



Article

An Efficient Adaptive Data-Link-Layer Architecture for LoRa Networks

Micael Coutinho ¹, Jose A. Afonso ^{2,*}  and Sérgio F. Lopes ^{3,*}¹ Center of Physics, University of Minho, 4804-533 Guimarães, Portugal; b8719@fisica.uminho.pt² CMEMS-UMinho/LABELS, University of Minho, 4800-058 Guimarães, Portugal³ Centro Algoritmi/LASI, University of Minho, 4804-533 Guimarães, Portugal

* Correspondence: jose.afonso@dei.uminho.pt (J.A.A.); sergio.lopes@dei.uminho.pt (S.F.L.)

Abstract: LoRa is one of the most popular low-power wireless network technologies for implementation of the Internet of Things, with the advantage of providing long-range communication, but lower data rates, when compared with technologies such as Zigbee or Bluetooth. LoRa is a single-channel physical layer technology on top of which LoRaWAN implements a more complex multi-channel network with enhanced functionalities, such as adaptive data rate. However, LoRaWAN relies on expensive hardware to support these functionalities. This paper proposes a LoRa data-link-layer architecture based on a multi-layer star network topology that adapts relevant LoRa parameters for each end node dynamically taking into account its link distance and quality in order to balance communication range and energy consumption. The developed solution is comprised of multiple components, including a LoRa parameter calculator to help the user to configure the network parameters, a contention-free MAC protocol to avoid collisions, and an adaptive spreading factor and transmission power mechanism. These components work together to ensure a more efficient use of the chosen ISM band and end node resources, but with low-cost implementation and operation requirements.

Keywords: LoRa; data link layer; spreading factor; transmission power; MAC protocol



Citation: Coutinho, M.; Afonso, J.A.; Lopes, S.F. An Efficient Adaptive Data-Link-Layer Architecture for LoRa Networks. *Future Internet* **2023**, *15*, 273. <https://doi.org/10.3390/fi15080273>

Academic Editors: Panagiotis Papageorgas, Dimitrios Piromalis and Dionisis Kandris

Received: 24 July 2023

Revised: 13 August 2023

Accepted: 15 August 2023

Published: 17 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Low-power wide-area networks (LPWANs) play a key role in the implementation of the IoT (Internet of Things) concept, and LoRa (which stands for “long range”) is among the most popular of these technologies [1]. LoRa defines a spread spectrum modulation technique based on the chirp spread spectrum (CSS) [2]. Covering the physical layer of the OSI (open systems interconnection) model, it makes use of the sub-gigahertz ISM (industrial, scientific, and medical) bands to enable low-power and long-range communication that can reach up to 10 km in rural areas.

The LoRaWAN open specification [3] is built over the LoRa physical layer, providing upper-layer protocols and the means to connect the LoRa devices (end nodes) to the Internet. It defines the device-to-infrastructure LoRa parameters and the LoRaWAN protocol, allowing for interoperability between devices. LoRaWAN has established itself as one of the main IoT communication stacks for low-power and lossy networks (LLNs).

The LoRaWAN network architecture consists of a star-of-stars topology (see Figure 8 in [2]), where gateways forward packets from end nodes (usually, data collected from sensors) to a network server and vice versa. The network server manages the network by performing functions such as deduplicating received packets and controlling the data rate. The data can then be accessed by the user through the Internet on application servers and cloud services.

While LoRa is a single-channel point-to-point (P2P) technology, LoRaWAN is a more complex multi-channel network where several low-cost end nodes may communicate

simultaneously with high-cost LoRaWAN gateways using different channels and spreading factors (SFs). The LoRaWAN protocol controls the LoRa physical layer through the implementation of useful functionalities such as adaptive data rate (ADR), which manages the transmission power and spreading factor according to the received packet SNR (signal-to-noise ratio). However, it relies on expensive hardware to support these functionalities.

A functional LLN can be implemented on top of LoRa, instead of relying on LoRaWAN, and this may be a reasonable approach for simple network topologies (such as star) in low-power applications. By following this method, one can detach the network from a specific LoRaWAN infrastructure, which leads to lower costs. However, most works that pursue this approach do not consider adopting a mechanism similar to ADR from LoRaWAN, relying instead on a fixed physical layer parameter set. This leads to inefficient use of networks and end node resources, such as bandwidth and energy, which may be critical for the lifetime of end nodes, since they are typically battery-operated in many application scenarios.

Implementing an efficient LLN on top of LoRa requires a deep knowledge of the physical layer parameters and the development of a data-link-layer protocol. Transmission power (TxP), spreading factor (SF) and bandwidth (Bw) are three important and intrinsically related LoRa parameters because they have impact on range, data rate, and energy consumption. For example, a high TxP provides a long communication range but increases the energy consumption, and a higher SF increases the range but reduces the data rate (which can impact throughput), whereas increasing the used Bw increases the data rate but reduces range.

Our contribution in this paper is the proposal and evaluation of a LoRa data-link-layer architecture based on a multi-layer star network topology that dynamically adjusts the LoRa parameters of each end node taking into account the SNR in order to balance its communication range and energy consumption. The designed architecture takes into account that the end nodes usually are placed at different distances from the gateways and, therefore, require different parameter configurations to achieve a satisfactory link quality. The proposed architecture ensures efficient use of the chosen ISM band and low-cost implementation and operation, i.e., without requiring the use of third-party LoRaWAN services or infrastructure.

The rest of this paper is organized as follows: Section 2 identifies the related work, whereas Section 3 provides the background regarding the relevant LoRa parameters. Section 4 describes the proposed link layer architecture and each of its components, whereas Section 5 presents the results and discussion. Finally, Section 6 outlines the conclusions and suggestions of future developments.

2. Related Work

The efficient implementation of LLNs and other IoT networks and technologies requires the contribution of several research topics. Among them, data aggregation mechanisms are important to avoid data duplication and increase the sensors' measurement precision, especially when they are vulnerable to environmental factors (e.g., radiation, temperature, pressure variations, potential mobility, etc.) [4]. Likewise, security mechanisms are important to mitigate different types of attacks (e.g., eavesdropping, denial of service, data integrity attacks, etc.) [5]. Although these aspects are highly relevant, they are out of the scope of this paper.

Several works have addressed the application scenarios of LLNs [6,7]. Among the works making contributions to LoRa, some propose alternative implementations [8] and improvements [9,10] to the physical layer. Other works step up in the protocol stack and propose improvements to the LoRaWAN architecture [11,12]. Many works have explored the LoRa physical layer as a means to communicate. However, unlike our paper, most of these works merely apply the technology with fixed parameters for each end node [13,14], trying to maximize the range of communication while maintaining a reasonable throughput. These and other more closely related works are analyzed next.

In [13], an agricultural field monitoring application with irrigation control was developed. It consists of a network of end nodes for sensing or irrigation and a gateway node to connect the application to the Internet. Both the end nodes and gateway employ the LoRa physical layer to communicate by using the same single-channel link. Each sensor node sends its data every five minutes, and every data packet receives a downlink acknowledgment (ACK).

In [14], a wireless sensor network (WSN) is used to collect environmental data used to optimize irrigation timings. The network uses the LoRa physical layer between sensor nodes and an aggregating node with a single-channel transceiver. The sensor nodes send environmental data every 10 min. The aggregator node is capable of sustaining a network of up to 250 nodes, configuring them automatically at boot, and all messages are encrypted. Since there are no ACK packets, there is no way of easily detecting a lossy link. Like in the previous work, the LoRa parameters are fixed, so the use of the resources from the network and nodes is not optimized.

In [15] the capabilities of a LoRa single-channel device are enhanced by enabling it to receive packets coming with different and unknown spreading factors. This is achieved by performing channel activity detection (CAD) on each SF and detecting the preamble of incoming packets, taking into consideration the probability of adjacent SF detection. The developed adaptive spreading factor selection (ASFS) algorithm can serve as a basis for having end nodes with different SF configurations in the same network. Although there is no mention of how the preamble length is calculated for each SF, it is indicated that it should be long enough so that the receiver can successfully receive the packets. Since there are no ACKs, a similar version to the ADR algorithm cannot be applied, so in this work, the SF is kept fixed for each node.

In [16], the authors propose a data-link-layer protocol to improve the communication reliability (packet delivery ratio) and energy efficiency for indoor LoRaWAN networks. The approach is the introduction of booster nodes, which may be a subset of the LoRa end nodes. These nodes perform two tasks: (i) repeat the same physical layer frame that was not acknowledged by the gateway in synchronization (within time constraints) with the transmitting nodes and, on the same channel, improving the RSSI (received signal strength indicator) at the gateway and (ii) re-send downlink ACKs that were not received (originated by a retransmission) by the recipient end node. This approach increases the energy efficiency of LoRaWAN networks since it reduces the number of possible packet retransmissions. It is specifically suited to indoor environments, since the synchronization is much more difficult to achieve for larger outdoor distances. This work uses LoRaWAN instead of LoRa and does not control the transceiver transmission power to optimize node range and energy efficiency.

In [17], a mechanism to improve the packet delivery ratio is proposed by adapting the utilization of the channels according to estimated interference. The amount of interference in a network is estimated by gateways, from the number of packets received on each channel. From it, the probability of channel usage per node (or geographical area) is computed and this information is broadcast to end nodes, which in turn change normal channel allocation scheme according to the usage probabilities. Simulation results with different geographical distribution of nodes and packet loss scenarios show improvements over a uniform allocation of channels. This work does not consider node energy consumption and range or the medium utilization over time.

In [18], the LoRa CAD is used to add controlled access to the medium and distribute communication load between different channels/spreading factors. Three incremental algorithms are proposed, the base one avoiding collisions using an interframe spacing mechanism with a fixed number of detections and random backoff. In the second algorithm, end nodes use the results of the CADs to determine which channels are less occupied and use them for sending messages. The third algorithm is applied by the gateway, which has a complete view of the network and broadcasts the global channels load to end nodes, avoiding local biased load perceptions. Laboratory tests show that those algorithms

improve goodput and radio energy consumption comparatively to LoRaWAN's ALOHA protocol. This work's priority is to improve goodput and not energy consumption and it does not consider the optimization of node range and the usage of available medium time availability.

3. LoRa Parameter Background

To develop an efficient LoRa data-link-layer protocol, it is necessary to control SF, Bw, and other main LoRa parameters. This is performed by LoRaWAN to implement useful functionalities, such as ADR, which manages the TxP and SF according to the received packet SNR.

LoRaWAN allows the selection of bandwidths in the range between 125 and 500 kHz for uplink packets. On the other hand, the range of available bandwidths for LoRa networks is larger, with values that can go as low as 7.8 kHz. The TxP impacts the link budget directly, and its value depends on the hardware implementation and the regional limitations, where its maximum value may range from 14 to 20 dBm. The spreading factor (SF), which can vary from 7 to 12, defines the number of chips per symbol, and each step up doubles the symbol period, as shown by Equation (1), where Bw is in Hz [19].

$$\text{Symbol period} = \frac{2^{SF}}{Bw} \quad (s) \quad (1)$$

For a LoRa node to receive a packet, it must perform CAD using the same SF and Bw of the sending node. This operation takes a time expressed in the transceiver datasheet [20] as a multiple of the LoRa symbol period.

These parameters have impact on communication throughput, range, and energy consumption. As shown by Equation (1), using a higher Bw decreases the symbol period (and, consequently, increases the data rate), which allows us to increase the network throughput. A higher data rate decreases the time on air (ToA) for the packets, which means that less energy is spent to transmit them. On the other hand, the communication range decreases when Bw increases.

A higher TxP provides a longer communication range but increases energy consumption. Therefore, this parameter should be controlled to adequately balance communication range and energy consumption.

Increasing the SF increases the symbol period (Equation (1)), so the data rate is reduced. Equation (2), derived from [19], shows that a higher SF increases the processing gain of the receiver, and thus communication range is also increased. However, as seen in Equation (3), each step up of SF divides the bit rate approximately by half and, consequently, the ToA of each packet is roughly doubled. This means that more energy is required to send packets with higher spreading factors. Therefore, the SF is another LoRa parameter that can be used to balance the range and the energy consumption.

$$\text{Processing gain} = 10 \times \log_{10} \left(\frac{2^{SF}}{SF} \right) \quad (dB) \quad (2)$$

$$\text{Bit Rate} = SF \times \frac{BW}{2^{SF}} \times CR \quad (bps) \quad (3)$$

Other relevant LoRa parameters are the coding rate (CR) and header mode, which is largely application dependent. The CR improves communication robustness by performing forward error correction (FEC); its possible values in Equation (3) range from 4/8 (higher robustness) to 4/5 (lower robustness). The header mode can be implicit, implying a fixed payload length, or explicit, giving the freedom of arbitrary payload sizes.

4. Data Link Layer Design

The link layer architecture proposed in this paper comprises multiple interrelated components: network topology, ASFS algorithm, LoRa parameters calculator, MAC protocol, and the adaptive data link (ADL) mechanism. These components are described with

more detail in the respective subsections. Figure 1 shows where the components that are executed inside the end nodes and/or the packet forwarders are located.

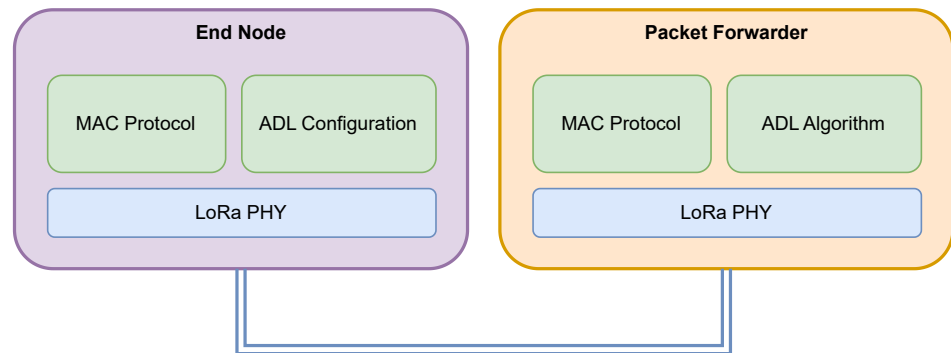


Figure 1. Components of the proposed data-link-layer architecture.

4.1. Network Topology

The designed link layer architecture uses single-channel LoRa devices as packet forwarders, as observed in Figure 2, instead of more complex multi-channel LoRaWAN gateways, in order to reduce the overall network cost.

The proposed solution is based on a two-level multi-layer topology where each set of end nodes EN_{PF_p} sends the collected data to their respective packet forwarder PF_p using a LoRa network based on the star topology. The packet forwarder relays the data received from its end nodes to a web server using any available connection to the Internet, such as Wi-Fi, cellular or Ethernet. In the developed prototype, a Wi-Fi connection was used. Through the use of multiple packet forwarders in range of the respective end nodes, the coverage area of the system can be increased.

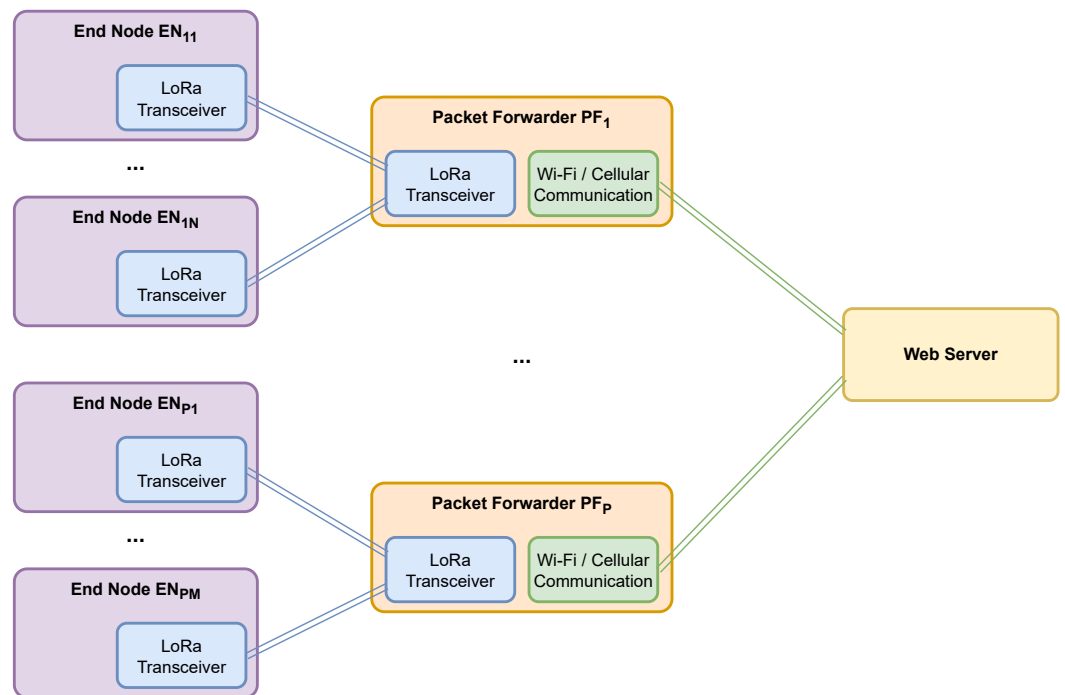


Figure 2. Network topology of the proposed link layer architecture.

In the following sections, for simplicity, the term “network” is used to refer to a single LoRa subnetwork composed of a packet forwarder and its associated end nodes instead of the larger network composed of all LoRa subnetworks.

4.2. Adaptive SF Algorithm

The solution proposed in this paper implements the ASFS algorithm, described in [15], to make it possible for the packet forwarders to correctly detect and decode packets transmitted by its end nodes using unknown spreading factors. Through this algorithm, if a preamble is not detected through CAD, the packet forwarder switches the SF until one is detected. If a preamble is detected, then it uses the same spreading factor and performs CAD again. At the third consecutive successful detection, it either prepares to receive the LoRa packet or performs CAD detection on an adjacent SF until the current SF is not able to find a valid preamble. This step is necessary for some spreading factors, because of the imperfect orthogonality between adjacent spreading factors. In the latter case, the SF is decreased by one, since it is the last valid candidate, and the LoRa transceiver is placed in reception mode. By adopting this algorithm, false SF selection is mitigated, solving the orthogonality problem stated before. The ASFS algorithm described in [15] was converted into the flowchart shown in Figure 3 and implemented in the developed prototype.

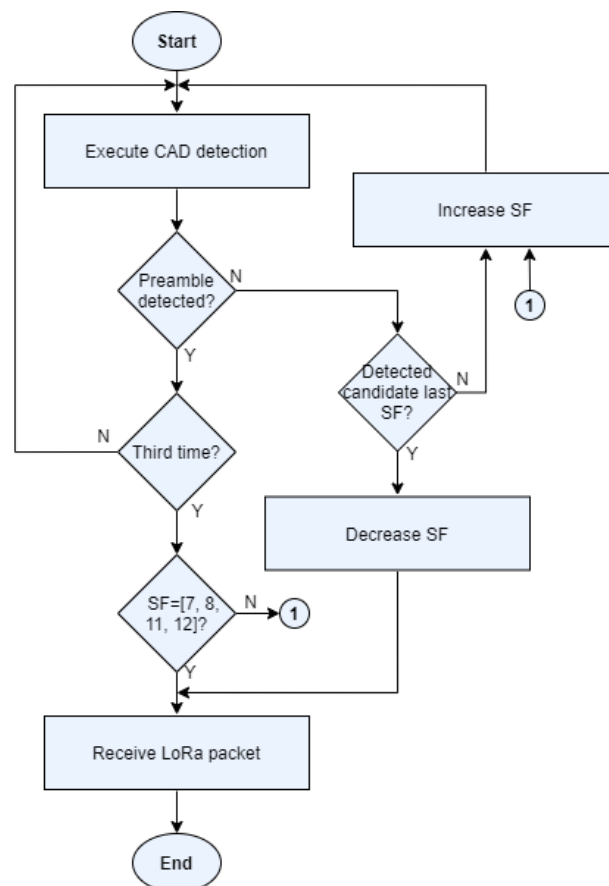


Figure 3. Flowchart of the version of the ASFS algorithm implemented in the developed prototype.

One improvement made to this algorithm is the introduction of a fixed preamble length for each SF, which has to be high enough so that the minimum amount of packets is missed. This can be calculated by analyzing the worst-case scenarios for packet detection with the ASFS algorithm, for the spreading factors that do not suffer from orthogonality ($SF = 7$) and for those that do ($SF = 10$). The preamble length needs to correspond to an entire cycle of cumulative CAD detection periods until it successfully detects the right SF. This process still has to take into account the necessary time to process the CAD detection interrupt service routine (ISR), which may include the reading of a register through SPI (serial peripheral interface) or an input GPIO (general purpose input/output) pin, along with the clearing of the necessary flags, for every CAD detection cycle. The maximum values (T_{max}) for $SF = 7$ and $SF = 12$ are shown in Equations (4) and (5), respectively,

where $T_{CAD_{sf}}$ (obtained from the transceiver datasheet [20]) is the time taken by the CAD detection period for the given SF and T_{ISR} is the period taken to process the interrupts to control the ASFS algorithm.

$$T_{max}(7) = 3T_{CAD7} + T_{CAD8} + T_{CAD9} + T_{CAD10} + T_{CAD11} + T_{CAD12} + 8T_{ISR} \quad (s) \quad (4)$$

$$T_{max}(12) = T_{CAD7} + T_{CAD8} + 3T_{CAD9} + 3T_{CAD10} + 2T_{CAD11} + T_{CAD12} + 11T_{ISR} \quad (s) \quad (5)$$

The communication parameters of the end nodes are calculated by the packet forwarder, similarly to the ADR mechanism of LoRaWAN, by defining a default SNR margin and relating the maximum SNR of the received packets to the SNR limit for each SF, as shown in Equation (6).

$$margin = \max(SNR_{packet}) - SNR_{limit}(SF) - margin_{default} \quad (dB) \quad (6)$$

4.3. LoRa Parameter Calculator

The developed calculator is a tool that aids its users (i.e., network administrators) in the selection of the values for LoRa parameters to be used in the initial configuration and deployment of the LoRa devices.

The calculator takes a set of LoRa parameters as input, namely the maximum ToA, the payload length and the coding rate (CR), as well as the estimated processing time for execution of the ASFS algorithm by the packet forwarder microcontroller and the transceiver power consumption (which are device dependent). Since the CR provides recovery from packet errors, it is eminently application-dependant. For example, in an application scenario that is loss tolerant, then CR can have the highest value (4/5), which corresponds to the minimum level of redundancy. On the other hand, in applications where it is critical to minimize the data loss, then the CR should have the minimum value (4/8).

Based on these input values, many important metrics are calculated and displayed by this tool: the energy consumption per packet, the effective ToA, the maximum link budget (difference between the transmitter output power and the sensitivity of the receiver), the maximum receiver (Rx) sensitivity (which determines the minimum strength of a signal that the packet forwarder can detect), the maximum time necessary for the ASFS algorithm to execute an iteration (used by the packet forwarder to detect incoming packets), the minimum time (Tmin) between messages (a restriction imposed by fair use of the ISM bands) and the maximum number of packets that can be sent in a day. It also calculates the Bw and the preamble length to be used in the transceiver configuration. The inputs and outputs of the calculator can be observed in Figure 4.

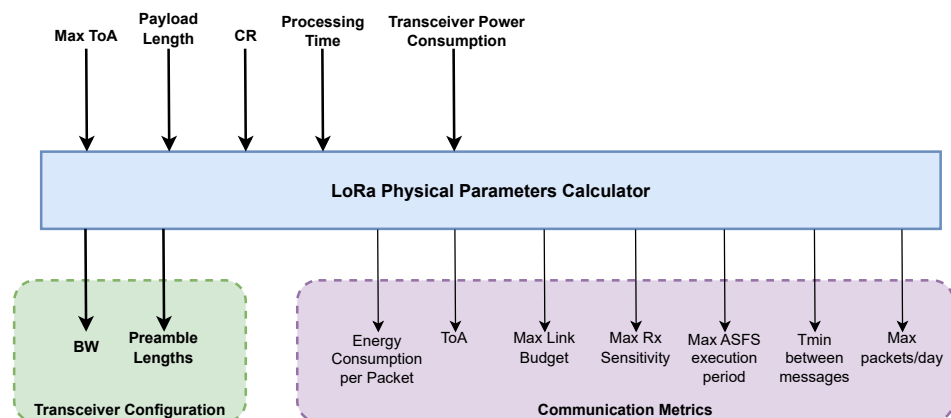


Figure 4. Inputs and outputs of the LoRa parameters calculator.

For the development of the LoRa parameter calculator, as a starting point, a LoRa air-time calculator was used [21], as it already contains many of the necessary parameters.

Then, it was adapted to receive the necessary inputs and calculate the output parameters iteratively, taking the ASFS algorithm requirements into consideration. The maximum time it takes for the ASFS algorithm to run an iteration can be defined as the cumulative CAD detection periods (T_{max}), as described in the previous section.

The calculator algorithm is depicted in Figure 5. It iterates over all possible bandwidths and returns the lowest Bw that fulfills the Max ToA requirement. To verify this, the calculator considers a maximum SF (12), the maximum TxP, and the application-specific payload and CR. Along with the Bw, the calculator outputs the calculated preamble lengths for each spreading factor and the important metrics related to the set of parameters in Figure 4.

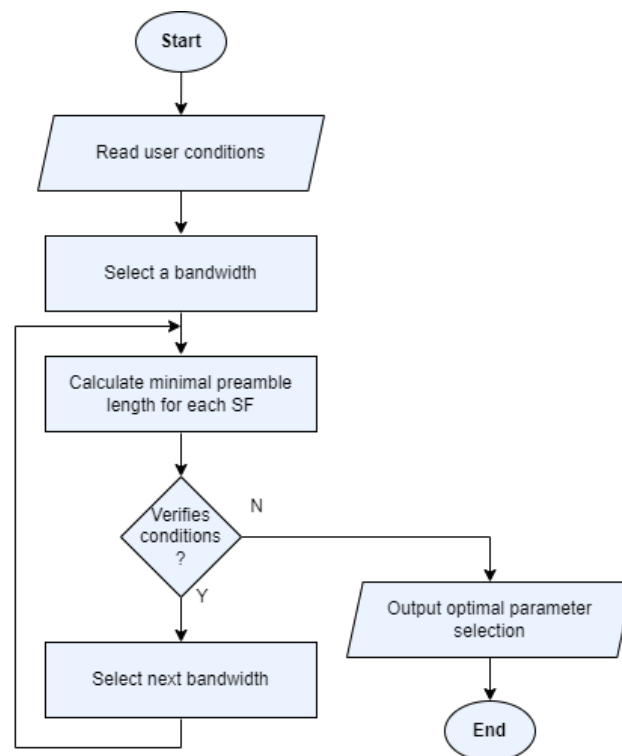


Figure 5. LoRa parameter calculator flowchart.

The user configures the node with the Bw output with the calculator, an SF, and a TxP of choice. It should be noted that the output Bw offers the highest possible range. The initial SF and TxP are not very important since they will be dynamically minimized according to the link quality by the proposed ADL mechanism. However, starting with the highest SF and TxP values ensures the node will work from the first moment.

4.4. MAC Protocol

This section describes the implemented medium access control (MAC) protocol, which controls the insertion of end nodes into the star LoRa network controlled by a given packet forwarder, defines the uplink data transmission time from each end node to the packet forwarder, and provides the transmission of downlink acknowledgments (ACKs) that allow the end nodes to adapt their SF and transmission power dynamically (according to the ADL mechanism described in the next section).

Most of the energy required by a wireless node is consumed by its MAC protocol [22], so its optimization is fundamental. This energy consumption is high when the node is in the active mode (either transmitting, receiving or idle) and very low in the sleep mode; therefore, the MAC protocol should ensure that the node stays in active mode only for short periods.

The main potential sources of energy waste in a MAC protocol are collisions, protocol overhead, and idle listening. A collision is a superposition (either total or partial) of the transmissions of different nodes. In this case, the energy is wasted because the receiver normally is unable to decode the packets, resulting in packet loss. The distributed ALOHA protocol has low overhead but is very vulnerable to collisions. Distributed contention-based MAC protocols, such as carrier sense multiple access (CSMA), are less vulnerable to collisions but at the cost of higher protocol overhead (carrier sense and random backoff periods), and they are not able to avoid all collisions.

On the other hand, the proposed centralized reservation-based MAC protocol is collision-free since each slot is reserved to a single end node. It also has low protocol overhead, since it does not perform carrier sense or backoffs. With this protocol, after the initial slot allocation, an end node only needs to wake up periodically to sample its sensors, transmit its data packet and, sporadically (as explained in the next section), receive an ACK packet, which means that it can sleep the rest of the time to save energy.

The third main source of energy waste, idle listening, occurs when a node stays active waiting to receive data packets destined for it. This may be the case, for example, when some end nodes are actuator nodes instead of sensor nodes. In this case, with conventional ALOHA or CSMA protocols, the node would need to be active all the time waiting for new data, whereas with our MAC protocol, the node would only need to be active during its slot at the superframe, sleeping the rest of the time.

As referred above, most of the energy of an end node is spent when the node is in active mode. Since the proposed MAC protocol minimizes the protocol overhead and eliminates collisions and idle listening, the active mode is used almost exclusively for the transmission of data packets during the steady-state operation of the end node, which means that its energy consumption becomes proportional to the ToA.

The proposed MAC protocol is divided into two key parts: slot allocation and data transmission. The first part starts with an allocation request from the end node, which is followed by an allocation response from the packet forwarder. After that, the node may send data periodically at the start of its allocated slot in every superframe.

The superframe period (T_s) of the MAC protocol is set by the network administrator taking into account the requirements of the application. With homogeneous end nodes, it corresponds to the data transmission interval required by the end nodes, whereas with heterogeneous nodes it can be set to the minimum data transmission interval required among all end nodes, and those with less demanding timing requirements may use their allocated slots only when necessary.

Considering that each node sends data at a predefined slot of the superframe, by limiting the number of nodes per LoRa network, a suitable scheduling can be defined. The method used to allocate slots to the end nodes is exemplified in Figure 6. This method starts by spreading the transmissions as much as possible, and new slot allocations are intercalated between the previous ones. The admission control mechanism of the MAC protocol prevents the allocation of slots to new end nodes in the LoRa network when its maximum capacity is reached, i.e., when it would not be possible for an end node to make a transmission with the maximum ToA without overlapping with a new end node's allocation.

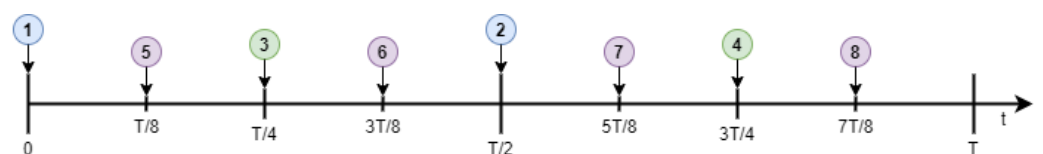


Figure 6. Timing for the configuration of end nodes.

A short end-node address (n) is attributed to each new node by the packet forwarder. The value starts from 1 and is increased at each new slot allocation request the packet forwarder receives. This address serves two purposes: it reduces the size of the data packets

in comparison to the long end node address, and it identifies the position of the allocated slot inside the superframe according to Equation (7), where T_{slot} is the start of the slot (relative to the start of the superframe) and T_s is the superframe period. The exception for this equation is the value of T_{slot} for the first node (1), which is zero (i.e., the start of the superframe). The maximum value of n (i.e., the maximum number of end nodes supported by the network) is not restricted by Equation (7) but is limited by the admission control mechanism of the MAC protocol, as explained before.

$$T_{slot}(n) = \left(\frac{n - 0.5}{2^{\lfloor \log_2(n-1) \rfloor}} - 1 \right) * T_s \quad (s) \tag{7}$$

Before the deployment of the network nodes, the application-specific parameter values obtained with the LoRa parameter calculator are configured into both the end nodes and the packet forwarders. Then, when the end nodes are turned on at their installation sites, they start by sending a slot allocation request in order to join the network.

The developed MAC protocol defines two sets of uplink and downlink packets, one for the initial slot allocation and the other for the periodic data transmission and for the ACK packet feedback. The format of these packets can be observed in Figure 7.

Slot Allocation Request (Uplink)

Packet Type (Slot Allocation Request)	Long End Node Address
---------------------------------------	-----------------------

Slot Allocation Response (Downlink)

Packet Type (Slot Allocation Response)	Network Address	Short End Node Address	Superframe Period	Superframe Synchronization Field
--	-----------------	------------------------	-------------------	----------------------------------

Data (Uplink)

Packet Type (Data)	Network Address	Short End Node Address	Data	Options
--------------------	-----------------	------------------------	------	---------

Acknowledgement (Downlink)

Packet Type (ACK)	Network Address	Short End Node Address	Resynchronization Field	Spreading Factor	Transmission Power
-------------------	-----------------	------------------------	-------------------------	------------------	--------------------

Figure 7. High-level description of the developed protocol.

The allocation request packet is composed of two fields: packet type (slot allocation request) and the long end node address, which is unique for each device. The allocation response includes the following five fields: packet type (slot allocation response), the network address (each packet forwarder has one), the short end node address (attributed to the node by the packet forwarder, as described before), the superframe period, and the superframe synchronization field.

The data packet contains the packet type (data), the network address, the short end-node address, the data (samples from its sensors), and an options field (used by the end nodes to request a response packet from the packet forwarder), whereas the ACK packet contains the packet type (ACK), the network address, the short end-node address, the superframe resynchronization field, and the updated parameter values (spreading factor and transmission power) calculated by the ADL mechanism.

The superframe synchronization field of the allocation packet contains an offset value (time elapsed since the start of the superframe). Using this offset value, the time of reception of the allocation response and the short end-node address, the end node can determine the precise time to transmit each of its data packets using Equation (7). In a similar manner, the purpose of the superframe resynchronization field of the ACK packets is to prevent the end node from losing synchronization over extended periods due to clock drift.

4.5. ADL Mechanism

The adaptive data link (ADL) mechanism is performed by both sides of each link between an end node and the packet forwarder and uses the MAC protocol described in the previous section for communication. The ADL mechanism adjusts the SF and the TxP dynamically for each end node according to the link quality conditions observed by the packet forwarder since the last ACK, namely the number of data packets lost and the stored SNR from the received packets. The feedback from the packet forwarder to the end node is inserted in the ACK packet, which is transmitted only after Np data packets are sent by the end nodes in order to reduce the protocol overhead (ToA and energy consumption) and to give time to collect a sufficient number of link quality observations. The value of Np is configurable by the network administrator.

As observed in Figure 8, where T is the end node’s data transmission interval, based on the link quality observations, the packet forwarder sends the new calculated values for the spreading factor (SF) and transmission power in the ACK packet, which allows adapting these parameters in order to reduce the energy consumption of the end nodes without compromising data reception. A timeout mechanism was also implemented at the end nodes, which increases these parameters to the maximum in case the ACK is missed, aiming to restore the link between the end node and the packet forwarder.

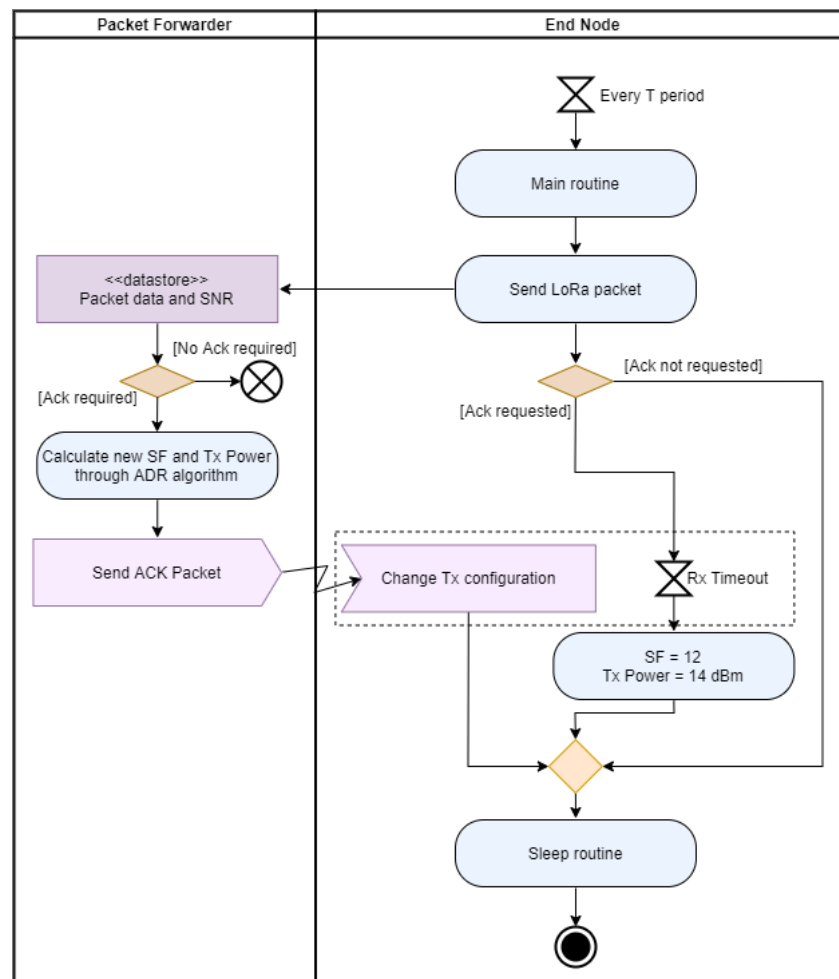


Figure 8. Activity diagram for the ADL mechanism.

Since the SF is the parameter that has more impact on the ToA (and energy consumption), the algorithm starts decreasing the SF. If SF = 7 is achieved and the link quality remains satisfactory, the TxP is also decreased. If the link quality deteriorates for any reason, the values of these parameters are increased again to restore the reliability of the

link. The Bw also has an effect on the ToA and range and, consequently, on the energy consumption and measured link quality, but less than the SF, so its adjustment was not included in the current version of the ADL mechanism.

5. Results and Discussion

For evaluation of the proposed solution in a real-world scenario, the end node functionalities were implemented in a printed circuit board (PCB) based on the STM32L0 microcontroller, with an RFM95W LoRa module and a soil moisture sensor connected to it, while the packet forwarder used the same LoRa module but was connected to a Raspberry Pi Zero W and resorted to Wi-Fi to send the received packets to the web server using the CoAP (constrained application protocol) application layer protocol. Figure 9 shows the hardware implemented for these two LoRa devices.

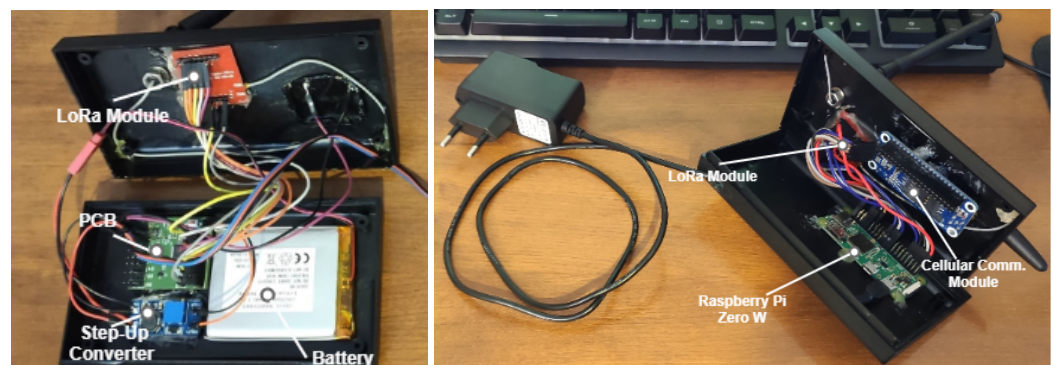


Figure 9. Hardware of the end node (on the left) and the packet forwarder (on the right).

The Raspberry Pi Zero W module is compatible with the IEEE 802.11 b/g/n wireless LAN standards. The maximum bit rate supported by the IEEE 802.11g standard is 54 Mbps, and the IEEE 802.11n allows even higher bit rates. According to Equation (3), the maximum LoRa bit rate is less than 22 kbps. That means that the Wi-Fi bit rate is several orders of magnitude higher than the necessary to forward the packets to the Web server. This is expected, since Wi-Fi is a wireless network optimized for performance, whereas LoRa is optimized for low resource requirements (such as energy consumption, since the LoRa-based end nodes are normally designed to operate with batteries). On the other hand, the packet forwarders are designed to be connected to the power grid, so they do not have the same restrictions.

Regarding the implementation of the MAC protocol, it supports up to 65,535 packet forwarders, where each network was configured to support up to 254 end nodes. Notice that this limitation is due to the use of a 8-bit short end node address and the reservation of two addresses. In terms of network scalability, the number of supported end nodes may be higher, as explained in the next paragraph. The data consists of a fixed 4-byte payload (implicit header mode) that contains soil moisture information, and the superframe period and data packet interval were set to the same value (1 h). Table 1 shows an example of the values of relevant fields of the packets presented in Figure 7 for this application scenario.

The maximum number of end nodes supported by a network in the proposed solution is application-dependent. It will depend on several parameters configured by the network administrators with the help of the LoRa parameter calculator. Each end node reserves a superframe slot in the proposed MAC protocol. According to Equation (7), the number of slots generated by the MAC protocol increases in powers of two (2, 4, 8, and so on). Therefore, as an example, with a maximum ToA of 4000 ms (4 s) and the MAC superframe period (and data packet interval) of 1 h (3600 s), which is the value adopted in the implemented prototype, the maximum number of end nodes supported by the network would be 512 (the highest power of two below 900).

Table 1. Protocol applied to a real-world scenario.

Field Name	Length (bits)	Value
Slot Allocation Request (Uplink)	4	0 × 0
Slot Allocation Response (Downlink)	4	0 × 1
Data (Uplink)	4	0 × 2
ACK (Downlink)	4	0 × 3
Long End Node Address	32	0 × 00000000–0 × FFFFFFFF
Network Address	16	0 × 0001–0 × FFFF
Short End Node Address	8	0 × 01–0 × FF
Superframe Period (s)	16	0–65,535
Superframe Synchronization Field (s)	16	0–65,535
Data (Payload)	32	Application-specific
Options	4	0–1

The network parameters were obtained by feeding the LoRa parameter calculator the maximum ToA of 1000 and 4000 ms, packet size of 8 bytes, coding rate of 4/5, and processing time needed to perform an iteration of the ASFS algorithm by the gateway of 10 μs. During each iteration, the algorithm finds the available options and selects those that fit the criteria, providing them to the user in a CSV file. The user may then decide if they want a lower link budget that provides a faster and more energy efficient option, allowing also more messages to be sent per superframe, or the opposite option, which maximizes the range at the cost of lower data rate and energy efficiency. These two options can be observed in Table 2.

Table 2. Results obtained with the LoRa parameter calculator.

Metrics	T = 1000 ms	T = 4000 ms
Time on Air (ToA)—min max (ms)	97.5 925.7	406.5 3964.9
Spreading Factor (SF)—min max	7 12	7 12
Coding Rate (CR)	4/5	4/5
Bandwidth (Bw)—kHz	125	31.25
$T_{preamble}$ —min max (ms)	68.9 335.9	275.5 1343.5
$N_{preamble}$ —min max	6 63	6 63
Sensitivity—min max (dB)	−121 −134	−127 −140
Max. Link Budget (dB)	−107 −120	−113 −126
Max. ASFS period (dB)	132.1	528.4
Min. time between messages—min max (mm:ss)	00:09 01:32	00:39 06:10
Max. packets/day—min max	933 8857	217 2125

By observing Table 2, we can also note how the proposed ADL mechanism is able to provide a more efficient use of the medium and a smaller end node’s energy expense, in comparison to approaches that do not adjust the SF dynamically and, especially those that used the same fixed SF value for all the end nodes. In many cases, the end nodes are located in close enough proximity to the gateway, enabling them to communicate with a smaller SF while sustaining a reliable link to the gateway. By comparing the ToA of the lowest SF to the highest, we can see that an adaptive approach may decrease the ToA up to around nine times, resulting in much smaller energy consumption and use of the ISM band. On the other hand, if a fixed small SF value is used, we are not taking advantage of a larger link budget to support longer communication ranges when necessary or to overcome link quality fluctuations.

In Figures 10 and 11, we can observe the relation between the sensitivity (where a lower value means more sensitive) and ToA for different spreading factors and CR fixed at 4/5. The four points of each curve/SF correspond to bandwidths of 125, 62.5, 41.7, and 31.25 kHz, respectively, from the top left to the bottom right of the graphs. The graphs

clearly show that both sensitivity and ToA increase as SF also increases, with the former increasing 2 to 3 dBm for each SF step up, while the latter increases very strongly as SF increases. Consequently, having smaller spreading factors when possible can heavily decrease ToA and, consequently, energy consumption. In case smaller values of CR are used (4/6 through 4/8), ToA increases (as is shown in Figures 12 and 13), but the sensitivity is not affected.

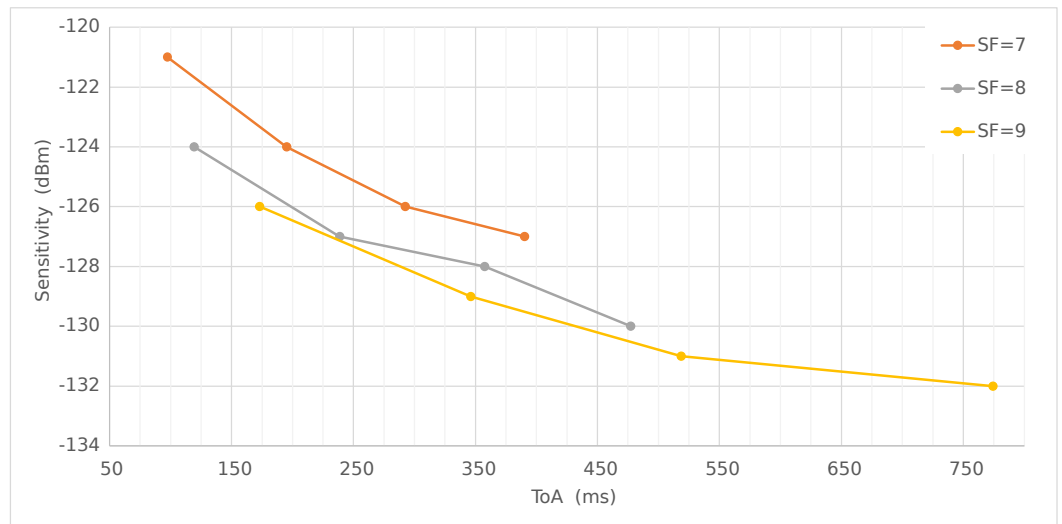


Figure 10. Relation between sensitivity and ToA for SFs 7 through 9, with CR = 4/5 and Bw ∈ {125, 62.5, 41.7 and 31.25} kHz.

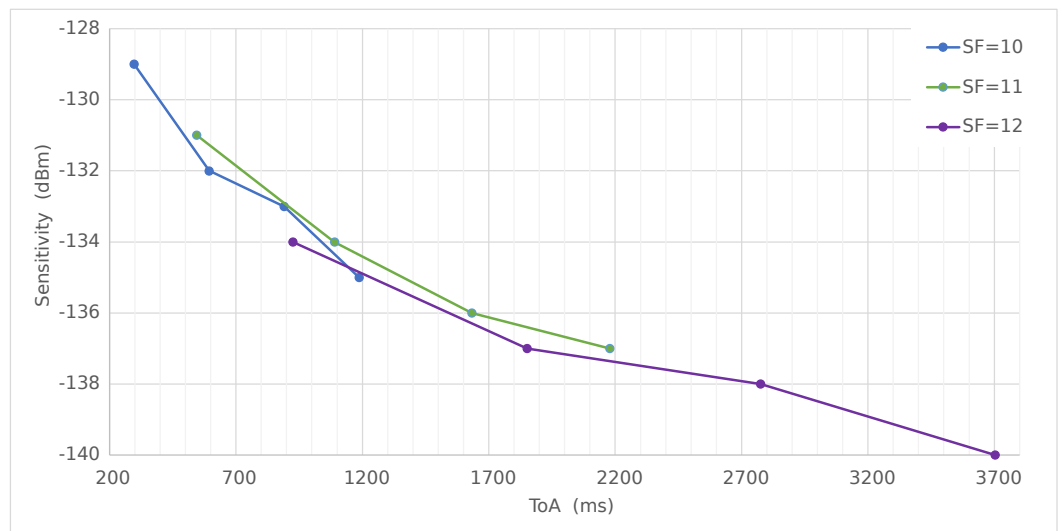


Figure 11. Relation between sensitivity and ToA for SFs 10 through 12, with CR = 4/5 and Bw ∈ {125, 62.5, 41.7 and 31.25} kHz.

Figures 12 and 13 show the relation between the coding rate and ToA for different spreading factors, where Bw is fixed at 125 kHz, where we can conclude the ToA increases proportionally with each increase in CR and that the extra error correction provided by the coding rate might be worth the extra ToA and power consumption.

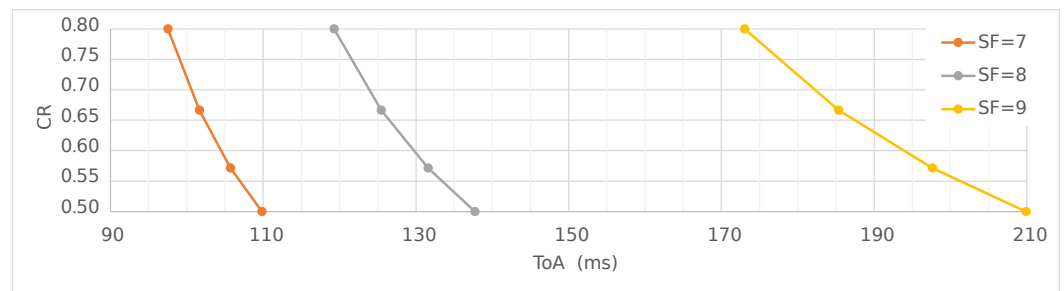


Figure 12. Relation between coding rate and ToA for SFs 7 through 9 (with Bw = 125 kHz).

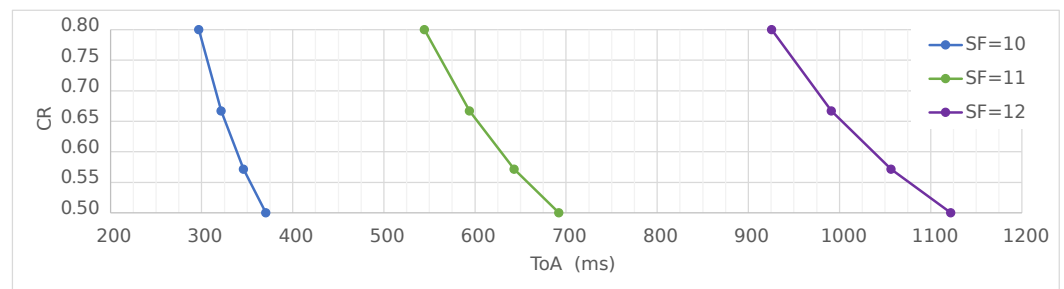


Figure 13. Relation between coding rate and ToA for SFs 10 through 12 (with Bw = 125 kHz).

6. Conclusions

In this work, the design and implementation of a data-link-layer architecture based on the LoRa physical layer was presented. The proposed solution is able to manage the spreading factor and transmission power dynamically according to the measured link quality, ensuring efficient utilization of the device and ISM band resources. Along with it, a LoRa parameter calculator was developed to aid the user to configure the physical layer parameters in accordance with the necessities of the target application.

The developed architecture poses itself as a viable and low-cost alternative to the LoRaWAN stack, where some of its important features are replicated, such as the ability to efficiently manage the network and end node resources. It also upgrades the P2P LoRa link by controlling the end nodes' link budget dynamically and individually, enhancing network robustness by adding downlink ACK packets and allocating resources for contention-free data transmission, so that network performance is not degraded with an increasing number of nodes while complying with the ISM band restrictions.

Even though the developed work is promising, some improvements can be made, such as the adaptation of the proposed architecture to a mesh network topology for increased network coverage and robustness. The integration of the packet forwarder into the LoRaWAN stack may also provide some advantages, such as easing the interoperability with already-developed applications that use this stack and extending the network coverage. The integration of security mechanisms in the data link protocol should also be a relevant step to improve its usefulness.

Author Contributions: Conceptualization, M.C., S.F.L. and J.A.A.; methodology, M.C. and S.F.L.; software, M.C.; validation, M.C.; investigation, M.C. and S.F.L.; resources, S.F.L.; writing—original draft preparation, M.C., S.F.L. and J.A.A.; writing—review and editing, M.C., S.F.L. and J.A.A.; supervision, S.F.L.; funding acquisition, S.F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by FEDER through COMPETE2020—Programa Operacional Competitividade e Internacionalização (POCI) grant number POCI-01-0145-FEDER-028247 and by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

Data Availability Statement: The data presented in this study are available within this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* **2019**, *5*, 1–7. [CrossRef]
2. Semtech. LoRa and LoRaWAN: A Technical Overview. Available online: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/> (accessed on 5 December 2022).
3. LoRa Alliance. LoRaWAN 1.0.4 Specification Package. Available online: https://lora-alliance.org/resource_hub/lorawan-104-specification-package/ (accessed on 5 December 2022).
4. Kenyeres, M.; Kenyeres, J. Average Consensus over Mobile Wireless Sensor Networks: Weight Matrix Guaranteeing Convergence without Reconfiguration of Edge Weights. *Sensors* **2020**, *20*, 3677. [CrossRef] [PubMed]
5. Adefemi Alimi, K.O.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S. A Survey on the Security of Low Power Wide Area Networks: Threats, Challenges, and Potential Solutions. *Sensors* **2020**, *20*, 5800. [CrossRef] [PubMed]
6. Ali, T.A.A.; Choksi, V.; Potdar, M.B. Precision Agriculture Monitoring System using Green Internet of Things (IoT). *Int. J. Res. Appl. Sci. Eng. Technol.* **2018**, *6*, 2961–2970. [CrossRef]
7. Co, J.; Tiausas, F.J.; Domer, P.A.; Guico, M.L.; Monje, J.C.; Oppus, C. Design of a Long-Short Range Soil Monitoring Wireless Sensor Network for Medium-Scale Deployment. In Proceedings of the TENCN 2018—IEEE Region 10 Annual International Conference, Jeju, Republic of Korea, 28–31 October 2018; pp. 1371–1376.
8. Nico, N.G. Development of Low-cost LoRaWAN Gateway for Private Deployments. Master’s Thesis, Instituto Superior Técnico, Lisbon, Portugal, November 2017.
9. Xia, X.; Hou, N.; Zheng, Y.; Gu, T. PCube: Scaling LoRa Concurrent Transmissions with Reception Diversities. *ACM Trans. Sens. Netw.* **2022**, *18*, 66. [CrossRef]
10. Wang, Y.; Zhang, F.; Zheng, X.; Liu, L.; Ma, H. Decoding LoRa Collisions via Parallel Alignment. *ACM Trans. Sens. Netw.* **2023**, *19*, 62. [CrossRef]
11. Loubany, A.; Lahoud, S.; Chall, R. Adaptive algorithm for spreading factor selection in LoRaWAN networks with multiple gateways. *Comput. Netw.* **2020**, *182*, 107491. [CrossRef]
12. Finnegan, J.; Farrell, R.; Brown, S. Analysis and Enhancement of the LoRaWAN Adaptive Data Rate Scheme. *IEEE Internet Things* **2020**, *7*, 7171–7180. [CrossRef]
13. Islam, A.; Akter, K.; Nipu, N.J.; Das, A.; Rahman, M.M.; Rahman, M. IoT Based Power Efficient Agro Field Monitoring and Irrigation Control System: An Empirical Implementation in Precision Agriculture. In Proceedings of the 2018 International Conference on Innovations in Science, Engineering and Technology (ICISSET), Chittagong, Bangladesh, 27–28 October 2018; pp. 372–377.
14. Gloria, A.; Dionisio, C.; Simoes, G.; Sebastiao, P.; Souto, N. WSN Application for Sustainable Water Management in Irrigation Systems. In Proceedings of the IEEE 5th World Forum on Internet of Things (WF-IoT’19), Limerick, Ireland, 15–18 April 2019.
15. Kim, S.; Lee, H.; Jeon, S. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. *Sensors* **2020**, *20*, 1008. [CrossRef] [PubMed]
16. Rahman, M.; Saifullah, A. Boosting Reliability and Energy-Efficiency in Indoor LoRa. In Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation (IoTDI ’23), San Antonio TX, USA, 9–12 May 2023; pp. 396–409.
17. Marjasz, R.; Polys, K.; Strzoda, A.; Grochla, K. Improving Delivery Ratio in LoRa Network. In Proceedings of the 19th ACM International Symposium on Mobility Management and Wireless Access (MobiWac ’21), Alicante, Spain, 22–26 November 2021; pp. 141–146.
18. Gamage, A.; Liando, J.; Gu, C.; Tan, R.; Li, M.; Seller, O. LMAC: Efficient Carrier-Sense Multiple Access for LoRa. *ACM Trans. Sens. Netw.* **2023**, *19*, 44. [CrossRef]
19. Semtech. LoRa Modulation Basics, AN1200.22. Available online: <https://web.archive.org/web/20190718200516/https://www.semtech.com/uploads/documents/an1200.22.pdf> (accessed on 10 March 2020).
20. Semtech. Reading Channel RSSI during a CAD, AN1200.21. Available online: <https://studylib.net/doc/18373650/an1200.21-reading-channel-rssi-during-a-cad> (accessed on 24 February 2020).
21. GitHub Repository. LoRa Air-Time Calculator. Available online: <https://github.com/ifTNT/lora-air-time> (accessed on 1 June 2020).
22. Karl, H.; Willig, A. *Protocols and Architectures for Wireless Sensor Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2007.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.