Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment

Tommaso Polonelli [1],[*] , Davide Brunelli [2],[*] , Achille Marzocchi [1] and Luca Benini [1],[3]

[1]   DEI, University of Bologna, 40123 Bologna, Italy; achille.marzocchi@studio.unibo.it (A.M.);
    luca.benini@unibo.it (L.B.)
[2]   DII, University of Trento, I-38123 Trento, Italy
[3]   IIS, ETH Zurich, 8092 Zürich, Switzerland
[*]   Correspondence: tommaso.polonelli@studio.unibo.it (T.P.); davide.brunelli@unitn.it (D.B.)

check for updates

Abstract: LoRaWAN is one of the most promising standards for long-range sensing applications. However, the high number of end devices expected in at-scale deployment, combined with the absence of an effective synchronization scheme, challenge the scalability of this standard. In this paper, we present an approach to increase network throughput through a Slotted-ALOHA overlay on LoRaWAN networks. To increase the single channel capacity, we propose to regulate the communication of LoRaWAN networks using a Slotted-ALOHA variant on the top of the Pure-ALOHA approach used by the standard; thus, no modification in pre-existing libraries is necessary. Our method is based on an innovative synchronization service that is suitable for low-cost wireless sensor nodes. We modelled the LoRaWAN channel with extensive measurement on hardware platforms, and we quantified the impact of tuning parameters on physical and medium access control layers, as well as the packet collision rate. Results show that Slotted-ALOHA supported by our synchronization service significantly improves the performance of traditional LoRaWAN networks regarding packet loss rate and network throughput.

Keywords: Internet of Things; wireless sensor networks; long range radio; LoRa; LoRaWAN; ALOHA; Slotted ALOHA; synchronization; low-energy wireless protocols

1. Introduction

In past years, industry and academia have dedicated significant efforts to developing low-power wide area networks (LPWAN), as a new category of wireless communication standard [1,2]. Among commercial protocols, LoRaWAN, and the LoRa modulation developed by Semtech, is gaining popularity [3–11]. The public availability of the specifications [12] combined with low-cost certified transceivers [13] and the fact that a LoRaWAN WSN (Wireless Sensor Network) can operate in the unlicensed radio spectrum, means that anyone can use the radio frequencies without having to pay million dollar fees for transmission rights, and has motivated the global attention on this protocol. LoRa offers a link range of up to 10 km with a single gateway, which is capable of supporting hundreds of thousands of different devices; moreover, a unique feature of LoRa is the ability of the gateway to simultaneously demodulate multiples transmissions, with different data rates, on the same channel and time. LoRaWAN specification describes three classes of operations: low-power bi-directional end devices (Class A), scheduled downlink transmission (Class B) and Class C, designed for always-on bi-directional actuators. In this work, we focus the attention on the most commonly used Class A.

Despite the initial success, multiple open issues remain concerning the real performance in terms of scalability and the maximum traffic supported in massive LoRaWAN installations. Most LPWAN protocols promise to manage thousands of devices, but the impact of the LoRaWAN parameters on

large-scale networks, with packet time that can be more than one second, is still not well explored. Indeed, in Refs. [14–16] the scalability of LoRaWAN is studied, and the results in [14] show that only 120 nodes can be managed with a packet transmission every 22 min with static configurations. The performance improves, growing up to 1600, with dynamic configurations that minimize the time of air. In Ref. [5], the authors show an impressive reduction of channel capacity, up to 2×, when message acknowledgments are required. However, these figures are not enough for future IoT deployments [5]. While many existing works have studied the scalability of LoRaWAN networks, most of them do not consider the impact of downstream traffic on channel throughput, interference, and issues related to real deployments, such as clock drifts and delays produced by the restricted computational resources of LoRaWAN gateways and servers.

The work presented in this article aims at providing an analysis of the channel throughput of the LoRaWAN protocol, and to investigate the issues associated with its scalability. The results can provide useful guidelines for large-scale deployments and physical configurations. A specific network simulator offers information about the real performance of LoRaWAN Class A with varying number of devices and packet time of air. We studied the effective payload throughput related to the protocol overhead, which decreases with a factor of 2.22× (in the worst case) compared to Pure-ALOHA (P-ALOHA). After that, we propose a solution to improve network performance based on Slotted-ALOHA (S-ALOHA). We developed a reliable synchronization algorithm optimized for low-power devices, with an accuracy of 5.37 ms and a success rate over 99%. This reference time is used to keep aligned all the end nodes in LoRaWAN network, allowing deployment of a Slotted LoRaWAN built on top of standard LoRaWAN specifications. Simulations and tests, performed with 24 end nodes, show a throughput improvement of 2×, and 5.8× in crowded conditions. The low-power capabilities of Class A are preserved, and our Slotted LoRaWAN can be implemented on existing commercial products.

The remainder of this work is organized as follows: Section 2 presents the related works, while Section 3 introduces the LoRa physical layer and the LoRaWAN protocol. Section 4 models the channel throughout in the worst case and verifies the statistical analysis with results generated from a network simulator. Section 5 provides the description and of the synchronization algorithm and Section 6 shows the implementation of the S-ALOHA in LoRaWAN network. Finally, Sections 7 and 8 investigate the Slotted LoRaWAN throughput and synchronization errors. Section 9 concludes the paper.

2. Related Works

The number of connected IoT devices is expected to rise at an annual rate of 32% and market forecasts anticipate 21B devices by the end of this decade [2]. Some of the newest applications require coverage of entire cities or huge buildings at very low energy use and cost. These requirements are hard to achieve using traditional methods and infrastructures such as WPAN or cellular [1]. Large-scale IoT installations are becoming a reality, as networks are being deployed for urban monitoring applications [2], smart city [17], intelligent transportation systems [18], and urban monitoring applications [19]. Several radio technologies, such as Sigfox, LoRa™, IEEE 802.15.4 NB-IoT, BLE 5.0 and DASH7, are currently competing in the arena of Device-to-Device (D2D) long-distance, low-power communication [20].

Different communication technologies aimed at low power have been proposed and deployed. It is possible, as discussed in [5], to split them in two categories: low power local area networks, such as IEEE 802.15.4, and Bluetooth Low Energy which are typically utilized in short-range personal area networks; low power wide area networks, with a transmission range greater than 1000 m; this category includes LoRaWAN, Sigfox, and DASH7 as major players. Sigfox is a protocol based on a variation of the cellular system which permits remote devices to connect to an access point through Ultra Narrow Band. It is a proprietary technology, developed and distributed by Sigfox. Full specifications are difficult to obtain, so it is difficult to carry out a clear and comprehensive comparison with other protocols. Each end-device can send 140 messages per day, with a payload size of 12 octets, at a data rate up to 100 bps. Each Sigfox access point can handle up to a million end-devices, with a coverage area of 30–50 km in rural areas and 3–10 km in urban areas [21]. The payload and number of packets

limits make Sigfox networks hardly usable in application scenarios where communication is not very sporadic: LoRaWAN is less prescriptive and more flexible.

DASH7 [22] is a wireless sensor and actuator full Open Systems Interconnection (OSI) stack protocol that operates in the 433 MHz, 868 MHz, and 915 MHz unlicensed ISM bands. It originates from the ISO 18,000-7 standard [23]. DASH7 can provide communication in the range of 2 km with low latency, mobility support, and multi-year battery life. A secure AES 128-bit shared key encryption is supported with a data rate up to 167 Kbit/s. However, DASH7 needs more power per bit than other protocols like LoRaWAN, and for applications that do not require synchronization or low latency uplink, LoRaWAN would be preferred. Moreover, during fast motion in rural areas, DASH7 connections are not as reliable as other LPWAN solutions.

In the LoRaWAN™ architecture [24], sensor nodes communicate with the gateway, which serves as a bridge between the nodes and a network server. Three types of functional classes are defined for end-devices: A, B, and C. Currently, several papers analyze LoRa performance [5–30]. In these works, LoRa deployments are compared in terms of network throughput and power consumption. In Ref. [3] the coverage of LoRa is studied and in Ref. [31], the authors analyze the LoRa network capacity and propose LoRa-Blink to support multi-hop communications. In Ref. [32], Casalas and Mir presented an analytical model that characterizes the current consumption, lifetime and energy cost in LoRaWAN Class A. The proposed model allows to quantify the impact of physical and MAC layers, as well as bit error rate and collisions, on energy performance.

As mentioned in the Class A ISO/IEC ISM regulations [24], the end nodes and gateways can transmit "at will" without any carry sensing. This makes the LoRaWAN's MAC similar to ALOHA [33]. The ALOHA mechanism can cause substantial inefficiency in LoRaWAN networks, as highlighted by the simulation study reported in Ref. [26], which confirmed that increasing the number of gateways can improve the global performance, but cannot eliminate fast saturation. Indeed, generating packets with Poisson law and uniform distribution of the payload lengths between one and 51 bytes, as simulated in Ref. [5], the maximum expected channel capacity usage is 18%, which is similar to the Pure-ALOHA (P-ALOHA) throughput, with a link load of 0.48. This may be an issue because several packets could be lost; indeed, at the maximum channel load, around 60% of the packets transmitted are dropped because of collisions. To verify the correct reception of each packet an acknowledgment (ACK) method could be applied, but it requires two successive transmissions in order to be successful, the payload in uplink and ACK in downlink, increasing the collision probability with other messages.

The S-ALOHA protocol is well known in local wireless communications [34] since the 70's. In S-ALOHA the channel time is divided into slots (Figure 1), which have fixed length T and are basically composed of two parts: transmission time (T_r) and the tolerance interval (T_b), as presented in Figure 1a. Every end node must transmit a packet only at the beginning of a slot. If two or more end nodes transmit their packet during the same period, a collision occurs; otherwise, no collision is generated, and the data are properly sent (Figure 1b). If no terminals access in one slot, it will be free, and no device will start the transmission in the middle of the time-slot. The maximum theoretical S-ALOHA channel throughput is 37%.
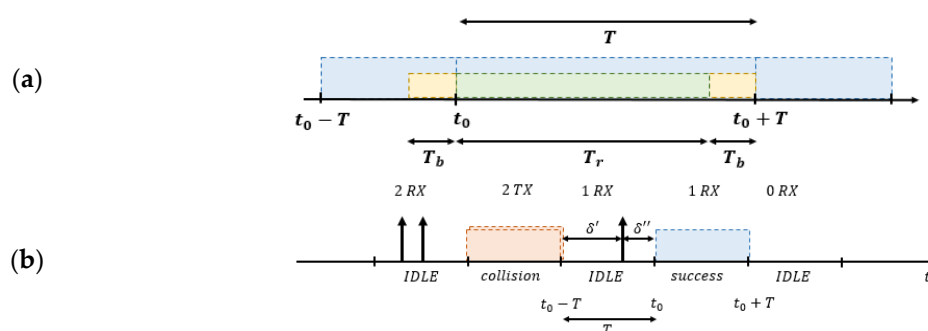

Figure 1. (**a**) Slot width definition in S-ALOHA, (**b**) example of free, collided and successful slot.

3. LoRa Physical Layer and LoRaWAN

LoRa is a chirp spread spectrum modulation. It is a signal which frequency increases or decreases with time. It was developed for radar applications; indeed, these signals have a constant amplitude and pass the whole bandwidth in a linear or non-linear way from one end to another end in a defined time. The frequency delta allowed between the receiver (gateway) and transmitters (end nodes) can reach 20% of the bandwidth without impacting decoding performance [5]. This factor is useful to design cheap devices with low-quality crystals, such as commercial products with 80 ppm/°C. LoRa typical transceiver, such as the SX1276 [35], offers a link budget up to 168 dB with a receiver sensitivity of −148 dBm. Since the symbol period is longer than a typical noise spike, errors generated from these interferences are easily filtered at the receiver side, through error correction codes. Code Rate (CR), Spreading Factor (SF) and Bandwidth (BW) parameters affect the LoRa modulation; indeed, a LoRa symbol is composed of two SF chirps, with a frequency variation that covers the entire band. In LoRa modulation the chirp symbol rate is directly dependent on the bandwidth, this parameter generates different consequences as specified in Ref. [5]: (i) increasing the SF, the chirp frequency span is divided by two; (ii) the period of each symbol is multiplied by two; (iii) since more bits are transmitted in a symbol, the bit rate is not decremented with a two ratio; (iv) the bit rate at given SF is proportional to the bandwidth, doubling the bandwidth will double the transmission rate. Over LoRa modulation, a forward error correction is used to enhance the communication reliability. This method is implemented through four different coding rate (CR) approaches, denoted CR1 to CR4. Lastly, a low data rate drift correction mechanism is applied to increase robustness to frequency variation over the timescale of the LoRa message. Equation (1) gives an estimation of the equivalent bit rate (EBR), detailed information can be found in Supplementary Materials and Appendix A:

$$EBR = SF \left(\frac{BW}{2^{SF}} \right) CR \tag{1}$$

The radio channels used in a LoRa deployment depend on the country rules. In this paper, all the tests are performed in Italy, Europe, where the Industrial Scientific Medical (ISM) band is at 863-870 MHz. LoRaWAN provides nine different channels, where channels from 0 to 9 have a bandwidth of 125 KHz and must support data rates between 0.3 kbps to 5 kbps; indeed, channel ten is allocated for FSK modulation with a bandwidth of 250 KHz. Figure 2 presents the frequency characterization of each LoRaWAN channel.
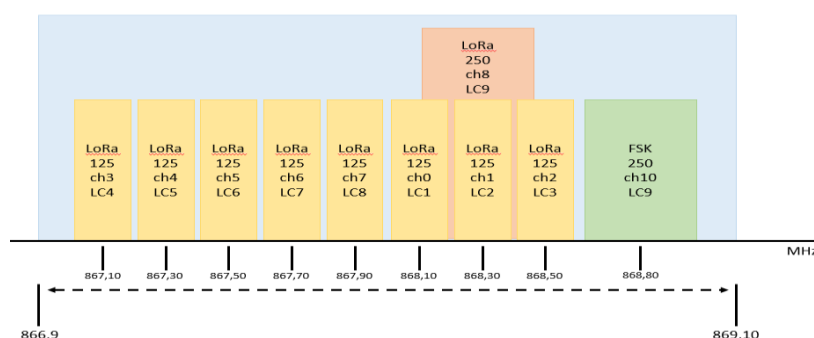

Figure 2. Description of LoRaWAN channels.

An important feature of LoRa modulation is that the different SFs are orthogonal, this means that a LoRaWAN gateway can receive a multiple transmission on different SFs simultaneously and can apply this methodology for each channel.

Due to local rules imposed by ISO/IEC ISM regulations, wireless sensor devices working on ALOHA MAC access cannot occupy more than 1% of the channel time. It is important to underline

that, as long as the limitations for each band are respected, each end device can transmit on different channels contained in different sub-bands in order to increase its overall throughput [9].

Class A in LoRaWAN is focused on the end node side; indeed, all transmissions are scheduled by the end node, whereas the server can transmit only in one of two receive windows opened after a previous uplink transmission, as presented in Figure 3. Therefore, any packet that the application level needs to transmit must wait until the next scheduled receive windows.
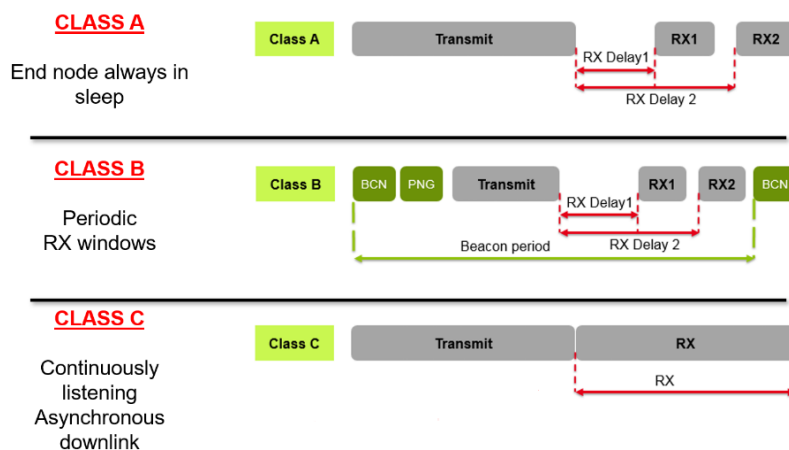

Figure 3. Uplink and Downlink methodology in Class A, B, and C.

In the MAC layer the typical data packet is composed by a preamble, a physical header, a variable payload length and, lastly, error detection bytes (CRC); additionally, the CRC is present only in the uplink packet, thus the downlink data is often used for ACKs. LoRaWAN MAC defines different strategies for the downlink channel and DR selection in RX1 and RX2 windows; in this paper, we set up the WSN to use the same DR and channel for uplink and downlink transmissions; moreover, we disable the RX2 window to evaluate the network capabilities with static conditions. Indeed, with the RX2 enabled, the overall downlink period could change up to 200%. The minimum time width for both receiving windows must guarantee at least the time required to effectively detect a downlink preamble [24], and if preamble symbols are detected, the transceiver must remain active until the reception of all data.

4. LoRaWAN MAC Model

Class A is designed for battery-powered end nodes that must be ultra-low power with very long expected lifetime. It permits bi-directional communications, but downlink transmissions are constrained to two short intervals after each uplink transmission to the gateway. The MAC of LoRaWAN is based on P-ALOHA. End-devices of Class B and Class C are designed for end nodes with no critical energy constraints, thus they allow for more receiving slots at specific scheduled times and synchronized beacons are sent from the gateway. In this paper, we focus on an S-ALOHA scheme for LoRaWAN battery-powered sensor nodes.

We define S as the average number of packets generated per transmission time interval; the traffic source λ consists of a high number of users who collectively form an independent Poisson source with an aggregate mean packet generation rate of X packets/s, the packet time width is supposedly fixed with a period of T seconds (Equation (2)). Moreover, we can consider that each user generates packets infrequently and that each packet can be successfully transmitted in an interval smaller than the average time between successive packets from a given node [36]. S can also be expressed as the channel throughput rate. Each node delays the transmission of a previously collided packet by some random time, chosen, for example, uniformly between 0 and $T_{\max}$. Therefore, the traffic injected into

the channel consists not only of new packets but also of previously collided packets: this increases the mean traffic generated, usually denoted with *G* (Equation (3)):

$$S = \lambda\,T \tag{2}$$

$$G \geq S \tag{3}$$

$$G(n) = \lambda(n)T \tag{4}$$

$$S = G(n)\cdot P_{\text{suc}} = \lambda(n)T\cdot e^{-\lambda(n)2T} \tag{5}$$

In P-ALOHA, a single transmission is successfully performed if during the time period 2*T* (vulnerability period) the channel remains free. The probability that there are no transmissions in the 2*T* period is P_{suc} [36]. The total channel traffic could be expressed as presented in Equation (4). With these hypotheses, it is possible to obtain the maximum channel throughput: 18%.

This model generates results that approximate to real deployments for LoRaWAN Class A end node in the unacknowledged configuration when there is only uplink transmission without the answer from gateway (s). To realistically model the single channel throughput in the half-duplex mode we achieved a statistical model based on measurements from a real LoRaWAN testbed, the network simulator and statistical analysis.

As shown in Figure 3, in LoRaWAN Class A there is one second delay between transmission and RX1 windows; this period could be used by other end nodes to transfer data, but the probability of a successful transmission in the RX Delay 1 is very low, since LoRa modulation generates a remarkable packet's time of air. With an SF 12, a BW 125 KHz, CR 4 and 25 bytes of payload the overall time of air is 1253 ms, with these setting the corresponding ACK time width is 530 ms.

With the proposed parameters, a transmission inside the RX1 Delay is not allowed, as shown in Figure 4, and the probability of collision is equal to one. In the proposed model the overall channel occupation time for each transmission is composed by time of air of both uplink (*T*) and downlink (T_{ack}) windows plus the RX1 Delay. If we consider T the channel time used to transfer the payload, we can define 2.22*T* the total channel time used for each sensor, considering the LoRaWAN overhead; this value is equal to T_{LoRaWAN}. Therefore, the ratio between the traffic injected and transmission successfully performed can be expressed as presented in Equation (6), where the vulnerability period scales up from 2*T* to 2(2.22*T*):

$$S = G(n)\cdot P_{\text{suc}} = \lambda(n)T\cdot e^{-\lambda(n)2(2.22T)} \tag{6}$$
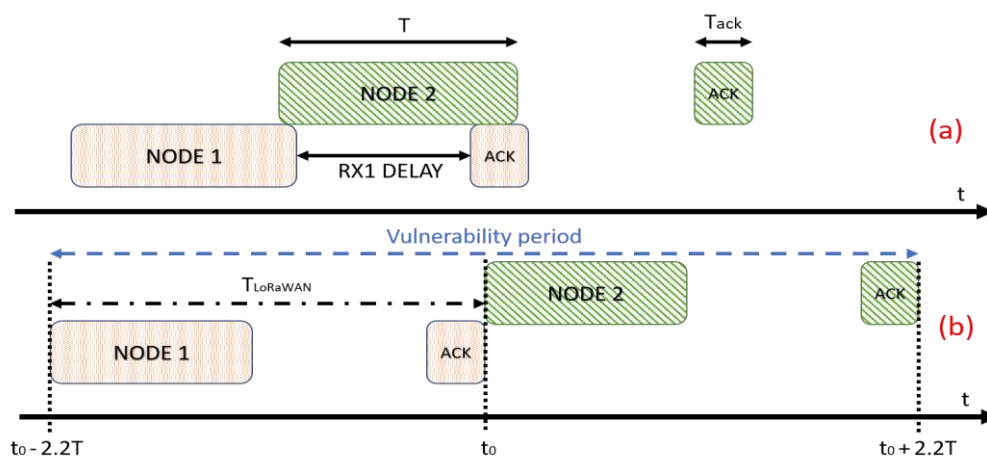

Figure 4. Vulnerability period in LoRaWAN with ACK. (**a**) *T* is greater than RX1 Delay, a collision must occur; (**b**) vulnerability period needed to perform a successful link between an end device and the gateway.

The effective LoRaWAN throughput decreases drastically with respect to the classic P-ALOHA, with a maximum of 8% at G equal to 0.25. This result shows that the LoRaWAN throughput (for each channel) is poor when half-duplex communication is required. To confirm this conclusion, we developed a simulator in MATLAB to evaluate the channel collision rate in a WSN with different time-of-air configurations with the uplink and RX1 windows. In details, the simulator generates traffic on a variable number of nodes and each one transmits following a Poisson uniform distribution. When a simulation starts, the number of nodes grows from 1 and stops as soon as the desired (*G*) is achieved. The maximum number allowed is 1000, and each element tries to access into the channel with a variable duty cycle and a maximum of 40 attempts. The simulator checks every 1ms if two end devices try to communicate simultaneously, with an overall simulation time of two hours. Figure 5 shows the *S*(*G*) for unconfirmed and confirmed LoRaWAN single channel link; indeed, in Figure 6a,b the results of two different simulations are presented. In Figure 6a the packet time is greater than the RX1 Delay, in Figure 6b the time of air is <<1 s, so the probability for a transmission successfully performed is not equal to zero, and the network performance slightly increases.
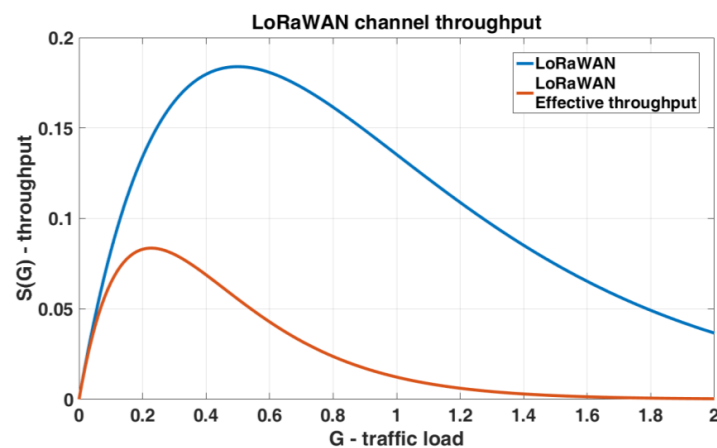

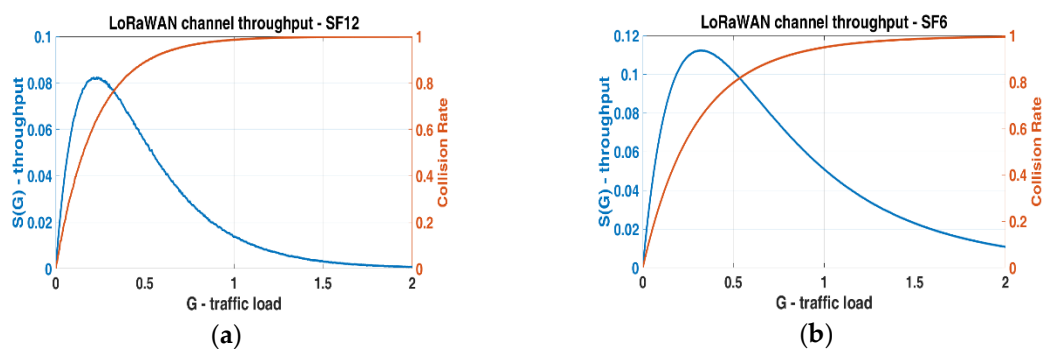**Figure 5.** Unconfirmed and confirmed LoRaWAN single channel throughput.

Figure 6. Confirmed LoRaWAN throughput and collision rate for one channel acquired by MATLAB simulation. (**a**) SF = 12, BW = 125 KHz, CR = 4 Maximum throughput is 8%; (**b**) SF = 6, BW = 125 KHz, CR = 4 Maximum throughput is 11%.

Figures 5 and 6 show that the proposed model is valid and that the effective channel throughput in LoRaWAN networks is tightly constrained. In S-ALOHA MAC protocol all the components of a LoRaWAN link can be fitted into a single slot (Figure 1a), uplink and downlink windows as well as the RX1 Delay, keeping the channel throughput of Slotted LoRaWAN (S-LoRaWAN) equal to standard S-ALOHA. As expressed for the P-ALOHA, if we consider *T* the channel time used to transfer the payload, we can define 2.22*T* the total slot time used, taking into account the LoRaWAN overhead. With these considerations, the maximum channel throughput in S-ALOHA is 16%, doubled respect to the classic half-duplex LoRaWAN.

5. Real Time Clock Synchronization over LoRaWAN Stack

Clock synchronization across end nodes is fundamental to define the slots used by S-ALOHA. In literature, articles experimented some synchronization schemes on the top of the bare LoRa radio [28], without any LoRaWAN protocol to demonstrate that LoRa modulation can be used even industrial environments [29]. Here we use a lightweight synchronization methodology, presented in Ref. [27] and tailored for LoRaWAN end nodes and successfully tested on an STM32L4 MCU. To generate a time reference for the S-ALOHA protocol, where the slots must be kept aligned in all the end devices, we use a real-time clock (RTC) with an inexpensive external crystal at 32.768 KHz with a thermal variability between 20 to 80 ppm/°C. Since this component drifts significantly with temperature, we need to re-synchronize end node clocks frequently to keep slots sufficiently aligned. The synchronization scheme is piggybacked on the predefined answer (ACK) of a LoRaWAN Class A RX1 window, which is opened with a maximum error of ±20 μS delay [27]. Common events for both devices, end node and gateway, are used as a shared reference for the synchronization algorithm. In a typical half-duplex link, the node and the gateway must know when to open the RX1 window after a successfully uplink transmission. A timestamp at the end of such packet is saved on both devices with the goal to open the RX1 window simultaneously; thus, this event (Figure 7) is used as the reference for our clock synchronization procedure [27].
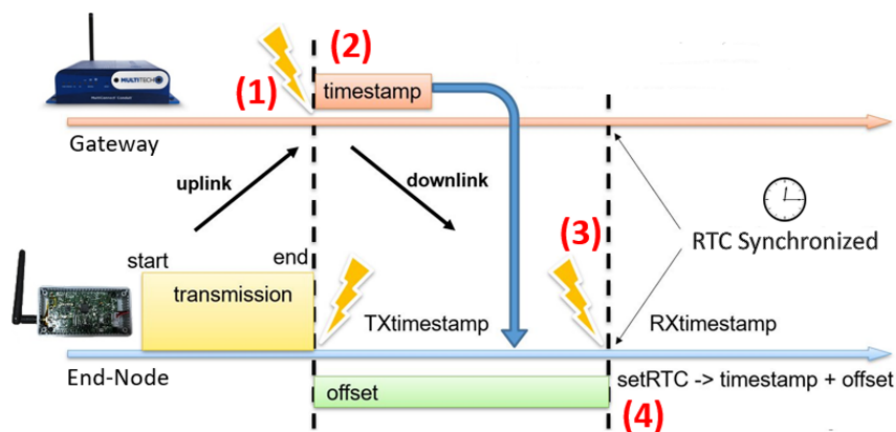

Figure 7. Synchronization procedure.

The synchronization algorithm is based on a clock reference distribution system: basically, at the end of every complete uplink transmission each node updates its real-time clock. We implemented an algorithm that works as follows: (1) gateway and node both save a timestamp of the moment when the uplink transmission has ended; (2) the node receives, in the first available RX window, gateway's timestamp piggybacked in the acknowledgment message; (3) due to the non-negligible flight times and the RX window's delay, the node must consider the offset between the timestamp when the uplink transmission has ended (TX timestamp) and the one when it receives the acknowledgement message (RX timestamp). Node calculates the difference between these two timestamps and adds it to the timestamp received by the gateway. (4) The node updates its real-time clock with the new timestamp, in this way both devices are synchronized.

The developed system must automatically update the end node RTC to guarantee an adequate alignment with the gateway's reference clock. Moreover, the protocol must be compliant with LoRaWAN specification without adding a significant overhead on the overall communications. The proposed approach uses the LoRaWAN uplink packet and the RX window already scheduled after 1 s; indeed, if the timestamp saved on gateway, received at the end of the uplink transmission, can be sent back with the ACK packet in the RX1, the end node has the time reference needed for a clock synchronization. Overhead is very low, only 8 bytes of timestamp, with good scalability and configurability [27]. In many applications the overhead is negligible, for example when the end node

generates many data packets, the synchronization comes for free adding the gateway timestamp at some downlink packet. In WSN where the end node is programmed to send few packets every day, some additional synchronization phases will be needed to ensure clocks do not drift too much. Maximum tolerable clock drift depends on three factors: (i) maximum uplink time of air; (ii) daily duty cycle of each end node; (iii) maximum throughput required for the specific application.

On the end node side, the purpose of the RTC synchronization function, implemented in firmware, is to update the time contained in the local RTC with the timestamp received from the gateway, this procedure requires two steps: the calculation of the offset and the management of the sub-seconds. Since the packet in response to the uplink transmission takes a variable delay before reaching the node, the offset calculation is needed to update the received timestamp that could be no longer current. The offset calculation procedure consists in measurement between the response packet and the initial instant when the transmission by the end node ends, this instant coincides also with the arrival of the packet to the gateway, which is the moment where the gateway "marks" the message with the timestamp included in the RX1. The electromagnetic propagation speed is negligible factor respect to the times involved.

A total of 30 synchronization tests were carried out with two, three and four nodes: the results are shown in Table 1, where the time difference in ms is expressed taking as reference the end node N0. The purpose of this evaluation is to verify the synchronization performance with many devices, indeed they are programmed to send a timestamp request every 15 s, which is comparable with 320 end nodes with a request-rate of 80 min each.

Table 1. Results of the measurements carried out to verify the RTC synchronization between multiples end node connected to a single Gateway.

Number of End Nodes	Min. Synch. Error [ms]	Max. Synch. Error [ms]	Avg. Synch. Error [ms]	Delta Avg. Error [%]
2	0.072	8.296	3.310	-
3	0.120	33.260	4.705	+42
4	0.243	37.080	5.370	+62

From Table 1 it is possible to notice how the increase of devices that request the synchronization from the gateway significantly reduces the synchronization accuracy; this result is probably due to the management of processes at the operating system level within the gateway. Since the management of the timestamps is done by the software working on the Linux operating system, the priority of the processes on the gateway is not well known, and consequently, the final accuracy of the timestamp associated with the received packet cannot be under one millisecond. The results achieved shows that the average error between all the devices is 5.37 ms, with a maximum value of 37.08 ms, such uncertainty must be considered in S-ALOHA deployment to avoid transmission overlaps between adjacent slots. To decrease the timestamp error some LoRaWAN gateways embed a GPS module, used by the physical layer to associate exactly the time reference at the end of the packet frame, but, since this feature is not mandatory from the protocol rules, we tested our Slotted LoRaWAN without this support.

Figure 8a presents the plots of synchronization deviation between end nodes, from two to four, where are indicated the median, and the bottom and top edges of the box, respectively the 25th and 75th percentiles. Figure 8a shows that, as the number of nodes increases, the mean and the width of the interquartile gap (blue box) increases, which is a dispersion index provides a measure of how far the points move away from a central value, such as standard deviation. As a result of the increase in the number of devices, the number of "off-average" synchronizations indicated in the graph with red crosses also increases. Figure 8b represents the maximum synchronization uncertainty between nodes since last clock alignment and provides fundamental information to calculate the T_b period as well as the RTC

refresh rate, which are dependent from application and transmission period. For example, in the worst case, when the end node integrates an 80 ppm crystal, an error of 200 ms is generated every 40 min.
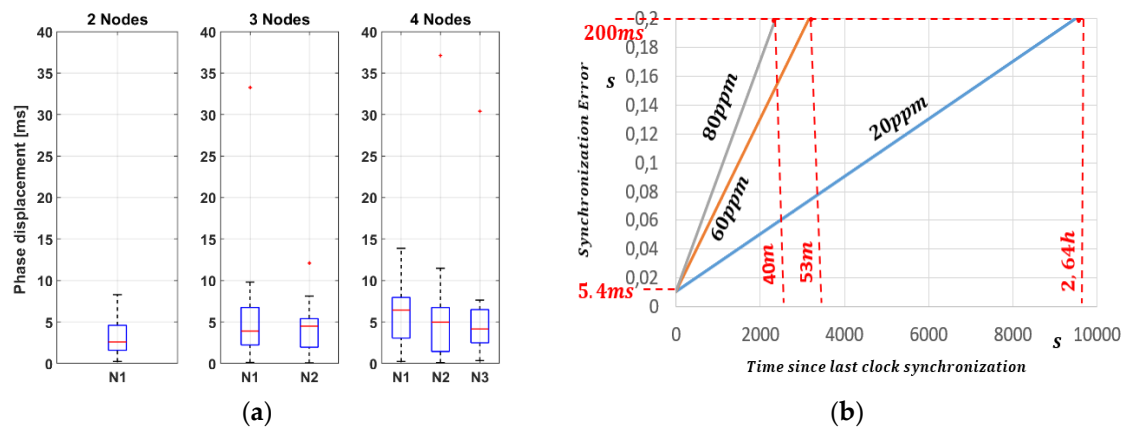

Figure 8. (**a**) Median, 25th, and 75th of synchronization error between 2–4 end nodes; (**b**) typical crystal drift over time.

6. S-ALOHA Implementation over LoRaWAN

Once implemented, at the application level, an efficient and accurate synchronization system that guarantees acceptable uncertainties of the order of milliseconds, it is necessary to introduce specific functions for the transmission management on top of LoRaWAN MAC level. On the end node side, the application layer must schedule the uplink transmission only at the beginning of each slot, starting an immediate transfer request when the internal RTC triggers the event. For the calculation of the time T_r it is necessary to consider the fact that the LoRa radios allow different types of configurations depending on the spreading factor, bandwidth, coding rate and, above all, depending on the length of the payload of the packet sent. The configuration used for the tests carried out in the following chapter is made up with these parameters: SF = 8, BW = 125 KHz, CR = 1, Preamble = 8 bytes, and Payload = 200 bytes. Following [32], we can estimate the uplink time of air, equal to 553.47 ms. This means that considering the eight bytes added into ACK needed for synchronization procedure, the lower bound of the slot duration is 1.615 s (T_r), which is equal to $T_{LoRaWAN}$ portrayed in Figure 4. The size of the packets is considerable and close to the maximum allowed of 255 bytes, this to maintain an acceptable ratio between the time used to transfer the data (T_r) and the overhead time (T_b). This result has been calculated considering that the response in the first window RX1 must occur with the same transmission parameters of the end node. Moreover, it is avoided that the transmission of the ACK message collides with other transmissions being completely included within the time of a slot. The calculation of the T_b time, the tolerance interval, takes place considering the average alignment error introduced by the synchronization algorithm and the clock drift introduced by the quartz oscillator embedded in the microcontroller. Considering a crystal with an error of 80 ppm/°C (worst case), a phase shift of 200 ms is generated every 40 min, a considerable and not negligible value in the case of devices that transmit a few times in a day. For the calculation of T_b, we considered a base-time of 5.4 ms given by the maximum alignment shift reached during the synchronization tests to which was added, to consider the clock drift, a time margin of about 400 ms in order to have an interval of at least 80 min between one synchronization and another. Lastly, a T_b of about 385 ms has been chosen, then a slot time T with a total duration of 2 s, with a T_b/T_r ratio = 25%.

In addition to the definition of the slot time, to implement the S-ALOHA protocol and make evaluations about the performances and improvements introduced, it was decided to force the transmission of the nodes on a single channel (channel 6). This is to avoid the random selection algorithm of the transmission channel by the LoRa MAC, which would involve the evaluation of the performance on several channels, which goes beyond the scope of this paper.

The S-ALOHA protocol must provide confirmation of the receipt of packets. In case of loss of packets, the end node needs to schedule retransmission in a random slot (random backoff). The backoff algorithm has been implemented in the LoRaWAN application layer, which allows calculating the wait time before retransmitting as a multiple of the single slot time. In the function, a random natural number is calculated within the interval that can be set by the user [0, NSLOTS]. The randomness in the choice of the slot in which to perform the retransmission is a fundamental feature to avoid devices entering pathological repeated interference corner cases.

S-LoRaWAN is implemented at the application layer through the block diagram presented in Figure 9, using two timers and the LoRa transmission libraries. The S-ALOHA state machine has five states: (i) Init: in this state, the timers used during firmware execution are initialized. There are three timers: *TxLoraSlotted* for the calculation of the time to wait to transmit in the first available slot; *TimerBackOff* for the choice of a random slot for retransmission, and *TimerAck* at the expiration of which packet acknowledgment receipt is checked. Following the initializations, the system transitions into the Wait state, in which a transmission request is expected. In this state, the MCU is in sleep. (ii) Wait: *TimeToSend* variable is periodically checked, when it assumes logical value true it is necessary to transmit a new packet to the LoRaWAN network. The state transition between Wait and Send occurs with the triggering of the *TxLoraSlotted* timer in correspondence of the first time slot in which to transmit (as foreseen by the Slotted Aloha protocol). (iii) Send: in this state, the transmission function is executed, which is responsible for sending the data and calling the *PrepareTxFrame* and *SendFrame* functions. These functions deal respectively with the preparation of the payload and the communication with the LoRa transceiver. Following transmission, the system automatically transits to the WaitForAck state. (iv) WaitForAck: the system remains in this state until the timer *TimerAck* is triggered or the acknowledgement is received. If the ACK is received, the system returns to the Wait state until the next transmission request, otherwise the system transitions to the Rand state where the backoff algorithm is executed. (v) Rand: as a result of non-acknowledgment reception, the MCU programs the transmission in a randomly selected slot. The *TimerBackOff* is programmed and the device remains in the Rand state until it expires. When it expires, the system automatically transits to the Send state where the message is retransmitted.
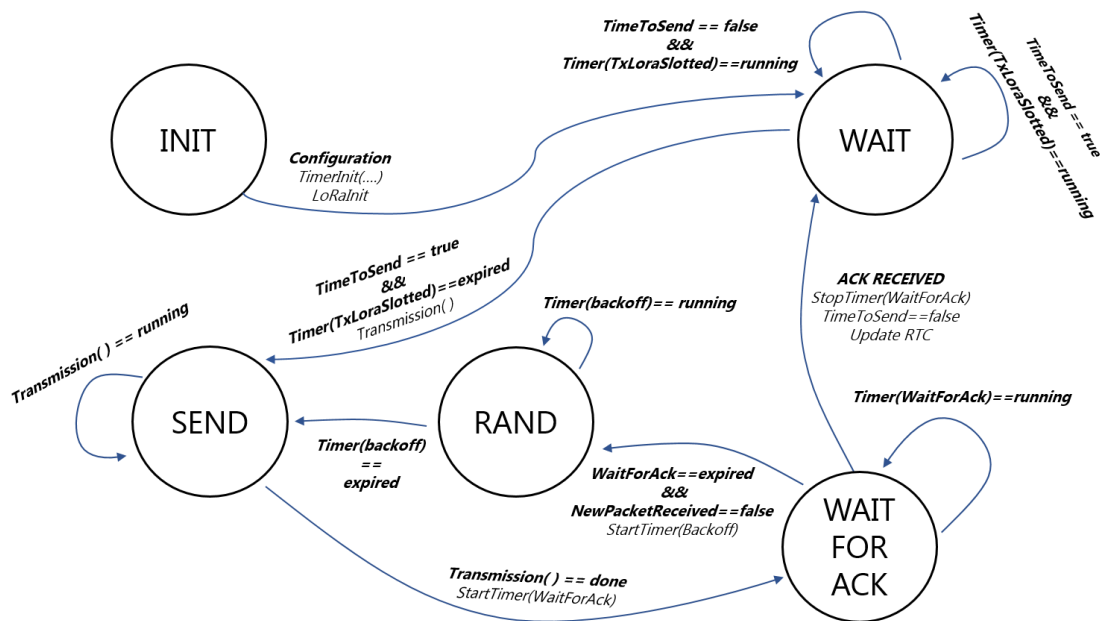

Figure 9. S-LoRaWAN block diagram.

7. S-ALOHA Evaluation in a Real Deployment

7.1. The end Device

The end node used in this paper is based on a custom board developed for multiple purposes. It embeds a STM32L476 (STMicroelectronics, Agrate Brianza, Italy) the LoRa RFM95W transceiver, an energy harvester sub-circuit (BQ25570) that manages the power supply, a temperature and humidity sensor (SHT21). Lastly, an expansion connector enables the board to be connected with several analogue and digital external sensors. The current in sleep mode is 4 µA at 3 V with the RTC enabled; the STM32L476 uses 8.25 mA in RUN mode @48 MHz. The RFM95W power consumption in TX and RX is respectively 76 (@10 dBm) and 11.5 mA. The LoRaWAN firmware comes from I-CUBE-LRWAN libraries package from STMicroelectronics, which is configured to work as our S-LoRaWAN. If the MCU is programmed to send a packet every 30 s, the battery (1000 mAh) lifespan is seven months. Figure 10 presents a picture of the board and the related block diagram.
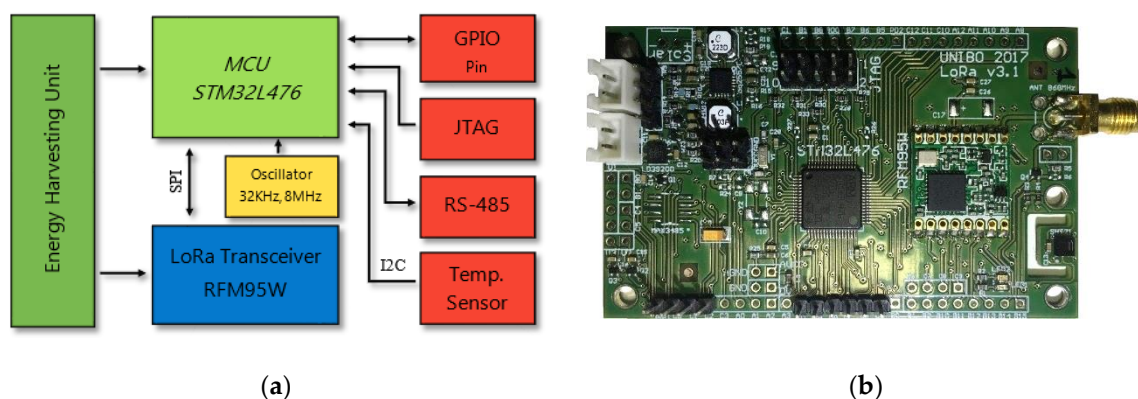

(a) (b)

Figure 10. End device HW. (**a**) Hardware block diagram; (**b**) picture of the custom board.

7.2. The Gateway

The LoRaWAN gateway is a MultiConnect Conduit device. It is a highly configurable gateway for industrial IoT applications. LTE, 3G and 2G, plus Ethernet, are available to deploy network connections and dataa management. The LoRaWAN radio module embeds the SX1301 and two SX1257 (Semtech, Camarillo, USA) which demodulate the packets received simultaneously on all channel and SFs. In our deployment, the gateway is connected through ethernet in LAN with the server, and the application software runs on top of Node Red [37].

7.3. The Environments

We tested our custom sensor node in different deployments and applications. The results presented in this paper are acquired placing the network indoor, allowing an easy setup and configuration management for all the tests presented in the following chapters. We tested our S-LoRaWAN in CINECA (Figure 11), the Tier0 supercomputing center for scientific research in Italy, where our end-nodes were used to monitor in a distributed fashion the temperature evolution of the Computer Air Conditioners with the goal to improve the cooling efficiency, as an improvement to other WSNs implementations [38].

We demonstrated [27] a packet collision reduction of 3.4× in comparison with a standard LoRaWAN WSN with 20 end-nodes. Moreover, we tested our S-ALOHA in a structural monitoring application with a custom sensor node developed to measure the width of cracks in building walls [25]. This application would require hundreds of sensors, which generate thousands of bytes per day. This traffic can overcrowd a standard LoRaWAN network; indeed, our S-ALOHA enables the possibility to double the data exchanged on the network or increase the overall number of sensors.
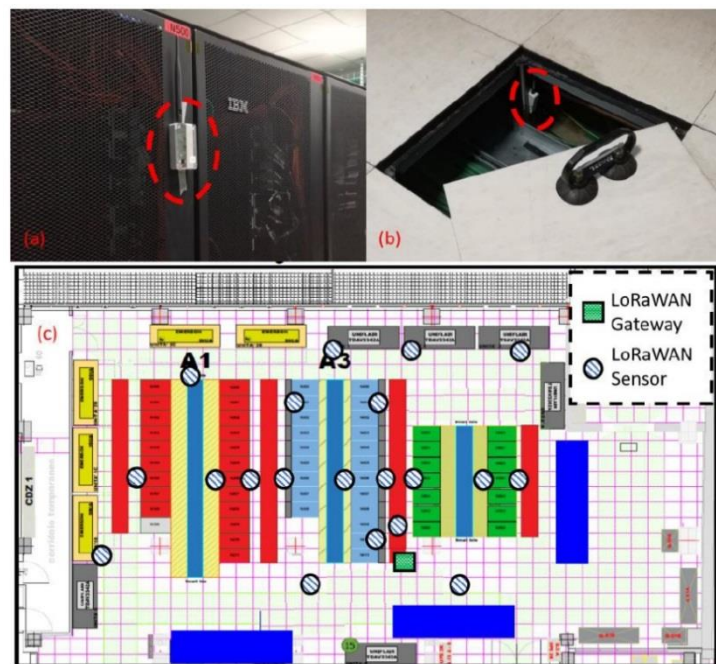
Figure 11. Sensor node deployment in CINECA. (**a**) Hallway positioning, (**b**) the sensor node is under the data center floor, (**c**) CINECA data center map.

7.4. S-ALOHA Evaluation

In order to accomplish a performance analysis of the S-ALOHA on LoRaWAN and to make a comparison with the standard LoRaWAN, a detailed log of the status of each device is collected by a database, which is managed by LoRaWAN server. Two packet-send and packet-loss counters were introduced in the state machine indicating the total number of packets that each node sent and those gone lost. The packet-send counter is incremented every time the microcontroller is in the sending state, while to check if the acknowledgment has been received a logical value is set to true a flag present in the receiver function, whenever it assumes logical value true it means that it is received a packet in response, otherwise the S-ALOHA state machine resumes the transmission by increasing the packet-loss counter. Moreover, the gateway detects the channel access time of each end-node, measuring the effective RTC synchronization of the WSN. The payload of each message sent by the end nodes contains the log information, allowing through a debug window to read in real-time the statistics of each node. In order to make the analysis possible, however, it was necessary to memorize this data on a support that can be analyzed remotely with numerical analysis programs such as Matlab or Python.

Tests have been carried out to assess whether the introduction of the S-ALOHA protocol in the LoRa stack leads to an improvement in the overall performance of the network. The LoRaWAN protocol is similar both in terms of channel access and as a transmission algorithm to the P-ALOHA, but it is necessary to consider the fact that the statistical analysis of the P-ALOHA does not take into account the channel occupation for acknowledgment messages. The transmission of the ACK by the gateway in the same channel can, in fact, collide with the other frames that have in the meantime been transmitted by the other devices.

We carried out three tests for the S-ALOHA protocol in three different network traffic conditions with the goal to evaluate the performances gap between the theory, statistical models, and a real deployment. The main variables of these tests are the number of devices as well as the transmission frequency of the messages. Indeed, other parameters, such as payload size and SF are kept constant. In all tests, confirmed messages are used and the same firmware S-ALOHA with a back-off algorithm is implemented in whole devices. The following tables and plots show the comparison between the

performance of both protocols in terms of traffic offered and disposed of by the network. These values are temporally normalized, so the results are independent of the duration of the experiment and the overall use of the channel is expressed in percentage value.

For the channel traffic calculation (G) the total number of packets sent in the channel in the time unit (packet/second) was considered; this means that, on average, values greater than $G = 1$ denote that multiples end nodes transmit simultaneously in the same slots. The overall throughput from the channel is therefore given by the product of G for the average percentage of packages successful disposed of, as described in Equations (7) and (8). The packet time is calculated considering the T_{LoRaWAN}:

$$G = \frac{\text{Packet}_{\text{sent}}}{T_{\text{test}}} \tag{7}$$

$$S = G * P_{\text{success}} \tag{8}$$

Values reported in Table 2 and Figure 12 confirm a correct implementation of the S-ALOHA over a LoRaWAN protocol, indeed the maximum theoretical throughput is doubled, with a trend similar to standard S-ALOHA statistical model.

Table 2. Summary of the results obtained from the tests with Slotted LoRaWAN.

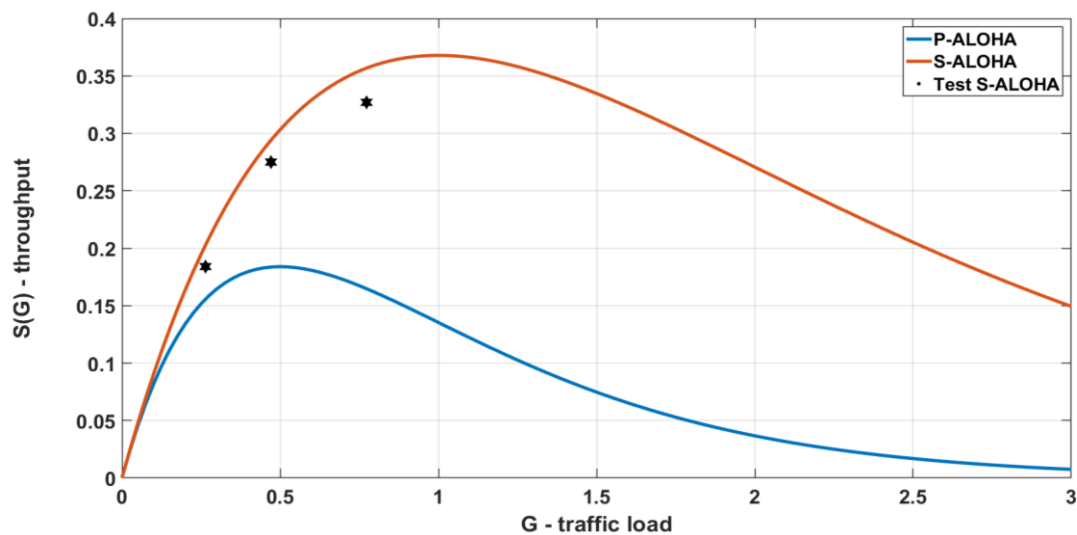| Test	Test 1	Test 2	Test 3
Test duration [s]	10273	6977	3524
Packets sent	2708	3285	2894
N of end nodes	10	18	24
TX period [s]	15	15	13
TX success rate [%]	69	55	43
Traffic generated (G)	0.264	0.471	0.773
Channel throughput (S)	0.184	0.275	0.332

Figure 12. Summary of the results obtained from the tests with Slotted LoRaWAN.
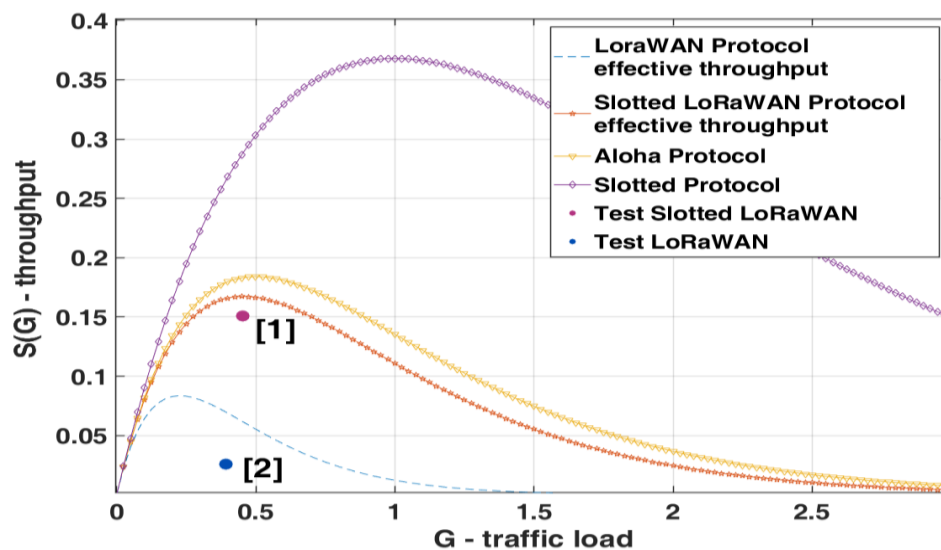
8. Comparison between LoRaWAN and Slotted LoRaWAN

A comparison between the standard LoRaWAN and the S-LoRaWAN is presented in this chapter. The same parameters of the radio configurations and the traffic injected are used for both protocols. In this test the channel traffic and the corresponding throughput are calculated considering only the payload time (T) as traffic generated, the following conditions applied in Equation (6). Moreover, with this approach the tolerance interval (T_{b}) inserted into the slot time is considered into the protocol overhead. The channel throughput in Table 3 describes the effective payload transmitted on the channel; this allows a more accurate comparison without considering the protocol overhead.

Table 3. Comparison between LoRaWAN and Slotted LoRaWAN.

Test	Slotted LoRaWAN	LoRaWAN
Test duration [s]	3524	9596
Packets sent	2894	6817
N of end nodes	24	24
TX period [s]	13	13
TX success rate [%]	33	7
Traffic generated (G)	0.455	0.393
Channel throughput (S)	0.150	0.026

Referring to the traffic analysis, Table 3 shows the comparison between the S-LoRaWAN performed with the S-ALOHA MAC (Figure 13 point 1) and the LoRaWAN standard protocol (Figure 13 point 2). Interestingly, the generated traffic resulting from the two protocols is slightly different, this is partly due to the statistical evolution of transmissions; in fact, in the case of the LoRaWAN standard protocol, the overlapping of the frames is much more probable, with consequent recourse to the backoff algorithm and an average reduction in the number of packets transmitted. In the proposed results, the measured throughput of S-LoRaWAN and standard LoRaWAN is respectively 15% and 2.6%. Finally, the measured improvement is 5.8x. This result is higher than the theoretical 2x between S-ALOHA and P-ALOHA; indeed, it has to be considered that the maximum throughput is not placed at the same *G* (Figure 13), and with the parameters used in this test, the P-ALOHA channel is heavily crowded with a TX success rate of 7%.

Figure 13. Comparison between LoRaWAN and Slotted LoRaWAN.

9. Synchronization Analysis in Test 3

An in-depth analysis of the synchronization between end nodes in the network is presented here. Test 3 of Section 7 is used as the reference, indeed the high generated traffic and the high packet collision rate on the channel tresses the synchronization procedure. The error is calculated between t_0 (Figure 1a) and the packet time of arrival measured by the gateway. With the T_b proposed in Test 3 the overall number of failed slots for all 24 nodes, is five, with a success rate of 99.6%. In this case, an overlap happens between adjacent slots because the tolerance interval is exceeded. As presented in Figure 14a, in point 1 and 2 two consecutive slot overlaps are shown, and both are generated from the same end node. This issue comes from end node reboot; indeed, when a firmware error is detected the MCU is restarted and the internal RTC is not configured. Therefore, the overall synchronization failures are three.
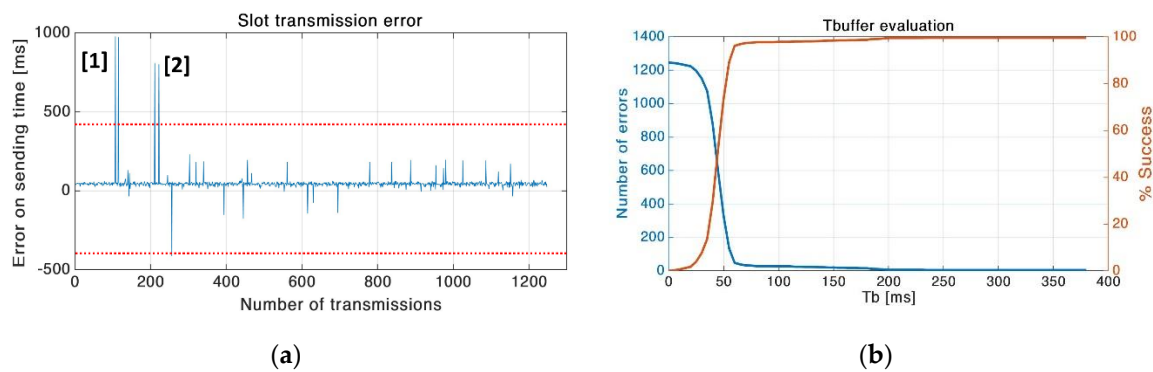
Figure 14. (**a**) Synchronization error in ms respect to the beginning of each slot. The maximum error allowed is 400 ms, which is exceeded five times. Two overflows are generated from end node reboot due to firmware issues; (**b**) number of slots overlapped as a function of tolerance interval length.

The previous tests are executed with a tolerance interval (T_b) calculated to fulfill the RTC drift in the worst case, we evaluated the S-LoRaWAN throughput in this condition to compare both protocols in a typical low power application. However, in Test 3 the T_b could be decreased ensuring a slot success rate over 90%; in Figure 14b and Table 4 is shown the relationship between T_b and the number of errors. For example, with a T_b of 100ms the error rate is practically the same as Test 3, and with 50 ms the error rate is still acceptable. The tolerance interval length affects the channel throughput; indeed, reducing T_b decreases the protocol overhead, and allows more slots in a unit time.

Table 4. The number of slots failed respect to tolerance interval shrinkage.

T_b [ms]	50	100	150	200	250	300
N slot errors	328	27	20	6	5	5
Success rate [%]	74	97	98	99	99	99

10. Conclusions

We have presented an analysis of the LoRaWAN protocol focused on channel throughput and issues associated with its scalability. Moreover, we have implemented a reliable synchronization method, used to develop an S-LoRaWAN protocol suitable for low-cost and low-power IoT devices. This approach gives a theoretical 2× network throughput improvement, but in high traffic set-up, the measured improvement is up to 5.8× with a demonstrated reduction of packet collisions of 26% in a real-life deployment with 24 nodes operating for hours. The overall overhead of the proposed S-LoRaWAN is only eight bytes in the downlink packet, with a negligible consequence in term of power consumption, which is a fundamental factor for battery operated devices. Finally, our S-LoRaWAN does not require any change to the LoRaWAN software stack and can be deployed on top of unmodified LoRaWAN firmware.

Appendix A

The duration of uplink and downlink transmissions depends on parameters of LoRa modulation such as SF (Spreading Factor), BW (Bandwidth), CR (Coding Rate, values from 1 to 4) and can be expressed as the sum of the time needed to transmit the preamble and the physical message (Equation (A1)):

$$T_{\text{tx}} = T_{\text{preamble}} + T_{\text{PHY Message}} \tag{A1}$$

Equations (A2) and (A3) represent how these two terms have been calculated, where N_{preamble} is the number of symbols used by the radio transceiver as the physical preamble of the message and N_{PHY} indicates the number of symbols transmitted in the physical message and can be determined as shown in Equation (5). T_{sym} (Equation (A4)) is the duration (in seconds) of a symbol and depends on SF and BW:

$$T_{\text{preamble}} = T_{\text{sym}} \cdot \left(N_{\text{preamble}} + 4.25 \right) \tag{A2}$$

$$T_{\text{PHY Message}} = T_{\text{sym}} \cdot N_{\text{PHY}} \tag{A3}$$

$$T_{\text{sym}} = \frac{2^{\text{SF}}}{\text{BW}} \tag{A4}$$

$$N_{\text{PHY}} = 8 + \max \left[ceil \left[\frac{28 + 8 \cdot \text{PL} + 16 \cdot \text{CRC} - 4 \cdot \text{SF}}{4 \cdot (\text{SF} - 2 \cdot \text{DE})} \right] \cdot (\text{CR} + 4), \, 0 \right] \tag{A5}$$

In Equation (A5) Payload Length (PL) denotes the number of bytes in the physical payload, CRC indicates the presence (value 1) or not (value 0) of the CRC field in the physical message and DE indicates if the mechanism to prevent issues about the clock drift of the crystal reference oscillator is used (value 1 for SF12 and SF11, 0 for others). The transmission data rate can be obtained as shown in Equation (A6):

$$\text{DR} = \text{SF} \cdot \frac{\text{BW}}{2^{\text{SF}}} \cdot \frac{4}{\text{CR} + 4} \tag{A6}$$

References

1. Raza, U.; Parag, K.; Mahesh, S. Low power wide area networks: An overview. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 855–873. [CrossRef]
2. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gen. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
3. Petajajarvi, J.; Mikhaylov, K.; Roivainen, A.; Hanninen, T.; Pettissalo, M. On the coverage of lpwans: Range evaluation and channel attenuation model for Lora technology. In Proceedings of the 2015 14th International Conference on ITS Telecommunications (ITST), Copenhagen, Denmark, 2–4 December 2015; pp. 55–59.
4. Adelantado, F.; Vilajosana, X.; Tuset-Peiro, P.; Martinez, B.; Melia-Segui, J.; Watteyne, T. Understanding the limits of LoRaWAN. *IEEE Commun. Mag.* **2017**, *55*, 34–40. [CrossRef]
5. Augustin, A.; Yi, J.; Clausen, T.; Townsley, W.M. A study of LoRa: Long range & low power networks for the internet of things. *Sensors* **2016**, *16*, 1466.
6. Nolan, K.E.; Guibene, W.; Kelly, M. An evaluation of low power wide area network technologies for the Internet of Things. In Proceedings of the 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, Cyprus, 5–9 September 2016; pp. 439–444.
7. Petäjäjärvi, J.; Mikhaylov, K.; Pettissalo, M.; Janhunen, J.; Iinatti, J. Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. *Int. J. Distrib. Sens. Netw.* **2017**, *13*. [CrossRef]
8. Petäjäjärvi, J.; Mikhaylov, K.; Yasmin, R.; Hämäläinen, M.; Iinatti, J. Evaluation of LoRa LPWAN technology for indoor remote health and wellbeing monitoring. *Int. J. Wirel. Inf. Netw.* **2017**, *24*, 153–165. [CrossRef]
9. Neumann, P.; Julien, M.; Thomas, N. Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study. In Proceedings of the 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, NY, USA, 17–19 October 2016.

10. Magno, M.; Aoudia, F.A.; Gautier, M.; Berder, O.; Benini, L. WULoRa: An energy efficient IoT end-node for energy harvesting and heterogeneous communication. In Proceedings of the Conference on Design, Automation & Test in Europe, Lausanne, Switzerland, 27–31 March 2017; European Design and Automation Association: Nice, France, 2017; pp. 1532–1537.

11. Sartori, D.; Brunelli, D. A smart sensor for precision agriculture powered by microbial fuel cells. In Proceedings of the Sensors Applications Symposium (SAS), Catania, Italy, 20 April 2016; pp. 1–6.

12. Sornin, N.; Luis, M.; Eirich, T.; Kramp, T.; Hersent, O. *Lorawan Specification*; LoRa Alliance, Inc.: San Ramon, CA, USA, 2015.

13. Alliance, L. LoRaWAN™ Certified Products. 2017. Available online: https://www.lora-alliance.org/Products/Certified-Products (accessed on 19 January 2019).

14. Bor, M.C.; Roedig, U.; Voigt, T.; Alonso, J.M. Do LoRa low-power wide-area networks scale? In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Miami, FL, USA, 21–25 November 2016; pp. 59–67.

15. Bankov, D.; Khorov, E.; Lyakhov, A. On the limits of LoRaWAN channel access. In Proceedings of the 2016 International Conference on Engineering and Telecommunication (EnT), Moscow, Russia, 29–30 November 2016.

16. Georgiou, O.; Raza, U. Low power wide area network analysis: Can LoRa scale? *IEEE Wirel. Commun. Lett.* **2017**, *6*, 162–165. [CrossRef]

17. Centenaro, M.; Vangelista, L.; Zanella, A.; Zorzi, M. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wirel. Commun.* **2016**, *23*, 60–67. [CrossRef]

18. Yang, C.; Liang, O.; Ontanon, S.; Ke, W. Predictive Modeling with Vehicle Sensor Data and IoT for Injury Prevention. In Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Philadelphia, PA, USA, 18–20 October 2018.

19. Porcarelli, D.; Brunelli, D.; Benini, L. Clamp-and-Forget: A self-sustainable non-invasive wireless sensor node for smart metering applications. *Microelectron. J.* **2017**, *45*, 1671–1678. [CrossRef]

20. Goursaud, C.; Gorce, J.-M. Dedicated networks for IoT: PHY/MAC state of the art and challenges. *EAI Trans. Internet Things* **2015**. [CrossRef]

21. Margelis, G.; Piechocki, R.; Kaleshi, D.; Thomas, P. Low throughput networks for the IoT: Lessons learned from industrial implementations. In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 181–186.

22. Weyn, M.; Ergeerts, G.; Wante, L.; Vercauteren, C.; Hellinckx, P. Survey of the DASH7 alliance protocol for 433 MHz wireless sensor communication. *Int. J. Distrib. Sens. Netw.* **2013**. [CrossRef]

23. ISO-IEC. *ISO/IEC 18000-7:2014 Information Technology "Radio Frequency Identification for Item Managemen "Part 7: "Parameters for Active Air Interface Communications at 433 MHz"*; ISO: Geneva, Switzerland, 2014.

24. *LoRaWAN™, Specification v1.0*; LoRa Alliance, Inc.: San Ramon, CA, USA, 2015.

25. Polonelli, T.; Brunelli, D.; Guermandi, M. An accurate low-cost Crackmeter with LoRaWAN communication and energy harvesting capability. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; Volume 1.

26. Van den Abeele, F.; Haxhibeqiri, J.; Moerman, I.; Moerman, J. Scalability analysis of large-scale LoRaWAN networks in ns-3. *IEEE Internet Things J.* **2017**, *4*, 2186–2198. [CrossRef]

27. Polonelli, T.; Brunelli, D.; Benini, L. Slotted ALOHA Overlay on LoRaWAN-A Distributed Synchronization Approach. In Proceedings of the 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), Bucharest, Romania, 29–31 October 2018; pp. 129–132.

28. Tessaro, L.; Raffaldi, C.; Rossi, M.; Brunelli, D. Lightweight Synchronization Algorithm with Self-Calibration for Industrial LORA Sensor Networks. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 259–263.

29. Tessaro, L.; Raffaldi, C.; Rossi, M.; Brunelli, D. LoRa Performance in Short Range Industrial Applications. In Proceedings of the 2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), Amalfi, Italy, 20–22 June 2018; pp. 1089–1094.

30. Rossi, M.; Tosato, P.; Gemma, L.; Torquati, L.; Catania, C.; Camalò, S.; Brunelli, D. Long range wireless sensing powered by plant-microbial fuel cell. In Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, 27–31 March 2017; pp. 1651–1654.

31. Bor, M.; Vidler, J.E.; Roedig, U. Lora for the Internet of Things. In Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN '16), TU Graz, Austria, 15–17 February 2016; pp. 361–366.

32. Casals, L.; Mir, B.; Vidal, R.; Gomez, C. Modeling the energy performance of lorawan. *Sensors* **2017**, *17*, 2364. [CrossRef] [PubMed]

33. Abramson, N. HE ALOHA SYSTEM: Another alternative for computer communications. In Proceedings of the ACM Fall Joint Computer Conference, Houston, TX, USA, 17–19 November 1970.

34. Jeong, D.; Jeon, W. Performance of an Exponential Backoff Scheme for Slotted-ALOHA Protocol in Local Wireless Environment. *IEEE Trans. Veh. Technol.* **1995**, *44*, 470–479. [CrossRef]

35. SX1276/77/78/79 Transceivers. (01/2019). Available online: https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf (accessed on 19 January 2019).

36. Kleinrock, L.; Tobagi, F. Packet switching in radio channels: Part I—Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Trans. Commun.* **1975**, *23*, 1400–1416. [CrossRef]

37. Conduit AEP: LoRa Communication and Node-RED. Hoboken, NJ, USA, 2019. Available online: http://www.multitech.net/developer/software/lora/conduit-aep-lora-communication/ (accessed on 19 January 2019).

38. Rossi, M.; Rizzon, L.; Fait, M.; Passerone, R.; Brunelli, D. Energy Neutral Wireless Sensing for Server Farms Monitoring. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2014**, *4*, 324–334. [CrossRef]