part of the time slot. The total number of

TABLE 2. The slotFrame of the scheduling mechanism.

channel Time slot	T_0	T_1	T_2	$\cdots$	$T_{UpSlotNb-1}$
ch0	g_1	g_3	g_5	$\cdots$	g_{ng}
ch1	g_2	g_4	g_6	$\cdots$	

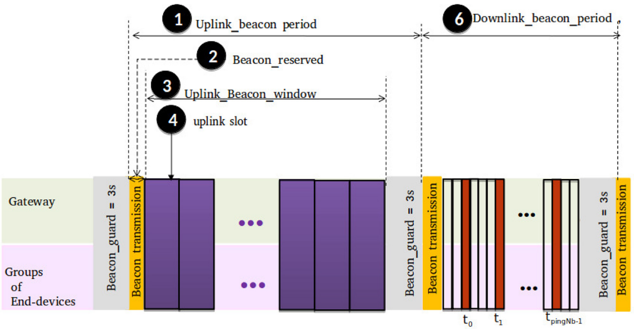


FIGURE 5. An uplink beacon period followed by a downlink beacon period structure.

uplink time slots is denoted as "UpSlotNb". It depends on the total number of groups n_g, used in the pre-configuration phase, which in turns depends on the number of end-devices and their distribution in the network cell area. This what explains the variable duration of Uplink_beacon_period.

Next, we show how end-devices derive channel and time slot parameters to schedule their uplink transmissions, based on g_i and SF, extracted from the received join-accept packet. First, the end-device waits for the reception of two successive beacon messages, to synchronize with the Uplink_beacon_period and time slots. Then, based on its group index g_i, it derives (i.) the channel, which is selected among two mandatory channels 868.1MHz and 868.3MHz, denoted as ch_0 and ch_1 respectively; and (ii.) the time slot for the uplink transmission T_k where k is in [0,UpSlotNb-1]. Channel ch_p and time slot T_k are allocated to the group g_i, where indexes p and k are derived as a function of the index i of the group, and expressed as follows:

$$\begin{cases} k = (i \textbf{ div } 2 + i \textbf{ mod } 2) \ -1 \\ p = (i+1) \textbf{ mod } 2 \end{cases} .$$

To better understand the allocation of ch_p and T_k to a group g_i, we consider $\mathbf{M}_{2,UpSlotNb}$, a 2-dimensional matrix representing the slotframe of our scheduling schema, illustrated in Table 2. The matrix has tow lines that represents the two channels ch_0 and ch_1 and $UpSlotNb$ rows representing the uplink time slots T_0 to $T_{UpSlotNb-1}$. The matrix cells contains the different groups g_1 to g_{n_g}. The number of uplink time slots can be expressed as a function of the total number of groups as follows:

$$UpSlotNb = n_g \textbf{ div } 2 + n_g \textbf{ mod } 2 \tag{3}$$

During an uplink time slot, the members of a group should be able to send their packets and receive their acknowledgment in case of confirmed traffic. According to LoRaWAN specification [14], after sending the uplink packet, the end-device opens a first receive window RX1 after a RECEIVE_DELAY1 and a second receive window RX2 after a RECEIVE_DELAY2 to receive the acknowledgment (refer to Section II). Hence, the length of an uplink time slot $UpSlotLen$ is expressed as follows (refer to Figure 5):

$$UpSlotLen = ToA_{max} + RECEIVE\_DELAY2 + T_{ack2} + T_{cd}, \tag{4}$$

where ToA_{max} is the maximum time-on-air of an uplink packet transmission in LoRaWAN, RECEIVE_DELAY2 is the delay to begin the second receive window RX2 set to $2s$ by default in LoRaWAN [14], T_{ack2} is the time-on-air of an acknowledgement sent in RX2 in LoRaWAN, and T_{cd} is the clock drift. The time T_{cd} depends on the clock accuracy A_c and it is expressed as:

$$\begin{aligned} T_{cd} = A_c \times (&Uplink\_beacon\_period \\ &+ Downlink\_beacon\_period), \end{aligned} \tag{5}$$

where

$$\begin{aligned} Uplink\_beacon\_period = \ &beacon\_reserved \\ &+ Uplink\_beacon\_window \\ &+ Beacon\_guard \end{aligned} \tag{6}$$

The term $Uplink\_beacon\_window$ is the product of the number of slots and the slot duration, i.e., $UpSlotNb \times UpSlotLen$. The slot duration $UpSlotLen$, can be estimated according to Eq. (4) without accounting the time clock drift T_{cd}.

Notice that all the group members send their packets and open their receive windows simultaneously during the allocated time slot. Hence, in our scheduling schema, the netServer multicasts an acknowledgment to the group members using SF12, to ensure that the acknowledgment reaches the farthest end-devices in the network. In LoRaWAN specification, downlink packets (acknowledgments) are sent either in RX2 using SF12, or in RX1 using the same SF as the uplink packet. The LoRaWAN specification gives the possibility to modify this default setting to make sending the acknowledgment during RX1 using SF12 whatever the SF of the uplink packet. Indeed, this configuration can be done by setting the RX1DRoffset in the DLSetting in the join-accept packet to 5 [14].

Hence, at the end of the scheduling phase, each end-device allocates autonomously a time slot and a channel, which are shared with all the members of its group. Hence, at each time slot two groups of end-devices transmit their packets simultaneously (8 packets) to the gateway, without a risk of collision. One acknowledgment is sent in multicast to all the group members by the netServer, in RX1 or RX2 and using SF12.

D. DISCUSSION

The proposed schedule-based schema does introduce some changes compared to the standard LoRaWAN class A and B. However, these changes can be considered non-major due to several key reasons. Indeed, compared to LoRaWAN specification, our proposed schedule-based schema introduces the following modifications:

- *Beacon timing and frequency:* In LoRaWAN class B, the gateway periodically generates a beacon to synchronize downlink data. Our schema introduces an additional beacon transmission dedicated to synchronize uplink data, which effectively mitigates collision risks. While the downlink beacon period retains the same structure and timing as specified in LoRaWAN class B, we make an adjustment in the timing of the uplink beacon period. This modification allows us to dynamically derive the duration of the uplink beacon window based on the number of end-device groups, a feature illustrated in Figure 5.
- *Scheduled channel access:* In LoRaWAN specification, uplink communication is based on pure ALOHA, where channel access is at the will of end-devices (random channel access). In our proposed schedule-based schema, each end-device derives autonomously and without the need of any additional overhead, a time slot within the uplink beacon period, to send the uplink packet without a risk of collision (scheduled channel access).
- *The exploitation of the optional field "CFList" in join accept packet:* In order to disseminate scheduling parameters, we use the two first bytes of "CFList" field in join-accept packet. This field includes 16-optional bytes to specify the list of channels. Hence, scheduling parameters are disseminated without the need of additional signaling overhead.
- *Multicast ACK transmission:* In LoRaWAN specification, the netServer transmits ACK in unicast to the end-device in question. Based on our proposed schedule-based schema, all the group members send their uplink packets simultaneously (use the same time slot but different SFs and channels). Hence, the server multicasts only one ACK to the group members using SF12 (highest value) to ensure that the acknowledgment reaches the farthest end-devices in the network. Note that LoRaWAN specification gives the possibility to set the SF.

We argue that these proposed changes do not disrupt compliance with LoRaWAN specification. For instance, these changes do not alter the packet structure or signaling mechanisms that are integral to the LoRaWAN protocol. In contrast to our approach, existing schedule-based optimization techniques often propose either a complete replacement of the existing LoRaWAN MAC protocol [39] or require significant overhauls to the LoRaWAN specification. Examples of such extensive alterations include introducing a synchronization phase for non-synchronized class A end-devices [28], [29], [30], [31], [32], [33] and suggesting a new format for the beacon packet [29], [36]. Next, we analyze the performance of our schedule-based scheme analytically in terms of latency and energy consumption.

V. ANALYTIC MODELING: LATENCY AND ENERGY CONSUMPTION

Based on analytic modeling, we evaluate our schedule-based scheme for LoRaWAN and also the LoRaWAN legacy class A and class B. Performance criteria includes (1) the average latency $L_{Scheduling}$, which is the time between an uplink transmission and the acknowledgment reception, and (2) the average energy consumption by the end-device $E_{Scheduling}$, during a synchronization period S_t.

A. OPTIMIZED LORAWAN USING THE SCHEDULING SCHEME

To develop our analytic model, we consider the following assumptions:

- Although our schedule-based scheme is designed for both confirmed and unconfirmed uplink traffic, we only consider confirmed traffic in our evaluation.
- In the LoRaWAN cell, end-devices are uniformly distributed around the gateway.
- The uplink traffic delivered by the application layer is periodic and higher than the traffic delivered by the MAC layer. Let's denote T_{app} the application layer traffic periodicity, and T_{mac} the MAC layer traffic periodicity, for uplink packet transmission. Then, $T_{app} \geq T_{mac}$.
- We assume a perfect links quality in the network as we aim to study the impact of collision as the main cause of packet losses. To implement this assumption, we fixed the LoRaWAN cell radius R to 1 km to guarantee a good signal reception. Further, the allocation of SF based on the distance between the end-device and the gateway (i.e., the concept of annulus) allows to have good signal reception and thus high quality links even at the disk perimeter.
- Acknowledgments are received in RX2 only.
- Packet transmissions are scheduled since the second uplink beacon period.

T_{mac} corresponds to the duration between two successive uplink time slots, which is also the duration between two uplink beacon message transmissions. Hence, it is given by the following equation:

$$T_{mac} = (Uplink\_beacon\_period + Downlink\_beacon\_period). \tag{7}$$

We recall that Downlink_beacon_period is fixed to 128s according to LoRaWAN specification.

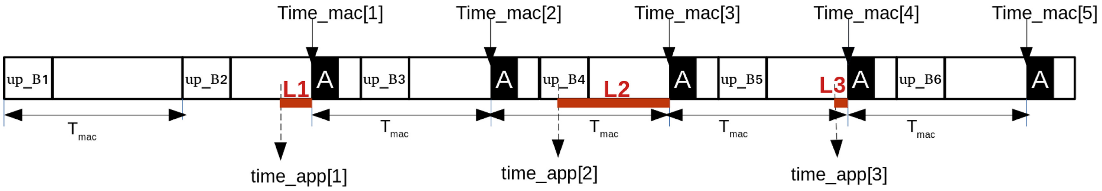


FIGURE 6. Latency of a successfull uplink transmission according to our scheme.

1) LATENCY DETERMINATION

To express the latency formula, we consider the following terms:

- *p_index* is the index of the packet delivered by the application layer.
- *Time_app*[*index*] refers to the time when the packet p_{index} is delivered by the application layer, where the time (00,00) starts at the reception of the first beacon message.
- *Time_mac*[*index*] refers to the time of the upcoming slot for the MAC layer to deliver the packet p_{index}.
- *SlotLen* is the length of a time slot and *slot_index* is the index of the slot in the schedule.

The Latency of a packet p_{index}, denoted as $L_{p_{index}}$ is the waiting time spent by a packet in the MAC layer queue. Hence, $L_{p_{index}}$ can be expressed as follows:

$$L_{p_{index}} = Time\_mac[i] - Time\_app[index], \tag{8}$$

where

$$\begin{cases} Time\_app[index] = T_{app} \times p_{index}, \\ Time\_mac[1] = T_{mac} + beacon\_reserved \\ +slot\_index \times slotLen, \\ Time\_mac[i] = Time\_mac[1] \\ +((Time\_app[index]\ \mathbf{div}\ T_{mac}) - 1) \times T_{mac}. \end{cases}$$

Figure 6 illustrates the computation of $L_{p_{index}}$, where *Up_Bi* represents an uplink beacon i and A is the time slot allocated for a given end-device within each beacon period delimited by the reception of two uplink beacons.

Finally, the average latency during a synchronization period S_t is given by the following equation:

$$L_{Scheduling} = \frac{\sum_{p_{index}=1}^{n_{packets}} L_{p_{index}}}{n_{packets}}, \tag{9}$$

where $n_{packets} = S_t\ \mathbf{div}\ T_{app}$, is the number of all the packets delivered by the end-device during S_t.

2) ENERGY-CONSUMPTION DETERMINATION

The synchronization period S_t is a succession of downlink and uplink beacon periods. During this synchronization period, there may be a certain number of unused uplink time slots (e.g., at the time slot, the end-device has no packets to send in the MAC buffer). Let's denote $n1$ the number of uplink beacon periods during S_t, where the scheduled time slot has been consumed to deliver an uplink packet and E_{bcUpTx} the energy consumption during one such uplink beacon period. Let's denote also $n2$ as the number of uplink beacon periods where the scheduled time slot was not used and $E_{bcUpOff}$ as the energy consumption during such uplink beacon period. The number $n3$ denotes the number of downlink beacon periods during S_t and E_{bcDown} is the corresponding energy consumption. Note that in downlink beacon periods the end-device remains in reception mode during the allocated donwnlink time slots (also called ping slots) even if no packet is received. The energy consumption of an end-device during S_t, $E_{scheduling}$ is then given as follows:

$$\begin{aligned} E_{scheduling} &= n1 \times E_{bcUptx} \\ &\quad + n2 \times E_{bcUpOff} + n3 \times E_{bcDown}, \end{aligned} \tag{10}$$

where E_{bcUptx} includes the energy consumption during (1) the beacon reserved time interval where the end-device is in reception mode E_{br}, (2) the uplink time slot where the end-device is alternating between the transmission mode, the sleep mode and the reception mode E_{upSlot} and (3)the remaining time of the beacon period where the end-device is in a sleep mode $E_{upSleep^{tx}}$.

Let's denote U the supply voltage, I_{tx} the transmission mode power, I_{rx} the reception mode power and I_{sleep} the sleep mode power. Hence, E_{bcUptx} can be derived as follows:

$$E_{bcUptx} = E_{br} + E_{upSlot} + E_{upSleep^{tx}}, \tag{11}$$

$$\text{where} \begin{cases} E_{br} = U \times I_{rx} \times Beacon\_reserved. \\ E_{upSlot} = U \times (I_{tx} \times (ToA_{max} + T_{cd}) \\ +I_{sleep} \times RECEIVE\_DELAY1+ \\ I_{rx} \times (T_{RX1} + T_{ack2})). \\ E_{upSleep^{tx}} = U \times I_{sleep} \times (Beacon\_guard \\ +Uplink\_Beacon\_window - UpSlotLen). \end{cases}$$

We consider $T_{RX1} = 1s$ as the time during which the end-device is in a reception mode during the first receive window RX1. It is considered as the time between the opening of the first receive window and the beginning of the second receive window.

When the end-device has no packets to send during its scheduled uplink time slot, it turns into a sleeping mode during all the uplink beacon window and the beacon guard as well. The corresponding energy consumption is $E_{upSleep}^{off}$. Hence, we derive $E_{bcUpOff}$ as follows:

$$E_{bcUpOff} = E_{br} + E_{upSleep}^{off}, \tag{12}$$

$$\text{where} \begin{cases} E_{br} = U \times I_{rx} \times beacon\_reserved. \\ E_{upSleep}^{off} = U \times I_{sleep} \times (Beacon\_guard \\ +Uplink\_Beacon\_window). \end{cases}$$

Next, we compute the energy consumption during the downlink beacon period, E_{bcDown} (refer to Figure. 2 and Section II-B). During this beacon period, the end-device first receives a beacon during the beacon_reserved time interval (E_{br}). Then, it turns on reception mode during the allocated ping slots ($E_{pingSlots}$). In the remaining time of the downlink beacon period, the end-device is in a sleep mode a ($E_{downSleep}$). Hence, E_{bcDown} is expressed as follows:

$$E_{bcDown} = E_{br} + E_{pingSlots} + E_{downSleep}, \tag{13}$$

$$\text{where} \begin{cases} E_{pingSlots} = U \times I_{rx} \times 0.03 \times PingNb. \\ E_{downSleep} = U \times I_{sleep} \times (beacon\_guard \\ +Downlink\_beacon\_window - 0.03 \times PingNb). \end{cases}$$

The parameters $n1$, $n2$ and $n3$ mentioned previously, are derived according to the following equations:

$$\begin{cases} n1 = \\ ((n_{totalBc} - 1) \times Uplink\_beacon\_period) \ \mathbf{div} \ T_{app}. \\ n2 = n_{totalBc} - n1. \\ n3 = n_{totalBc}. \end{cases} \tag{14}$$

where $n_{totalBc}$ is the total number of the couple Uplink/downlink beacon periods during S_t and we express it as:

$$n_{totalBc} = S_t \ \mathbf{div} \ (Uplink\_beacon\_period + Downlink\_beacon\_period) \tag{15}$$

We recall that the first Uplink_beacon_period is not used to schedule uplink communications. It is dedicated to define the length of the uplink beacon period for the end-device.

B. LORAWAN LEGACY CLASS A AND B

To assess the latency and energy consumption of the reliable uplink communication in LoRaWAN legacy (class A and B), we adopt the Markov chain based model proposed in [45]. Next, we give a short overview of this model.

Based on a finite state diagram representing the different states of a LoRaWAN end-device (class A and B) when communicating reliably an uplink frame to the gateway, we elaborate the transition matrix $\mathbf{P_n}$ where n is the number of possible states. Then, we write $\mathbf{P}$ in a canonical form by partitioning it into four submatrices: (1) The null sub-matrix; (2) The identity matrix $\mathbf{I_r}$ of dimension $r \times r$, where r is equal to 1 as in our model we have only one absorbing state representing the reception of an acknowledgement; (3) The matrix $\mathbf{R_{m,r}}$ that represents the transition probabilities from transient states to the absorbing state; and (4) The matrix $\mathbf{Q_{m,m}}$ that represents the transition probabilities from one transient state to another. Then, we derive the fundamental matrix, $\mathbf{M} = (\mathbf{I_m} - \mathbf{Q})^{-1}$, where $\mathbf{I_m}$ is an identity matrix of dimension $m \times m$. The entry $m_{i,j}$ of $\mathbf{M}$ is interpreted as follows: suppose that the end-device starts at the transient state s_i. To reach the absorbing state, the end-device needs to visit the state s_j, $m_{i,j}$ times. Given $\mathbf{w}$ a vector which entries represent the duration spent by the end-device at each state, we can derive the vector $\mathbf{t}$ according to the following formula: $\mathbf{t} = \mathbf{M} \times \mathbf{w}$. An entry t_i represents the number of time units (e.g., seconds) needed by the end-device to reach the absorbing state when it starts from the state s_i. Hence, it can be inferred that the entry t_0 represents the latency for an uplink data frame to be acknowledged that we denote $L = t_0$.

Similarly, to compute the end-device energy consumption, we replace the vector $\mathbf{w}$ by the vector $\mathbf{y}$ where each entry represents the energy consumption at a given state. Then we derive the vector e as follows: $\mathbf{e} = \mathbf{M} \times \mathbf{y}$. We denote the energy consumption for an end-device to reliably communicate an uplink frame as $\mathbf{E} = e_0$.

VI. EVALUATION AND NUMERICAL RESULTS

Based on the developed analytic models for latency and energy consumption, we compare the optimized version of LoRaWAN using our schedule-based scheme to the original LoRaWAN legacy meeting reliable uplink communication (class A and class B), in terms of latency and energy consumption.

The analytic model with respect to our schedule-based schema has as input the layout of end-devices in the LoRaWAN cell area, which follows uniform distribution. This distribution has an impact of the performance of our proposed schema: The total number of groups varies according to the distribution of end-devices and not the number of end-devices. As the number of groups increases, the number of slots in the uplink beacon window also increases and thus the duration of the uplink beacon period increases as well. Therefore, the latency increases but energy consumption decreases because beacons are transmitted less frequently. Hence, while the analytical model is deterministic, one of the model inputs is a random variable, namely the emplacement of end-devices, which follows a uniform distribution. Hence, each numerical result is an averaged value derived from 10 simulations in order to increase the confidence of our results.

In what concerns the analytic model with respect to LoRaWAN legacy, we use the Markov-chain model introduced in [45] and briefly described above. We establish the latency $L_{A/B}$ and the energy consumption $E_{A/B}$ with respect to class A and class B and we consider the following settings:

- Class A end-devices with NbTrans (maximum number of retransmission attempts) fixed to 8. The end-devices keep resetting the uplink packet transmission and the SF value to the lowest one (SF7) each 8 failed attempts till receiving an acknowledgment from the netServer.
- Class B end-devices with NbTrans fixed to 8 and a downlink periodicity $p = 7$. The end-device reset the packet transmission and the SF value to the lowest one each 8 failed attempts till receiving an acknowledgment.

The settings for the evaluation of our schedule based LoRAWAN are summarized in Table 3.

As for the evaluation scenario, we analyze the latency and the energy consumption as a function of (1) the number of end-devices when T_{app} is fixed to $1800s$, and (2) as a function of T_{app} when the number of end-devices is fixed to 1000.

TABLE 3. Settings for the evaluation of the schedule based LoRAWAN.

ToA_{max} [43]	2.793s
T_{ack2} [43]	1.8s
Synchronization period S_t	24h
End-devices internal clock accuracy A_c	10ppm
End-devices number N	1000/900/800/700/600/500
Voltage U	3.3 V
I_{tx}	28mA
I_{rx}	11.2mA
I_{sleep}	0.1µA

A. LATENCY AND ENERGY-CONSUMPTION AS FUNCTIONS OF THE NUMBER OF END-DEVICES

Fig. 7 shows that the optimized version of LoRaWAN using our scheduling schema presents an outstanding improvement of the network performance in terms of latency and energy consumption, compared to the original LoRaWAN class A or class B. For example, the uplink communication latency and energy consumption are reduced respectively by 89% and 78%, compared to the original LoRaWAN legacy class A. Indeed, when using our scheduling scheme each end-device has specific resources (channel frequency, SF and time slot) to transmit uplink packets to the LoRa Gateway without any risk of collision. Moreover, radio links are supposed of perfect quality (refer to Section V-A assumption 4). Hence, there is no packet losses because of signal attenuation. Therefore all packets are successfully received from the first transmission attempt which conserves energy and reduces latency. On the other hand in LoRaWAN class A or B, uplink transmission is AlOHA-based and thus the probability of collision increases exponentially with the increase of the network size. This what explains the shape of latency and energy consumption curves. Further, collisions lead to excessive packet retransmissions till the reception of an acknowledgment especially at large scale (1000 nodes). This represents the cost of attempting reliable uplink communications in LoRaWAN, as confirmed in [45]. This cost is traduced by the increase of latency and energy consumption.

It is important to notice that the optimized version of LoRaWAN using the proposed scheduling schema is definitely more scalable than the original LoRaWAN legacy as network performance are not significantly affected by the increase of network scale. In what concerns latency, the increase of the number of end-devices leads to the increase of the number of groups n_g derived in the pre-configuration phase (refer to Section IV), and consequently the increase of the number of uplink time slots *UpSlotNb*, and the increase of the uplink beacon period, which in turns lead to a moderated increase of latency. As for energy consumption, more the number of end-devices increases, more the uplink beacon period increases leading to a reduced number of beacon messages received during the synchronization period S_t. Thus, the end-device will spend more time in a sleep mode. That's why we have this slight decrease in energy-consumption curve but overall the energy consumption curve also presents a good scalability potential. In fact, what makes the difference between our proposal and the original LoRAWAN is retransmissions which increase exponentially with the increase of the network size in the case of LoRaWAN legacy, and kept null or minimal in the case of optimized LoRaWAN using our scheduling schema; due the deterministic and collision-free channel access.

Fig. 7 also shows that LoRaWAN class A outperforms LoRaWAN class B in terms of latency and energy consumption despite the fact that both use an ALOHA-based channel access technique for uplink transmissions. This is due to the following: LoRaWAN class B is designed to improve downlink (not uplink) latency by the allocation of ping slots for downlink communications (e.g., MAC commands transmission). However, this feature constraints uplink communications. For example, the acknowledgments of uplink packets cannot be transmitted by the gateway when a beacon is already in transmission or during the allocated ping slots for MAC command transmission, which leads to additional latency. Further, an additional energy is consumed by the end-device for the reception of periodic beacons.

B. LATENCY AND ENERGY-CONSUMPTION AS A FUNCTION OF T_{APP}

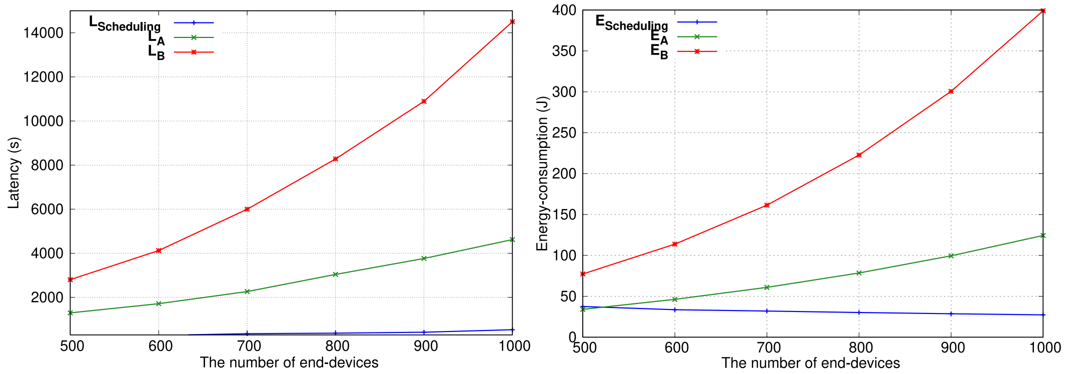
Three main observations can be retained from Fig. 8(a) and Fig. 8(b):

First, the optimized LoRaWAN using our scheduling schema outperforms significantly the original LoRaWAN class A and class B. Again, this outperformance is justified by the deterministic and collision-free channel access provided by the scheduling schema.

Second, the latency and the energy consumption of the optimized LoRaWAN is almost insensitive to the increase of T_{app}, i.e., the periodicity of the application layer packet transmission. This is due to the following: Latency is mainly affected by the number of groups, and energy consumption is mainly affected by the number and the duration of beacon periods. When increasing T_{app} with a fixed number of end-devices (1000 end-devices), all these parameters remains constant; that is why the decrease of latency and energy consumption in case of our schedule-based scheme is not noticeable in Figures 8(a) and 8(b). In what concerns LoRaWAN legacy, the traffic rate decreases with the increase of T_{app} which reduces the probability of collisions. Consequently, the performance of LoRaWAN legacy increases when the traffic rate decreases.

Third, when T_{app} is higher than $5500s$, i.e., the application layer generates a packet each $/1.5h$, LoRaWAN class A provides comparable or slightly better network performance as the optimized LoRAWAN. This is due to the fact that the collisions are considerably avoided when $T_{app} \geq 1.5h$, despite that the number of end-devices is still high (1000).

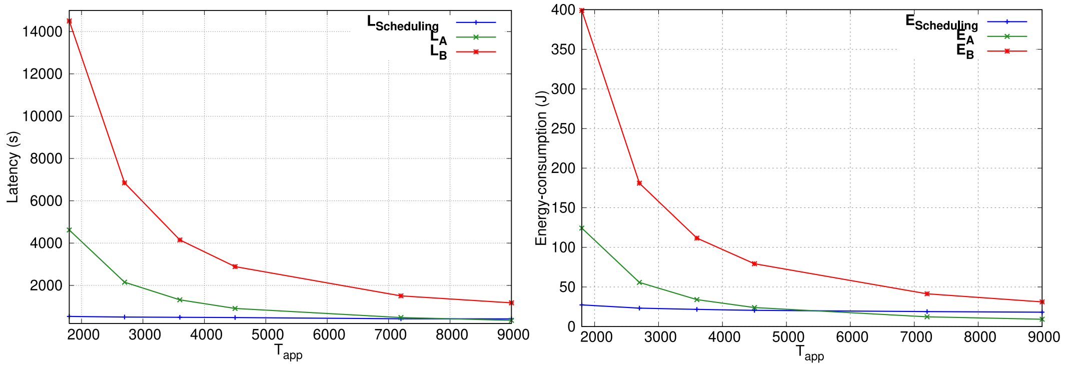
Consequently, our schedule based-scheme of LoRaWAN is efficient for IoT applications deployed in large scale network and having a packet transmission periodicity less than 1 packet$/1.5h$. We can find a variety of IoT applications where the packet transmission periodicity is lower than $1.5h$ [44], including traffic management (one message per 10 minutes),



(a) Latency evaluation as a function of number of the end-devices.

(b) Energy-consumption evaluation as a function of the number of end-devices.

FIGURE 7. **Latency and energy consumption as a function of network size.**



(a) Latency evaluation as a function of T_{app}

(b) Energy-consumption evaluation as a function of T_{app}.

FIGURE 8. **Latency and energy consumption as a function of T_{app}.**

air pollution monitoring, urban conditions monitoring, structural health monitoring and smart metering (1 message per 15 minutes), and wildlife tracking(1 message per 30 minutes).

VII. CONCLUSION

In this paper, we introduced a new schedule-based schema for reliable uplink communications in large scale LoRaWAN. In contrast of most existing studies in the literature, our schema does not involve any additional synchronization phase for end-devices, or major changes in LoRaWAN specification. We assume that the network is operating under perfect SFs orthogonality conditions.

Our scheduling schema consists in two phases. In the pre-configuration phase the network end-devices are organized into groups where a group contains a maximum of 4 end-devices having disjoint SFs. Based on the over-the-air activation procedure, each end-device is configured with a group index and a suitable SF depending on its location. Then, we make end-devices switch to class B to take profit from the existing synchronization using downlink beacon messages. Hence, time is structured by an alternation of uplink/downlink beacon periods having the same structure but different timing. The uplink beacon period is dedicated to schedule uplink communication and receive acknowledgement in case of confirmed traffic. Downlink beacon period used to schedule downlink traffic (MAC commands) is kept unchanged as in LoRaWAN class B specification.

Results based on analytic modeling showed an outstanding improvement in terms of latency and energy consumption, brought by our schedule-based schema compared to the LoRaWAN legacy class A and B. When the traffic transmission rate decreases (one packet per 1.5h), the LoRaWAN legacy class A starts to conquer the performance of the schedule-based mechanism.

As a future work, we plan to evaluate our scheme using ns-3 simulation and real-world deployment.

REFERENCES

[1] F. Gu, J. Niu, L. Jiang, X. Liu, and M. Atiquzzaman, "Survey of the low power wide area network technologies," *J. Netw. Comput. Appl.*, vol. 149, Jan. 2020, Art. no. 102459.

[2] M. L. Liya and D. Arjun, "A survey of LPWAN technology in agricultural field," in *Proc. 4th Int. Conf. I-SMAC (IoT Soc. Mobile Anal. Cloud) (I-SMAC)*, Palladam, India, 2020, pp. 313–317.

[3] Y. Wang, Y. Huang, and C. Song, "A new smart sensing system using LoRaWAN for environmental monitoring," in *Proc. Comput. Commun. IoT Appl. (ComComAp)*, Shenzhen, China, 2019, pp. 347–351.

[4] J. Råžička, M. Sliacky, Z. Purkrábková, and E. Hajčiarová, "Opportunities of LoRaWAN technology for smart cities–A review," in *Proc. Smart City Symp. Prague (SCSP)*, Prague, Czech Republic, 2023, pp. 1–6.

[5] M. Abbasi, S. Khorasanian, and M. H. Yaghmaee, "Low-power wide area network (LPWAN) for smart grid: An in-depth study on LoRaWAN," in *Proc. 5th Conf. Knowl. Based Eng. Innov. (KBEI)*, Tehran, Iran, 2019, pp. 022–029.

[6] *LoRaWAN Specification v1.0.3.*, LoRa Alliance, Inc., Beaverton, OR, USA, 2017, pp. 1–72.

[7] N. Poursafar, M. E. E. Alahi, and S. Mukhopadhyay, "Long-range wireless technologies for IoT applications: A review," in *Proc. 11th Int. Conf. Sensing Technology (ICST)*, 2017, pp. 1–6.

[8] H. Wymeersch and A. Eryilmaz, "Chapter 12–Multiple access control in wireless networks," in *Academic Press Library in Mobile and Wireless Communications*, S. K. Wilson, S. Wilson, and E. Biglieri, Eds. New York, FL, USA: Academic, 2016, pp. 435–465.

[9] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *Proc. Int. Conf. Eng. Telecommun. (EnT)*, 2016, pp. 10–14.

[10] A. Farhad, D.-H. Kim, and J.-Y. Pyun, "Scalability of LoRaWAN in an urban environment: A simulation study," in *Proc. 11th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2019, pp. 677–681.

[11] G. Ferre, "Collision and packet loss analysis in a LoRaWAN network," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, 2017, pp. 2586–2590.

[12] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *Proc. Symp. Commun. Veh. Technol. (SCVT)*, Mons, Belgium, 2016, pp. 1–5.

[13] "LoRa Alliance." [Online]. Available: https://lora-alliance.org/

[14] *RP002-1.0.0 LoRaWAN Regional Parameters*, LoRa Alliance, Inc., Beaverton, OR, USA, Nov. 2019, pp. 1–88.

[15] P. Mahajan and S. Abhishek, "A study of encryption algorithms AES, DES and RSA for security," *Global J. Comput. Sci. Technol.*, vol. 13, no. 15, pp. 1–9, 2013.

[16] K.-H. Phung, H. Tran, Q. Nguyen, T. T. Huong, and T.-L. Nguyen, "Analysis and assessment of LoRaWAN," in *Proc. 2nd Int. Conf. Recent Adv. Signal Process. Telecommun. Comput. (SigTelCom)*, 2018, pp. 241–246.

[17] M. Capuzzo, D. Magrin, and A. Zanella, "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures," in *Proc. 17th Annu. Mediterranean Ad Hoc Netw. Workshop (Med-Hoc-Net)*, 2018, pp. 1–7.

[18] B. Paul, "A novel mathematical model to evaluate the impact of packet retransmissions in LoRaWAN," *IEEE Sensors Lett.*, vol. 4, no. 5, pp. 1–4, May 2020.

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

[20] T. Polonelli, D. Brunelli, and L. Benini, "Slotted ALOHA overlay on LoRaWAN-A distributed synchronization approach," in *Proc. IEEE 16th Int. Conf. Embedded Ubiquitous Comput. (EUC)*, 2018, pp. 129–132.

[21] G. Yapar, T. Tugcu, and O. Ermis, "Time-slotted ALOHA-based LoRaWAN scheduling with aggregated acknowledgement approach," in *Proc. 25th Conf. Open Innov. Assoc. (FRUCT)*, 2019, pp. 383–390.

[22] L. Beltramelli, A. Mahmood, P. Österberg, M. Gidlund, P. Ferrari, and E. Sisinni, "Energy efficiency of slotted LoRaWAN communication with out-of-band synchronization," *IEEE Trans. Instrument. Meas.*, vol. 70, pp. 1–11, Jan. 2021.

[23] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for LoRa," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–13.

[24] T.-H. To and A. Duda, "Simulation of LoRa in NS-3: Improving lora performance with CSMA," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.

[25] N. Kouvelas, V. Rao, and R. V. Prasad, "Employing p-CSMA on a LoRa network simulator," 2018, *arXiv:1805.12263*.

[26] S. Ahsan, S. A. Hassan, A. Adeel, and H. K. Qureshi, "Improving channel utilization of LoRaWAN by using novel channel access mechanism," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2019, pp. 1656–1661.

[27] A. Loubany, S. Lahoud, and R. El Chall, "Adaptive algorithm for spreading factor selection in LoRaWAN networks with multiple gateways," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107491.

[28] N. Chinchilla-Romero, J. Navarro-Ortiz, P. Muñoz, and P. Ameigeiras, "Collision avoidance resource allocation for LoRaWAN," *Sensors*, vol. 21, no. 4, p. 1218, 2021.

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

[30] T.-H. To and A. Duda, "Timemaps for improving performance of LoRaWAN," in *Proc. IEEE Int. Conf. Communications (ICC)*, 2020, pp. 1–7.

[31] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3097–3109, Apr. 2019.

[32] D. Zorbas and B. O'Flynn, "Autonomous collision-free scheduling for LoRa-based industrial Internet of Things," in *Proc. IEEE 20th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2019, pp. 1–5.

[33] D. M. Ibrahim, "Improving LoRaWAN performance using reservation ALOHA," *J. Inf. Technol. Manag.*, vol. 12, no. 2, pp. 70–78, 2020.

[34] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the industrial Internet of Things," *Comput. Commun.* vol. 153, pp. 1–10, Mar. 2020.

[35] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "FREE–Fine-grained scheduling for reliable and energy efficient data collection in LoRaWAN," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 669–683, Jan. 2020.

[36] A. Triantafyllou, P. Sarigiannidis, T. Lagkas, I. D. Moscholios, and A. Sarigiannidis, "Leveraging fairness in LoRaWAN: A novel scheduling scheme for collision avoidance," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107735.

[37] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-demand TDMA for energy efficient data collection with LoRa and wake-up receiver," in *Proc. 14th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2018, pp. 1–4.

[38] H. Mroue et al., "LoRa+: An extension of LoRaWAN protocol to reduce infrastructure costs by improving the quality of service," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100176.

[39] L. Leonardi, F. Battaglia, and L. Lo Bello, "RT-LoRa: A medium access strategy to support real-time flows over LoRa-based networks for industrial IoT applications," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10812–10823, Dec. 2019.

[40] J. Moraes et al. "An efficient heuristic LoRaWAN adaptive resource allocation for IoT applications," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2020, pp. 1–6.

[41] "Wireless and sensing products." 2017. [Online]. Available: https://www.mouser.com/datasheet/2/761/sx1301-1523429.pdf

[42] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, "LoRa throughput analysis with imperfect spreading factor orthogonality," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 408–411, Apr. 2019.

[43] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *Proc. Eur. Wireless; 22th Eur. Wireless Conf. VDE*, 2016, pp. 1–6.

[44] J. Mocnej, A. Pekar, W. K. G. Seah, and I. Zolotova, *Network Traffic Characteristics of the IoT Application Use Cases*, School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand, pp. 1–20, 2018.

[45] C. E. Fehri, N. Baccour, and I. Kammoun, "The cost of reliable uplink communication in large scale LoRaWAN," in *Proc. 19th IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, 2022, pp. 1–8.

CHÉKRA EL FEHRI received the Engineer Diploma degree from the National engineering School of Sfax (ENIS), Tunisia, in 2013, and the Ph.D. degree in computer systems engineering from ENIS in 2022. Her research interests consist in evaluating and boosting the long range wide area network (LoRaWAN). She works on improving the mac layer aspect by exploring the possible scheduling and resource allocation techniques that may be applied in a large scale LoRaWAN.

NOUHA BACCOUR received the engineering, master's, and Ph.D. degrees in computer science from the National Engineering School of Sfax, University of Sfax, Tunisia, in 2005, 2006, and 2012, respectively. She is currently an Assistant Professor with the National Engineering School of Sfax and a Senior Researcher with the ReDCAD Research Unit, University of Sfax. She has several publications in reputed conferences and journals, such as *ACM Transactions on Sensors Networks* with more than 1600 citations. She is also a coauthor of a book: "*Radio Link Quality Estimation in Low-Power Wireless Networks*" (Springer 2013). Her research interests are wireless sensor networks, communication protocols, link quality estimation, Internet of Things, and smart grid.

INÈS KAMMOUN (Senior Member, IEEE) received the Engineer Diploma degree from the Ecole Polytechnique de Tunis, Tunisia, in 1999, and the master's and Ph.D. degrees from the Telecom ParisTech (National School of Engineering in Telecommunications), Paris, France, in 2000 and 2004, respectively. From 2004 to 2008, she was an Assistant Professor with Enet'Com (Ex ISECS), Sfax, Tunisia. In 2008, she joined the National Engineering School of Sfax (ENIS), where she is currently a Full Professor. She also has been the Head of the Research Laboratory LETI, ENIS. Her research interests are in the areas of digital and wireless communications with special emphasis on MIMO and massive MIMO systems, channel estimation, relaying, wireless sensor networks, cooperative networks, and cognitive radio.