Time-Slotted LoRa Networks: Design Considerations, Implementations, and Perspectives

Zorbas, Dimitrios; Fafoutis, Xenofon

Published in:
IEEE Internet of Things Magazine

Link to article, DOI:
10.1109/IOTM.0001.2000072

Publication date:
2021

Document Version
Peer reviewed version

Time-Slotted LoRa Networks: Design Considerations, Implementations, and Perspectives

Dimitrios Zorbas* and Xenofon Fafoutis†
*Tyndall National Institute, University College Cork, Ireland
†Technical University of Denmark, Denmark
Authors' email: dimzorbas@ieee.org, xefa@dtu.dk

***Abstract*—In recent years, LoRa and LoRaWAN have emerged as a promising Low-Power Wide Area Networking (LPWAN) technology for long-range low-throughput applications. LoRaWAN, however, is based on unregulated medium access and, thus, cannot provide high packet delivery guarantees. This limitation, indeed, hinders the suitability of LoRaWAN for Industrial Internet of Things applications. In this article, we explore time-slotted medium access protocols as a reliable alternative to LoRaWAN, focusing on the considerations, challenges and perspectives for designing time-slotted protocols that leverage LoRa at the physical layer. The article is particularly focused on protocols with either a proof-of-concept implementation or protocols that have been deployed in the real world.**

I. Introduction

LoRa is one of the most known LPWAN radio technologies. Due to its long range, low cost, low-power operation, and resistance to external interference, it can be employed to support a wide range of Internet of Things (IoT) applications. Examples of such applications are the environmental monitoring of remote areas, localisation, and industrial applications.

The current LoRa-based standard proposed by LoRa Alliance, called LoRaWAN, is designed to support multiple concurrent applications without any of them having specific characteristics in terms of payload size, data generation periodicity, and Quality of Service (QoS) requirements. Thus, LoRaWAN is designed having in mind deployment simplicity combined with network longevity. It gives to the user the freedom to select among numerous radio and network parameters such as the Spreading Factor (SF), the Coding Rate (CR), the payload size, the radio frequency, and the packet rate. As a consequence, LoRaWAN adopts a Pure Aloha-based MAC layer while a level of fairness is guaranteed by regional radio duty cycle rules.

However, due its unregulated medium access mechanism, LoRaWAN cannot guarantee application-specific requirements such as a higher than 99% packet delivery ratio and a guaranteed delay in industrial applications [1]. This restriction made researchers think about other medium access approaches to support remote applications with strict network requirements. Unlike Aloha, time-slotted communications can provide the desired level of network reliability.

In time-slotted communications, the time is divided in repeated frames whereas a frame consists of a number of time-slots. The size of a time-slot is usually fixed and depends on the selected payload size and the radio characteristics. In this way, multiple users can share the same radio frequency without colliding with each other when assigned to different time-slots. The time-slot assignment is a fundamental process of every time-division protocol and it is usually handled by a central coordinator (e.g., in cellular networks). Apart from this, time-slotted protocols base their operation on time synchronisation which allows end-devices to wake-up, receive, and transmit at strictly defined timings.

Even though time-slotted communications have extensively been studied in the literature since decades and many real-world applications already rely on them, the unique characteristics of LoRa radios as well as the duty cycle restrictions in sub-GHz ISM bands make the design of new Time-Slotted LoRa (TSL) MAC layers a very challenging problem. The purpose of this paper is to highlight these unique characteristics of LoRa technology that need to be taken into account when designing a TSL MAC, present the progress of current TSL implementations, and point out their issues and perspectives.

II. LoRa and LoRaWAN

LoRa (**Lo**ng **Ra**nge) is a proprietary chirp spread spectrum (CSS) modulation. LoRa's main characteristic is that it trades data rate with receiver sensitivity by selecting the amount of spread to use in the CSS modulation. This is controlled by a radio parameter called Spreading Factor. The higher the SF, the lower the sensitivity and, thus, the larger the coverage. However, data rate decreases substantially with higher SFs resulting in longer transmission times and higher energy consumption. Apart from the SF, the transmission time or range of a LoRa packet are also affected by some other parameters such as the channel bandwidth (BW), the CR, the preamble size, and the cyclic redundancy check. Finally, transmissions performed over different SFs are almost orthogonal to each other increasing the network capacity.

LoRaWAN is an open standard developed by LoRa Alliance. It provides a number of services for LoRa-enabled devices such as registration, uplink and downlink communication, end-to-end encryption, adaptive data rate, and localisation [2]. According to the standard, an end-device can be part of one of three classes, however, the majority of devices belong to Class A. These are energy constrained devices whose transmissions are performed at sparse intervals for monitoring purposes. Class A devices adopt a very simple MAC mechanism which is Pure Aloha-based. Every transmission in this class of devices is optionally followed by

TABLE I
TRANSMISSION POWER (TP) AND DUTY CYCLE REGULATIONS PER SUB-BAND FOR THE EU868 BAND [4].

Frequency	TP	Duty Cycle
863 – 865 MHz	25 mW ERP	$\leq$ 0.1% or LBT
865 – 868 MHz	25 mW ERP	$\leq$ 1% or LBT
868 – 868.6 MHz	25 mW ERP	$\leq$ 1% or LBT
868.7 – 869.2 MHz	25 mW ERP	$\leq$ 0.1% or LBT
869.4 – 869.65 MHz	500 mW ERP	$\leq$ 10% or LBT
869.7 – 870 MHz	5 mW ERP	No requirement
869.7 – 870 MHz	25 mW ERP	$\leq$ 1% or LBT

one or two downlink receive windows for acknowledgments and LoRaWAN commands (e.g., to suggest a node to use a different channel). However, downlink availability may be very limited [3].

III. Enabling Time-Slotted LoRa Transmissions

This section describes two main particularities of the LoRa radio technology and how these particularities affect the entire design of a TSL system, such as the scheduling, the synchronisation, and the acknowledgements. It also discusses an additional number of factors that need to be reconsidered in a TSL network. Overall, this section highlights the key differences between TSL and traditional time-slotted protocols.

A. Radio Duty Cycle Limitations

LoRa mainly operates in the sub-GHz ISM bands, where radio duty cycle and transmit power regulations are imposed by regional authorities. For example, in Europe LoRa devices use the 868 MHz ISM band ranging from 863 to 870 MHz [4]. The majority of the sub-bands in this range have a 1% radio duty cycle limit and 25 mW (14 dBm) maximum allowed Equivalent Isotropically Radiated Power (EIRP), as shown in Table I. This results in a total uplink time of 36 seconds within an hour. When this time is divided in successive transmissions, an inactive time period of 99 times the transmission time of the last transmission must be followed in between them. In LoRaWAN, the same duty cycle rules hold for the gateways with the exception of an extra 10% duty cycle channel dedicated to the second receive window.

The radio duty cycle restriction can be bypassed if a Listen Before Talk (LBT) method is used to access the medium. However, to date there is no LBT mechanism to allow duty cycle-free LoRa transmissions in sub-GHz ISM bands. The Channel Activity Detection (CAD) mechanism cannot be used as a CSMA-style method to replace the duty cycle rules because it cannot detect channel activity of packets with a SF different to the currently selected, and moreover, it cannot detect signals transmitted from other radio technologies on the same frequency.

B. Unequal Slot Length

As mentioned in Section II, unlike other radio technologies, LoRa has a large number of settings whose configuration may lead to longer or shorter transmission times for the same payload. For example, the transmission time increases substantially with higher SFs, lower BWs, and lower CRs. In a time-slotted system, those irregular transmission times impose the use of unequal slot lengths unless it is explicitly decided to use specific fixed settings for all the transmissions (e.g., the same SF, BW, and CR). However, such a choice would lead to a big loss of flexibility and would decrease capacity. For example, if all the nodes use a low SF or high BW, there will be a loss of range. On the contrary, a high SF and low BW would cause a very limited network capacity and long delays.

C. Consequences on the system design

A number of consequences on a TSL system design due to the duty cycle restrictions and the multiple LoRa configuration settings are described below.

1) Downlink Activity & Acknowledgments: The radio duty cycle restriction causes many issues in downlink transmissions. Apparently, when the uplink traffic is high, the gateways may not be able to acknowledge all the transmissions one-by-one or send out command packets, which causes extensive retransmissions and increased energy consumption. Moreover, the problem gets harder considering that LoRa transceivers are half-duplex, so a gateway cannot receive data as long as it is being used for downlink transmissions.

In a time-slotted environment, an acknowledgement can be sent either during the data transmission slot or in separate future time-slot. In the first case, it is difficult to control the available downlink time resources because a receiver may get several packets in a short amount of time. The second case offers higher flexibility but increases delay. Apart from that, additional overhead is required with every downlink interaction so downlink transmissions need to be as compact as possible.

2) Data Periodicity & Capacity: By default, time-slotted networks have a limited capacity. In LoRa networks, due to duty cycle restrictions, the data transmission periodicity may be very sparse. This actually means that certain applications that require frequent packet generation may not be supported by a LoRa-based system. To satisfy a certain data periodicity in a time-slotted system, the number of slots in the frame has to be limited. Given the low-data rate of LoRa in combination with the duty cycle restrictions, the network capacity may be strictly bounded. Nevertheless, a time-slotted LoRa system must be designed in such a way that a large number of applications is supported even though that means that some of the network capacity must be sacrificed.

3) Scheduling: Time-division protocols base their functionality on scheduling of transmissions. Scheduling is part of the resource allocation mechanism, a fundamental mechanism of every TDMA-based system. The job of this mechanism is to reserve a number of slots in the frame for every node in the network so that the latter can use them to perform transmissions without interfering with other nodes sharing the same resources. Scheduling can be performed either centralised or distributed depending on the nature of the network.

However, scheduling of transmissions in a TSL system – as well as any kind of resource allocation in such networks – suffers from extremely limited downlink availability and low data rates of LoRa radios. Indeed, a LoRa gateway would require several minutes to disseminate a schedule of a few kilobytes to all the nodes given a 10% duty cycle and the lowest SF [5]. Apart from that, having nodes joining and leaving the network at random times would require often re-computation and re-dissemination of the schedule. The alternative example of cellular networks where nodes request slots on demand would not work either for similar reasons.

Moreover, as it is already mentioned, in a time-slotted environment, the time is divided in repeated frames and a number of slots is accommodated in each frame. Assuming that a slot can be allocated to only a pair of nodes (transmitter-receiver), one could say that the number of slots in the frame has to be equal to the number of pairs, eventually with some additional slots dedicated to control packets. However, due to the duty cycle restriction, empty space (slots) may need to be added in the schedule to respect the duty cycle rules.

4) Time Synchronisation: An integral part of every time-division protocol is the time synchronisation. Because each node's clock runs at a slightly different rate, the nodes need to periodically synchronise their clock according to a reference clock so that transmissions and other time-based activities are performed as expected. In a LoRa-based system, synchronising a large number of nodes is not trivial because the uplink transmissions may be sparse, the downlink may not be available at all times, and the nodes may not send data with the same rate. Moreover, the guard times that are added between successive slots to tolerate slight de-synchronisations need to be longer and be adapted to different LoRa configuration settings.

5) Routing: In star networks, like LoRaWAN, routing is trivial; however, time-slotted LoRa communication enables more complex mesh topologies and multi-hop communication, and such topologies require routing. IETF RPL is arguably the most commonly used routing protocol in low power networks. The primary challenge with using RPL in a mesh LoRa context is the overhead that it introduces for establishing and maintaining the routes. Alternatively, the problem of routing can also be solved with flooding. However, flooding does not scale well in large networks. In both cases, routing introduces transmission overhead and consumes part of the limited capacity and duty cycle resources.

6) Large Number of Registrations: Nodes network join times in a LoRa network are expected to be longer than other systems. This is because of (i) the longer transmission times that cause higher probabilities of collisions when multiple nodes try to join in a short period of time and (ii) the duty cycle restrictions that impose delays between successive tries. Considering a TSL network may not cause any difference in the join process since the latter may remain Aloha-based. Allowing the nodes to join-register over different SFs and channels using a single gateway is not a viable solution either, because of the half-duplex nature of LoRa transceivers. The most profound solution to that problem is to add several gateways, increasing however the cost of the deployment.

D. General Considerations

1) Propagation Time: LoRa is a long range radio technology which can achieve a several-kilometre range. Due to this long distance, the propagation time may not be negligible. Taking into account that signals travel at speed of light, the propagation time may reach 30 μs for distant nodes leading to desynchronisation problems if the design does not take into account this extra time. A straightforward solution to that problem is to include a maximum propagation time into the guard times. The solution exhibits negligible – compared to the transmission time – delay, does not require additional packets to be sent (as with cellular networks), and alleviates the programming complexity.

2) Battery Lifetime: Due to time synchronisation, the nodes may consume more energy because they have to periodically turn their radio on to receive the synchronisation packet. However, simulation studies have shown that in high traffic scenarios the energy cost of re-transmissions in an Aloha-based approach may be higher than the synchronisation cost [6]. In any case, the design of a TSL system should include as light and short as possible synchronisation mechanisms due to (i) the duty cycle restrictions, and (ii) to achieve as short as possible wake-up times.

3) Mobility, Multiple Gateways, & Roaming: A characteristic that we often meet in industrial and smart-city scenarios is the user mobility. A node may move at different locations and report information received by multiple gateways at the same time. A practical issue in a TSL network is how a node can switch from one gateway to another with the minimum possible cost. A solution could be to have multiple gateways sharing a number of common slots to make this transition smoother. However, a number of other issues arise here such as how to make the schedule as efficient and fair as possible, how to make the schedule work over different LoRa configuration settings, and how to periodically re-compute and maintain that schedule. Perhaps, this would also require synchronisation between gateways but how gateways can globally synchronise to a reference clock and what this reference clock should be, are some new questions that need to be answered.

4) Security: The security of IoT networks has attracted a lot of attention in the last years due to their particularities in limited hardware capabilities compared to conventional wireless networks. LoRa communications exhibit a number of additional issues due to the low data rate of the technology as well as due to the duty cycle restrictions which make on-the-fly negotiation and exchange of security keys a hard task. Moreover, the authentication and encryption mechanism of LoRaWAN is restricted to a pair of nodes (node-gateway), while it does not consider distribution of the key material associated with one-to-many communications. However, in every time-slotted system, control packets need to periodically be transmitted (e.g., for synchronisation) that rely on the one-to-many way of communication.

Moreover, time-slotted wireless networks are vulnerable to various attacks, such as selective jamming attacks. An attacker could eventually synchronise with the network and jam the downlink synchronisation packet causing a network desynchronisation. The key in the system design is to avoid such attacks by utilising multiple channels or even different SFs but to do that as autonomously as possible with the minimum possible downlink information.

TABLE II
FEATURES OF CURRENT TSL IMPLEMENTATIONS

	Property	TS-LoRa	TSCH-over-LoRa	Synchronous LoRa Mesh	LoRaBlink	Multi-Hop LoRa
Features	Collision-Free	Yes	Depends on schedule	Yes	No	Yes
	Acknowledgements	Yes	Yes	Yes	Optional	No
	Topology	Star	Mesh	Mesh	Tree	Tree
	Multiple Gateways	Multiple 1-channel	Yes	Yes	No	N/A
	Addressing	LoRaWAN-based	IPv6	Fixed	Fixed	Fixed
	Routing	N/A	RPL, Static	Flooding	Flooding	Custom
	Security	LoRaWAN-based	802.15.4, TinyDTLS	No	No	No
	Compatibility	N/A	UDP, CoAP, MQTT	N/A	N/A	N/A
	Protocol Overhead	Low	Very high	High	Medium	High
	Scalability	Medium to High	Medium	Low	Low	Medium
	Timeslot Size	Fixed per SF	Fixed	Fixed	Fixed	Fixed
	Channel Hopping	No	Yes	No	No	No
	Joining / Registration	Slow	Slow	Medium	Slow	N/A
Code	Open Source	Yes*	Yes†	Yes**	Yes‡	No
	License	GNUv3	BSD-3	GPL-3.0	EPL-v1.0	N/A
	Radio Supported	SX1276	SX1272	SX1276	SX1272/76	SX1272
	OS / SDK	Pycom SDK	Contiki-NG	FreeRTOS	IBM LMiC	Mbed OS
	Reference	[6]	[7]	[8]	[9]	[10]

*https://github.com/deltazita/ts-lora
†https://github.com/dtu-ese/contiki-ng-lora
**https://github.com/Eawag-SWW/loramesh
‡https://www.lancaster.ac.uk/scc/sites/lora/lorablinkkit.html

IV. CURRENT IMPLEMENTATIONS

This section presents the current implementations and proof-of-concepts of TSL systems along with their main features and weaknesses. It also explains how these implementations tackle or mitigate some of the challenges described in the previous section and presents ideas to address some of the unresolved issues. It should be noted that a few more TSL approaches have been presented in the literature [11]–[15] without, however, being experimentally tested or validated.

A. Implementations

Table II summarises the current implementations and their main features. The following subsections present details of the adopted designs.

1) TS-LoRa: TS-LoRa [6] has been proposed as an alternative to LoRaWAN for applications that require frequent and very reliable transmissions. It can run over LoRaWAN as an additional layer once some modifications have been made to the mechanism that generates the device address (`DevAddr`) and the network address (`NwkAddr`). In LoRaWAN, `DevAddr` is a 32-bit number consisting of the network id (8 bits) and the network address (24 bits). The network id is a unique number assigned by the LoRa Alliance, while the network address is a random number generated by the network server. In TS-LoRa, `NwkAddr` (or entirely `DevAddr`) is chosen at random until the desired slot number is produced. Each node can determine its slot by executing a same simple modulo calculation once `DevAddr` is received during registration. The advantage of this method is that no scheduling information needs to be disseminated to the nodes. Thereby, the protocol exhibits a very low overhead.

In order to increase coverage and network capacity, TS-LoRa exploits several repeated parallel frames (one for each SF) using an equal number of single channel gateways and an additional one for join requests. Each frame consists of a number of slots whose number depends on the application requirements and the radio duty cycle rules. One of the novelties of TS-LoRa is that it uses a single slot at the end of each frame for synchronisation and acknowledgements (SACK slot). The nodes wake-up at a predefined time which is slightly earlier before the SACK packet. They can adjust their clock by counting the time until the SACK packet arrives. The SACK packet contains a series of binary digits, one for each assigned slot in the frame. Each binary digit indicates if the transmission performed on the corresponding slot was successful or not. In this way, TS-LoRa tackles the problem of limited downlink time of the gateway and extends scalability. A TS-LoRa example is illustrated in Fig. 1.

TS-LoRa suggests the use of six orthogonal radio channels, one for each SF. This is suggested to avoid inter-SF interference caused due to imperfect SF orthogonality. However, this means that the nodes (and the gateways) must be configured to use only the assigned frequencies per SF. No channel hopping mechanism is currently implemented. In terms of security, TS-LoRa shares the same key generation mechanism with LoRaWAN OTAA, utilizing an application key (`AppKey`) and a join unique identifier (`JoinEUI`). `AppKey` and `JoinEUI` are used to generate the session encryption key (`AppSKey`). Nevertheless, `AppSKey` is only used to encrypt the uplink transmissions, while the downlink ones are sent unencrypted.

2) TSCH-over-LoRa: TSCH-over-LoRa [7] is a layer that connects the implementation of TSCH (Time-Slotted, Channel Hopping) in Contiki-NG to a LoRa radio driver.

TSCH is a synchronous MAC protocol that is typically used on top of the IEEE 802.15.4 standard. In a TSCH network, all nodes are globally synchronised and transmissions are orchestrated by a schedule. A schedule defines a repeating sequence of timeslots, in which a particular device can operate as a transmitter, receiver, or sleep to save energy. Moreover,

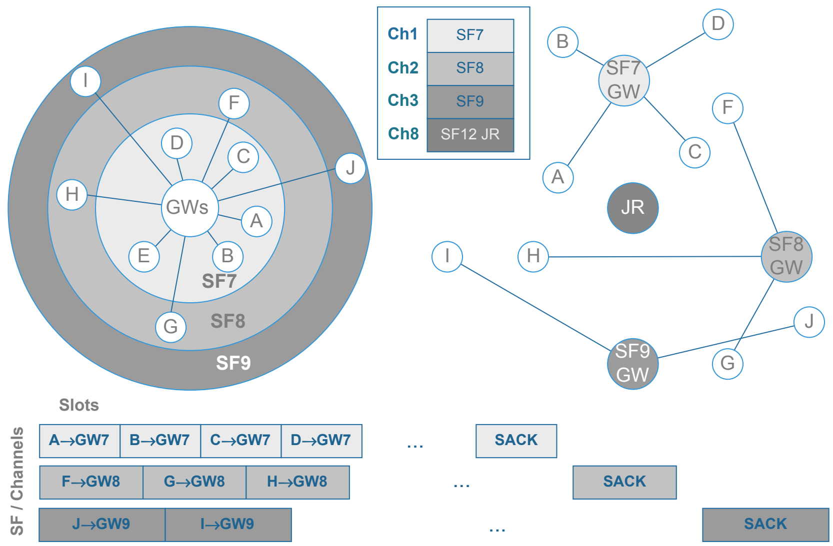


Fig. 1. TS-LoRa with two different arrangements of nodes/gateways utilizing the first 3 SFs. In the first arrangement, all the gateways (GW) are placed next to each other. In the second one, the placement is random, however, all the nodes have to be in the range of the join request (JR) server (SF12).

the schedule specifies a channel offset for parallel orthogonal transmissions on different channels. In addition, TSCH implements channel hopping using a pseudo-random channel hopping sequence. The schedule itself can be static or be generated dynamically in a centralised or distributed manner. If the scheduler allocates a timeslot to no more than a single transmitter, then the timeslot is collision-free. The TSCH-over-LoRa mechanics are illustrated in Fig. 2.

The TSCH-over-LoRa implementation maintains full compatibility with the implementation of TSCH; indeed, the authors only adapt the defined timings, such as the timeslot size. As a result, TSCH-over-LoRa brings a lot of the off-the-shelf functionality of 6LoWPAN-based TSCH networks to LoRa devices. These include: (i) high reliability with collision-free scheduling, channel hopping, and unicast acknowledgements both at the uplink and downlink; (ii) arbitrary mesh topologies including multi-hop LoRa networks; (iii) IPv6/6LoWPAN addressing and compatibility with several higher-layer protocols including RPL, UDP, CoAP and MQTT; and (iv) state-of-the-art hop-by-hop security (using the IEEE 802.15.4 security layer) and end-to-end security (with TinyDTLS). A key difference between LoRa and the IEEE 802.15.4 PHY is that the transmission time of LoRa is highly variable depending on the configuration. TSCH is, indeed, designed for fixed timeslots and TSCH-over-LoRa inherits this property. As a result, it supports homogeneous networks of fixed SF, CR and BW. Moreover, TSCH-over-LoRa inherits the joining process of TSCH which is notoriously slow. Overall, the key limitation of TSCH-over-LoRa is its very high protocol overhead: the 6LoWPAN protocol stack adds significant overhead in terms of headers and control packets, which further limit the already limited bandwidth (see Table I).

3) Multi-hop LoRa: A similar to TSCH-over-LoRa approach is also presented in [10]. In that approach, a custom routing protocol based on overhearing and the CAD mechanism is introduced. The data scheduling is done based on local information gathered by the parent nodes. However, the authors do not clearly describe their approach. It seems that there is no channel hopping is performed and it is not clear how the nodes respect the duty cycle rules. Since no name is given by the authors, we named this approach as “Multi-hop LoRa”.

4) Synchronous LoRa Mesh: Ebi *et al.* [8] present a 2-hop monitoring system consisting of underground end-nodes, relay nodes, and gateways. The communication between the end-nodes and the relay nodes is done over a custom LoRa mesh time-slotted protocol, while the relay nodes communicate with the gateways using LoRaWAN. The proposed LoRa mesh protocol builds a topology based on periodically transmitted beacons initiated by the relay nodes. A node can join the network by scanning the list of available channels to detect a beacon. It can select to connect directly to a relay node or connect via another end-node based on the received signal strength. It then sends a join request to the selected parent and receives a time-slot in due time. If the selected parent is not a relay node, the request is forwarded to the relay node in a multi-hop manner. The relay nodes may reserve several slots to accommodate this mechanism. Moreover, the time-slot allocation mechanism accommodated on the relay nodes reserves a number of slots for uplink and downlinks within the LoRa mesh sub-network as well as multiple slots for LoRaWAN transmissions between the relays and the gateways. Each frame starts with a flooding method to disseminate synchronisation timestamps and align the end-nodes’ clock with that of the relay nodes. An example of its mechanics is illustrated in Fig. 2.

The proposed synchronous sub-network has been designed only for the specific scope of the paper; that is the underground monitoring system consisting of a limited number of repeater devices as well as end nodes. The system relies on flooding and multi-unicast communications (e.g., for slot allocation) which restrict the scalability of the network (given also the duty cycle constraints). Moreover, no security/encryption mechanism is available for both uplink and downlink traffic even though

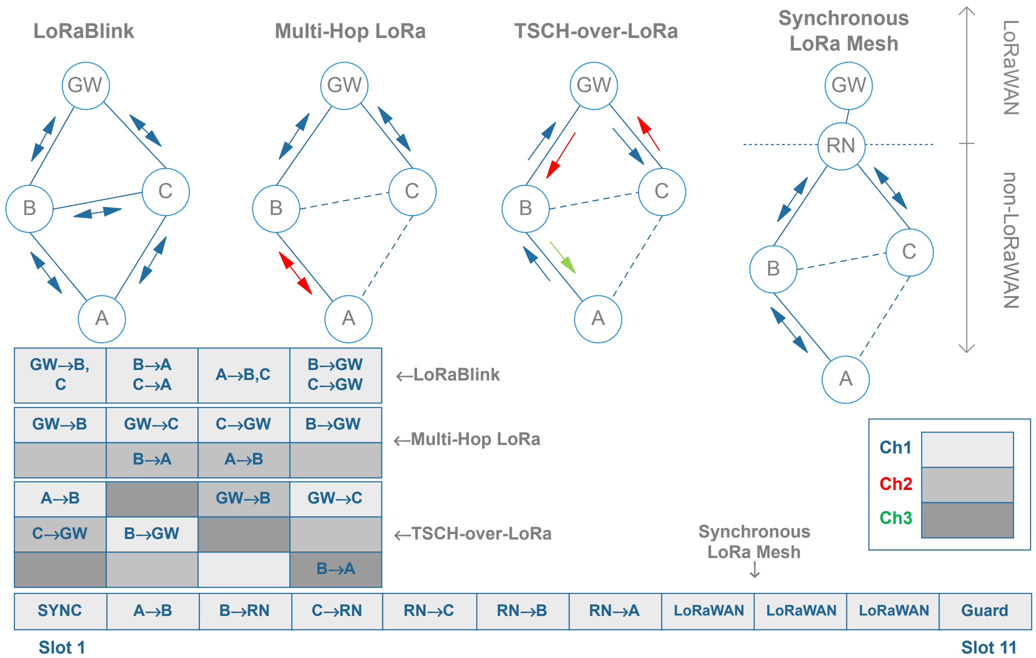


Fig. 2. Four LoRa-mesh solutions along with their indicative time-slot schedule (join slots are omitted). "Synchronous LoRa Mesh" introduces one or more relay nodes (RN) in between gateways (GW) and nodes. TS-LoRa is presented separately in Fig. 1.

pre-shared keys can be installed on the repeater nodes.

5) LoRaBlink: LoRaBlink [9] builds upon the principle of concurrent transmissions. It implements a MAC/routing protocol that is based on synchronous flooding on the downlink (sink to nodes) and directed synchronous flooding on the uplink (nodes to sink). Concurrent transmissions take place within repeating epochs. Each epoch is organised in two phases: in the first phase concurrent beacons are flooded in the downlink direction enabling multi-hop communication. In the second phase concurrent transmissions in the uplink direction transfer the data. Link-layer acknowledgements of data packets are supported but they are optional. A node joining the network turns on their radio and listens to the channel until it receives a beacon: the expected joining time is determined by the duration of an epoch. LoRaBlink is also illustrated in Fig. 2.

LoRaBlink aims to support multi-hop tree topologies whereby the forwarding nodes can duty-cycle to conserve energy. The authors target LoRa networks of low density and low traffic volume, and assume a limited number of nodes. The protocol is low-latency and resilient to interference; however, collisions are possible and, indeed, the evaluation presented in the paper demonstrates a packet delivery ratio of 80%. The protocol overhead is relatively moderate; in particular, beacon packets occupy some of the slots and nodes periodically turn their radios on for CAD on every timeslot, even when there is no transmission taking place. On the other hand, LoRaBlink requires no scheduling and, thus, does not have any of the overhead of generating, updating and distributing schedules.

B. Current Issues & Perspectives

Current implementations tackle or mitigate some of the issues presented in Section III, however, a few of them remain unresolved. Table III summarises how these implementations confront the corresponding challenges while more details are given in the following paragraph.

Limited downlink availability is tackled by grouping multiple acknowledgments in a single short slot. This solution saves a lot of the additional overhead and better controls the duty cycle resources. Autonomous scheduling algorithms are being used to avoid downlink bottlenecks that otherwise would lead to significant delays. Moreover, one of the fundamental challenges is how to support as many LoRa settings (i.e., different SF/BW/CR/Payloads) as possible. Most of the implementations consider fixed settings for all the nodes which may limit the application flexibility and capacity. TS-LoRa supports multiple SFs by utilizing additional 1-channel gateways. Different settings in terms of BW, CR, and payloads can be used per SF because multiple independent frames can run in parallel. All the approaches perform synchronisation in a single slot – within data transmission slots or with a separate slot. In most of them, synchronisation is achieved by counting the time from the moment the receiver turned on its radio to the moment it received the sync info. This solution saves some duty cycle time and depicts lower energy consumption. To tackle the longer transmission times of LoRa radios as well as the sparse communication due to the duty cycle restrictions, all the approaches use large enough guard times. The guard times also incorporate the minor but important propagation time. Encryption is still an issue since most of the implementations either neglect it or support only uplink encryption. Only TSCH-over-LoRa provides a full encryption mechanism as it relies on the already mature security layers of Contiki-NG. Multi-hop solutions still suffer from increased overhead and delays during the construction of the routing tables. This is because they are usually based on periodic dense beacon transmissions that rapidly vanish the available duty

TABLE III
SUMMARY OF CHALLENGES IN CURRENT TSL IMPLEMENTATIONS

Challenge	**TS-LoRa**	**TSCH-over-LoRa**	**Synchronous LoRa Mesh**	**LoRaBlink**	**Multi-Hop LoRa**
ACK slots	Grouped in 1 slot	Grouped in 1 slot	1 per slot	Grouped in 1 slot	N/A
Multiple settings (SF/BW/CR)	6 SFs/Fixed/Fixed	Fixed/Fixed/Fixed	Fixed/Fixed/Fixed	Fixed/Fixed/Fixed	Fixed/Fixed/Fixed
Capacity	Fixed per SF	Fixed	Fixed	Fixed	Fixed
Scheduling algorithm	Autonomous	Various options	On demand	Contention-based	Distributed
Synchronisation slots (timestamps)	up to 6 in parallel (No)	During transmissions (No)	1 dedicated (Yes)	Beacon-based (No)	During transmission (No)
Routing mechanism	N/A	RPL, static	Beacon-based	Beacon-based	Beacon-based
Join method	Aloha + CAD	Beacon-based	Slotted Aloha	Beacon-based	CAD- & beacon-based
Roaming	Re-registration	Rejoin	N/A	Rejoin	Rejoin
Propagation delay	in guard time	in guard time	N/A	$<$ 3 symbols period	N/A
Encryption (Uplink/Downlink)	Uplink only	Both	Optional	Unresolved	Unresolved
Reference	[6]	[7]	[8]	[9]	[10]

cycle resources. Finally, a last problem we have identified is the lack of a roaming mechanism which would allow devices to interoperably move across multiple cells. Current designs assume that this can happen by allowing a device to rejoin the network, however, this is not efficient in terms of both delay and energy consumption.

V. CONCLUSION

This paper explored important considerations and challenges for designing time-slotted medium access protocols for LoRa networks. In addition, we surveyed various time-slotted solutions that have been proposed in the literature, mainly focusing on protocols that have been implemented, experimentally evaluated, and are open source. Time-slotted communication offers high reliability and has the potential to transform LoRa networks into a dependable option for Industrial IoT applications. Yet, there are numerous issues that remain to be addressed, particularly in reducing the protocol overhead, eliminating security barriers, and leveraging the flexibility of LoRa at the physical layer.

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.

REFERENCES

[1] 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.

[2] LoRa Alliance Technical Committee, "LoRaWAN^{TM} 1.0.3 Specification." lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf, 2018. Online; accessed 17-Oct-2019.

[3] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "A Survey on the Viability of Confirmed Traffic in a LoRaWAN," *IEEE Access*, vol. 8, pp. 9296–9311, 2020.

[4] ETSI ERM TG28, "Electromagnetic compatibility and radio spectrum matters (erm); short range devices (srd); radio equipment to be used in the 25 mhz to 1000 mhz frequency range with power levels ranging up to 500 mw," *European harmonized standard EN*, vol. 300, no. 220, p. v2, 2012.

[5] D. Zorbas, K. Q. Abdelfadeel, V. Cionca, D. Pesch, and B. O'Flynn, "Offline Scheduling Algorithms for Time-Slotted LoRa-based Bulk Data Transmission," in *IEEE 5th World Forum on Internet of Things (WFIoT)*, pp. 1–6, IEEE, 2019.

[6] 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, Mar. 2020.

[7] M. Haubro, C. Orfanidis, G. Oikonomou, and X. Fafoutis, "TSCH-over-LoRA: Long Range and Reliable IPv6 Multi-hop Networks for the Internet of Things," *Internet Technology Letters*, 2020.

[8] C. Ebi, F. Schaltegger, A. Rüst, and F. Blumensaat, "Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure," *IEEE Access*, vol. 7, pp. 57663–57677, Sep 2019.

[9] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," in *EWSN '16 Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, pp. 361–366, Junction Publishing, feb 2016.

[10] D. L. Mai and M. K. Kim, "Multi-Hop LoRa Network Protocol with Minimized Latency," *Energies*, vol. 13, no. 6, p. 1368, 2020.

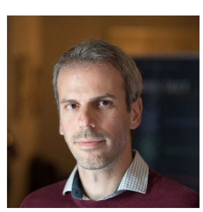
[11] J. Lee, W. Jeong, and B. Choi, "A Scheduling Algorithm for Improving Scalability of LoRaWAN," in *International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1383–1388, IEEE, Oct 2018.

[12] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of lorawans through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, pp. 1830–1842, June 2018.

[13] S. Gao, X. Zhang, C. Du, and Q. Ji, "A Multichannel Low-Power Wide-Area Network With High-Accuracy Synchronization Ability for Machine Vibration Monitoring," *IEEE Internet of Things Journal*, vol. 6, pp. 5040–5047, June 2019.

[14] 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 of Things Journal*, vol. 7, pp. 669–683, Jan. 2020.

[15] A. Abrardo and A. Pozzebon, "A multi-hop LoRa linear sensor network for the monitoring of underground environments: the case of the Medieval Aqueducts in Siena, Italy," *Sensors*, vol. 19, no. 2, p. 402, 2019.



Dimitrios Zorbas is an Assistant Professor at Nazarbayev University in Kazakhstan. He holds a PhD in Computer Science from the University of Piraeus in Greece. He has worked as post doctoral researcher at Inria Lille – Nord Europe and at University of La Rochelle in France. He was also a researcher at Tyndall National Institute – University College Cork in Ireland after receiving a Marie Curie fellowship. He is author of more than 50 peer-reviewed publications in the area of wireless communications, Internet of Things, and distributed computing. He has worked in several national as well as FP7 and H2020 projects. He is a member of the IEEE.

Xenofon Fafoutis (S'09-M'14-SM'20) received a PhD degree in Embedded Systems Engineering from the Technical University of Denmark in 2014; an MSc degree in Computer Science from the University of Crete (Greece) in 2010; and a BSc in Informatics and Telecommunications from the University of Athens (Greece) in 2007. From 2014 to 2018, he held various researcher positions at the University of Bristol (UK), and he was a core member of SPHERE: UK's flagship Interdisciplinary Research Collaboration on Healthcare Technology. He is currently an Associate Professor with the Embedded Systems Engineering (ESE) section of the Department of Applied Mathematics and Computer Science of the Technical University of Denmark (DTU Compute). His research interests primarily lie in Wireless Embedded Systems as an enabling technology for Digital Health, Smart Cities, and the (Industrial) Internet of Things (IoT).