TREMA: A traffic-aware energy efficient MAC protocol to adapt the LoRaWAN capacity

Laurent Chasserat, Nicola Accettura, Balakrishna Prabhu, Pascal Berthou

TREMA: A traffic-aware energy efficient MAC protocol to adapt the LoRaWAN capacity

Laurent Chasserat, Nicola Accettura, Balakrishna Prabhu and Pascal Berthou
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
Email: {firstname.lastname}@laas.fr

***Abstract*—The emerging LoRa technology is quickly becoming the *de facto* standard for Low Power Wide Area Networks upon unlicensed frequencies. Herein, the LoRaWAN medium access sets up a lightweight network architecture able to connect very low power devices to the Internet. Traffic flows in such deployments can be variable, or even unpredictable, depending on the needs of the monitoring applications using the network. As an example, to track air quality in cities, some applications can trigger an increased need of fine grained pollution data during the daytime. However, the network capacity is currently limited by the default LoRaWAN pure ALOHA access scheme. A time synchronized scheduled access would considerably improve the achievable throughput, at the cost of an increased power consumption for synchronization duties. In such a context, this contribution introduces the traffic-aware energy efficient Medium Access Control (TREMA) protocol for LoRa networks, capable of seamlessly switching between asynchronous and synchronous schemes according to the probed traffic variations. TREMA ultimately increases the maximum capacity of LoRa deployments while always selecting the most energy efficient access scheme.**

I. Introduction

In the last decade we have witnessed relevant technological progresses in radio communications and electronic miniaturization techniques. Together they unveiled the possibility to monitor physical phenomena at large scale [1]. Low power wireless devices have been then leveraged to deploy widely spread radio networks. However, battery replacement can be very expensive, and in most cases unfeasible in networks with potentially thousands of devices, so maximizing their lifetime becomes the only viable solution. Herein, the Low Power Wide Area Network (LPWAN) architecture has been largely adopted for long range, low throughput and energy efficient data collection with relatively simple and cheap devices. Several LPWAN technologies have emerged, including Long Range (LoRa) networks. They are promoted by the LoRa Alliance, which is a consortium gathering more than 500 companies to drive the open development of the LoRa Wide Area Network (LoRaWAN) specification [2]. LoRa deployments display many interesting features such as bidirectional communications and end-to-end encryption, making them suitable for a large variety of applications. Moreover, LoRaWANs are built upon unlicensed Industrial, Scientific, Medical (ISM) radio bands enabling easy prototyping and deployment.

The LoRaWAN Medium Access Control (MAC) scheme was designed as pure ALOHA for uplink communications, meaning that upon frame generation, devices immediately start transmitting without checking whether the radio channel is free. Such a random access scheme does not require synchronization, so power is saved and circuitry is kept simple. However, this scheme suffers from high collision rates that naturally limit the channel throughput to 18% at most [3]. A low throughput is thus traded-off for a low power consumption. Several works in the literature have proposed channel access improvements to increase the scaling capabilities of LoRa, many of them based on synchronization and the consequent possibility to slice time into slots. Such enhancement allows to increase the maximum throughput, but we showed in a previous contribution [4] that the additional energy consumption induced by the process only makes it beneficial for high traffic situations.

In short, pure ALOHA is more energy efficient when the traffic load is low, and the synchronized access is preferable for high frame generation rates. Therefore, in scenarios where the traffic load varies over time, the access scheme should be adapted dynamically in order to maximize the lifetime of batteries. As an example, one of the many LoRaWAN applications is pollution tracking. Interestingly, the air quality in a city varies much more during the daytime than during the night [5]. Efficient monitoring of this kind of phenomena would thus require a time-varying number of measures. In the considered example scenario, switching between the asynchronous pure ALOHA access scheme and a time-synchronized one would permit a higher data reporting rate during daytime, and conversely energy savings during night hours.

In such a context, this paper timely presents the traffic-aware energy efficient MAC protocol (TREMA), able to seamlessly switch between access modes depending on the probed conditions. In order to fully exploit the capabilities of TREMA in high traffic scenarios, a time-synchronized scheduled access relying on the LoRaWAN beaconing system is introduced as well. In addition, this contribution also features a beacon skipping strategy able to save energy while ensuring that cumulative clock drifts do not generate transmission misalignment. Interestingly, configuring TREMA requires a prior performance evaluation of the considered LoRa deployment, aiming at characterizing several metrics such as the throughput or energy efficiency for a wide range of frame generation rates. These metrics are assessed for both the asynchronous and synchronous access schemes. This preliminary study in fact provides the deployment fingerprint, that is then leveraged to maximize energy efficiency in all situations.

In more details, the design of TREMA includes *(i)* a *time-*

synchronized scheduled access to be used when the traffic load is high, *(ii)* a detailed description of the required deployment *fingerprint*, *(iii)* a *probing strategy* that aims at estimating the generated traffic, *(iv)* a *decision mechanism* that determines whether or not a switching should be triggered, and *(v)* the *signaling protocol* required to adapt the access scheme of all devices from asynchronous to synchronous and *vice versa*.

The performance of such a protocol is assessed through simulations. To this aim, a preliminary analysis is done on a large scale single-gateway LoRa deployment to figure out its fingerprint in terms of expected throughput and energy efficiency. Then, TREMA is tested in an example scenario to show how the network capacity is adapted in reaction to traffic load variations. A final performance assessment has been pursued to feature its behavior under any traffic condition. To the best of the authors' knowledge, TREMA is the first protocol aiming at dynamically synchronizing and desynchronizing LoRa deployments according to the traffic load variations, thus adapting the network capacity and always maximizing energy efficiency.

The rest of this work is organized as follows. Sections II and III give an overview of LoRaWAN and highlight other significant contributions about medium access enhancements for these networks. Section IV presents the design of TREMA in detail. In Section V, a simulation environment is used to assess the performance of TREMA. Finally, Section VI concludes this contribution and envisages research perspectives and future works.

II. Background on LoRa

The LoRa LPWAN technology is well known for its capability to gather data from a large number of widely-spread sensors. The term LoRa specifically designates the physical layer technology, while LoRaWAN refers to the MAC layer. The proprietary LoRa modulation, owned by Semtech, relies on a Chirp Spread Spectrum (CSS) technique, resilient to multi-path interference and channel fading [6]. It allows transmissions to reach up to 10 km ranges under certain conditions [7], with a relatively low power operation. The chirp frequency sweeping duration defines the Spreading Factor (SF) of the modulation [8]. Increasing the SF lengthens the transmission (TX) range, but also increases the frame time-on-air, inducing a lower data rate and a higher energy consumption.

Even though the LoRa modulation is proprietary, the upper layers are open and well documented. The LoRa Alliance regroups many industrial actors and facilitates the interoperability of LoRa networks, especially through the development of the LoRaWAN specification [2], that provides a common view on the MAC layer implementation. This standard defines a 2-tier topology, where devices communicate with gateways via LoRa communications, and gateways interact with servers using IP. The standard also specifies the behavior of several device classes, as well as parameter guidelines such as Data Rates (DR) derived from the SFs.

A notable LoRaWAN feature is the use of the Industrial, Scientific and Medical (ISM) license-free radio spectrum, which facilitates the prototyping and deployment of distributed applications. However, these public frequency bands are subject to Duty Cycle (DC) limitations. This means that, after the transmission of a frame, any LoRa device must wait for a specific amount of time before transmitting again. These rules are region-specific, in Europe they are enforced by the European Telecommunications Standards Institute (ETSI). The LoRaWAN specification thus provides directives in their regional parameters document [9] to ensure legal compliance.

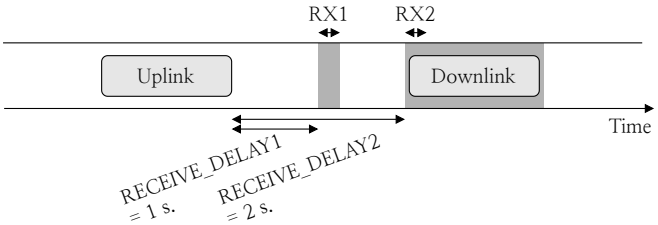


Fig. 1: *Class A* transaction.

The LoRaWAN standard defines three device classes, respectively *Class A*, *B* and *C*, that will now briefly be presented. They are featured with specific communication modes able to fit different types of applications. *Class A* is the default scheme that all devices must implement. In this mode, the uplink (device to server) transmissions are carried-out in a pure ALOHA manner, and followed by two 30 ms reception slots `RX1` and `RX2` that occur respectively 1 and 2 seconds after the end of the transmission. The `RX1` window uses the same channel as the preceding uplink, while `RX2` occurs on a dedicated downlink channel. In Europe, the uplink channels are subject to a 1% DC, and the downlink one to a 10% DC. During the RX slots, the device switches its radio to a listening state, providing the server with an opportunity to return a downlink message. In other words, if the server has got a pending message for a given device, it will wait for an indeterminate time before having a transmission opening. This behavior is represented in Figure 1, for the particular case of a downlink being received during `RX2`. The device radio is then forced to keep listening until the end of the message. In order to reduce the downlink delay, *Class B* was introduced as an extension to *Class A*. *Class B* devices still transmit uplinks according to the *Class A* strategy, but they also periodically open additional reception slots, offering frequent downlink opportunities. This requires a precise synchronization with the server time, so they listen to beacons broadcast by the gateways every `BEACON_PERIOD` $= 128$ seconds. Finally, *Class C* devices continuously listen for incoming frames when they are not transmitting, thus consuming a significant amount of power. This class is thus suitable for actuation purposes.

III. Related works

The design of efficient MAC layers enhancements to the pure ALOHA LoRaWAN uplink scheme is a hot research topic, as clearly pictured in what follows. The variety of contributions in this sense even triggered higher level analyses

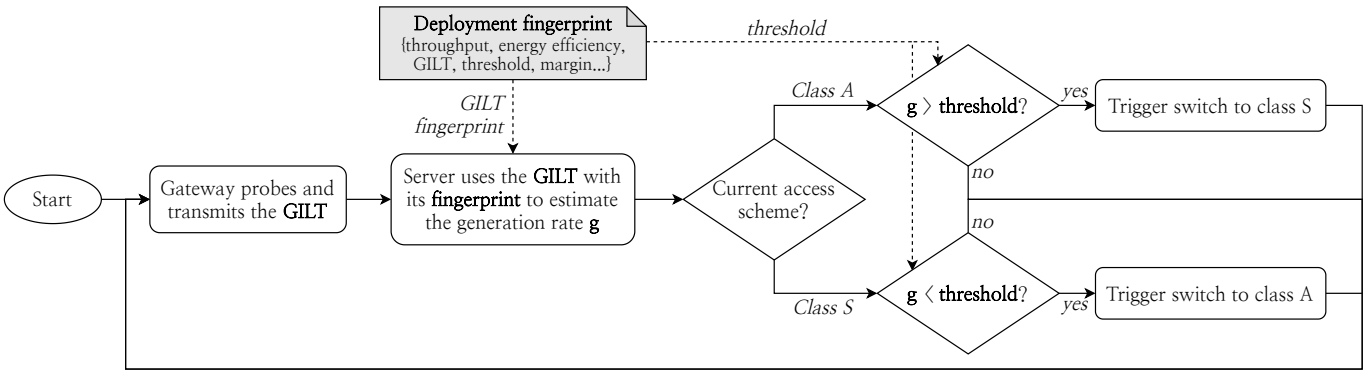


Fig. 2: TREMA's switching mechanism flowchart

such as the one led by Beltramelli *et al.* [10], where a model was proposed to compare them. This shows the clear interest of the research community in testing new LoRaWAN access schemes. Interestingly, a recent contribution [11] proposed a method to temporarily alter the protocol in order to ease firmware updates. This is a first example showing how the MAC layer can be changed to suit the context.

If some Carrier Sense strategies have been explored [12], most of the novel protocols somehow rely on synchronization and the induced possibility to slice time into transmission slots. A first work in this direction has been provided by Rizzi *et al.* in [13], where class A transactions are placed within a coarse Time Slotted Channel Hopping scheme to reach real-time requirements. However, the timeslot size is not tailored to maximize the throughput and no explanation is provided on how to achieve synchronization.

There are several ways to give all devices a common time reference, and they can be used to discriminate some of the proposed protocols. For instance in [14] and [15], respectively proposing a slotted ALOHA and a scheduling scheme over LoRaWAN, acknowledgements are used to synchronize the devices. This option has not been chosen here because the gateway DC limitations and collisions induce a poor downlink reliability [16], which may lead to synchronization difficulties when increasing the network load. The authors of [17] proposed to use low-power wake-up receivers to setup an on-demand Time Division Multiple Access (TDMA) that completely avoids frame collisions. In [18], FM-RDS is used to synchronize the devices out-of-band and enable a slotted ALOHA access. The main drawback in these two contributions is that they require adding specific circuitry to the devices. In [19], a time-slotted scheme is achieved by using a Synchronization and ACKnowledgement (SACK) packet, requiring dramatic changes to the legacy protocol.

In a previous contribution, we proposed to rely on the LoRaWAN beacons for synchronization. In that, we defined *Class S* as an extension of *Class B* in order to introduce uplink transmission timeslots fitting the longest frame time-on-air of the network [4]. The beacons were also leveraged in [20], with the goal of providing a contention-free access during a sub-portion of the beacon period. Yet, the goal there was to reserve channel resources to critical traffic, not to increase the overall network capacity. Compared to the alternatives, *Class S* has the advantages of requiring very little changes to the legacy protocol and keeping the device complexity low. It is therefore the basis on which we build the scheduling scheme used throughout this contribution.

IV. Design of TREMA

The goal of TREMA is to adapt the LoRaWAN capacity to the traffic load while always maximizing energy efficiency. Herein, a time-synchronized scheduled access is presented to increase the achievable throughput. It provides a good energy efficiency when the generated traffic is high. On the other hand, the legacy pure ALOHA asynchronous scheme is more efficient when the generated traffic is low. TREMA thus implements a mechanism capable of seamlessly switching between the two aforementioned schemes depending on the situation. To achieve this goal, the radio medium is probed in order to estimate the current traffic load. The network server then uses this estimation to select the most appropriate access scheme. To do so, this decision mechanism refers to a pre-established deployment fingerprint that relates to the network behavior under different traffic loads. When the access scheme needs to be changed, the signaling protocol embedded in TREMA is leveraged to transmit switching commands to the end devices. This behavior is pictured in the flowchart of Figure 2.

A. A time-synchronized scheduled access over Class S

The proposed time-synchronized resource scheduling protocol is built upon *Class S*, a TDMA scheme already introduced in a previous contribution [4]. *Class S* takes advantage of the *Class B* synchronization beacons to define transmission timeslots fitting the longest frame time-on-air of the considered deployment. Given the focus on delay tolerant applications that require only unconfirmed uplink traffic in a single-gateway deployment, a simple resource scheduling scheme has been designed to maximize the achievable network throughput.

Specifically, `NumDevices` defines the number of devices in the deployment, `NumSlots` the number of slots

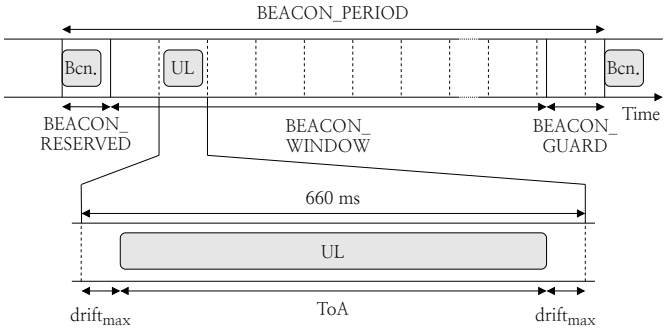


Fig. 3: Scheduled access over *Class S*

in the inter-beacon period, `NumChannels` the number of available channels and $\texttt{JoinIndex}_k$ the joining index of device k into the network. In the presented scheduling scheme, each device k is assigned to the slot number $(\texttt{JoinIndex}_k \ mod \ \texttt{NumSlots})$. It starts its transmission on the channel of index $\lfloor \texttt{JoinIndex}_k / \texttt{NumSlots} \rfloor$, then a round-robin channel hopping is used to prevent a possible channel fading from making a given device unreachable. This guarantees that the frames generated by a device may collide with frames from a maximum number of $\lceil \texttt{NumDevices}/(\texttt{NumSlots} \cdot \texttt{NumChannels}) \rceil$ other devices. The frame collision probability is therefore lower than when using slotted ALOHA. This scheduled access is represented in Figure 3. In the given scenario, an uplink is sent by a device assigned to the second slot. As specified in [4], the last slot (which corresponds to the 187^{th} when using 660 ms-wide slots) overlaps onto the `BEACON_GUARD` period. The uplink transmission is centered within the slot to prevent a possible clock drift from letting it overlap onto another slot.

B. Prior fingerprinting of the deployment

In order to wisely select the most energy efficient access scheme for a given the network load, it is beforehand necessary to study the deployment behavior for each scheme and for a wide range of frame generation rates g. This analysis ultimately allows to derive g from the probed traffic information, and provides a threshold value for g which allows to decide which access scheme should be chosen. The use of this deployment fingerprint in TREMA's switching mechanism is depicted in the flowchart of Figure 2.

The **throughput** T, expressed in bytes per second, represents the amount of data successfully transmitted during a given time interval. It must be assessed in order to understand the maximum capacity of the network. The **Gateway Idle Listening Time** (GILT) is characterized as well. It plays a role in the traffic load estimation, which is detailed hereafter in Section IV-C. The **energy efficiency** E [21] is obtained by dividing the throughput T by the average power consumption. In order to increase E, devices skip some of the synchronization beacons. Indeed, the LoRaWAN specification states that devices should be able to maintain beacon-less operation during at least two hours if the RX slots are widened by the worst-case drift time since the last synchronization. Hereafter, this upper bound to the allowed drift is indicated with $drift_{max}$, while the parameter n_{skip} is introduced to represent the number of beacons skipped by devices. For instance, if $n_{skip} = 1$, the device will listen to one beacon out of two. The relationship between the maximum time-on-air within the network and the chosen *Class S* slot size determines $drift_{max}$, which in turn bounds the maximum value for n_{skip}. As we will show later on in Section V-A, the fingerprint of E allows to determine a traffic load tipping point, under which pure ALOHA is the most energy efficient access scheme, and above which the synchronized access becomes preferable. This tipping point defines the frame generation rate **threshold** guiding TREMA's decision mechanism.

C. Network probing and decision mechanism

TREMA's probing strategy aims at estimating the generated traffic thanks to a measurable value. The probed metric must respect two conditions: *(i)* the generated traffic should be deductible from its value, and *(ii)* all gateways must be able to compute it easily. The GILT is an excellent choice for this task, as it checks both requirements.

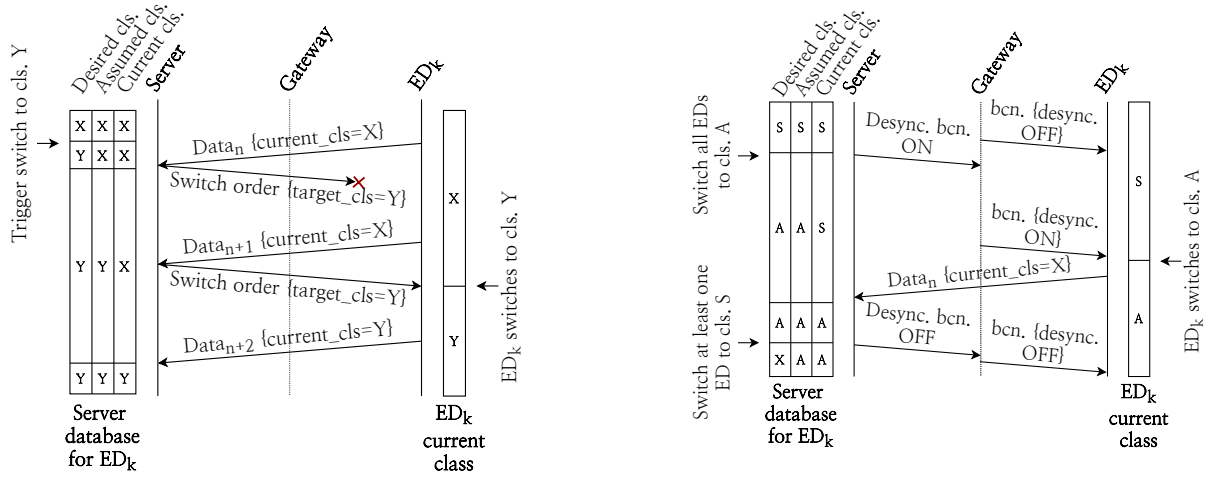
Condition *(i)* is respected by any metric that proves to be a bijective function of the considered generation rate range. Indeed, the GILT metric strictly decreases when the traffic load increases. The GILT also matches *(ii)*, as all gateways can keep track of their idle time. This is why this metric was chosen over the collision time for instance, that checks *(i)* as well, but would require discriminating the successes from the failures among the total reception duration. It is interesting to note that the introduction of the GILT within the fingerprint was necessary, as the T matches only *(i)*, while E complies only with *(ii)*. The gateway keeps track of the GILT and transmits it to the server every 20 minutes. Upon reception of the probed GILT, the server uses it to deduce an estimation of the offered traffic through the GILT fingerprint. Such an estimate is checked against the threshold value to decide whether the access scheme should be adapted following the reasoning depicted in Figure 2.

D. Signaling protocol

The switching commands detailed in this part may be used to decide the relative portion of synchronous devices out of the total number of LoRaWAN end devices into the network. However, in Section V-A, it will be shown that the transient states in which access schemes coexist are not energy efficient. Therefore, this contribution will focus on switching between a 100% *Class A* and a 100% *Class S*.

The switching command requires the definition of a `SwitchCommand` MAC command that will be transmitted through the `FOpts` field within the MAC payload [2]. As there are only two possible access modes for now, one bit will suffice to transmit the command value.

The server keeps track of three class-related variables for each registered device. `DESIRED_CLASS` is updated when a class switching event is triggered and contains the target



(a) General-case packet exchange for class switching.

(b) Desynchronization beacon broadcasting.

Fig. 4: TREMA's switching mechanism signaling protocol.

access scheme for this device. `CURRENT_CLASS` keeps track of the class the considered device was using when its sent the last uplink frame. `ASSUMED_CLASS` is updated when a switching command has been sent, but with no guarantee that it has been correctly received. This last one is used to provide an optimistic estimation of the current class division within the deployment, which is necessary to estimate the metrics more accurately when using the fingerprint as will be shown in Section V-A. Indeed, when a device receives a switching order, it does not send an acknowledgement immediately but waits the next uplink frame to inform the server that it has switched. This restricts the traffic induced by the mechanism, at the cost of adding some delay to the server database updates.

Fig 4a holds a switching command transmission chronogram. First, a switch from any class X to any class Y is triggered, so the server updates the `DESIRED_CLASS` entry. With no prior assumption on the node's current class, a RX window has to be used to transmit the order. The server thus waits for an uplink transmission $Data_n$ to send the order, which is done every time the values in `CURRENT_CLASS` and `DESIRED_CLASS` differ. When doing so, it updates `ASSUMED_CLASS`, but then the downlink frame is lost. At the next uplink message $Data_{n+1}$, the current class of the device is still X so another switching order is transmitted, and this one arrives. The device updates its access mode, and when finally it sends $Data_{n+2}$ `CURRENT_CLASS` is updated.

The switching process can be sped up in the particular case where a switching event aims at setting all devices to *Class A*. Indeed, all synchronized devices receive beacons periodically, thus broadcasting a desynchronization command through the beacon will allow to reach this goal instantly. This process is schematized in Figure 4b. When the switching event is triggered, the server informs the gateway that the desynchronization beacon flag should be set. At the next beacon reception, the device updates its class, and at its next data frame it informs the server that it has switched. Then, a switching to another class distribution is triggered and the desynchronization beacon flag is disabled.

TABLE I: Simulation parameters

Parameter	Value
Number of devices	1000
Simulation duration	24 hours
Frame time-on-air	626.94 ms
Beacon time-on-air	173.06 ms
Timeslot size	660 ms
Data rate	DR5 (SF7 with bandwidth 125 kHz)
Frame generation rate	Varies from ~ 0.5 to ~ 19 pkts. / h.
Channels	868.1, 868.3 and 868.5 MHz
Uplink Duty Cycle	1%
Device buffer size	1
Downlink data messages	Disabled
Acks and retransmissions	Disabled
Sensor voltage	3.3 V
Sensor current intensity	20 mA TX, 10.8 mA RX [22]

V. Performance evaluation

In order to assess the goodness of the TREMA design, an ad-hoc event-based simulator has been used. Since the deployment fingerprint is needed to kick off TREMA, it was established beforehand through large simulation campaigns using the considered network parameters. After that, an example scenario in which the frame generation rate varies over time has been used to feature the online TREMA behavior. Finally, TREMA has been evaluated for a wide range of traffic loads to quantify its performances in a more general frame.

The simulator frame-arrival process is modeled as a Poisson process with a variable rate to represent the fluctuations of the aggregated load. The network features have been set to resemble a large scale single-gateway deployment. The frame generation rates match realistic values for real-world use-cases and the number of devices has been set sufficiently high to

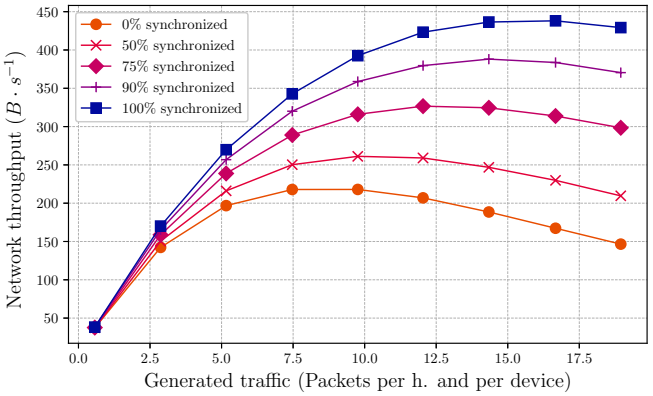
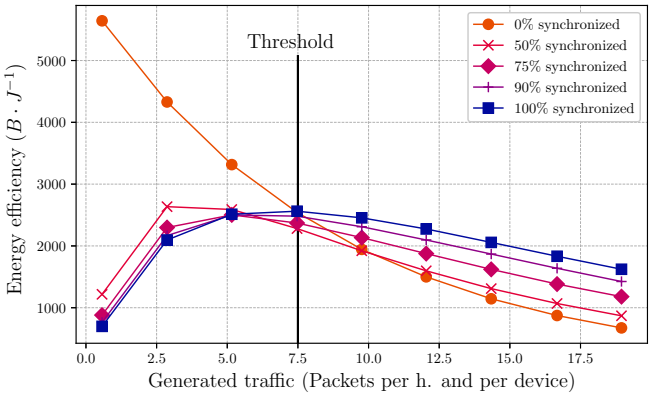
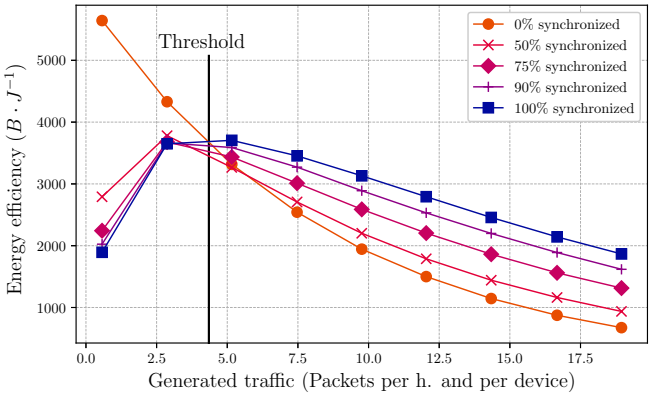


Fig. 5: Throughput as a function of the generated traffic.



(a) $n_{skip} = 0$



(b) $n_{skip} = 3$

Fig. 6: Energy efficiency as a function of the generated traffic.

challenge the network capacity boundaries. Exact details of the settings are listed in Table I.

In this paper, the traffic is restrained to uplink data communications only, thus the downlink data frames have been disabled along with acknowledgements and re-transmissions. In this way, the downlink capabilities are left free to be used without interference by the switching mechanism. Impact of downlink traffic on switching delays and overall performances will be tackled in future works. Only the three mandatory LoRaWAN channels are implemented, which are subject to a Duty Cycle of 1%. The highest priority in LoRa networks is to maximize the lifetime of the device batteries. The transmission time must therefore be minimized in order to reduce the energy required to send a frame and the collision probability. This is achieved by using the highest LoRaWAN data rate available, DR5, which corresponds to the smaller SF. Results can however safely be extended to the other orthogonal SFs by following a similar reasoning.

In the scope of this paper, all frames are considered to be sized according to the maximum MAC payload allowed (250 bytes), with the Coding Rate set to 4, and explicit header and Message Integrity Code are enabled. This all in all results in a time-on-air of 626.94 ms for data frames and 173.06 ms for beacons [23]. The *Class S* timeslot size is set to 660 ms in order to comfortably fit this frame length. In fact, this value is a multiple of the 30 ms LoRaWAN receive window, so that hardware implementations will require minimal modifications to the firmware.

A. Preliminary deployment analysis

In order to assess which access scheme performs better for different traffic loads, the deployment fingerprint will now be established when using *Class A* and the scheduled access built over *Class S*. Different degrees of coexistence of the two modes are also assessed to evaluate the network behavior during the transient states of the switching For the sake of statistical significance, for each configured scenario, 10 different realizations have been considered by feeding the pseudo-random number generator with 10 different seeds.

T has been plotted in Figure 5, for different percentages of synchronized devices. The figure shows that for this deployment, *Class A* is capable of offering a maximum of 220 $B{\cdot}s^{-1}$, while the scheduled access peaks at 440 $B \cdot s^{-1}$. It therefore provides a 100% capacity increase, which is better than the 88% increase offered by slotted ALOHA over *Class S* in [4].

Another notable observation is that the advantage of the scheduled access only appears clearly when a significant portion of the devices are synchronized. Indeed, a capacity increase of solely 18% is observed when 50% of devices operate over *Class S*, and with 90% of synchronized devices the curve peaks at 390 $B \cdot s^{-1}$ which corresponds to a relative improvement of simply 77%.

When using *Class S*, the chosen slot size sets the value of $drift_{max}$, and therefore an upper bound for n_{skip}, because any transmission overlapping onto another slot should be strictly prohibited. A typical low-cost LoRa device crystal has a 30 ppm quality, which means that in the worst-case scenario, it may drift of 1 ms every 33.3 s. With our parameters, the transmission is framed by a margin of $drift_{max} = (660 - 626.94)/2 = 16.53$ ms within its slot. It would take a minimum of 550.5 seconds to get a similar drift, so the maximum number of synchronization-less beacon periods that

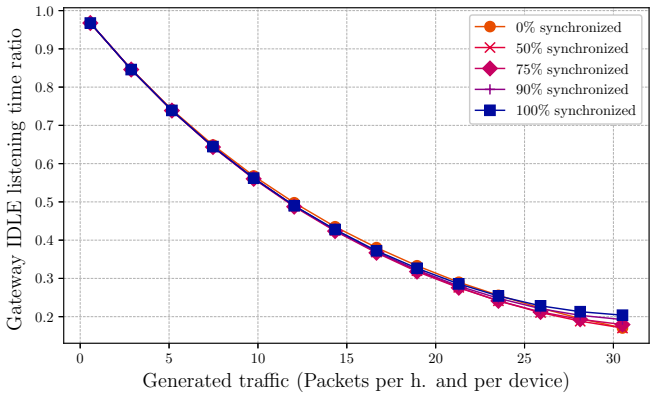


Fig. 7: GILT as a function of the generated traffic.

a device can maintain is $\lfloor 550.5/\texttt{BEACON\_PERIOD} \rfloor = 4$, which bounds n_{skip} to 3.

A higher energy efficiency may be achieved by widening the slots to enlarge the margin, but this would have the effect to reduce the maximum achievable throughput. Once again, designing the deployment requires trading-off network capacity for battery life.

From these considerations, E has been plotted for the minimum and maximum values of n_{skip}, respectively 0 and 3, in Figure 6a and 6b. When using the maximum value, the RX slots opened for beacon receptions have been widened by w, defined as:

$$w = 2 \cdot n_{skip} \cdot \texttt{BEACON\_PERIOD} \cdot 3 \cdot 10^{-5} \qquad (1)$$

With $3 \cdot 10^{-5}$ representing the 30 ppm crystal quality, $n_{skip} \cdot \texttt{BEACON\_PERIOD}$ the synchronization-less time induced by the skipping, and the multiplication by 2 accounts for a possible positive or negative drift.

TREMA aims at maximizing the energy efficiency at all times. A first remark that could be made on Figure 6 is that the maximum value for E is always attained with either 0% or 100% of synchronized devices. Therefore, the coexistence of the two access schemes should remain transitory.

The point at which the 0% and 100% curves cross defines the frame generation rate threshold over which the scheduled access becomes more energy efficient than pure ALOHA. This is the tipping point used by TREMA to decide whether the network should be synchronized or not. An interesting impact of setting n_{skip} to 3 is that it shifts the threshold to the left. Indeed, without beacon skipping, the scheduled scheme becomes more efficient than pure ALOHA when generating 7.5 pkt/h per device, but when receiving only 1 beacon out of 4 the threshold is reduced to 4.5 pkt/h per device. This allows to profit from the gain in network capacity induced by synchronization at a lower rate, and therefore to ameliorate the frame delivery ratio between 4.5 and 7.5 pkt/h per device.

Finally, the GILT fingerprint has been plotted in Figure 7. With the specific parameters used in this deployment, curves for all class divisions overlap. However in the general case, the network class division provided by the `ASSUMED_CLASS` parameter of the database is used to interpolate the appropriate traffic load from the closest class division curves. Further studies about the impact the network parameters on the GILT fingerprint shape shall be led in future works.

B. Online testing

In order to test TREMA, a 24 hour scenario has been simulated to employ day and night frame generation rates. In the interval between midnight and 8 AM, a low rate of 1.5 pkt/h per device is used, lower than the switching threshold. Then, the devices employ a higher rate of 10 pkt/h per device until 8 PM, higher than the tipping point, and finally go back to the night rate. No beacons are skipped, therefore the threshold is 7.5 pkt/h per device. The network behavior is evaluated through the plotting of T and E over time using a sliding window averaging over the last 15 minutes, with and without the switching mechanism. When it is enabled, the number of devices using each access mode is plotted over time as well. All these test results are disclosed in Figure 8

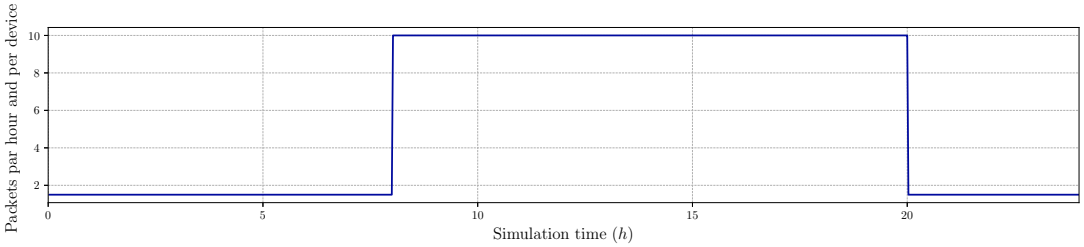
Without TREMA, the network always operates with *Class A*. When the rate increases, the throughput stabilizes at $220B \cdot s^{-1}$ and the energy efficiency at 2000 $B \cdot J^{-1}$, which were indeed the values observed for the 0% synchronized fingerprint curves in figs. 5 and 6 for 10 pkt/h per device. But, when switching is enabled, the rate increase is automatically detected by the server that starts synchronizing the devices. The process is not instantaneous, and takes about 3 hours, as seen in Figure 8b. This transition time explains the tipping point observed at around 9 AM in the energy efficiency curve. Once again these values are perfectly consistent with the fingerprint data. This time, the throughput T remains stable at 400 $B \cdot s^{-1}$, while the energy efficiency E sets up at 2500 $B \cdot J^{-1}$. When the rate decreases at nightfall, the desynchronization beacon allows to switch the whole deployment back to *Class A* instantly, and the metrics return to their initial values.

Two additional remarks can be made about the E plot. First, the energy efficiency drops faster when the rate increases if TREMA is enabled. This is due to the additional energy consumption induced by the reception of the switching commands. On the other hand, when the rate decreases, E increases a bit later when TREMA is enabled than when it is disabled. This is because the system takes some time to probe the traffic load reduction and trigger the switching to *Class A*.

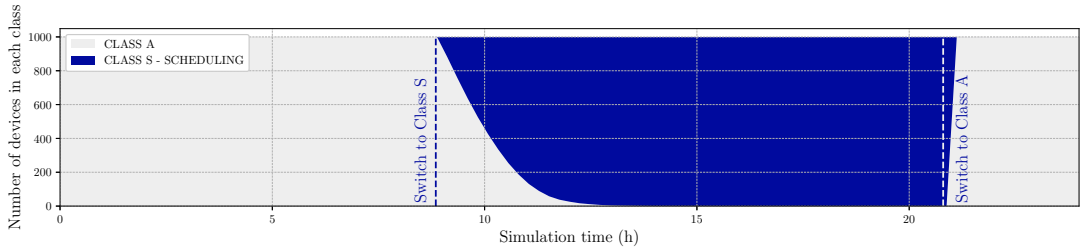
C. Switching time and performance gain

This part aims at estimating the performance of TREMA for the deployment parameters listed in Table I.

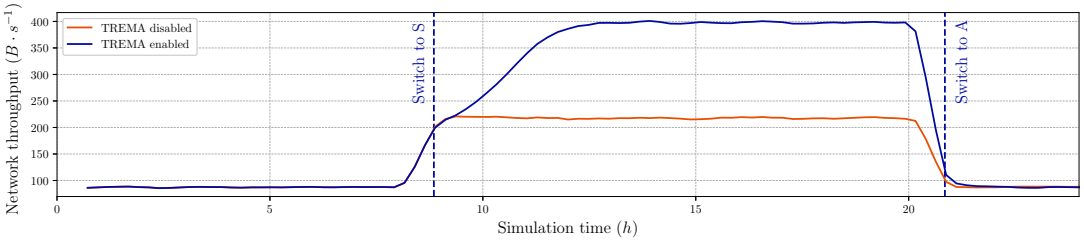
First, the switching duration from *Class A* to *Class S* has been plotted in Figure 9. It is defined as the time between the triggering and the moment when 90% of the devices have changed their access mode. A first remark that can be made is that its value is always within the two and three hours range for this specific deployment. Another interesting fact is that it starts by decreasing between 0 and 9 pkt/h per device,



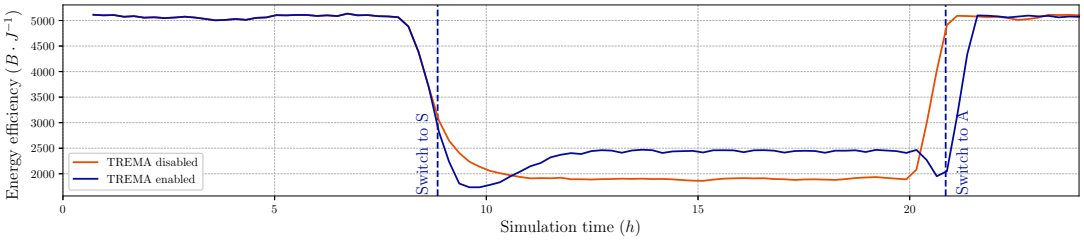
(a) Generation rate variation.



(b) Evolution of the number of devices in each class, with TREMA enabled.



(c) Evolution of the T over time, with and without TREMA.



(d) Evolution of the E over time, with and without TREMA.

Fig. 8: TREMA testing with one example scenario ($n_{skip} = 0$).

because the increase of the *Class A* throughput reduces the average delay the server has to wait before sending a switch command to a device. The value of 9 pkt/h per device actually corresponds to the rate at which the maximum pure ALOHA throughput is attained in Figure 5. It then increases again because the number of collisions causes the uplink throughput to decrease. The switching duration from *Class S* to *Class A* has not been plotted because it may easily be estimated thanks to the desynchronization beacon mechanism. In the worst case scenario, it is $(n_{skip}+1)\cdot$ `BEACON_PERIOD`. With $n_{skip} = 3$ it represents a duration of 512 seconds only, which is much shorter than the switching from *Class A* to *Class S*.

The advantage of using TREMA compared to the legacy LoRaWAN access in terms of throughput and energy efficiency has been plotted in Figure 10 as a function of the generated traffic, when using $n_{skip} = 3$. When the rate is below the 4.5 pkt/h per device threshold, the gains are 0% because TREMA selects the *Class A*. However, when the rate is above the threshold, the relative gains increase because the synchronized access performs better than pure ALOHA with these high traffic loads. The energy efficiency curve is continuous because the 100% and 0% synchronized curves cross in the E fingerprint, which is not the case for the throughput.

VI. Conclusions and Future Works

The TREMA protocol was developed to dynamically adapt the LoRaWAN MAC layer to traffic load variations. More specifically, the legacy pure ALOHA is used when the generated traffic is low, and the access is switched to a time-synchronized scheduling of the transmissions when the

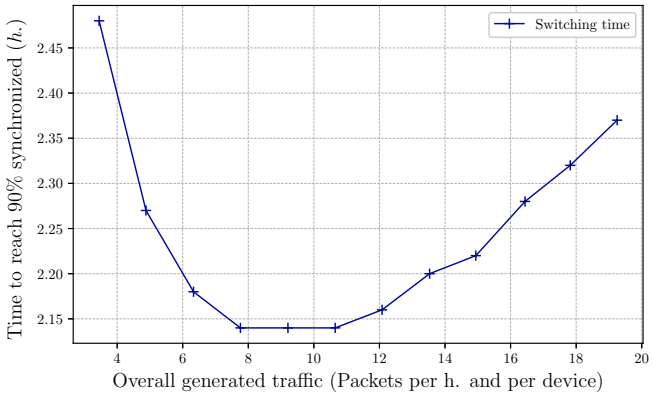


Fig. 9: Switching duration from *Class A* to *S*.

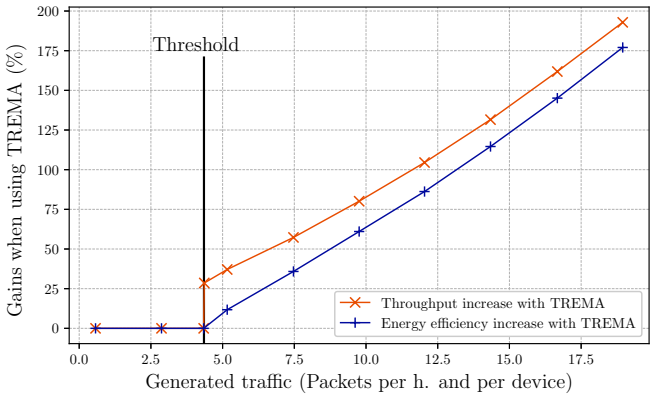


Fig. 10: Increase in T and E ($n_{skip} = 3$).

network gets more congested. A threshold-based decision mechanism is used to seamlessly switch between the schemes according to the probed traffic load. Results show that this mechanism increases the maximum achievable throughput while always maximizing the device energy efficiency, by synchronizing and desynchronizing the devices. One limitation of this preliminary approach is that it requires large simulation campaigns to establish the deployment fingerprint before implementing the mechanism. Future contributions will evaluate the tools capable of establishing such fingerprints on the fly for any network parameters. Research directions include probability models and machine-learning based approaches.

Further studies will also investigate the mitigation of network churn under unstable traffic conditions through hysteresis, assuring that a switching event is triggered only when observing a significant traffic load change for a sufficient amount of time, thus justifying the access scheme adaption. The impact of downlink data traffic on the system performances will be assessed as well. Finally, a real-hardware implementation with a deeper analysis of the clock drifting of devices will also shed some light on the behavior of time-synchronized access schemes in real-world deployments.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128601003024

[2] LoRa Alliance, "LoRaWAN® Specification v1.1," 2017.

[3] N. Abramson, "THE ALOHA SYSTEM: another alternative for computer communications," in *Proc. of AFIPS '70 (Fall)*, ser. AFIPS '70 (Fall). Houston, Texas: ACM, Nov. 1970, pp. 281–285.

[4] L. Chasserat, N. Accettura, and P. Berthou, "Short: Achieving energy efficiency in dense LoRaWANs through TDMA," in *IEEE WoWMoM 2020*, Cork, Ireland, Aug. 2020.

[5] P. I. Jalava, Q. Wang, K. Kuuspalo, J. Ruusunen, L. Hao, D. Fang, O. Väisänen, A. Ruuskanen, O. Sippula, M. S. Happo, O. Uski, S. Kasurinen, T. Torvela, H. Koponen, K. E. J. Lehtinen, M. Komppula, C. Gu, J. Jokiniemi, and M. R. Hirvonen, "Day and night variation in chemical composition and toxicological responses of size segregated urban air PM samples in a high air pollution situation," *Atmospheric Environment*, vol. 120, pp. 427–437, Nov. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1352231015303332

[6] S. El-Khamy, S. Shaaban, and E. Thabet, "Frequency-hopped multi-user chirp modulation (FH/M-CM) for multipath fading channels," in *Proceedings of the Sixteenth National Radio Science Conference. NRSC'99 (IEEE Cat. No.99EX249)*, Feb. 1999.

[7] K. Mikhaylov, J. Petäjäjärvi, and T. Haenninen, "Analysis of capacity and scalability of the lora low power wide area network technology," in *European Wireless 2016; 22th European Wireless Conference*, ser. Proceedings of EW '16, Oulu, Finland, May 2016, pp. 1–6.

[8] M. Knight and B. Seeber, "Decoding LoRa: Realizing a Modern LPWAN with SDR," *Proc. of the GNU Radio Conference*, vol. 1, no. 1, Sep. 2016.

[9] LoRa Alliance, "LoRaWAN Regional parameters v1.1," 2017.

[10] L. Beltramelli, A. Mahmood, P. Osterberg, and M. Gidlund, "Lora beyond aloha: An investigation of alternative random access protocols," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[11] K. Abdelfadeel, T. Farrell, D. McDonald, and D. Pesch, "How to make Firmware Updates over LoRaWAN Possible," *arXiv:2002.08735 [cs]*, Feb. 2020, arXiv: 2002.08735. [Online]. Available: http://arxiv.org/abs/2002.08735

[12] C. Pham, "Robust CSMA for long-range LoRa transmissions with image sensing devices," in *2018 Wireless Days (WD)*, Apr. 2018, pp. 116–122.

[13] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund, "Using LoRa for industrial wireless networks," in *2017 IEEE 13th WFCS*, May 2017, pp. 1–4.

[14] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment," *Sensors*, vol. 19, no. 4, p. 838, Jan. 2019.

[15] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low Overhead Scheduling of LoRa Transmissions for Improved Scalability," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3097–3109, Apr. 2019.

[16] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?" *arXiv:1704.04174 [cs]*, Dec. 2017.

[17] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-Demand TDMA for Energy Efficient Data Collection with LoRa and Wake-up Receiver," in *2018 14th WiMob*, Oct. 2018, pp. 1–4.

[18] L. Beltramelli, A. Mahmood, P. Osterberg, M. Gidlund, P. Ferrari, and E. Sisinni, "Energy efficiency of slotted lorawan communication with out-of-band synchronization," *IEEE Transactions on Instrumentation and Measurement*, p. 1–1, 2021.

[19] 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.

[20] L. Leonardi, F. Battaglia, G. Patti, and L. L. Bello, "Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2018, pp. 4141–4146.

[21] E. Björnson and E. G. Larsson, "How energy-efficient can a wireless communication system become?" in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 1252–1256.

[22] Semtech Corporation, "SX1276/77/78/79 datasheet," Jan. 2019.

[23] ——, "AN1200.13 SX1272/3/6/7/8: LoRaModemDesigner'sGuide," 2013.