Improving Delay and Capacity of TS-LoRa with Flexible Guard Times

Dimitrios Zorbas
Tyndall National Institute, University College Cork, Ireland
dimzorbas@ieee.org

***Abstract*—TS-LoRa is an autonomous time-slotted communication protocol for LoRa-based networks. As with other synchronous protocols, TS-LoRa bases its operation on repeated frames where each frame consists of a number of slots while guard times are added in between successive transmissions to tolerate slight desynchronisations until the next synchronisation. The number of slots in the frames depends on the payload size, the radio duty cycle rules, the application duty cycle requirements, and the number of nodes in the network. Due to the low bitrate of LoRa transmissions, the guard times are much longer compared to other traditional time-division protocols. Those longer guard times lead to larger in size frames and, thus, to increased delay and lower capacity. In this paper, a delay and capacity analysis of TS-LoRa is provided and the possibility of using flexible guard times to mitigate the problem of the increased delay and the reduced network capacity is explored. The potential of the solution to improve the above-mentioned performance metrics is shown through numerical simulations while the practicality of the flexible guard times is assessed through testbed experiments.**

I. INTRODUCTION

The fourth industrial revolution envisions the reduction of costs, the increase of operation speeds, the increase of operational uptime, and the increase of the level of safety. To achieve all these, Industry 4.0 relies on the design of reliable and cost-effective Internet of Things (IoT) networking solutions that can achieve high reliability, low-latency, low power consumption, and low maintenance [1].

Existing wireless IoT protocols in industry such as the BLE, the WirelessHART, and the ISA 100.11a, are short to medium range radio technologies and, thus, they require an extensive number of relays and gateways to be deployed increasing the deployment cost. Apart from that, these technologies can barely support mobility in a large deployment area.

In contrast with the current solutions, a long range technology such as LoRa can tackle the problem of limited mobility as well as of the installation cost while exhibiting a similar energy consumption. Even though LoRa cannot achieve high bitrates and its transmissions must obey radio duty cycle rules (for the sub-GHz ISM bands), it can still be used for low-bitrate industrial monitoring applications, such as predictive maintenance, asset tracking, and smart grid systems [2]. However, LoRaWAN which is the current LoRa-based standard, is designed for battery longevity, interoperability between devices with different QoS needs, and deployment simplicity. Its MAC layer is Aloha-based and, thus, it cannot guarantee packet delivery even with a moderate network traffic [3]. Moreover, LoRaWAN can get heavily congested and can lead to heavy waste of energy when a high number of packets needs to be acknowledged in a short amount of time [4].

TS-LoRa [5] has been proposed as a LoRa-based candidate for industrial applications. It uses time-slotted communications and collision-free transmissions between nodes of the same network. It also adopts a novel mechanism to achieve synchronisation and acknowledgements with a single downlink transmission. TS-LoRa in its current form, wastes a lot of time resources (i.e., slots) in guard times used to tolerate clock desynchronisations. Hence, in this paper, an analysis of the current delay and capacity capabilities of TS-LoRa is presented along with an improved version of guard time arrangement. It is shown that using this new arrangement of guard times and 16 bytes of payloads, the frame capacity can be improved by up to 29%. Moreover, experimental results on a 8-node testbed show that the new method can achieve similar to the traditional TS-LoRa results in terms of packet delivery ratio.

II. LORA AND LORAWAN

LoRa is a proprietary spread spectrum modulation (SSM) technology currently owned by Semtech [6]. LoRa can trade data rate with sensitivity by adjusting the amount of spread in the SSM. The spread is controlled using a radio parameter, called Spreading Factor (SF) which in sub-GHz bands ranges from 7 to 12. Assuming a fixed channel bandwidth (BW) and payload, the higher the SF, the higher the sensitivity and, thus, the longer the transmission range. Moreover, the higher the SF, the lower the data rate and, thus, the longer transmission time. In addition, parallel transmissions performed on different SFs can be simultaneously decoded by the gateway. LoRa mainly uses license-free sub-gigahertz radio frequency bands (e.g., EU868, US915) that are restricted to radio duty cycle regulations. For example, in EU the nodes are allowed to transmit only for 1% of the time.

LoRaWAN is currently the only open LoRa-based protocol. It is proposed and maintained by the LoRa Alliance, a non-profit association consisting of Semtech as well as other companies and universities from across the world. LoRaWAN supports a number of features such as device registration, acknowledgments, end-to-end encryption, synchronisation for specific applications, and localisation services [7]. Moreover, LoRaWAN distinguishes three classes of devices. The majority of the nodes belong to the first class (class A). Class A devices follow an Aloha-based medium access mechanism to transmit packets. They can optionally wait for an acknowledgement using two predefined time windows. On the contrary, Class B devices perform some kind of synchronisation but only for the dowlink activity. The purpose of this synchronisation is to have those devices ready for long downlink periods (e.g., firmware updates). However, uplink transmissions in Class B are still Aloha-based. Finally, devices that belong to Class C, are always available for downlink data.

III. DELAY & CAPACITY ANALYSIS OF TS-LORA

A. TS-LoRa Frame Structure

TS-LoRa follows a frame structure as it is depicted in Fig. 1. The transmissions are organised in sequential slots, while guard times are added before and after each transmission. These guard times are fixed for all the nodes and their length depends on the frame size. An additional slot is used at the end of the frame for synchronisation and acknowledgements. In this slot, the gateway transmits a packet called "SACK" to synchronise the nodes and acknowledge transmissions performed in the current frame. This is done using a series of zeros and ones whose order and size correspond to the number of nodes in the frame. An "1" indicates successful transmissions and a "0" indicates non-delivered packets. During the SACK transmissions, all the nodes of the same SF have their radio on. If a packet was not successfully delivered in the current frame, its transmission is repeated in the next frame. The same is done when a SACK is not received from a node.

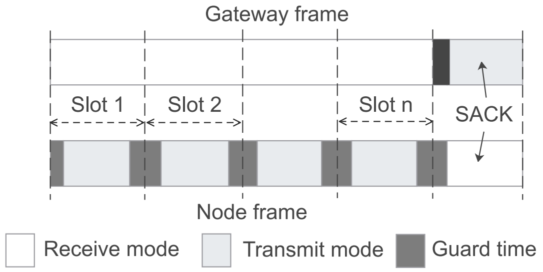


Fig. 1. Frame structure of TS-LoRa for the nodes and the gateway.

The decision of how long a frame should last, depends on the application requirements. TS-LoRa supports two types of applications as well as any combination between them. In first type, data delivery is not considered as a critical process and, thus, it can be postponed in order to accommodate more nodes in the frame. For example, the data periodicity of a temperature monitoring system of a building can be extended from every 5 to every 8 minutes in order to support more devices without sacrificing a lot of its accuracy. In this case, the frame size is flexible and may expand as more nodes are added into it. However, the delay starts increasing after a certain point as the frame expands. This point depends on the radio duty cycle restrictions, the payload size, and the LoRa radio characteristics (e.g., the SF). If a node has data to transmit, it has to wait until its allocated slot in the forthcoming frame. In the second strategy, the frame size is defined by strict application duty cycle requirements. For example, an alert has to be transmitted within a certain amount of time once an event has been detected. Given this strict requirement, the delay is kept constant; however, the capacity gets limited due to the fixed frame size.

TABLE I

NOTATIONS AND THEIR MEANING.

Notation	Meaning
f	Spreading Factor (SF)
n_f	Number of nodes in the frame of SF f
$\mathcal{T}_f$	Transmission time for SF f
g	Guard time (for the fixed guard time scenario)
g_i	Guard time associated with slot i
$\mathcal{C}_f$	Capacity of the frame with SF f
$\mathcal{D}_f$	Downlink packet transmission time for SF f
$\mathcal{P}$	Gateway processing time per occupied slot
$\mathcal{L}$	Application delay requirements
$\mathcal{F}_f$	Frame size for the frame with SF f

B. Analysis

In this subsection, the delay and capacity analysis of the network for the two aforementioned scenarios is presented. Table I summarises the notations used in the rest of paper and their meaning.

1) Time-Flexible Applications: When no time-critical constraints exist, the frame size can be expanded to support more nodes. As it has previously been explained [8], [5], the frame size depends on the radio duty cycle or how many nodes have the same SF. In the first case, if only a few nodes exist, the frame is filled with empty slots until the radio duty cycle rules are satisfied. If we assume that T_f is the data transmission time for each SF $f \in [7, 12]$ [9] and g is the guard time, then the slot length is equal to $T_f + 2g$. Moreover, let $\mathcal{D}_f$ denote the SACK slot length, and n_f denote the number of accommodated nodes in the frame. Since 1 bit of information is used per node for acknowledgments and another 8-10 bytes for overhead (to indicate the gateway id, the frame size, and the processing time), the total payload size of a SACK packet is $\lceil \frac{n_f}{8} \rceil + 8$. The transmission time of the uplink and downlink packets are computed by the function $airtime()$ as it is detailed in [9]. $airtime()$ takes two arguments, the SF and the payload size, while other parameters such as the channel bandwidth and the coding rate are set equal for all the nodes. Furthermore, it has been experimentally found that the gateway needs some time to process and prepare the SACK packet before transmission [5]. This extra time depends on the processing capabilities of the gateway and the number of nodes in the frame. We set $\mathcal{P}$ the processing time per node. Taking into account the previous parameters and assuming an 1% radio duty cycle, the frame size – denoted by $\mathcal{F}_f$ – is computed by Eq. (1).

Moreover, g depends on the maximum inaccuracy of the crystal clock as well as on the crystal age. A typical crystal has a maximum clock drift of 100 ppm (or 100 μsec per second). This means that for a frame size equal to $\mathcal{F}$ seconds, the guard time should roughly be equal to $100\mu\mathcal{F}$ seconds. The problem here is that g appears in the $\mathcal{F}$ equation and at the same time it depends on $\mathcal{F}$. This practically means

$$\mathcal{F}_f = \begin{cases} (T_f + 2g)\left\lceil \dfrac{100T_f - \mathcal{D}_f - \mathcal{P}\mathcal{C}_f}{T_f + 2g} \right\rceil + \mathcal{D}_f + \mathcal{P}\mathcal{C}_f, & \text{if } n_f(T_f + 2g) + \mathcal{D}_f + \mathcal{P}\mathcal{C}_f \leq 100T_f, \\ \\ n_f(T_f + 2g) + \mathcal{D}_f + \mathcal{P}\mathcal{C}_f, & \text{if } n_f(T_f + 2g) + \mathcal{D}_f + \mathcal{P}\mathcal{C}_f > 100T_f. \end{cases} \tag{1}$$

that $\mathcal{F}$ cannot be computed analytically but only numerically. The issue here is that the nodes cannot do this computation by their own since it may be a time and energy consuming process. So the problem is translated to a problem of how the gateway can let the nodes know about this guard time and how the nodes get informed about potential changes of the guard time value over time as new nodes may be added in the frame. Another issue is that if a node does not receive a SACK packet, it will not be perfectly synchronised when it will wake-up for the next transmission and, thus, it may cause a collision by overlapping with one of the adjacent slots. TS-LoRa allows up to 2 re-transmissions per packet which means that the guard time must be long enough to tolerate clock desynchronisations as long as 3 frames. A simple solution to those problems is to use a fixed guard time value based on a maximum frame length (e.g., a frame length consisting of 1000 slots). Another solution is to use fixed guard time values for predefined frame ranges. Those values and ranges can be flashed in the nodes non-volatile memory. Both solutions are very efficient in terms of software development effort as well as in terms of computational complexity. The problem is that a lot of time is wasted in guard times and, thus, the capacity may decrease considerably. TS-LoRa follows another approach. The guard time is computed at the gateway (or at the network server) and it is communicated to the nodes through the downlink SACK packet. This requires only 4 additional bytes to be sent, thus, the frame size is only slightly affected.

The delay (denoted with $\mathcal{L}$) in the case of the non-time-critical scenario is equal to the frame size $\mathcal{F}_f$ given the SF and the number of nodes n_f. The delay is constant when only a few nodes are accommodated in the frame as it is imposed by the duty cycle rules but increases with higher frame sizes. Any additional slot, once all the empty slots have been filled with transmissions, increases the delay by $\mathcal{S}_f$ amount of time, plus some extra time due to the longer SACK slot.

2) Time-Critical Applications: In time-critical applications, the frame size is fixed as it is dictated by the delay constraints $\mathcal{L}$. As a consequence, the number of accommodated devices is limited. The guard time can be easily calculated using the maximum allowed frame size for three consecutive frames (i.e., $g = 3 \cdot 10^{-4} \cdot \mathcal{L}$). In that case the capacity of the frame can be computed by Algorithm 1.

According to the regional radio duty cycle rules in the EU, a LoRa node can transmit in total 3.6 to 360 seconds within one hour depending on the selected frequency. In TS-LoRa, in order to better handle the flow of transmissions, a node has to wait for at least 99 times the duration of the last transmission in order to be allowed to transmit again. This means that TS-LoRa cannot support applications whose delay requirements are higher than the data periodicity imposed by the radio duty cycle rules (see lines 1–3). Apparently, this restriction diminishes in 2.4GHz LoRa and, thus, almost any application can be supported.

Algorithm 1: Frame capacity computation with fixed guard times

```
   require: $f, \mathcal{L}, payload, \mathcal{P}, airtime()$
1  $T_f = airtime(f, payload)$;
2  if $\mathcal{L} < 100T_f$ then
3      return 0;
4  end
5  $g = 3 \cdot 10^{-4} \cdot \mathcal{L}$;
6  $\mathcal{C}_f = \lfloor \frac{\mathcal{L}}{T_f + 2g} \rfloor$;
7  $\mathcal{D}_f = airtime(f, \lceil \frac{\mathcal{C}_f}{8} \rceil + 8)$;
8  $\mathcal{F}_f = \mathcal{C}_f \cdot (T_f + 2g) + \mathcal{D}_f + \mathcal{P} \cdot \mathcal{C}_f$;
9  while $\mathcal{F}_f > \mathcal{L}$ do
10     $\mathcal{C}_f = \mathcal{C}_f - 1$;
11     $\mathcal{D}_f = airtime(f, \lceil \frac{\mathcal{C}_f}{8} \rceil + 8)$;
12     $\mathcal{F}_f = \mathcal{C}_f \cdot (T_f + 2g) + \mathcal{D}_f + \mathcal{P} \cdot \mathcal{C}_f$;
13 end
14 return $\mathcal{C}_f$;
```

IV. EXPLORING FLEXIBLE GUARD TIMES

In this section, the possibility of using flexible guard times in order to reduce the total reserved time for guard times and, thus, to increase the frame capacity is explored. Due to the economy of space, the study is focused on the second scenario, where the frame size is fixed, however the approach can easily be extended to the first scenario as well.

In TS-LoRa, all the nodes in the frame get synchronised at the same time but each individual transmission is performed at a different time. This means that each slot has different maximum clock drift times that depend on the position of the slot in the frame. For example, the first transmission after the synchronisation is performed only milliseconds later, while the last transmission may happen several seconds or minutes later. Hence, the idea is to use guard times which gradually increase as we move to larger slot numbers. This solution can save some of the wasted time of the reserved guard times, freeing some space for extra data slots. The new frame structure is illustrated in Fig. 2. The frame capacity under the new structure can be computed numerically by Algorithm 2.

Flexible guard times save significant amount of time in the first frame. However, since clock drift tolerances of two additional frames have to be considered, the total savings are finally reduced. Nevertheless, an advantage of this method is that each node can compute its guard time individually and,

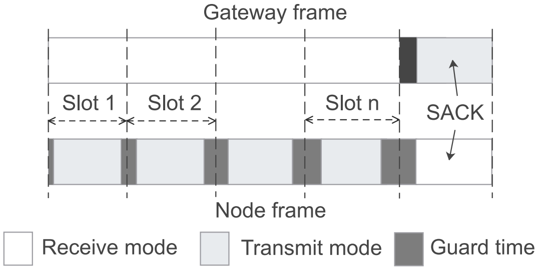


Fig. 2. The proposed frame structure.

Algorithm 2: Frame capacity computation with flexible guard times

```
require: $f, \mathcal{L}, payload, \mathcal{P}, airtime(), g_0$
$T_f = airtime(f, payload)$;
if $\mathcal{L} < 100T_f$ then
    return 0;
end
$\mathcal{C}_f = 1$;
$g = g_0$;
$\mathcal{D}_f = airtime(f, \lceil \frac{\mathcal{C}_f}{8} \rceil + 8)$;
$\mathcal{F}_f = T_f + 2g + \mathcal{D}_f + \mathcal{P} \cdot \mathcal{C}_f$;
$prevF = \mathcal{F}_f$;
while $\mathcal{F}_f < \mathcal{L}$ do
    $\mathcal{C}_f = \mathcal{C}_f + 1$;
    $g = \mathcal{F}_f \cdot 10^{-4} + 2 \cdot 10^{-4} \cdot \mathcal{L}$;
    $\mathcal{D}_f = airtime(f, \lceil \frac{\mathcal{C}_f}{8} \rceil + 8)$;
    $\mathcal{F}_f = \mathcal{F}_f + T_f + 2g + \mathcal{D}_f + \mathcal{P} \cdot \mathcal{C}_f$;
    if $\mathcal{F}_f > \mathcal{L}$ then
        $\mathcal{C}_f = \mathcal{C}_f - 1$;
        $\mathcal{F}_f = prevF$;
        break;
    end
    else
        $prevF = \mathcal{F}_f$;
    end
end
return $\mathcal{C}_f$;
```

thus, no additional bytes need to be sent in the SACK packet. Let g_i denote the guard time associated with slot i, where $i \geq 1$. g_i can be computed by the following formula:

$$g_i = \left(T_f + 2g_0 + \sum_{j=2}^{i-1} [2(g_j + 10^{-4}\mathcal{L}) + T_f] \right) 10^{-4} + 2 \cdot 10^{-4}\mathcal{L}, \tag{2}$$

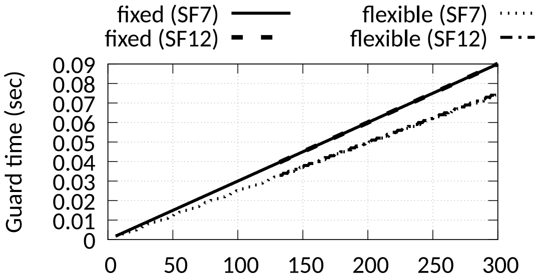
where g_0 is the (fixed) guard time of the first slot ($g_0 >> g_i$).

V. NUMERICAL RESULTS

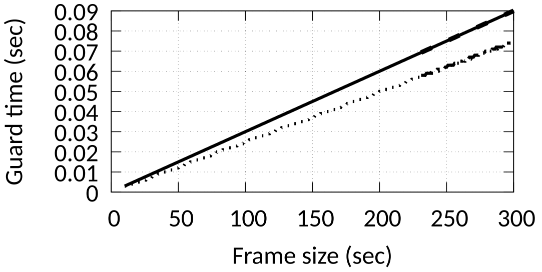
In this section, numerical results of the frame capacity are presented. Due to the economy of space, the results are focused on the second application scenario described in Section III-B. A comparison between the default TS-LoRa frame structure and the one that bases its operation on flexible guard times is made. Table II summarises the evaluation parameters.

TABLE II
EVALUATION PARAMETERS

Parameter	Value
Spreading Factor (SF)	7–12
Channel Bandwidth	125 KHz
Preamble Symbols	8
Coding Rate	4/5
Packet size (*payload*)	16 or 48 Bytes
Radio duty cycle	$\leq$1%
Guard time for the 1st slot (g_0)	5 ms
Minimum guard time for the other slots	1μs
Processing time per node ($\mathcal{P}$)	1 ms
Maximum clock drift per second	100μs



(a) 16Bytes payload



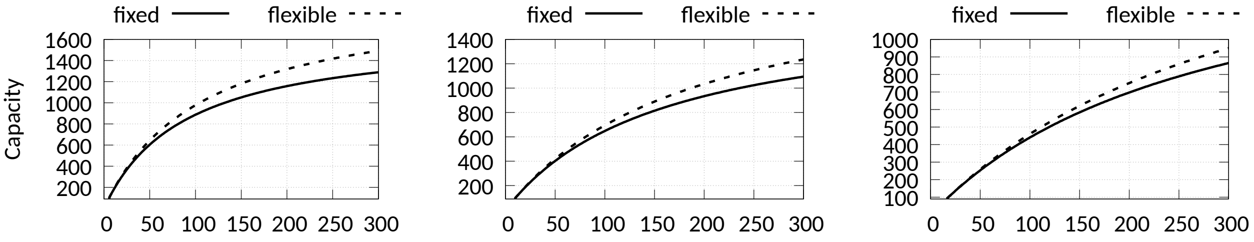
(b) 48Bytes payload

Fig. 3. Guard time for different delay requirements and payloads.

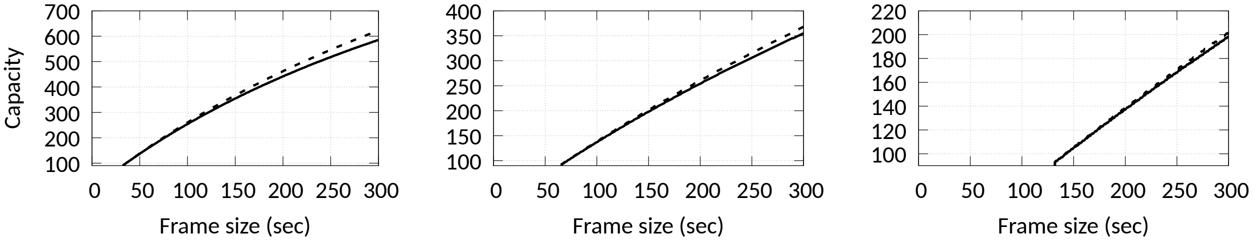
Fig. 3 depicts the average guard time for different frame sizes and different payloads. The two extreme SF cases are presented. The results show an over 25% improvement for SF7 and 20% improvement for SF12 by using the flexible guard time scheme. We must note that due to the radio duty cycle limitation, some frame sizes can not be supported by all SFs. This is the reason SF12 lines seem to start later in the plot.

Fig. 4 shows the maximum frame capacity for all SFs for a scenario with 16 Bytes payload. The capacity can be improved by up to 29% for SF7, 18% for SF8, 13% for SF9, 8% for SF10, 5% for SF11, and 2% for SF12 with flexible guard times. In general, the more available slots in the frame, the more the improvement. Thus, the improvement is lower in high-SF frames since a fewer slots can be allocated. Assuming orthogonality of transmissions over different SFs, the new scheme can provide more than 800 new slots for the examined payload size.

Finally, Fig. 5 compares TS-LoRa capacity with flexible guard times to the theoretical optimal frame capacity. In the latter case, we measure the upper bound in terms of capacity assuming that the nodes are perfectly synchronised such as no guard times are needed. The results show that the gap is

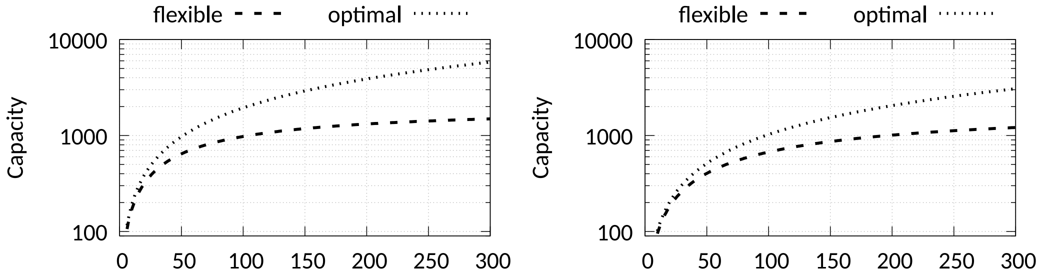


(a) Spreading Factor 7 (b) Spreading Factor 8 (c) Spreading Factor 9

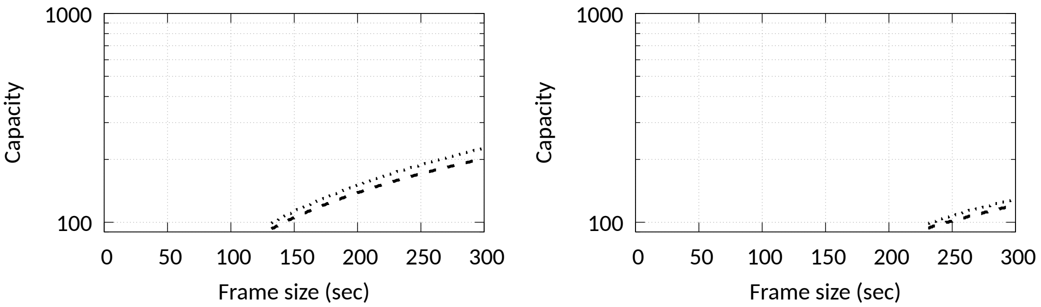


(d) Spreading Factor 10 (e) Spreading Factor 11 (f) Spreading Factor 12

Fig. 4. Frame capacity (in slots) for different spreading factors and delay requirements (16Bytes payload).



(a) 16 Bytes payload, Spreading Factor 7 (b) 48 Bytes payload, Spreading Factor 7



(c) 16 Bytes payload, Spreading Factor 12 (d) 48 Bytes payload, Spreading Factor 12

Fig. 5. TS-LoRa frame capacity (in slots) compared to the theoretically optimal capacity.

big for large frame sizes and low SFs. This practically means that more and more time resources are spent for guard times as the frame size increases. However, the gap is smaller for the larger payload scenario since less time is wasted for guard times. A solution to the long guard times issue would be to have multiple synchronisation packets within a single frame, however, this would increase the energy consumption as the nodes should wake-up more often to get synchronised.

VI. PROOF OF CONCEPT

In this section, the TS-LoRa platform[1] is used to assess the flexible guard times method. The purpose of these experiments is to provide a proof of concept of the new guard time scheme and see whether this scheme can provide the same reliability with the default TS-LoRa approach. Due to the covid-19 pandemic and the prohibited access to the labs, the evaluation is restricted to 8 nodes and a smaller deployment area. However, in this series of experiments, the focus is on

[1] https://github.com/deltazita/ts-lora

TABLE III
EXPERIMENTAL PARAMETERS

Parameter	Value
Spreading Factor	7
Bandwidth (BW)	125 kHz
Preamble Symbols	8
Coding Rate	4/5
Frequency	EU868
Radio duty cycle	1% for data and SACKs
Data packet size	16 Bytes
Guard time	By Eq. (2)
Tx power (nodes)	14 dBm
Delay requirement	6 sec
Frame size	5.98 sec
Guard time for the 1st slot (g_0)	5 ms
Minimum guard time for other slots	2 ms
Nodes (n_7)	8
Packets sent per node	1000
Number of runs	10

the overlap between successive slots while the overall capacity will be evaluated in a future study since more than 100 nodes are required in that case. For convenience, all nodes in these experiments use SF7. Moreover, the minimum guard time was set equal to 2 ms. This is because the LoRa module in the specific tested hardware requires 4.7 to 6.7 ms to switch from the sleep mode to the active mode and vice-versa. Thus, an extra 2 ms was imposed to tolerate this random radio wake-up time. Each experiment is repeated 10 times and the average results are presented along with the 95% confidence intervals. Table III summarises the parameters used in the experiments.

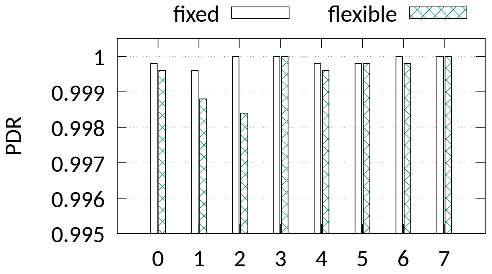
Fig. 6 depicts the average packet delivery ratio and the number of retransmissions for all the 8 slots. We can observe that there is no remarkable change in performance when switching to flexible guard times. What we can only see is that there is a higher number of retransmissions for some slots due to a firmware bug which causes unexpected clock jumps. This is something that will be investigated further in an extended version of the paper.

VII. CONCLUSIONS & FUTURE WORK

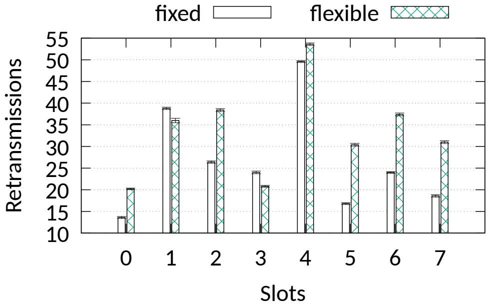
In this paper, a capacity and delay analysis of TS-LoRa, a time-slotted protocol for LoRa-enabled IoT devices was presented. An enhanced version of TS-LoRa which utilizes flexible guard times was also introduced and analysed. Theoretical results showed that this method can increase the frame capacity by up to 29%. Finally, the approach was implemented on a real testbed and was compared to the fixed guard times approach. The results did not show any considerable change in performance as the overall average packet delivery ratio was over 99.9%. However, some further investigation of the maximum drift times will be required as part of the future work. Moreover, additional frame design techniques with multiple synchronisation slots in the same frame will be explored.

ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077. It has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713567.



(a) Packet Delivery Ratio



(b) Retransmissions

Fig. 6. Experimental results with 8 nodes (slots 0–7).

REFERENCES

[1] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless networks*, vol. 23, no. 1, pp. 23–41, 2017.

[2] A. Mahmood, M. I. Ashraf, M. Gidlund, and J. Torsner, "Over-the-Air Time Synchronization for URLLC: Requirements, Challenges and Possible Enablers," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–6, IEEE, 2018.

[3] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '16, pp. 59–67, ACM, 2016.

[4] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[5] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things," *Computer Communications*, vol. 153, pp. 1 – 10, 2020.

[6] Semtech Corporation, "LoRa Modulation Basics." https://lora-developers.semtech.com/library/product-documents/, 2015. Online; accessed 11-May-2020.

[7] LoRa Alliance Technical Committee, "LoRaWAN$^{\text{TM}}$ 1.0.3 Specification." lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf, 2018. Online; accessed 11-May-2020.

[8] D. Zorbas, "Design Considerations for Time-Slotted LoRa(WAN)," in *International Conference on Embedded Wireless Systems and Networks (EWSN), 1st Workshop on Massive LoRa Deployments: Challenges and Solutions*, pp. 1–6, ACM, Feb 2020.

[9] Semtech Corporation, "LoRa Modem Design Guide." https://lora-developers.semtech.com/library/product-documents/, 2013. Online; accessed 11-May-2020.