Internet of Things 11 (2020) 100233

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Synchronization and efficient channel hopping for power efficiency in LoRa networks: A comprehensive study

Ritesh Kumar Singh*, Rafael Berkvens, Maarten Weyn

University of Antwerp - imec, IDLab - Faculty of Applied Engineering, Sint-Pietersvliet 7, 2000 Antwerp, Belgium

ARTICLE INFO

Article history:
Received 15 May 2020
Accepted 25 May 2020
Available online 31 May 2020

Keywords:
LPWAN
LoRaWAN
Time synchronization
Channel hopping

ABSTRACT

The usage of Low-Power Wide Area Networks (LPWAN) is increasing rapidly in many sectors of Internet of Things (IoT) applications. Long Range (LoRa) Wide Area Network (LoRaWAN) is a LPWAN, which has a long-range, low-cost and acts as a connectivity enabler. Most of the applications with real-time requirements, need reliable communication. This can be achieved by sharing a common time base across the network. However, making an efficient collaborative service of clock synchronization in LoRaWAN is challenging due to a lack of time notion. In this paper, We comprehensively study different approaches and design considerations for synchronization and tackle two problems of effective robustness in a LoRaWAN network. First, current research typically focuses on the benefits of LoRaWAN but ignores the requirement of reliability. To tackle this problem, we introduce a novel time synchronization scheme for radically reducing the usage of existing Aloha type protocol that handles energy consumption and service quality. Second, we look into the security space of LoRaWAN network, i.e. channel selection scheme for the given spectrum. Several security attacks are possible in LoRaWAN because the entire spectrum space is not used, and the utilization of few channels are comparatively higher. To tackle this problem, we present a channel hopping scheme that integrates cryptographic channel selection with the time notion. Finally, we evaluate the proposed time synchronization and channel hopping scheme for a real-world deployed peer to peer (P2P) model using commodity hardware. This paper concludes by suggesting the strategic research possibilities on top of this platform.

1. Introduction

Low power and low-cost communication technologies have made a great impact on the development of applications and services for the Internet of Things (IoT). Research progression in IoT has evolved with resource constraint challenges, generating new demands for wireless networks. Over the last decades, the number of applications is increasing exponentially, such as precision agriculture, healthcare and, traffic updates [1,2], etc. mainly due to miniaturization of IoT communication devices in parallel with energy-efficient computing. Around 29 billion connected devices are speculated by 2022 of which 18 billion will be related to IoT [3].

With the wide spectrum of services and projected inevitable scaling of applications in IoT, quality measures like scalability and reliability remain a challenge. This causes an increased demand for wide communication coverage, which in turn

* Corresponding author.
E-mail address: riteshkumar.singh@uantwerpen.be (R.K. Singh).

https://doi.org/10.1016/j.iot.2020.100233
2542-6605/

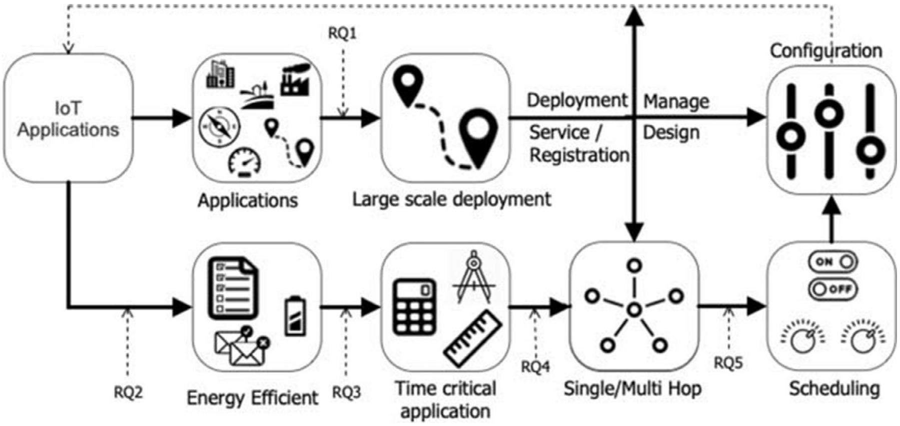

Fig. 1. Progression of IoT application and inline Research Questions (RQ)

led to the advent of Low-Power Wide Area Networks (LPWAN). It provides energy efficiency, scalability and, coverage as key ingredients. Various LPWAN systems [4] are evolving like LoRaWAN [5] by the LoRa Alliance, Narrow Band Internet of Things (NB-IoT) by 3GPP and ultra-narrow band communication by Sigfox [6]. LoRa, the radio modulation technology of LoRaWAN, is widely deployed because of its key enablers such as low power, long-range, and low-cost chipset [7].

LoRa has brought in a lot of interesting applications such as utilities, precision agriculture and smart city [7,8]. It has attracted many researchers for evolving LoRaWAN infrastructure [9], the network stack on top of LoRa physical layer. LoRaWAN features a low data rate and multi-kilometer communication range and enables a star network topology. This eventually simplifies deployment and maintenance cost of the network. Resulting, into greater momentum behind LoRaWAN with deployment all over the world by network operators and solution providers [10].

However, along with those advantages, LoRaWAN has some limitations and problems [11]. In particular, a key problem arises due to communication latency and using Aloha type protocol [12,5]. This in turn results in high energy consumption with low reliability [13] and limits the command and control scenario applications in the LoRaWAN network [14]. Moreover, for application such as data fusion and event monitoring, the timing of events becomes critical. In Wireless Sensor Networks (WSN), this is well implemented by dedicated strategies of clock synchronization [16,15]. However, it is hard to deploy these type of applications on top of LoRaWAN because there is no notion of time in the network on the end devices.

In recent literature, the paper [11] highlights the important research challenges in LoRaWAN network as - exploring new channel hopping methods, scheduling deterministic traffic along time and limiting interference/collisions by enabling coordination mechanisms between gateway and end devices. In addition, it specifically highlights the requirement of time synchronization and channel hopping as a mitigation technique to revamp the security and robustness of LoRaWAN network.

In this paper, we address the aforementioned challenges by proposing a novel time synchronization platform for LoRaWAN network and establish a proof of concept along with design consideration and state of the art for time synchronization. This paper is an extension of our work [17]. In Fig. 1, we try to exhibit the potential Research Questions (RQ) for the requirement of time notion, optimal channel utilization, energy consumption and managing time-critical applications for the LoRaWAN network (RQ1 to RQ5). Given RQ, are intended for both the research community and solution providers to further exploit the time sync platform and open new opportunities of applications. We try to answer RQ1 to RQ3 (time-critical requirement, energy-efficient and reliable communication respectively) through the proposed solution and RQ4, RQ5 (multi-hop and scheduling task) remains open for future work. Inclusion of a time notion in LoRa network can tackle the most critical drawback of the availability of down link communication. It can help to save energy by shortening the necessary guard time and turning off the radio for better duty cycling schemes. It can open a gamut of research opportunities for complex LoRaWAN based networks like dynamic scheduling, time-sensitive middleware, and multi-hop solution [18]. On top of this, a LoRaWAN time sync platform can be the silver bullet to reduce the deployment cost of small scale applications such as in a greenhouse [19] or smart home [2,14]. This can be achieved by replacing costly gateways with a low cost end node that listens to the dedicated sensor devices at a given allocated time slot. Also, we propose lightweight Cryptographic Channel Hopping scheme (CCH) to enhance the security and reliability (collision rate) of the network. This unique distribution and utilization of available channels can act as a mitigation technique for security challenges such as selective jamming [20] and increase the robustness of network.

The paper is structured as follows. Section 2, presents the background on time synchronization and channel hopping in the context of LoRa and LoRaWAN. It also presents a comprehensive study for different approach for synchronization and crucial design considerations. Section 3, proposes design and implementation of the system using real LoRa devices. We first perform few initial investigation and then take its finding into account to propose time synchronization and CCH algorithm. In Section 4, We evaluate preliminary the synchronization resolution and optimal guard time to receive the message. For CCH, we evaluate the channel utilization and distribution along with total memory consumption, and statistical test to

Time sync delivery delay	
Time involved	Field
Build reference time	Sync time
Radio Command	Radio
Encoding	Antenna
Propogation	Antenna
Decoding	Radio
Byte allignment	Radio
Interrupt handling	Software
Write/Read	RTC handle
RTC interrupt	Software

Fig. 2. Delay in time sync message delivery

check the uniformity of channel distribution followed by relative study of energy consumption. Finally, we conclude this paper along with the application use case in Section 5 and discuss the plans and set of possibilities that can be build upon time synchronized LoRaWAN platform as future work.

2. Background

This section gives an overview of time synchronization and CCH along with its key elements. Also, we try to analyze and explore LoRa and LoRaWAN in the same context.

2.1. Time synchronization in IoT

Time synchronization in IoT is a decisive part of computing to coordinate the tasks for more exhaustive interoperability [21]. With the orderly notion of time, the service offering from IoT devices becomes more productive by resolving the conflicts of packets, thereby increasing the reliability of the network. It is required to provide a common timescale for seamless integration of IoT devices along with the management and control of the network. The basic motivation behind synchronizing the clocks is to impose order of streams or sequencing of data from scattered devices. IoT applications require time synchronization to address different application scenarios like - coordinating operations, collaboration to achieve a task, power saving schemes to increase network lifetime or to share the transmission medium appropriately for avoiding collisions. Hence, the inclusion of time notion in IoT results into better energy efficiency, scalability, precision, robustness and improved lifetime of the network. Solutions such as [22], presents a flooding architecture for time synchronization with Glossy, Time slotted channel hopping (TSCH) [23,24], and have demonstrated enhanced reliability. On top of this, for the low power IPv6 stack, 6TiSCH [25] enables schedule management. Most of the algorithms perform message exchange for time synchronization, which involves many elements resulting into sync error. The decomposition of these errors is demonstrated in Fig. 2. Important elements contributing to time sync error is the time lost while building reference time, accessing the channel and radio, propagation time, handling interrupts, reading and writing clock, etc. It is important to understand the requirement of time precision for applications and accordingly keep an account of potential time lost during the synchronization phase.

The basic building block of the any time sync platform is the synchronization phase which consists of message exchange between nodes. There are traditional time synchronization protocols, like Network time protocol (NTP) and Precision time protocol (PTP) [26], Reference broadcast synchronization (RBS) [27], or Time-sync protocol for sensor network (TPSN) [28], which rely on different static sync phase and message sequence to the most recent contributions like [29] having dynamic root selection for reference time. However, it is very difficult to use the aforementioned protocols in LPWAN because of the different possible consequences of offset calculation and possible delay attacks, as in NTP and PTP [30]. In resource constraint IoT devices, the cost of opening the receive window for more period than reception duration results in more energy consumption. So it is important to have optimized synchronization phase and guard time. The passable tuning of the guard time helps to minimize the probability of missing frames. It impacts the throughput and lifetime of the network by energy consumption.

Post synchronization phase, the participating node sets their clock with reference time. However, the clock does not run at the same rate as reference clock and drifts gradually. Thereby, it is important to inspect the clock drift and accordingly re-sync the clock with a reference time.

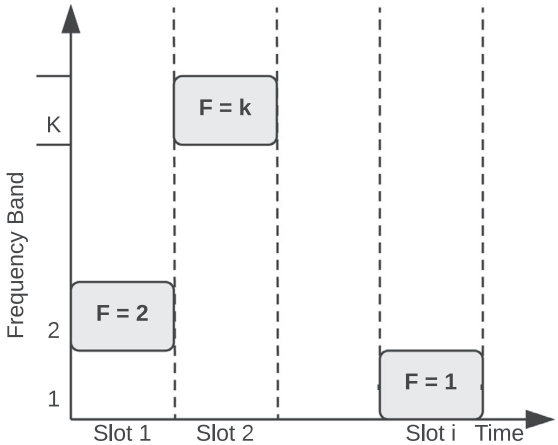

Fig. 3. Channel Hopping

2.2. Cryptographic channel hopping

Frequency hopping helps the devices to communicate using divergent channels at discrete times from the designated spectrum as shown in Fig. 3. CCH is basically a technique of picking pseudo-random channel amid accessible channels as a carrier. At any arbitrary time to communicate, transmitter and receiver should comprehend the channel selection itinerary or alternatively select a common frequency for communication. Generally, for resource-restricted devices in IoT, maintaining discrete lists of channels is inappropriate so software based channel hopping is endorsed. The usage of all the viable channels helps in avoiding collisions and strengthen the reliability [31]. In [32], CCH is mentioned as counter measure for interference issues in networks.

CCH can be obtained by periodically hopping between channels at the same global time using symmetric and synchronous procedure. Example, transmitter and receiver nodes share the key and produce cryptographic pseudo-random channel at given instance. It helps to maintain the *authenticity* by using the shared keys, *integrity* by selection of the same channel on both nodes for communication and finally the *freshness of messages* is achieved by the inclusion of time stamp for communication.

2.3. LoRa and LoRaWAN

LoRa is a radio modulation technology, proprietary and licensed by Semtech Corporation. It operates on 863-870 MHz ISM band in Europe and different frequencies between 470 MHz-925 MHz in variant countries. It exploits a Chirp Spread Spectrum (CSS) modulation method, to empower long-range connectivity [33].

The span of chirp is determined by a parameter called spreading factor (SF) that also manages the sensitivity and transmission speed in LoRa modulation [11]. Higher SF provide slow transmissions with a greater sensitivity that causes longer transmission range and lower data rate due to longer on-air time. LoRa modulation defines the physical layer in the network stack with six different SF to tune the signal according to various application requirements. LoRaWAN is the popular protocol currently used in both research and industry [34]. It explicates the data-link layer and network layer in the stack [5] as in Fig. 4 and utilizes a star networking topology, in which all end-devices transmit the data to the LoRa gateway.

LoRaWAN defines ten channels, eight of which are multi data rate from 250bps to 5.5 kbps, a single high data rate LoRa channel at 11kbps, and a single FSK channel at 50kbps. There are 3 common channels (868.1 MHz, 868.3 MHz, 868.5 MHz) that must be supported by all end-devices and that all gateways must always be receiving on. These three channels are used to join to a network by end-devices.

2.3.1. Time Sync in LoRaWAN

Time synchronization protocols or likewise mechanisms are generally implemented on top of the physical layer to define MAC layer of the network stack. As mentioned earlier, LoRaWAN is implemented as a networking protocol from the LoRa physical layer to the application layer. However, it does not incorporate a time synchronization mechanism. It does not have time-related information in message headers, but only slack timing requirements because of long transmission duration. Class A devices, utilize Aloha-type protocol [12] to send messages to gateways which comes with major bottleneck as : First, Messages from a Class A end device are application-specific. Besides, when there is a message from a gateway to an end-device (downlink message), the end-device should wait until the next uplink transmission to receive it as in Fig. 5 and Fig. 6. Second, After transmitting each message, an end device opens up two receive window for one second each irrespective whether or not it will receive a message from the gateway, which results into a power loss of 14.2 mA [35].

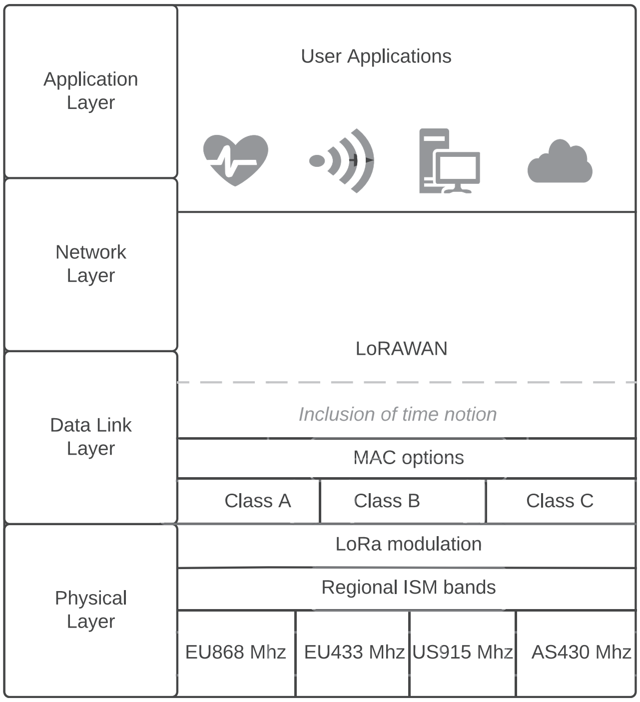

Fig. 4. The LoRaWAN stack

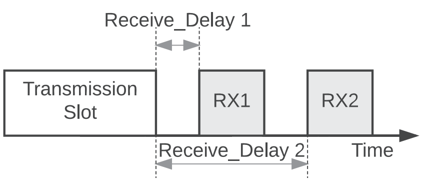

Fig. 5. Class A device Latency

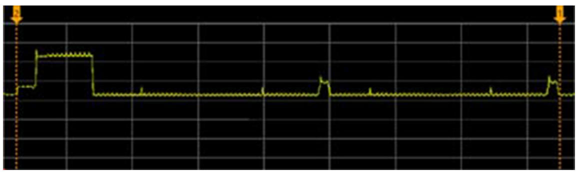

Fig. 6. LoRaWAN message overview on power analyzer

LoRa Alliance has introduced class B type of devices, as time beacon-like devices. It has a periodic receiving window to synchronize the time between end-devices and gateways. However, they still have an additional receive window same like Class A device type. This results in consuming even more power though there is no message for the end-device. Also, Class C type of devices always stay in receive mode, except when they are transmitting. Thereby, spending maximum power. Currently, there are no products of class B or C devices in the market for deployment.

2.3.2. Necessity for synchronization in LoRaWAN network

The combination of integrating time notion with effective and efficient slot allocation helps in achieving reliability and scalability. By characterizing the concept of time, the network becomes more consistent and feeds the possibility for coordination among tasks. Some of the potential benefits of synchronization over the LoRa network is shown in Fig. 7. There are a lot of IoT applications demanding operations guided by coordination and collaboration. This helps in bringing down the collision rate of the network, providing more reliability. LoRaWAN network, is not synchronized, thereby it is difficult to implement command based or event-based applications or in implementing efficient utilization of channels. With the network synchronization, the LoRaWAN network can support the feasibility for multi-hop by using harmonized scheme among nodes. Also, it can boost the security by node specific channel hopping along with power optimization by allowing node

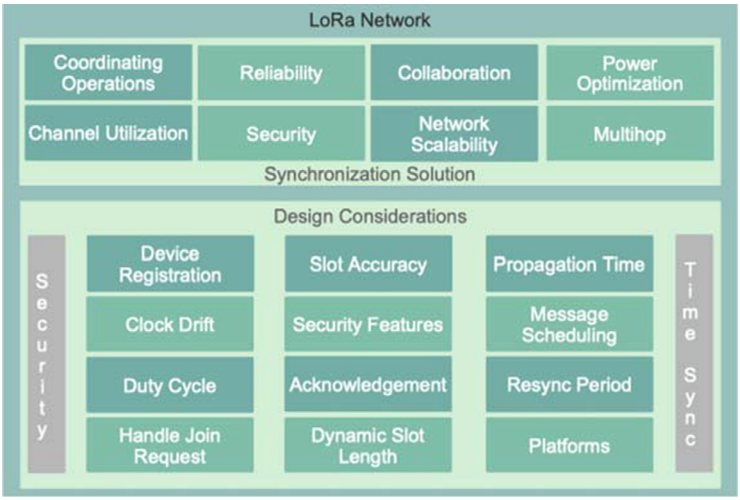

Fig. 7. Design consideration for synchronization in LoRaWAN network

to wake only at dedicated slot. IoT applications stipulating enhanced limits of LoRaWAN, industrial wireless networks and services such as scalable deployment, power efficiency and multi-hop connectivity requires synchronization.

2.3.3. Design consideration for synchronization in LoRaWAN network

There are factors as shown in Fig. 7, which are crucial in designing the synchronized LoRa network. Author Zorbas, [36] describes the important parameters that are required for LoRaWAN based time-slotted protocols, which are discussed in this section. LoRa currently follows strict duty cycle regulations, which results in issues such as transmission delays. Thereby, synchronization design should abide by the regulations as the foremost priority. Synchronization requires exchange of messages wherein slot length in LoRa is crucial as transmission time increases with SF. Usage of multiple frames can be a solution, but will result into collisions [37], this can be further solved by using different power configurations [34] or dedicated control channels. Another, perspective for time sync is utilization of acknowledgements (ACK) for transmissions which should adhere by the limited duty cycle along with the appropriate frame size [38]. Since ACK packets are short, so grouping can be a possible solution to save energy and abide by duty cycle. Joining is an important process in LoRa network, for large scale deployment the responsiveness and activation mechanism needs to be energy and time-efficient. Once the network is synchronized, scheduling is the fundamental issue which requires fair arrangement of the slots [39]. Due to limited downlink availability, assigning or scheduling to large number of nodes can be time consuming. This needs, an efficient mechanism from gateway side. After scheduling the network, two important aspects are handling of propagation time and clock drift. Mainly since, LoRa supports long range communication with larger on air time and variable size of time slot. This can be solved by either localizing the end device to anticipate the distance from the gateway or integrating propagation delay to the guard time. Lastly, the security aspect for authentication, encryption and integrity has to be done with an efficient manner by secure registration of device and channel hopping technique [40]. Currently, security is achieved by pre-installed security keys or during registration mechanism which needs better mechanism as attacker can synchronize with the network and can jam the network synchronization slot.

2.3.4. Relevant contribution

Recently, there has been rise in contribution to fill the gap of time notion in LoRa devices. The motivation for achieving synchronization in LoRa network along with the potential approaches are demonstrated in the Fig. 8. In [11], the explicit requirement for synchronization and scheme for channel selection is mentioned as mitigation for limits of LoRaWAN. Koichi et al. [41], proposes the crop specific communication protocol as field server for time synchronization and to save power consumption. In the recent articles [42,43], authors used time-slotted communication approach over LoRaWAN enabling nodes to self organize the time slot schedule. The experiment showed high packet delivery but lacked to tackle the cost to maintain acknowledgements by analyzing the simulation result on the networking stack of Contiki-NG. Author Laurent et al. [44], introduces a new Class S type of device dedicated for synchronization enabling TDMA access in LoRaWAN network.The authors of the paper [40,45], proposed the possibility for multi-hopping using synchronization by adaptation of TSCH protocol, diversifying transmissions over different channels and using different topologies like clustering and forwarding relay node. Another important perspective while evolving synchronization is the trade-off between energy efficiency and the uncertainty in the proposed algorithm. In the paper [46], author proposes a posteriori algorithm to study the trade off curve for energy consumption and paper [47] proposes LongShoT scheme for synchronization with drift correction. The authors of paper [37] focuses on energy consumption by retransmitting bitmaps instead of whole frames and in

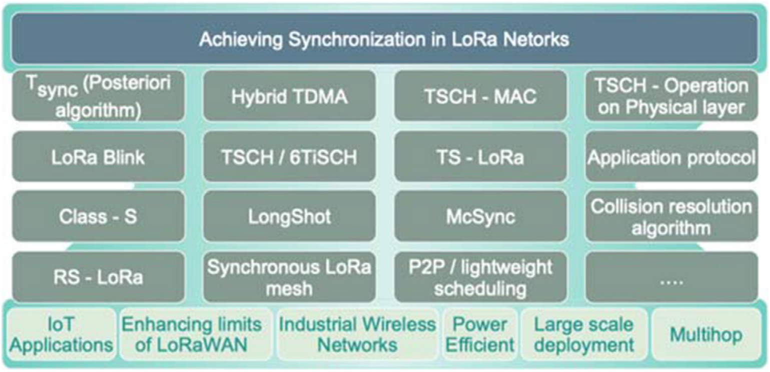

Fig. 8. Different approach to gain synchronization in LoRa networks.

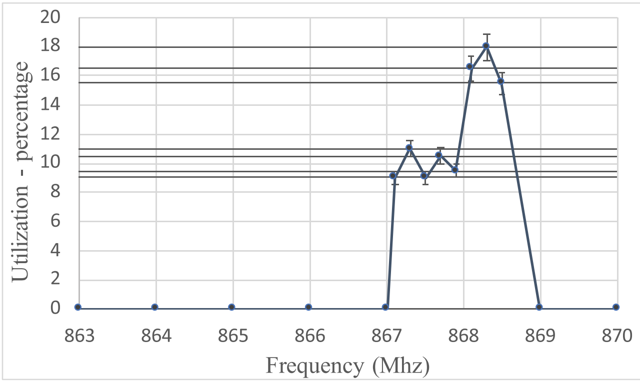

Fig. 9. LoRaWAN traffic Analysis

paper [48], new multi-channel synchronization-McSync approach is proposed for high performance. There are few contributions in regard to the specific application use case. Author Christian et al. [38] has addressed the challenges for time slot based transmissions for underground sensor nodes and paper [49], targets on Industry 4.0 applications by adaptation of TSCH mechanisms. Most of the aforementioned approaches are simulated or they need a server but none provides the provision for time synchronization among LoRaWAN nodes.

2.3.5. Channel hopping in LoRaWAN

LoRaWAN uses a pseudo-random channel hopping method to amplify overall network magnitude and avoid accidental collisions by disseminating transmission over the pool of available channels. The possibility of a channel selection is decided by duty-cycle limitations (i.e., the duty-cycle is 1% in EU 868 MHz for LoRaWAN end-devices) [11,5]. As mentioned in Section 2.3, LoRaWAN operates on 8 channels between 863-868 MHz in Europe. Each channel is partitioned by 200 kHz from each other to support 125 kHz bandwidth as defined in the LoRaWAN specification. In order to examine channel utilization in LoRa frequency spectrum, an analysis of LoRaWAN traffic was performed using a MultiTech Gateway[1] and its logs to scope the channel usage of LoRa transmissions in regular use. LoRaWAN traffic was recorded for 3 days, totalling 692 messages. The result of traffic analysis is shown in Fig. 9.

We observe: 1. Most of the spectrum is not utilized. Moreover, even though there are eight channels defined in the protocol, only three default ones are mostly used by end-devices; 2. Also, it affects security of legitimate communication and gives a scope for better channel hopping scheme.

3. System Model - Implementation

In this section, we unfold the proposed solution for time synchronization and CCH in LoRaWAN network.

[1] http://www.multitech.com/brands/multiconnect-conduit.

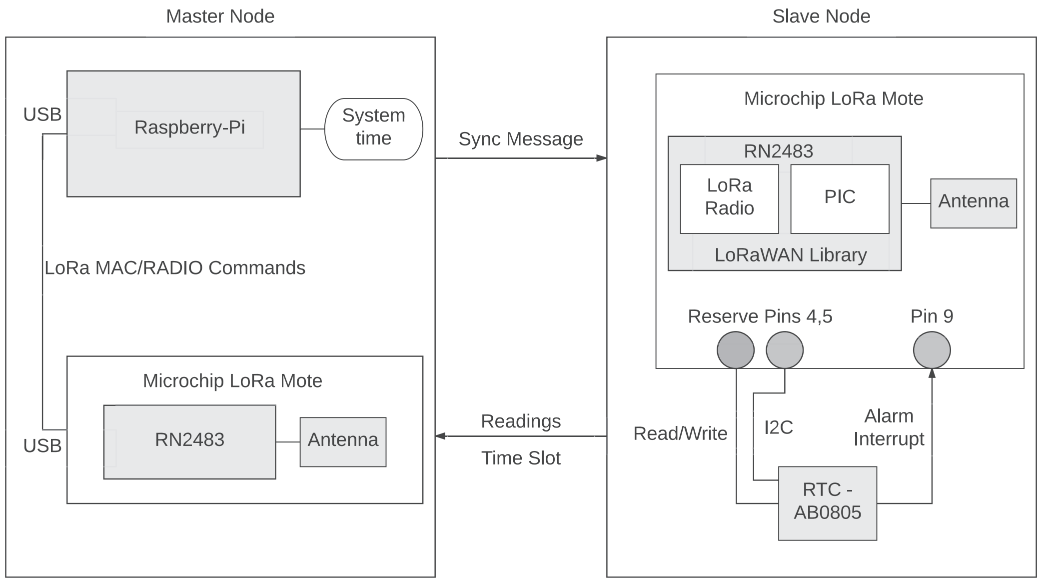

Fig. 10. System Architecture

3.1. Proof of concept and System architecture

For the proof of concept setup and to demonstrate the possibility of bringing time notion in the LoRaWAN network, we have considered a P2P model (two devices) which can be further scaled up to a full fledged network. Also, we calculate the next pseudo-random frequency on both the master and slave before performing the subsequent communication between them to adhere to CCH. The motivation behind this setup is to sync both the devices and reduce the guard time while guaranteed message delivery. This can eventually result in saving power and increasing the reliability of the network. Once both the devices are time synchronized, it becomes easy to develop a time-critical or time constraint application.

For the implementation we have taken, two Microchip Class A LoRa mote as end devices that support 868 MHz High-Frequency[2], demonstrated in Fig. 10. Detailed hardware setup using LoRa mote as master and slave is as follows:

Master Node This device is connected to the Raspberry Pi, which picks the system time as a reference time and manages the ongoing messaging between end devices. A script (high priority process) does the sending and receiving of data from the LoRa end device using LoRa radio and mac commands [50].

Slave Node This LoRa device has PIC inside RN2483 (PIC-18LF46K22) which is connected with a RTC (ABRACON-AB0805 [51]) and programmed using Pickit3 tool and MPLAB IDE. Programming PIC helps to avoid any further inclusion of hardware. The RN2483 module have two reserved pins as mentioned in its datasheet [35]. These pins are used to set up an I2C communication with RTC. This RTC is significantly ultra low powered (as low as 14 nA [51]) with an I^2C and SPI interface having precision up to centiseconds. The clock and alarm feature of the RTC is mainly used for time sync platform.

3.2. Clock synchronization

Clock synchronization is done in two steps using P2P model. Firstly, the initial investigation in most simplified way and then considering the insights of former approach for developing refined algorithm. Network joining of the device is out of the scope for this paper and considered to be a prerequisite. Simplified messaging between master and slave is shown in Fig. 11. Both nodes starts with default frequency, power and SF as defined in the joining process. Then the master node sends the reference time (R_t) as sync message to the slave. This R_t is composed of 7 bytes of data, which contains time stamp in the form of centiseconds, seconds, minutes, hours, date, month, and year. The slave, on receiving the sync packet writes the time value to the appropriate register in RTC. For better understanding and evaluation, we have predefined the alarm configurations to get the interrupt after each minute.

While sending a R_t to slave device, some time gets lapsed as computation time (C_t) and on air time (Section 2.1). We calculated the on air time (OA_t) for different SF and payloads as in Fig. 13. These two values, C_t and OA_t, are constant as in Fig. 14 for a given type of message, which is necessary to be added with the R_t. Thereby, instead of writing the received sync message we wrote "α"; $\alpha = R_t + OA_t + C_t$. Next, we calculated the clock drift over a given time by mapping it with R_t. It is important to contemplate and calculate the clock drift to maintain the optimal guard time and re-sync period. Fig. 15,

[2] www.microchip.com/RN2483LoRaMote4233989.

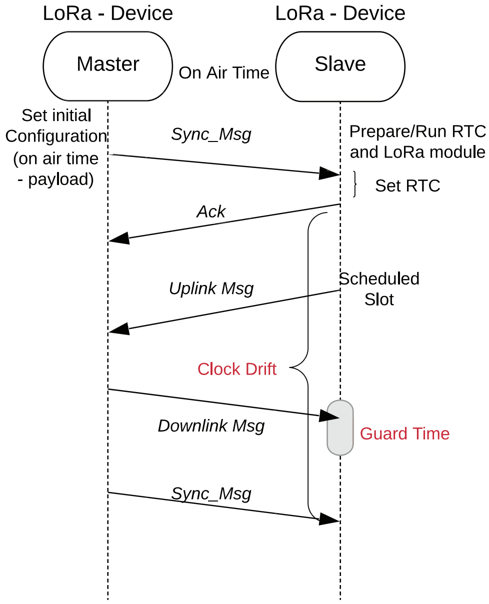

Fig. 11. Message sequence for Time Sync

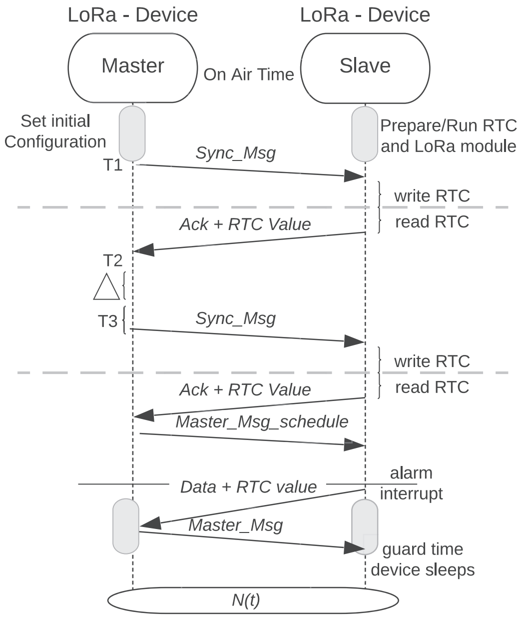

Fig. 12. Time sync protocol

shows drift for all SF, which is clock drifts for 50 millisecond in one hour. Thereby, it can drift to 1 second in 20 hours. For a scenario where application requires time precision of one second then in worst scenario device has to sync up with in every 20 hours in the current setup.

Further, we observed prospective effects of different parameters like power configuration or changing device internal frequency on C_t or OA_t. Power configuration has no footprint. However, changing internal frequency from 16 MHz to 4 MHz resulted variation in the C_t but not on the clock drift. We got the time synchronization with a resolution of 55 ms between master and slave. This variation in resolution was due to hardware dependency. Key take away points for developing the algorithm further were that it should be hardware independent, tackle automated re-sync, no impact of drift, and provision of message classification from master.

Second step, we developed time sync algorithm considering the aforementioned learning. The message sequence for this algorithm is shown in Fig. 12 with the detailed steps mentioned as below. Acronym used are: sync message (S_m),

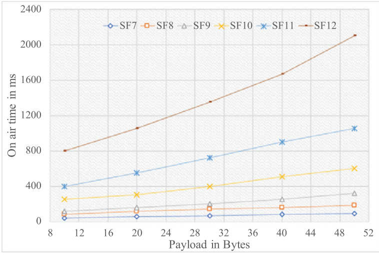

Fig. 13. On air time for different payload

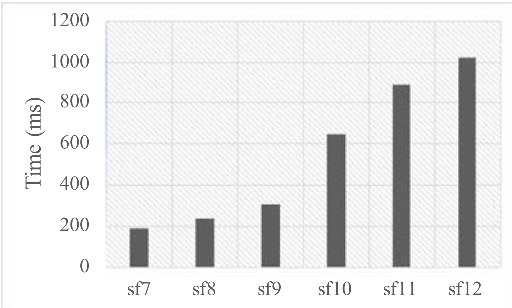

Fig. 14. OA_t+C_t for different SF

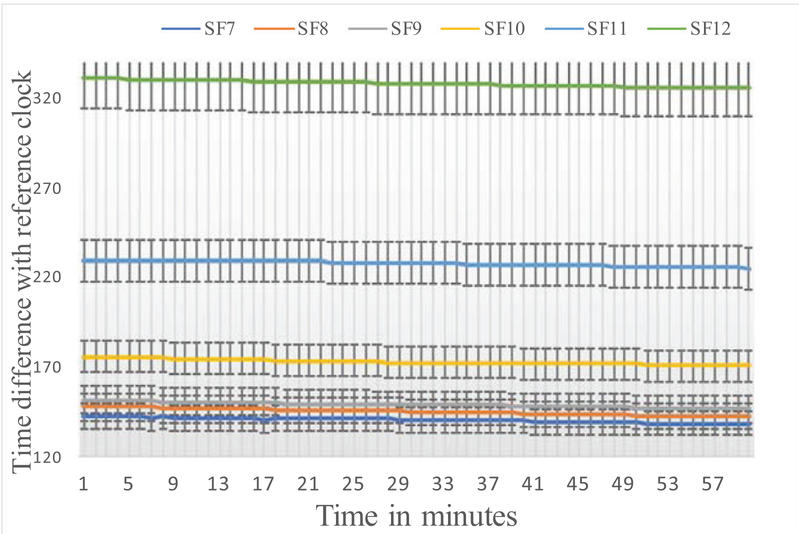

Fig. 15. Clock Drift for different SF

Table 1
Format for sync message

78	23	34	04	08	12	18	11	12
cs	sec	min	hr	day	mn	yr	id	type

acknowledgement (A_k), RTC value (R_v), RTC read (R_r), RTC write (R_w), RTC value (R_v), schedule message (M_s), network time (N_t), alarm interrupt (A_i), guard time (G_t), network time (N_t), master device (M_d), slave device (S_d).

This algorithm is hardware agnostic. The M_d sends S_m to S_d in the format displayed in Table 1 comprising of one byte for each field and stores time (T_1). The 'id' and 'type' field represents the network/master id and type of message respectively as mentioned in Fig. 16. Each message carries additional information apart from payload for message identification and required processing in the format given in Fig. 16. It carries the information of sender (ID), type of message to accordingly read the payload and type of message. The id details are stored in joining process and helps in avoiding anonymous

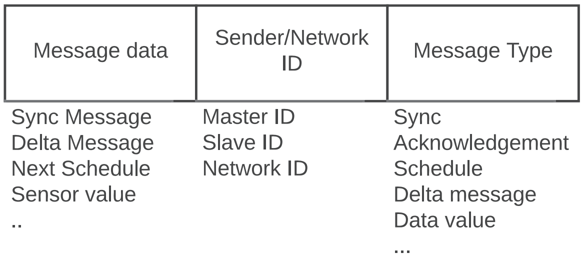

Fig. 16. Message format

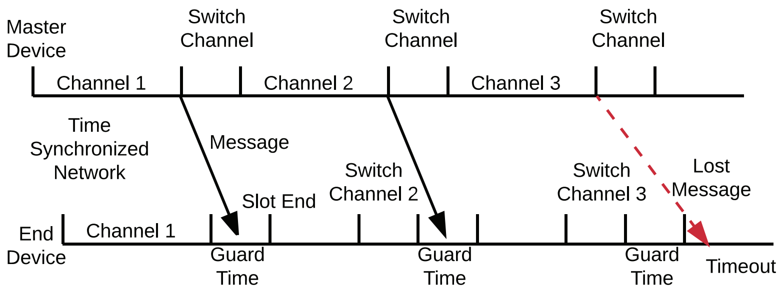

Fig. 17. Channel Hopping

messages. S_d, on receiving the message parses id and message type to perform R_w and immediately R_r followed by sending A_k. M_d receives A_k and stores this time (T_2). It calculates the difference of both the stored time slot and divides by two to get the delta (OA_t+C_t) which can be added to R_t for sending the S_m again. M_d sends the M_s as in Fig. 12, for receiving the sensor data. S_d, configures alarm registers to receive interrupt. We have configured one of the pins for device wakeup on alarm interrupt and send the message. On receiving the interrupt, S_d sends sensor data and R_v. M_d uses the R_v to map with application requirement for required resync and accordingly replies back. For each communication, S_d and M_d open their guard window to receive message and set their common frequency using CCH.

3.3. CCH

During time synchronization and further for each communication we implemented and used CCH to select next channel. After each message communication, master and slave close their radio modules (state:off) and select their next channel as demonstrated in Fig. 17. We have written a common function (*select_freq*) for CCH, that runs on both master and slave device. Both the end devices are loaded with common keys (K_n), plain text (P_t) and nonce (part of joining process). The *select_freq* function uses AES CTR mode (AES is already present in LoRaWAN stack, so using the same will result into light weight solution) to generate a pseudorandom channel from the available spectrum because of its important random access property. The cost perspective of using this cryptographic function (AES CTR) is very low [52], since AES and record of utilized channel is already part of LoRaWAN stack. The CCH, algorithm for channel selection works as follows:

1. Frequency spectrum for EU, $F = \{f_i\}$ for $863\,000\,000 \leq i \leq 870\,000\,000$, $f \in I$ where f_i has to be selected for the next communication.
2. Use AES-CTR mode with 256 *bit* K_n to produce stream cipher using sequential counter block of size 16 octets, comprising of counter, nonce and Initialization vector.
3. Give input, P_t of 16 octets for XORing with key stream and to get *cipher*
4. Calculate the sum(*cipher*), i.e., $\sum_{i=1}^{16} cipher$ which is always random because of inherent basic property of using 14 rounds.
5. With above results, we calculate the f_i using logic - $Sum()\%(upperbound - lowerbound) + lowerbound$
6. The lower and upper bound are the f_i range.

4. Evaluation

In this section we unfold the achieved sync resolution and guard time using our algorithm. Also, we analyze the channel statistical uniformity and accomplished channel utilization and distribution using CCH.

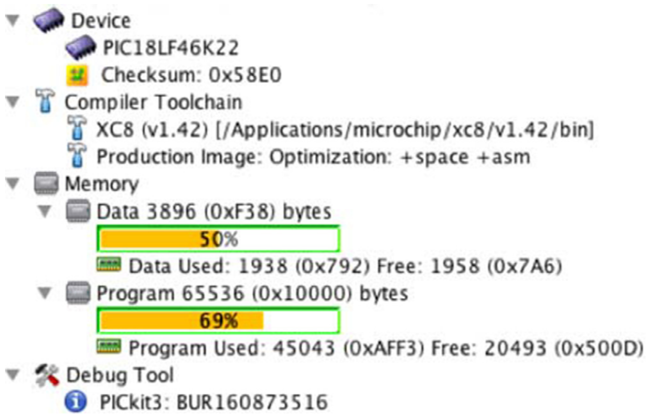

Fig. 18. Memory consumption for time sync and CCH

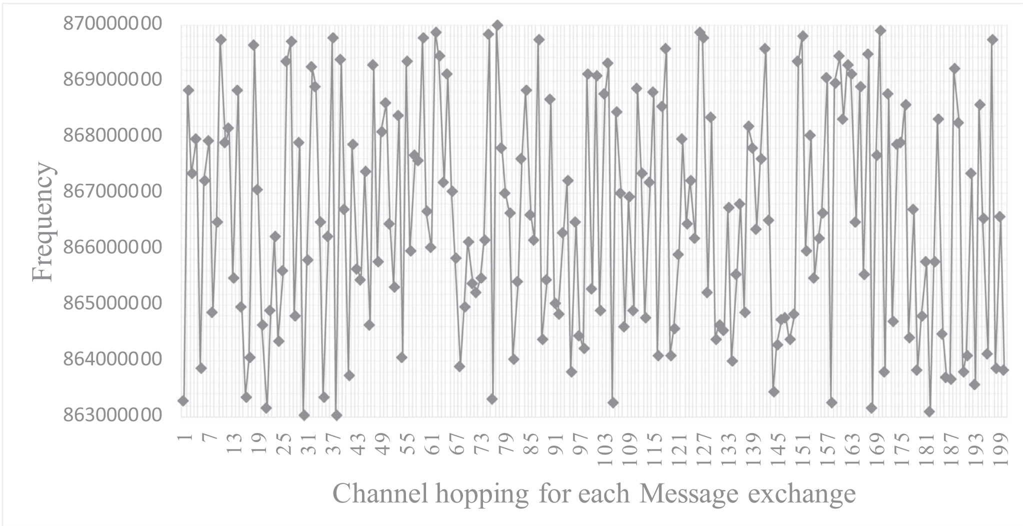

Fig. 19. CCH for different rounds of message

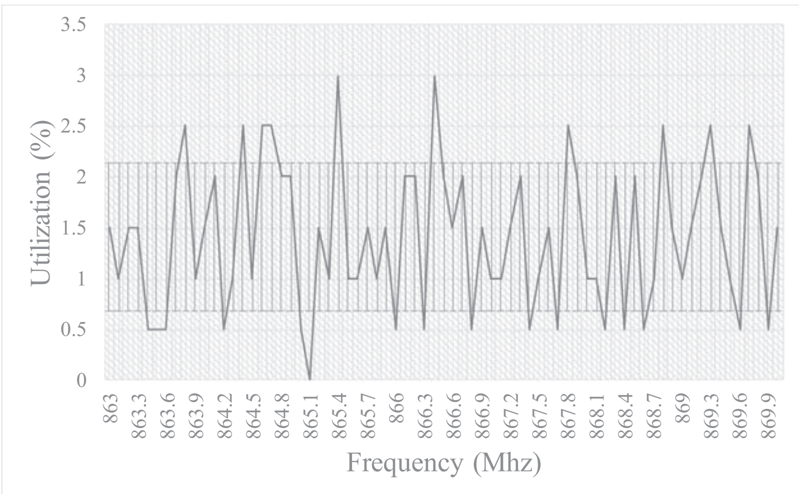

Fig. 20. Channel utilization and distribution

Algorithm 1 Time Sync algorithm

1: **procedure** JOIN NETWORK(S_d, $network$) ▷ pre-requisite
2: **procedure** TIME SYNC(S_d, M_d) ▷ time sync with master
3: $S_d \leftarrow config$ ▷ default network freq,Power,SF..
4: $S_d \leftarrow M_d(S_m, T_1)$ ▷ T_1 is the master reference time
5: $R_w(S_m, Reg) \leftarrow S_d$; $R_r(value, Reg) \leftarrow S_d$ ▷ RTC
6: $M_d(T_2) \leftarrow S_d(A_k, R_v)$ ▷ ack and RTC value
7: $\triangle((T_2 - T_1)/2) \leftarrow M_d$ ▷ $\triangle$ i.e delta is calculated
8: $S_d \leftarrow M_d(S_m(T_3 + \triangle))$ ▷ T_3 is new reference time
9: repeat step(5 & 6) ▷ Set rtc with new reference time
10: $S_d \leftarrow M_d(M_s)$ ▷ schedule information
11: $S_d(ISR) : M_d(ISR)$ ▷ Alrm interrupt on both device
12: $M_d(S_d : Ch) \leftarrow S_d(data, R_v)$ ▷ data and RTC value
13: $S_d \leftarrow M_d(A_k, S_m \text{ or } M_s)$ ▷ application specific

4.1. Guard time and Sync resolution

In Section 3, we calculated the on air time for different SF and payload size. Also, we calculated the clock drift for the RTC which is important for re-sync period. These results help to determine further the optimal guard time and the sync resolution with reference clock between master and slave using the proposed time sync algorithm. We got time sync resolution of 10 ms(clock difference with reference time) which is the minimum measurement resolution with the given RTC (centiseconds). Changing the RTC may even bring down the sync resolution to a lower value. We calculated the guard time to be 30 ms in addition to delta on master side which is dynamically picked for each instance of communication with the end device. Also, the guard time is kept to be twice delta in addition to 30 ms on slave side. This dynamic change in guard time with respect to SF can save energy by reducing the default receive window. We performed this experiment for a longer duration with several communications and observed no packet loss. Although, the sync message itself requires some amount of energy but when compared to efficient guard time and added benefits, the cost of sync message becomes negligible. To find the energy consumption added cost required due to time sync mechanism, we referred to the paper [53] on energy consumption model along with the analysis of LoRaWAN message on power analyzer as shown in Fig. 6. We found that transmission of one LoRaWAN message from end node was taking 0.911mJ of energy and received downlink message from the gateway. Also, 1.193mJ energy was spent on packet transmission without receiving downlink message. The extra cost for time sync in respect to overhead for transmission of packets is 1.822mJ. For a scenario when applications like, agricultural drones or livestock monitoring sends a 20 message each hour then they spend 572.64mJ of energy in a single day, while using time sync platform they initially spend some energy on sync but later can save energy by avoiding redundant messages thought out the day and sending only limited scheduled messages.

4.2. Channel utilization and Distribution using CCH

We evaluated the CCH frequency selection for each round of communication as in Fig. 19. This graph shows the randomness in channel selection. We did the statistical information entropy calculation [54] to check the uniformity which came as 5.9478 bits out of maximum 6.1497 bits. It shows nice uniformity in channel selection. The Fig. 20 shows equal distribution and utilization of the available channels for each round of communication. In research perspective, we have not considered EU duty cycle regulation for now. Also, we calculated the cost of implementing time sync and CCH scheme in regards to memory consumption on the device as in Fig. 18. It was 242 bytes for the implementation of time sync and 717 bytes for CCH. The computational cost added by by time sync and cryptographic function is 6% and 18% respectively. This stands out to be very low and a perfect fit for LoRa end devices.

5. Conclusion

Inclusion of time synchronization in a LoRaWAN network can be beneficial for several application domains. Time-critical applications like transportation, precision agriculture, or smart homes require a time critical and bidirectional scheduled approach for messaging. This platform can allow LoRa devices to send and receive messages in allocated slots, reducing interference and collision. Optimal utilization of the full-spectrum will increase the reliability. More intensive experiments to calculate the drop in collision will be done as a part of future work. For instance, we are in a EU Interreg project on greenhouse monitoring (Grow!) [55], where we can have many slave LoRa devices and one master LoRa device as demonstrated in Section 3. This enables a reduction in the deployment cost by replacing the expensive gateway with an end device and giving a dedicated slot for all devices along with connectivity to the back-end. In this paper, we presented a novel time synchronization platform for LoRaWAN network, to achieve reliability and robustness. The objective of this paper is to fill two gaps in LoRaWAN; time sync and CCH by proposing a solution and later plan to perform extensive experiments on power utilization and collision improvement (reliability). It will be interesting to deploy several master/slave nodes in an environment to check the fault tolerance and perform the empirical calculation on reliability and energy cost. We address a key problem of LoRaWAN for the availability and listening time of down link message. On top of the time cooperative approach, we propose cryptographic channel hopping to reduce the packet collision and enhance the security of the network. In addition, energy saving is achieved by lowering the guard time for message listening.

Encouraged by the results of the proposed time sync platform, we plan to extend our work towards more strategic research. This scope becomes even more substantial after the release of the recent LoRaWAN specification, with new commands for timing which reflects the inclination of the community towards the inclusion of time notions. However, it will take time before this is commercially available. We will focus on scaling the time sync network and measure the performance, especially coverage, capacity, and collision. We will first focus on dynamic task scheduling or time sensitive middleware which will reside on top of this platform and extend our evaluation. Next, we will try to explore joining procedures for end devices in time sync networks. Finally, we want to use this time crucial system for specific applications. On the other hand, on top of CCH we will try to come up with a channel hopping scheme to reserve set of channels for critical packets. We will also take into consideration the limitation of available device classes defined by LoRaWAN, and would try to evolve with an efficient new time-sensitive device type; enriching the possible use-case and applications of LoRa network.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Centenaro, L. Vangelista, A. Zanella, M. Zorzi, Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios, IEEE Wireless Communications 23 (5) (2016) 60–67.
[2] A.-A. A. Boulogeorgos, P. D. Diamantoulakis, G. K. Karagiannidis, Low power wide area networks (lpwans) for internet of things (iot) applications: Research challenges and future trends, 2016,
[3] Niklas Heuveldop, Ericsson mobility report, AB Ericsson, Stockholm, Sweden (2017).
[4] R.S. Sinha, Y. Wei, S.-H. Hwang, A survey on lpwa technology: Lora and nb-iot, ICT Express 3 (1) (2017) 14–21, doi:10.1016/j.icte.2017.03.004.
[5] LoRa Alliance, Inc. lorawantm specification, LoRa Alliance, Inc.: San Ramon, CA, USA(2015).
[6] Sigfox, Sigfox homepage (2016).
[7] U. Noreen, A. Bounceur, L. Clavier, A study of lora low power and wide area network technology, in: 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2017, pp. 1–6, doi:10.1109/ATSIP.2017.8075570.
[8] A. Mdhaffar, T. Chaari, K. Larbi, M. Jmaiel, B. Freisleben, Iot-based health monitoring via lorawan, in: IEEE EUROCON 2017 -17th International Conference on Smart Technologies, 2017, pp. 519–524.
[9] L.A.T.M. Workgroup, A technical overview of LoRa and LoRaWAN, Technical Report, LoRa Alliance, 2015.
[10] i scoop, Lorawan across the globe (2017).
[11] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, T. Watteyne, Understanding the limits of lorawan, IEEE Communications Magazine 55 (9) (2017) 34–40.
[12] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, J. Alonso-Zarate, Goodbye, aloha!, IEEE Access 4 (2016) 2029–2044, doi:10.1109/ACCESS.2016.2557758.
[13] P. Neumann, J. Montavont, T. Noël, Indoor deployment of low-power wide area networks (lpwan): A lorawan case study, in: 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2016, pp. 1–8.
[14] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, A comparative study of lpwan technologies for large-scale iot deployment, ICT Express (2018), doi:10.1016/j.icte.2017.12.005.
[15] L. Lin, S. Ma, M. Ma, A group neighborhood average clock synchronization protocol for wireless sensor networks, Sensors (2014).
[16] S.M. Lasassmeh, J.M. Conrad, Time synchronization in wireless sensor networks: A survey, in: Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp. 242–245.
[17] R.K. Singh, R. Berkvens, M. Weyn, Time synchronization with channel hopping scheme for lora networks, in: L. Barolli, P. Hellinckx, J. Natwichai (Eds.), Advances on P2P, Parallel, Grid, Cloud and Internet Computing, Springer International Publishing, Cham, 2020, pp. 786–797.
[18] I. Hussain, Z.I. Ahmed, D.K. Saikia, N. Sarma, A qos-aware dynamic bandwidth allocation scheme for multi-hop wifi-based long distance networks, EURASIP Journal on Wireless Communications and Networking 2015 (1) (2015) 160, doi:10.1186/s13638-015-0352-z.
[19] R.K. Singh, M. Aernouts, M. De Meyer, M. Weyn, R. Berkvens, Leveraging LoRaWAN Technology for Precision Agriculture in Greenhouses, Sensors 20 (7) (2020) 1827, doi:10.3390/s20071827.
[20] A. Proano, L. Lazos, Selective jamming attacks in wireless networks, in: 2010 IEEE International Conference on Communications, 2010, pp. 1–6, doi:10.1109/ICC.2010.5502322.
[21] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M.L. Wymore, A.K. Sangaiah, A robust time synchronization scheme for industrial internet of things, IEEE Transactions on Industrial Informatics 14 (8) (2018) 3570–3580, doi:10.1109/TII.2017.2738842.
[22] F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, Efficient network flooding and time synchronization with glossy, in: Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2011, pp. 73–84.
[23] T. Watteyne, V. Handziski, X. Vilajosana, S. Duquennoy, O. Hahm, E. Baccelli, A. Wolisz, Industrial wireless ip-based physical systems, Proceedings of the IEEE 104 (5) (2016) 1025–1038.
[24] IEEE, Standard for low-rate wireless networks(802.15.4-2015).
[25] X. T. E. et al., An architecture for ipv6 over the tsch mode, IEEE 802.15.4 - draft-ietf-6tisch-architecture (10, June 2016).
[26] T. Neagoe, V. Cristea, L. Banica, Ntp versus ptp in com puter networks clock synchronization, in: 2006 IEEE International Symposium on Industrial Electronics, 1, 2006, pp. 317–362, doi:10.1109/ISIE.2006.295613.
[27] J. Elson, L. Girod, D. Estrin, Fine-grained network time synchronization using reference broadcasts, SIGOPS Oper. Syst. Rev. 36 (SI) (2002) 147–163, doi:10.1145/844128.844143.
[28] S. Ganeriwal, R. Kumar, M.B. Srivastava, Timing-sync protocol for sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, in: SenSys '03, ACM, New York, NY, USA, 2003, pp. 138–149.
[29] T. Qiu, X. Liu, M. Han, M. Li, Y. Zhang, Srts : A self-recoverable time synchronization for sensor networks of healthcare iot, Computer Networks 129 (Part 2) (2017) 481–492. Special Issue on 5G Wireless Networks for IoT and Body Sensors
[30] M. Ullmann, M. Vögeler, Delay attacks implication on ntp and ptp time synchronization, in: 2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2009, pp. 1–6, doi:10.1109/ISPCS.2009.5340224.

[31] X. Ji, Y. He, J. Wang, K. Wu, D. Liu, K. Yi, Y. Liu, On improving wireless channel utilization: A collision tolerance-based approach, IEEE Transactions on Mobile Computing 16 (3) (2017) 787–800, doi:10.1109/TMC.2016.2567380.
[32] H.K. Kalita1, A. Kar, Wireless sensor network security analysis, International Journal of Next-Generation Networks(IJNGN) 1 (2009).
[33] L. Vangelista, A. Zanella, M. Zorzi, Long-range iot technologies: The dawn of lora, Future Access Enablers of Ubiquitous and Intelligent Infrastructures (2015).
[34] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, M. Gidlund, Using lora for industrial wireless networks, in: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), 2017, pp. 1–4, doi:10.1109/WFCS.2017.7991972.
[35] Microchip, RN2483 LoRa Technology Transceiver Module Data Sheet, Technical Report, 2015.
[36] Dimitrios Zorbas, Design Considerations for Time-Slotted LoRa(WAN), MaDeLoRa 2020: 1st Workshop on Massive LoRa Deployments: Challenges and Solutions, ACM, Lyon, France, 2020.
[37] S. Abboud, N.E. Rachkidy, A. Guitton, Efficient Decoding of Synchronized Colliding LoRa Signals (2019).
[38] C. Ebi, F. Schaltegger, A. Rust, F. Blumensaat, Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure, IEEE Access 7 (2019) 57663–57677, doi:10.1109/ACCESS.2019.2913985.
[39] Brecht Reynders, et al., Improving Reliability and Scalability of LoRaWANs through Lightweight Scheduling 5 (2018) 1830–1842, doi:10.1109/JIOT.2018.2815150.
[40] M. Bezunartea, R. Van Glabbeek, A. Braeken, J. Tiberghien, K. Steenhaut, Towards energy efficient LoRa multihop networks, IEEE Workshop on Local and Metropolitan Area Networks 2019-July (2019) 22–24, doi:10.1109/LANMAN.2019.8847137.
[41] K. Tanaka, M. Sode, T. Ozaki, M. Nishigaki, T. Mizuno, I. Science, F. Server, A Study on Time Synchronization Method for Field Servers for Rice Cultivation 11 (1) (2019) 33–44.
[42] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, D. Pesch, TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things, Computer Communications 153 (October 2019) (2020) 1–10, doi:10.1016/j.comcom.2020.01.056.
[43] M. Haubro, C. Orfanidis, G. Oikonomou, X. Fafoutis, TSCH-over-LoRA: Long Range and Reliable IPv6 Multi-hop Networks for the Internet of Things, Internet Technology Letters (2020) e165, doi:10.1002/itl2.165.
[44] L. Chasserat, N. Accettura, P. Berthou, L. Chasserat, N. Accettura, P. Berthou, S. Achieving, L. Chasserat, N. Accettura, P. Berthou, Short : Achieving energy efficiency in dense LoRaWANs through TDMA To cite this version : HAL Id : hal-02551973 Short : Achieving energy efficiency in dense LoRaWANs through TDMA(2020).
[45] M.S. Aslam, A. Khan, A. Atif, S.A. Hassan, A. Mahmood, H.K. Qureshi, M. Gidlund, Exploring Multi-Hop LoRa for Green Smart Cities, IEEE Network 34 (2) (2020) 225–231, doi:10.1109/MNET.001.1900269.
[46] M. Rizzi, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, E. Sisinni, Synchronization Uncertainty Versus Power Efficiency in LoRaWAN Networks, IEEE Transactions on Instrumentation and Measurement 68 (4) (2019) 1101–1111, doi:10.1109/TIM.2018.2859639.
[47] C.G. Ramirez, A. Sergeyev, A. Dyussenova, B. Iannucci, Longshot: Long-range synchronization of time, IPSN 2019 - Proceedings of the 2019 Information Processing in Sensor Networks (2019) 289–300, doi:10.1145/3302506.3310408.
[48] K. Zhang, A passive multi-channel synchronization solution for IoT, Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM 03-07-Octo (2016) 42–44, doi:10.1145/2987354.2987366.
[49] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, M. Gidlund, Using LoRa for industrial wireless networks, IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS (2017) 13–16, doi:10.1109/WFCS.2017.7991972.
[50] Microchip, RN2483 LoRaTM Technology Module Command Reference User's Guide, Technical Report, 2015.
[51] Abracon, AB08X5 Real-Time Clock Family, Technical Report, October 16, 2014.
[52] A. Prathiba, V.S.K. Bhaaskaran, Fpga implementation and analysis of the block cipher mode architectures for the present light weight encryption algorithm, Indian Journal of Science and Technology 9 (38) (2016).
[53] T. Bouguera, J.F. Diouris, J.J. Chaillout, R. Jaouadi, G. Andrieux, Energy consumption model for sensor nodes based on LoRa and LoRaWAN, Sensors (Switzerland) 18 (7) (2018) 1–23, doi:10.3390/s18072104.
[54] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, UK, 2003, doi:10.2277/0521642981.
[55] Interreg grow! project, (https://www.grensregio.eu/projecten/grow). Accessed: 2020-02-20.