electronics

Article

Dynamic Selection and Detection of Spreading Factors and Channels for End-Node Devices of LoRa Networks

Carles Aliagas [1], Roger Pueyo Centelles [2], Roc Meseguer [2,*,†], Pere Millán [1] and Carlos Molina [1]

[1] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, 43007 Tarragona, Spain; carles.aliagas@urv.cat (C.A.); pere.millan@urv.cat (P.M.); carlos.molina@urv.cat (C.M.)
[2] Department of Computer Architecture, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain; roger.pueyo@upc.edu
* Correspondence: roc.meseguer@upc.edu; Tel.: +34-93-4137102
† Current address: Computer Architecture Department, Universitat Politècnica de Catalunya—BarcelonaTech (UPC), Edifici C4, Despatx 005P. C. Esteve Terradas, 7, 08860 Castelldefels, Spain.

Abstract

LoRa and LoRaWAN facilitate effective long-range communication for IoT, with LoRa concentrating on transmission efficiency and low-power usage, while LoRaWAN addresses network architecture. Custom LoRa networks are ideal for small-scale applications due to their control and cost benefits. Nevertheless, large-scale deployments can experience message collisions, impacting efficiency and latency. LoRaWAN addresses this with Adaptive Data Rate (ADR), enhancing capacity and power efficiency. Our research introduced a novel strategy to improve LoRa network efficiency through a decentralized method. We employed multiple channels and spreading factors on client chips. This minimized contention and collisions. This approach allowed for dynamic adjustments, ensuring comprehensive communication control and enhanced performance in diverse environments. Our two-step mechanism, integrating heuristics and selection policies, provided flexible communication. We optimized parameters such as message size, transmission power, and bandwidth. This enhanced data rate, RSSI, and SNR, and reduced energy consumption. These results underscore the relevance of precise parameter tuning in achieving optimal LoRa performance.

Keywords: LoRa; dynamic selection; spreading factors; channel optimization; end-node devices; IoT networks; communication efficiency; network performance; wireless communication

check for updates

Academic Editor: Pingyi Fan

Received: 8 July 2025
Revised: 15 August 2025
Accepted: 17 August 2025
Published: 22 August 2025

Citation: Aliagas, C.; Pueyo Centelles, R.; Meseguer, R.; Millán, P.; Molina, C. Dynamic Selection and Detection of Spreading Factors and Channels for End-Node Devices of LoRa Networks. *Electronics* **2025**, *14*, 3341. https://doi.org/10.3390/electronics14173341

1. Introduction

LoRa, which stands for Long Range, is a wireless transmission technology developed in France at the end of 2010 and patented by the company Semtech. This technology allows communication over long distances and with low power consumption. It is especially designed for environments where there is little or no communication coverage and where access to electrical power is not always available. LoRa uses ISM (Industrial Scientific and Medical) radio bands, which, although limited in power and transmission time, are free to use. LoRaWAN is a protocol that operates on top of the physical layer of LoRa, defining network architecture, communication protocols, and security standards for LoRa-based LPWANs. It enables bidirectional communication between LoRa devices, gateways, and the Internet as maintained by the LoRa Alliance. Within the OSI model, LoRa operates in the physical layer (Layer 1), providing long-range communication at low power, while

LoRaWAN, operating at the link and network layers (Layers 2 and 3), defines the network architecture and communication protocols. Together, they offer an efficient solution for IoT applications.

Implementing a LoRaWAN network can be costly and may not be suitable for small-scale applications. For applications that need more control over network architecture and data management, a custom LoRa network might be more appropriate. In cases where simple point-to-point communication or existing LoRa-based infrastructure is sufficient, a custom LoRa network is preferable over migrating to LoRaWAN. This approach allows the use of LoRa technology without LoRaWAN. It suits applications requiring long-range communication but not the complexities of LoRaWAN. This network, without gateways, allows end nodes to gather information and make collective decisions, contributing to a global solution [1]. The provided text describes various types of sensor network and their applications.

- Traffic Sensor Network: Sensors in a city collect traffic data to adjust traffic light timings, reducing congestion and improving flow.
- Air Quality Sensor Network: Distributed sensors in urban areas measure real-time air quality, providing data to assess pollution and improve air quality.
- Agricultural Sensor Network: Farm field sensors gather data on soil moisture, temperature, and humidity to optimize irrigation and fertilization for better crop growth.
- Industrial Sensor Network: Factory sensors collect machine performance data for maintenance and production efficiency decisions.
- Smart Building Sensor Network: Sensors in smart buildings adjust lighting, heating, and cooling based on environmental conditions and occupancy. This maximizes energy efficiency and comfort.
- Health Sensor Network: Medical devices and patient sensors in healthcare settings collect vital information, aiding healthcare professionals in informed decision-making.
- Environmental Sensor Network: Sensors in remote areas monitor environmental variables like temperature and water quality for ecosystem studies and conservation decisions.

LoRa is a wireless communication technology noted for its long range and low energy consumption, despite having limited bandwidth. It uses low-power modulation to transmit signals over long distances, enabling devices to operate with long-lasting batteries. However, in high-density node deployments, the nodes compete for the transmission channel, causing message collisions that degrade network quality, reduce efficiency, and increase latency.

In LoRa and LoRaWAN networks, end devices (or nodes) transmit data at a specific data rate that determines how long it takes to send a message. This data rate is typically determined by various factors. These include the distance between the node and the gateway, the radio environment, and any interference present in the area. If the data rate is too low, the node takes too long to transmit its data. This delay causes other nodes to wait longer before transmitting their own data. Moreover, transmitting data at a low data rate takes more time than transmitting data at a high data rate. This is because the data are spread over a wider frequency band, resulting in a longer transmission time for low data rates. As a result, low data rate transmissions lead to increased power consumption and decreased battery life.

Adaptive Data Rate (ADR) is a mechanism that optimizes communication between nodes and gateways in LoRaWAN. It works by continuously adjusting data rate parameters such as spread factor, bandwidth, and transmission power. This ensures optimal data rates, maximizes network capacity, and minimizes power consumption. ADR is managed by the network server, which monitors the quality of the signal and adjusts the data rates accordingly, allowing efficient network operation under different conditions.

LoRa modulation does not support ADR (Adaptive Data Rate) similarly to cellular or Wi-Fi protocols. This is due to its non-linear, chirp-based modulation, where the data rate and signal bandwidth are interdependent. As the data rate increases, the chirp signals compress, shortening the duration of the signal. To preserve signal quality, the frequency of the chirp signal must increase, which broadens the signal bandwidth. Therefore, the LoRa modulation scheme does not allow dynamic changes in the data rate without also changing the signal bandwidth. As a result, LoRa networks cannot use the ADR mechanism in the same way as other wireless communication protocols.

LoRa has several configurable parameters. The main ones are bandwidth (BW), spreading factor (SF), coding rate (CR), carrier frequency (CF), and transmission power (TP). Furthermore, in a LoRa network, a channel is a specific frequency band within the overall BW of the system that is used for communication between devices. The CF is the center frequency of the channel and the BW is the range of frequencies around the CF that are used to transmit data. Even without the LoRaWAN protocol, LoRa networks still use low-level channels for communication between end nodes. The use of channels allows multiple nodes to transmit data simultaneously without interfering with each other, increasing the overall network capacity and reliability.

The selection of parameters such as BW, SF, CR, CF, and TP significantly influences both the data rate and the communication range. Lower BW increases the range but decreases the data rate, while higher BW does the opposite. Higher SF and CR improve reliability in interference-prone environments, but reduce data transmission rates. In contrast, lower SF and SR raise transfer rates but may cause packet loss. CF affects the frequency range of the chirp signal, with higher frequencies enhancing the data rates but reducing the range. TP governs both rate and range, where higher power increases both, but at the cost of higher energy consumption and potential interference.

In LoRa networks without LoRaWAN, there is greater flexibility and control over the network architecture by adjusting the SF and channel using various techniques. SF can be set using fixed, user-defined, or hybrid methods, while the channel can be adjusted through frequency hopping, fixed frequency, adaptive selection, or user-defined plans. These adjustments are made in the firmware and tailored to application needs, balancing factors such as data rate, range, and reliability. Unlike LoRaWAN, where gateways can dynamically adapt SF and channel, non-LoRaWAN networks require node-specific configurations from the start, without dynamic adjustments.

This work focuses on improving the efficiency of LoRa network communications using a decentralized approach within the LoRa layer. Using multiple channels and SFs simultaneously on client chips, the proposed method reduces node competition, increases bandwidth, and minimizes collisions, especially in networks with many nodes. Unlike existing methods that mainly adjust SFs, this approach also modifies channels, managing orthogonal channels and SFs to ensure smooth operations, dynamically selecting optimal SFs and channels for reliable communication in various environments. The method includes a two-step mechanism that involves heuristic value generation and policy-based selection, offering flexibility and adaptability to changing network conditions. Furthermore, a feasibility analysis of LoRa technology is provided in different scenarios and parameter configurations, marking a significant improvement over current methodologies.

The paper is organized as follows. Section 2 conducts a comprehensive review of current literature in the field of long-range communication technologies. Section 3 explores parameterization and measurements within the specific context of LoRa. In Section 4, the focus shifts to a critical analysis of LoRa parameterization, examining key factors that significantly influence its operational performance. Section 5 introduces a novel proposal to optimize LoRa parameterization, detailing the approach and theoretical foundations.

Section 6 validates the proposal through detailed simulations and analysis of the results. Finally, in Section 7, conclusions derived from the research are presented, highlighting key contributions and suggesting possible future directions.

2. Related Work

This section discusses existing work related to LoRa optimization strategies, with particular attention to spreading factor and channel selection.

Low-Power Wide-Area Networks (LPWANs) are wireless systems that enable long-distance data transmission and allow devices to be connected with a low-energy consumption approach. They appear as a valid transport protocol option for certain Internet of Things (IoT) needs. Examples of LPWAN technologies are LoRa, Sigfox, Weightless, and Ingenu (especially for IoT communications [2]). In particular, the LoRaWAN protocol has been widely demonstrated to be effective for transmitting and receiving quick data [3–8]. It works well with battery-operated devices in scenarios with low-power requirements.

For example, Pietrosemoli et al. [9] outline the design of a simple and affordable device to obtain weather data from commercial weather stations with limited wireless range and retransmit them to cover long distances using LoRaWAN. Dayana et al. [10] integrate a LoRaWAN module inside an electrical meter to send user consumption data. Their system achieves a wide transmission range, lower energy use, and negligible data loss. Finally, Nakamura et al. [11] propose a flexible architecture for scenarios with limited resources located in remote rural areas. The main idea of their proposal is to adopt LoRa technology to enable long-range transmissions with low power consumption from sensor nodes to a central node. This central node then transfers the data to the server via the Internet.

LoRa networks face significant scalability challenges as the number of end devices increases. Simulation models based on interference measurements show that with 1000 nodes per gateway, packet losses can reach up to 32% [12]. The scalability issue is primarily due to signal interference and concurrent transmission collisions in dense networks [13]. Studies using ns-3 simulations reveal that downstream traffic negatively impacts the delivery ratio of confirmed upstream traffic. Increasing gateway density only partially mitigates this effect due to stringent duty cycle requirements [14]. Stochastic geometry analysis demonstrates that coverage probability decreases exponentially as the number of end devices grows, mainly due to interfering signals using the same spreading sequence [15]. These findings highlight the need for innovative solutions to enhance LoRaWAN scalability, such as improved channel assignment strategies and new network topologies [13].

Some other papers explore decentralized approaches to enhance LoRa network performance without relying on traditional gateways. Haif et al. [16] propose a dynamic relaying framework using genetic algorithms for optimal relay placement and an equal-area-based SF allocation scheme, improving coverage and throughput. Liao et al. [17] focus on multi-gateway systems, dynamically selecting gateways and allocating SFs based on packet error rates. Alahmadi et al. [18] introduce an autonomous time-slotted protocol allowing nodes to determine optimal transmission parameters using geographical coordinates. Zhu et al. [19] present a tree-based SF clustering algorithm for mesh LoRa networks, enabling parallel transmissions across subnets with different SFs. While all approaches involve SF adjustments, they differ in their specific mechanisms: relay-based (Haif et al. [16]), multi-gateway (Liao et al. [17]), time-slotted (Alahmadi et al. [18]), and network clustering (Zhu et al. [19]). Unlike the above approaches, which primarily focus on spreading factor (SF) allocation or scheduling mechanisms, our proposal introduces a dual-parameter optimization strategy that simultaneously considers both SF and channel (CH) selection and detection.

Several studies on the performance potential of LoRa and LoRaWAN have been conducted. They evaluate their ability to provide low-power, long-range communication for IoT applications. These studies typically involve experimental measurements of signal strength, data rate, and network coverage, under different deployment scenarios and environmental conditions. The results of these studies provide valuable insight into the limitations and benefits of LoRa and LoRaWAN technologies, as well as recommendations for optimizing their performance in specific applications. For example, Gkotsiopoulos et al. [20] provide an overview of LoRa radio technology and its popularity in the IoT industry through the analysis of many research studies on LoRa and LoRaWAN networks. In this way, they categorize the various factors that affect the capacity metric into five categories and discuss the relevant research for each factor, highlighting its impact on network efficiency. In particular, they show how factors such as modulation techniques, signal BW, and SF influence the network's ability to transmit data efficiently. This applies to large numbers of end devices while meeting application requirements. In addition, Bor et al. [21] present a framework for evaluating the performance of different LoRa transmission parameters, such as TP, CF, SF, BW, and CR. All available parameter combinations are tested to evaluate the impact of LoRa transmission parameter selection on communication performance. Based on this study, they propose a mechanism that can be added to LoRaWAN to select the optimal transmission parameters based on the trade-off between energy consumption and network throughput. Furthermore, Li et al. [22] propose a two-dimensional taxonomy for LoRa networking (four network layers: PHY, Link, MAC, App; and four performance metrics: range, throughput, energy, security). They compare and classify different LoRa techniques from the literature and identify open issues and remaining challenges, followed by future trends. Finally, Dix-Matthews et al. [23] analyze the more than 6000 combinations of parameters, considering the metrics of path loss and energy consumption. The number of combinations of available parameters was reduced to 47 by eliminating the points that theoretically provide no advantage but at a higher cost. The Pareto front are the only parameter combinations that should be considered for applications that need to minimize energy. These 47 parameter combinations perform better than those used by ADR in LoRaWAN. They also apply payload replication and data-aware compression to improve data delivery ratios without using extra energy.

In order to improve LoRa performance, Pirayesh et al. [24] present MaLoRaGW (Multi-antenna LoRa GateWay) that enables MU-MIMO LoRa communications both uplink and downlink, operating at the PHY layer and compatible with current LoRa client devices. A two-antenna prototype achieves throughput increases of 10% uplink and 95% downlink.

Other previous studies propose methods for the dynamic selection of the LoRa SF. For example, Kim et al. [25] propose the Adaptive SF Selection (ASFS) scheme. ASFS enables every device to achieve a multi-data rate and allows a transmitter and a receiver to synchronize their SFs without any packet exchanges. The ASFS performance is evaluated on both single-hop and a multi-hop network configurations. The ASFS scheme outperforms other schemes in terms of end-to-end throughput. Likewise, Zorbas et al. [26] model a LoRa network with variable SF/BW combinations for the nodes. They formulate and solve an optimization problem to maximize the capacity of the node. The numerical results show that the number of LoRa users using close to optimal settings (only SF 7 and 8) can be increased by more than 700% compared to using equal node populations per SF. The improvement is up to 16% compared to using only nodes with the minimum possible packet air time. Similarly, Reynders et al. [27] present a new scheme that optimizes the Packet Error Rate (PER) in LoRaWAN cells. Their algorithm optimizes power and SF for each node and avoids near–far problems using different channels for far nodes. The simulation results show decreases in PER up to 50% in a moderate contention scenario. Finally, Shayo et al. [28] propose MFMSF TDMA, a dynamic multi-frame multi-SF scheduling for LoRaWAN with slotted synchronization. The proposed algorithm dynamically assigns time slots to end devices on a first-to-synchronize-first-to-be-assigned basis, reducing collisions during data transmission and improving the energy efficiency of battery-powered devices. The simulation results show higher packet delivery, less energy consumption, and better escalability for MFMSF over FREE.

Other authors have focused their research on LoRa preamble detection. For example, Kang [29] presents a new approach to optimizing detection thresholds for LoRa systems, resulting in improved performance and effectiveness. The proposed scheme maximizes the preamble detection probability while satisfying a constraint on false alarm rate and includes both coherent and non-coherent procedures for preamble detection. The simulation results demonstrate the superiority of the proposed scheme over comparable baseline schemes. Furthermore, Tian et al. [30] present RSSF, a method that analyzes RSS samples of the preambles to identify the SF used for transmission. This allows an off-the-shelf LoRa device to receive and decode a packet without prior knowledge of the SF used by the transmitter. SF is determined by a lightweight algorithm that partitions the RSS samples into sliding windows of different size (one for each possible SF) and measuring the zero-crossing intervals for each window size. Experimental results show that this method is able to accurately identify the SF used by examining at most five preamble chirps.

Articles have also been published that analyze the parameters and performance obtained, but at the LoRaWAN level. For example, Sanchez et al. [31] evaluated several LoRaWAN PHY layer configurations in three environments with varying levels of adversity against wireless transmissions: urban (high adversity), suburban (medium adversity), and rural (low adversity). The results show that there is a trade-off between link robustness and data rate, and the LoRaWAN configuration parameters (SF, CR, and BW) should be tuned accordingly. Likewise, Sisini et al. [32] propose a methodology to measure the real-world performance of LoRaWAN/SCLC solutions, which complements existing simulation-based studies. SCHC is a protocol developed for efficient transmission of IPv6 and UDP packets in low-power IoT networks. The proposed methodology was found to be effective in providing information to the user or administrator of IPv6-based IoT applications over LoRaWAN, allowing them to optimize backend configuration parameters to meet their needs. In conclusion, Fragkopoulos et al. [33] present another experimental study that assesses the performance of LoRaWAN technology in suburban and rural areas, identifying crucial factors such as distance, terrain, interference, and number of devices. The results show better performance in rural areas due to lower interference and clear line of sight, while suburban areas are affected by terrain, interference, and number of devices.

Adaptive Data Rate (ADR) [34] is a feature in LoRaWAN that automatically adjusts data rate and transmit power based on signal quality to optimize network performance and energy consumption, reducing data rate for devices far from the gateway or in areas with high interference to save battery and extend the network range. This technique was first proposed by the LoRa Alliance as part of the LoRaWAN 1.0 specification, which was released in June 2015. Since then, ADR has become a standard feature of LoRaWAN and is widely used in various IoT applications to optimize network performance and energy efficiency. Numerous studies have attempted to improve LoRaWAN ADR control using various approaches.

Several authors have proposed enhancements to LoRaWAN's Adaptive Data Rate (ADR) mechanism. These aim to better handle congestion, mobility, or energy optimization [35–41]. However, these methods still depend on centralized coordination, contrasting with our fully decentralized architecture.

In addition to the Adaptive Data Rate (ADR) mechanism or its variants, other alternatives exist. These aim to control data rate and power consumption in LoRaWAN.

Some authors have investigated the use of Machine Learning (ML) methods. For example, Minhaj et al. [42] benefit from ML techniques to present two distinct learning approaches to assign SF and TP to devices through a combination of centralized and decentralized methods. Specifically, SF allocation is achieved through a contextual bandit problem using Reinforcement Learning (RL), while TP is centrally assigned via a supervised ML approach. Similarly, Chen et al. [43] propose another ML-based approach to control the data rate in LoRaWAN. They introduce an RL-based method that uses probability and score tables to dynamically evaluate and adjust transmission parameters. This approach enables a decentralized dynamic SF allocation process that can be updated in real time during transmission, allowing the LoRaWAN network to adapt to varying conditions and optimize its data rate.

Several recent works have explored intelligent and learning-based strategies to improve the assignment of spreading factors (SF) in LoRa networks. For example, Hong et al. [44] propose a centralized reinforcement learning (RL) algorithm that uses temporal-difference methods to dynamically adjust the SF of end devices in order to maximize network throughput and fairness. Their method relies on global performance metrics such as packet delivery ratio and requires centralized coordination—conditions that are not met in infrastructure-less networks. In a different line, Scarvaglieri et al. [45] present a distributed multi-agent Q-learning framework, where each node maintains local Q-tables and updates its transmission parameters (including SF and transmission power) based on shared reward signals and observed contention. While this approach is more decentralized, it still involves non-trivial memory and communication overhead and requires exploration over time to converge to efficient configurations.

Separately, the LoRaWAN standard includes an Adaptive Data Rate (ADR) mechanism that dynamically adjusts SF and transmission power based on link quality indicators (e.g., SNR, RSSI) reported to the network server. ADR is effective in managed, static deployments with a stable gateway infrastructure. However, its centralized nature and reliance on network feedback make it unsuitable for gateway-free scenarios or mobile and ad hoc LoRa deployments.

In contrast to these approaches, our work proposes a fully decentralized framework that can operate autonomously in the absence of any infrastructure or coordination among nodes. The heuristics presented in this paper are intentionally simple, reactive, and stateless, ensuring compatibility with resource-constrained devices and eliminating the need for convergence, memory storage, or explicit communication. Although our goal is not to optimize for global performance, the selected strategies validate the feasibility and responsiveness of the mechanism in realistic embedded environments. More importantly, the proposed architecture is modular and extensible: advanced heuristics—including energy-aware, history-based, or learning-driven approaches—can be incorporated in future iterations without modifying the core detection and selection mechanism.

Other authors have also proposed alternatives to ADR, but not based on ML. First, Ochoa et al. [46] propose an algorithm to assign a SF to end devices in realistic multi-gateway deployments. The assignment is based on metrics such as link PDR (Packet Delivery Ratio), network PDR, and network distribution of SF per gateway. The algorithm improves network performance and almost doubles PDR compared to LoRaWAN ADR, thus enhancing LoRa deployments by adapting communication parameters of end devices according to network size and estimated metrics. Furthermore, Ivoghlian et al. [47] also introduce a new algorithm for LoRaWAN networks that adapts to individual device requirements and automatically selects the best network parameters based on real-time channel conditions. The algorithm is semi-decentralized and minimizes communication overhead, making it more efficient than a centralized approach. The proposed approach outperforms the current ADR approach used in LoRaWAN and takes into account scenarios where there are multiple applications with different communication requirements in the same network.

There are other approaches that try to improve the performance of LoRaWAN in specific scenarios. For example, Marais et al. [48] propose an adaptive congestion scheme (ACS) to improve the sustainability of confirmed traffic in LoRaWANs. ACS monitors congestion caused by DL traffic and activates the groupedPackets algorithm to reduce it, resulting in improved successful delivery of both unconfirmed and confirmed traffic. The groupedPackets algorithm consists of requesting nodes to aggregate their application packets to reduce the amount of confirmed traffic. In addition, Leonardi et al. [49] present MRT-LoRa, a multi-hop real-time (RT) communication protocol over LoRa for Industrial IoT applications. MRT-LoRa provides bounded delays to the RT flows typical of industrial applications. It enables long-range communications while maintaining shorter time on air at each hop. This lowers the impact on the duty cycle of each node. This multi-hop protocol is better than a single-hop approach for industrial environments. These environments typically cover large areas and require bit rate values of a few thousand bits per second for LoRa-based networks. In addition, Manzoni et al. [50] present a mesh network over LoRa that allows node-to-node multi-hop communications with self-configuration, multi-hop routing protocol, and an adaptive SF mechanism. Several experiments, both outdoor and indoor, demonstrate the feasibility and limitations of their proposal. Furthermore, Nowak et al. [51] propose two algorithms to reduce energy consumption in LoRaWAN communications. Their first algorithm predicts next data values that are not sent. Their second algorithm adjusts antenna gain power according to RSSI. Combining the two algorithms, they achieve up to 5% reduction in energy consumption. On the other hand, Adelantado et al. [52] identify the limitations of LoRaWAN pseudo-random channel hopping method. New channel hopping methods should be able to reserve a set of channels for retransmissions of critical packets (both uplink and downlink). Likewise, Singh et al. [53] present a novel time synchronization scheme to reduce energy consumption and packet collisions in LoRaWAN. They also introduce a new Cryptographic Channel Hopping scheme to increase communications security and use of the full frequency spectrum. Moreover, Figueiredo et al. [54] present a fair channel hopping scheme that uses multiple low-cost single-channel gateways. End devices use the most appropriate channel to transmit at a higher rate and spending less energy. Simulation results show 8x to 9x PDR improvement over the pure-ALOHA approach used in LoRaWAN. Similarly, Leenders et al. [55] propose, implement, and evaluate a new LoRa Multi-Hop protocol that uses prolonged preamble sampling and forwarded data aggregation to improve energy efficiency in remote sensing scenarios. Their open-source proposal achieves up to 61% energy savings whilst keeping high realibility levels (90% of nodes achieve at least 70% PDR values). To conclude, Prade et al. [56] introduce Multi-LoRa, a multi-radio and multi-hop LoRa communication architecture, to enhance the coverage and service for large-scale IoT deployment in rural areas. Area coverage is increased thanks to multi-hop communications. The authors increase throughput and decrease collisions by multiplexing data in multiple LoRa radios (channels). This way, Multi-LoRa reduces the delay by 60% and packet loss by 2.9%.

Sadly, ADR, its variants, and new algorithmic approaches cannot be used with LoRa because they lack the necessary network infrastructure. Instead, ADR requires manual adjustment of transmission power and data rate settings for each end device to optimize performance.

3. LORA Parameterization and Measurements

In this section, we describe the key configurable parameters of LoRa and how they impact communication performance, laying the groundwork for the heuristic-based strategy proposed later.

3.1. LoRa Parameters

LoRa deals with Chirp Spread Spectrum (CSS) modulation, which encodes data using a chirp—essentially a broadband, frequency-modulated sinusoidal signal that varies over time. This technology has five configurable parameters that determine the data rate and communication range: (1) bandwidth, (2) spreading factor, (3) coding rate, (4) carrier frequency, and (5) transmission power. All of these parameters define the sensitivity of the LoRa receivers. The higher the sensitivity, the greater the data range. They also influence the transmission speed of the LoRa transmitters, i.e., the data rate.

Transmission Bandwidth (TBW): Defines the frequency range of chirp signals, influencing the amplitude of transmission and the frequency range for baseband data. Practically, bandwidths of 125 kHz, 250 kHz, and 500 kHz are used. Lower bandwidths (e.g., 125 kHz) improve communication range and signal-to-noise ratio but reduce data transfer rates. Higher bandwidths (e.g., 500 kHz) increase data transfer rates but shorten communication range and reduce signal-to-noise ratio due to higher susceptibility to noise and interference.

Spreading factor (SF): In LoRa systems, the SF determines the number of chirps per symbol, which in turn affects the symbol duration and data rate. Lower SF values result in a higher data rate, but they reduce communication range and increase susceptibility to noise. In contrast, higher SF values offer longer range and better sensitivity to weak signals, but reduce data rate. LoRa supports SF values from SF7 to SF12, balancing this trade-off in data rate, signal-to-noise ratio, sensitivity, time on air, and battery life.

Carrier frequency (CF): Determines the central frequency of the transmitted signal in LoRa communication systems, typically ranging from 433 MHz to 915 MHz depending on the region. In the U.S., it is 902 MHz to 928 MHz, and in Europe, it is 863 MHz to 870 MHz. Lower carrier frequencies are better for long-range communication due to better obstacle penetration and less attenuation, while higher frequencies are suited for short-range communication and support higher data rates. Higher frequencies can accommodate wider bandwidths, allowing for faster data transmission.

Transmission Power (TP): In LoRa communication systems, it measures the power used to transmit signals and is usually expressed in decibels per milliwatt (dBm), ranging from -30 dBm to +30 dBm. TP impacts both the communication range and the data rate. Higher TP levels can extend the communication range and support higher data rates but consume more energy and increase interference risks. Hence, selecting the appropriate TP involves balancing communication range, energy efficiency, data rate, and data reliability.

Coding Rate (CR): This parameter determines the amount of error correction added to the payload data, with values of 4:4, 4:6, 4:7, and 4:8. A higher coding rate improves communication reliability by increasing error correction, but reduces the transmission rate. Conversely, a lower coding rate increases the transmission rate but may result in occasional packet loss due to reduced error correction.

Preamble (PRE): This parameter has the purpose of synchronizing the receivers and enabling them to adjust their settings to receive the main message. The preamble length in LoRa ranges from 6 to 65,535 symbols and is influenced by factors like transmission distance, data rate, and environmental conditions. Shorter preambles are used for high-speed configurations, enabling faster data transmission, while longer preambles are suitable for long-distance transmissions at lower data rates.

The choice of these parameters affects the sensitivity of the receiver, which is the lowest power level at which the receiver can receive or demodulate the signal. Higher SF and CR values result in a higher level of error correction, which can improve the receiver's sensitivity by reducing the impact of noise and interference on the received signal. However, this comes at the cost of lower data rates. Similarly, a lower bandwidth can also improve the sensitivity of the receiver, as it reduces the impact of noise and interference in a narrower frequency range. However, increasing the carrier frequency or reducing the transmission power can reduce the sensitivity of the receiver, as it increases the impact of noise and interference on the received signal.

3.2. LoRa Channels

LoRa communication uses multiple channels that occupy specific frequency ranges, with device capabilities ranging typically from 8 to 16 channels or more. Each channel operates at a central carrier frequency (in MHz) and a specified bandwidth (in kHz), which determines data transmission capacity. Channels can be configured differently to handle various data types or sources, allowing simultaneous data transmission and increasing network capacity and efficiency. The channel parameters are set by the network operator or user. Common bandwidths are 125 kHz, 250 kHz, and 500 kHz. Higher spreading factors (SFs) provide longer range but lower data rates.

These are the frequency bands and available channels for LoRa in some common regulatory regions:

- Europe: The European Union allows LoRa operation in the 863–870 MHz frequency band, with up to 16 channels and a bandwidth of 125 kHz.
- North America: In the United States, LoRa can be used in the 902–928 MHz frequency band, with up to 64 channels and a bandwidth of 125 kHz. In Canada, the frequency band is 902–928 MHz, and up to 16 channels are available.
- Asia: LoRa operates in various frequency bands based on the country. For instance, in Japan, it operates in the 920–928 MHz band, while in China it operates in the 470–510 MHz band.

In summary, selecting a transmission channel in a LoRa network is crucial due to its importance in optimizing performance, mitigating interference, saving energy, expanding coverage, and adapting to changing conditions. Channels play a fundamental role in enabling effective and reliable communication in diverse environments, underscoring their relevance in improving LoRa network efficiency and adaptability.

3.3. LoRa Devices

LoRa technology offers versatility in various applications and industries, making it a preferred choice for IoT deployments. A typical LoRa network comprises three main components:

- LoRa nodes are endpoint devices, usually powered by batteries, designed for wireless communication with LoRa gateways. These can include sensors, actuators, or other IoT devices.
- LoRa gateways serve as intermediaries between LoRa nodes and the network server. They have more robust hardware and are capable of managing multiple nodes, performing packet filtering, decoding, and encryption.
- The network server manages the network, processing data from the gateways and making it accessible to end-user applications. It also handles functions such as device registration, security, and data storage.

LoRa nodes communicate with gateways using a unique modulation scheme and frequency band, transmitting data periodically or upon specific events. The data are then forwarded to the network server for further processing. LoRa network infrastructure is scalable, suitable for both small deployments with a few nodes and gateways, and large deployments with thousands of nodes, multiple gateways, and advanced network servers.

3.4. LORA Spreading Factor (SF) Selection and Detection

Parameterization is crucial for the configuration of a LoRa device, which affects its performance, functionality, and efficiency in communication over LoRaWAN or other LoRa-based networks. The key areas affected include network compatibility, range, coverage, data rate, and power efficiency.

In a LoRa network, it is critical to ensure that messages are received using the same SF used for transmission. This is because SF affects the speed, sensitivity, communication range, and spectral efficiency of data transmission. The consistent use of SF across the network improves effective communication, prevents data loss, and simplifies network management. Thus, SF coherence is essential for optimal performance and efficient and reliable communication in a LoRa network.

In a LoRa network, client chips typically operate with a fixed SF that is pre-configured. This fixed SF is set in the end devices (clients) and determines how data are transmitted for the rest of communications. However, it is possible to modify this SF before deploying the network, taking into account the environment in which it will be used.

For example, (1) environmental monitoring uses a high SF for extended range and energy savings; (2) long-term asset tracking uses a low SF for fast and precise communication; (3) water level measurement employs a high SF due to less frequent monitoring; (4) agricultural irrigation control helps optimize the network; (5) waste management control can use a high SF to meet specific data transmission needs.

An essential feature of LoRa is the ability for gateways and LoRaWAN networks to request client devices to adjust their SF using ADR. This enables gateways to send SF adjustment commands to client devices based on network requirements or changing environmental conditions. When network conditions change, such as increased congestion or longer transmission distances, the gateway can instruct the client to use a lower SF for improved communication reliability or a higher SF to extend the range.

However, in a LoRa network without gateways, the modification of the SF does not occur automatically as in the Adaptive Data Rate (ADR) process in a LoRaWAN network. In this scenario, the SF configuration is typically performed manually on LoRa devices. Users or administrators must adjust the SF on each device as needed, taking into account network and environmental conditions.

3.5. LORA Channel Selection and Detection

The channels in a LoRa network play a crucial role in defining specific frequencies within the operating frequency band of the LoRa technology, allowing for the regulation of wireless communication between LoRa devices. Their essential functions include minimizing interference and preserving signal integrity. They also aim to optimize performance through proper channel selection and to maximize spectral efficiency for effective data transmission. In addition, they prevent collisions by assigning different channels to devices and ensure regulatory compliance for legal operation. Furthermore, channel adaptability allows the network to adapt to changing conditions, resulting in reliable and effective communication in diverse environments.

To select a communication channel, a LoRa transmitter node must first be configured with a list of accessible channels. Then, a particular channel is selected depending on factors like availability and interference reduction. After that, transmission parameters are adjusted to align with the chosen channel, including the central frequency and bandwidth.

Finally, data transmission is initiated on the selected channel and sometimes confirmation of reception is awaited.

In LoRa networks, devices generally stay on a single channel for long durations to maintain stable communication. However, they can switch channels if significant interference is detected, which could degrade signal quality. Channel switching can improve communication by avoiding such interference. Performance analyses can also indicate the need to switch channels for better service or efficient data transmission. Furthermore, environmental changes, such as new interference sources, can trigger channel switching to adapt and maintain effective communication. In general, while LoRa devices primarily use a single channel, the ability to switch channels is essential for adapting to changing conditions and ensuring reliable communication.

In typical LoRa networks, receiver nodes do not have an automatic mechanism to detect channels used by specific transmitter nodes, as they are manually set during initialization. More advanced systems like LoRaWAN use network infrastructure to dynamically manage and coordinate channel allocation, with servers instructing nodes on which channels to use for transmission and reception, enabling adaptive channel changes as required.

3.6. LORA Performance Measures

In the context of LoRa (Low Power Wide Area Network), various crucial performance metrics are used to assess the effectiveness and feasibility of this long-range wireless communication technology. These metrics are essential for comprehending and optimizing the performance of LoRa communications across a wide range of applications, from the Internet of Things (IoT) to telemetry systems and beyond. Among them, the following can be considered:

- Packet Delivery Rate (PDR) is the percentage of successfully received packets relative to the total packets transmitted over a wireless link. It is a crucial metric for evaluating the reliability and effectiveness of communication in noisy and interference-prone environments, such as those where LoRa operates. The key factors influencing PDR in LoRa systems include signal strength, SF, channel selection, bandwidth, coding rate, interference, noise, and collisions. Lower SF values result in higher data rates but increased vulnerability to noise, decreasing PDR, particularly in challenging environments. Higher SF values improve noise resilience and communication range, but reduce data rates. Proper channel selection mitigates interference and collisions, leading to higher PDR. Channels with minimal congestion facilitate more reliable communication, thus improving PDR. Ensuring channel selection adheres to regulatory requirements also minimizes interference from other devices operating in the same frequency band.
- Data Rate refers to the speed at which devices transmit or receive information in LoRa, measured in bits per second (bps). LoRa allows for adjustable data rates to balance speed and energy efficiency, making it ideal for IoT applications requiring long-range communication and low power consumption. The data rate in LoRa is influenced by SF and channel selection. Higher SF improves noise immunity and range but reduces data rate, while a lower SF does the opposite. Channel selection is also critical; less congested channels enhance signal quality and data rate, whereas interfered channels degrade it. Adjustments are made based on application needs to optimize transmission speed and range.
- Time On Air refers to the duration a data packet is in the air during transmission in a LoRa network. It represents the total time a LoRa signal occupies the electromagnetic spectrum from source to destination. This metric is vital for communication efficiency and power consumption. A higher SF increases time on air, as longer transmission

symbols take more time, while selecting a less congested channel can reduce the duration. Balancing SF and channel selection is essential for optimizing energy efficiency and transmission speed in LoRa applications.

- Received Signal Strength Indication (RSSI) represents the signal power in decibels (dBm) at the receiver. It is crucial for evaluating signal quality and estimating the distance between the transmitter and receiver in LoRa networks. High RSSI values indicate stronger signals and potentially more reliable communication, while low values suggest weaker signals. Factors such as SF and channel selection affect RSSI; higher SF generally increases RSSI by prolonging the duration of the symbol, and less congested channels lead to higher RSSI. However, a higher RSSI does not always equate to better signal quality due to possible interference.
- The Signal Noise Ratio (SNR) in a LoRa communication channel measures the quality of the signal relative to the noise and is expressed in decibels (dB). Higher SNR values indicate better signal quality and more reliable communication, while lower values indicate weaker signals and increased susceptibility to interference. Both SF and channel selection can influence SNR; a higher SF and a less congested channel typically result in higher SNR, thus enhancing signal quality. These factors are essential to evaluate and improve signal reliability in a LoRa network.
- Energy Consumption refers to the amount of electrical energy a LoRa device uses during operation, particularly during data transmission and reception. It is a crucial metric in LoRa applications, especially in battery-powered Internet of Things (IoT) devices and remote monitoring systems. Minimizing energy consumption is essential in low-power applications to extend battery life and ensure continuous, reliable operation. In LoRa devices, both the spreading factor (SF) and channel selection significantly impact energy consumption. A higher SF supports long-range communication but increases transmission duration and energy use. Lower SF values conserve energy by reducing transmission duration but limit communication distance. Channel choice is also important because channels with interference may require more retransmissions, thus increasing energy consumption, while less congested channels are more energy-efficient. Balancing these factors is necessary to optimize energy efficiency without compromising communication quality in LoRa applications.

4. Understanding LoRa Parameterization

Having established how LoRa parameters influence communication performance in the previous section, we now introduce a set of heuristics to guide their selection under varying conditions. Following the technical characterization of LoRa parameters and communication metrics in Section 3, our objective in this section is to provide a detailed examination of the benefits and limitations of LoRa behavior in various communication scenarios. We conducted a comprehensive analysis of the various configuration parameters involved in a LoRa transmission while exploring performance metrics and associated costs. The parameters to be considered for modification and study include: message size, preamble size, transmission power, spreading factor, channel, and bandwidth. Additionally, we examined cost and performance measures, which encompass the detection rate of a transmission, successful reception rate, data rate, time on air, RSSI, SNR, and energy consumption. This analysis was essential to understand the effectiveness and efficiency of LoRa transmissions. It allowed us to evaluate their behavior in specific communication contexts within our study.

4.1. Analyzing the Influence of LoRa Parameters

In this subsection, we sought to gain a comprehensive understanding of the impact of different LoRa parameters on its overall performance. To achieve this, we simulated an ideal scenario in which we systematically varied all relevant parameters to determine the most favorable configuration. This approach allowed us to explore how each parameter influenced LoRa's behavior and performance, providing valuable insights for optimizing its effectiveness in various communication scenarios.

To accomplish this, communication was established between two adjacent LoRa devices within a Faraday cage. This scenario represents an ideal controlled environment with specific limitations. (1) The distance between transmitter and receiver is not considered, as they are placed next to each other. (2) External electromagnetic interference is absent because the devices are enclosed in a Faraday cage. (3) There are no collisions, since only one transmitter and one receiver are involved.

The LoRa parameters adjusted in the simulations and their considered values are as follows:

- Transmission bandwidth: 125, 250, and 500 kHz.
- Spreading factor: 7, 8, 9, 10, 11, and 12.
- Carrier frequency [channels]: for 125 kHz: [868.1, 868.3, 868.5, 867.1, 867.3, 867.5, 867.7, and 867.9 MHz]; for 250 kHz: [868.3, 867.1, 867.5, and 867.9 MHz]; and for 500 kHz: [868.1 and 867.5 MHz] [57].
- Transmission power: 2, 5, 8, 11, and 14 dBm.
- The following transmission parameters remained constant throughout the simulations: coding rate set at 4/5 and a fixed preamble length of 24 symbols.

Each of the different parameter values examined results in a set of 1280 unique combinations, each with its corresponding simulations. This comprehensive dataset provided valuable insights into how each parameter influences LoRa's behavior and adaptability in different communication applications.

Each simulation involved transmitting packets of different sizes (1, 2, 5, 10, 20, 50, 100, and 200 bytes) from the sender to the receiver. In total, 1280 unique combinations were systematically tested. This means that for each of the 1280 possible different configurations, eight packets of different sizes were sent with a waiting time of tenths of a second. Additionally, the transmission of each packet size was repeated 30 times before moving on to the next size. Therefore, for each configuration, a total of 240 packets were transmitted (eight different sizes, each repeated 30 times). All of this was carried out with the purpose of calculating the average values and standard deviations for various performance metrics, including data rate, time on air, RSSI, SNR, and energy consumption.

After conducting the experiment and confirming 100% efficiency, it becomes evident that there are no significant discrepancies in transmission outcomes when altering certain parameters. Specifically, the choice of carrier frequency and transmission power does not influence transmission, given the close proximity of the two devices. However, transmission characteristics are notably influenced by the selected bandwidth, spreading factor, and message size. Figure 1 shows how the data rate in bps (or its inverse time on air in miliseconds (ms)) varies depending on the bandwidth (BW) used, the spreading factor (SF), and the size of the message sent. On the other hand, Figure 2 shows the RSSI and SNR values of these transmissions.

First, it is found that the relationship between bandwidth and data rate in LoRa communication systems is directly proportional. When the bandwidth is doubled, the data rate also doubles, highlighting the significant impact of bandwidth on transmission speed. Similarly, a parallel effect is observed with the spreading factor: increasing the spreading factor leads to a doubling of the data rate. This underscores the importance of spreading factor selection in optimizing data transmission efficiency in LoRa networks.

Furthermore, the analysis reveals an interesting trend regarding message size and data rate. As the size of the transmitted message increases, we observe a notable increase in the data rate. This suggests that larger messages can be transmitted more efficiently in LoRa systems, potentially due to the utilization of available bandwidth more effectively or the optimization of spreading factor settings.

It can also be observed that the Received Signal Strength Indication (RSSI) values exhibit remarkable stability across transmissions, regardless of changes in bandwidth and spreading factor. This consistency underscores the reliability of RSSI as an indicator of signal strength in LoRa communication. Interestingly, we note that RSSI values tend to be lower for smaller messages but stabilize as message size increases beyond 20 bytes. This trend suggests that the signal strength measurement becomes more robust and reliable for larger data packets, possibly due to improved signal-to-noise ratio (SNR) and reduced susceptibility to noise interference.

In contrast to RSSI, the signal-to-noise ratio (SNR) varies significantly between transmissions, particularly in response to changes in spreading factor. Our analysis reveals that SNR tends to improve with larger message sizes, reaching a plateau of stability around 20/50 bytes. This phenomenon implies that larger messages benefit from better noise suppression and signal clarity, resulting in higher SNR values. Additionally, we observe that specific spreading factors, namely 8, 11, and 12, consistently exhibit superior SNR ratios compared to SF 7, 9, and 10. This suggests that selecting an optimal spreading factor is crucial for maximizing signal quality and minimizing noise interference in LoRa communication systems.

Finally, the analysis reveals intriguing insights into the impact of different bandwidth settings on LoRa system performance. When transitioning from a bandwidth of 125 to 250 kHz, we observe relatively minor differences in system behavior, suggesting that these two bandwidth options may offer similar performance characteristics under certain conditions. However, as we shift to a wider bandwidth of 500 kHz, notable differences emerge, indicating a distinct effect on system performance. This phenomenon can be attributed to a specific technical limitation related to LoRa chips 1276/77/78, as outlined in the SEMTECH documentation [58]. The reduced sensitivity of these chips when operating at a 500 kHz bandwidth could lead to compromised signal reception and transmission capabilities compared to narrower bandwidth options. As a result, the wider bandwidth setting may introduce greater susceptibility to noise interference and degrade overall system performance, particularly in scenarios where signal quality is critical.

In general, maximizing data rate, RSSI, and SNR values is best achieved by prioritizing the transmission of fewer, larger messages over numerous smaller ones. This approach enhances the data/header ratio, thereby optimizing transmission efficiency. Additionally, favoring lower spreading factors (7, 8) and wider bandwidths is recommended. These settings occupy less time on the communication medium (time on air), resulting in improved data rate and overall efficiency in data transmission.

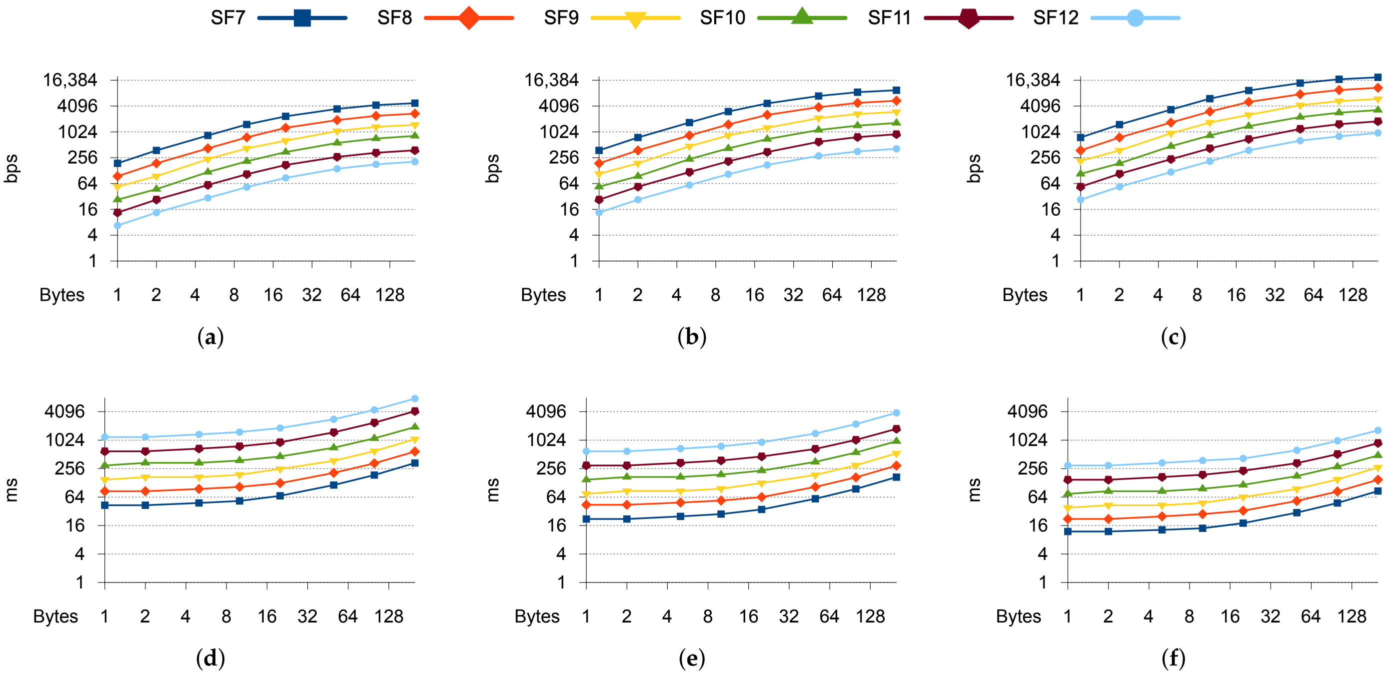

Figure 1. Data rate and time on air send (log scale) for different message sizes and spreading factors. (**a**) Data rate 125 kHz. (**b**) Data rate 250 kHz. (**c**) Data rate 500 kHz. (**d**) Time on air 125 kHz. (**e**) Time on air 250 kHz. (**f**) Time on air 500 kHz.

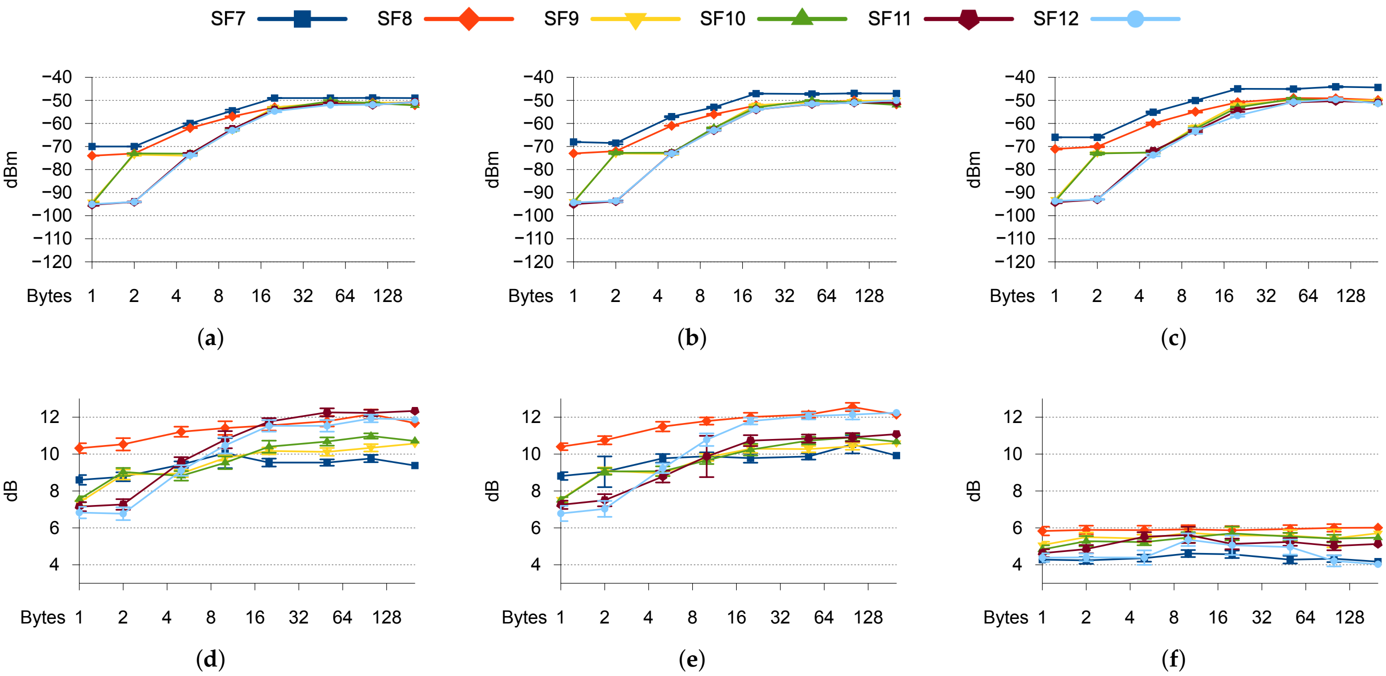

Figure 2. RSSI and SNR receive for different message sizes and spreading factors. (**a**) RSSI 125 kHz. (**b**) RSSI 250 kHz. (**c**) RSSI 500 kHz. (**d**) SNR 125 kHz. (**e**) SNR 250 kHz. (**f**) SNR 500 kHz.

4.2. Energy Consumption of a Transmission

In this experiment, the aim is to assess the energy consumption associated with data transmission in LoRa, encompassing both transmission and reception costs. Data transmissions were conducted with variations in packet size ranging from 1 byte to 253 bytes, utilizing each of the eight channels within the 125 kHz bandwidth. Energy consumption measurements were obtained using a ‘DollaTeck TC66’ USB ammeter/voltmeter, capable of measuring energy consumption in mWh. This device boasts a current measurement resolution of 0.01 mA and records consumption at an approximate frequency ranging from half a second to one second.

Figure 3 depicts three consumption measurements conducted in the experiment. The first measurement (Figure 3a) assesses the device's energy consumption when idle, without any message transmission (0.30 W). The second measurement (Figure 3b) evaluates the energy consumption during message transmission, with messages transmitting for a minimum of five seconds, while varying message size and spreading factor (0.64 W). However, it is important to clarify that during the measurement shown in that figure, the hardware remained active for a fixed duration of five seconds. This duration was applied regardless of the selected spreading factor (SF) or message size. As a consequence, configurations using lower SFs and smaller payloads (e.g., SF7 with 1–8 bytes) needed to transmit many more messages to fill the 5 s interval. In contrast, configurations with higher SFs and larger payloads (e.g., SF12 with 250 bytes) required fewer transmissions during the same period. This ensured consistency in energy sampling duration across configurations. The third measurement (Figure 3c) examines device consumption during message reception mode (0.45 W). Overall, it is observed that transitioning to transmission mode to send a message increases device consumption by 0.34 W, whereas the device maintains a consistent consumption of 0.45 W in reception mode.

Furthermore, the total energy consumption is directly proportional to the time the radio hardware remains active, which is itself strongly linked to the time on air (ToA) of the transmitted messages. As illustrated in Figure 4, configurations with longer ToA—such as those using higher spreading factors (SFs) and larger payloads—result in higher energy consumption per message, while configurations with shorter ToA—typically those using lower SFs and smaller payloads—are more energy-efficient. Consequently, the energy expenditure associated with message transmission can be effectively managed by adjusting the message size and the SF used. For example, transmissions with brief durations, small messages, and low SF values (e.g., SF7) consume considerably less energy than transmissions with long durations and high SF values (e.g., SF12).

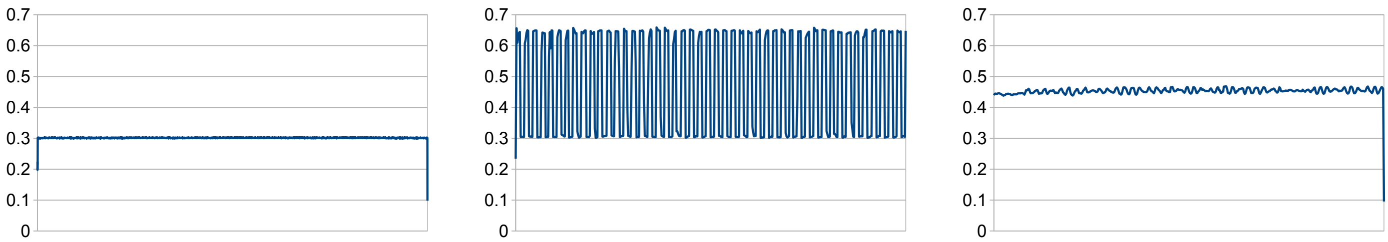

Figure 3. Power consumption in Watts. (**a**) Idle. (**b**) Sending. (**c**) Receiving.

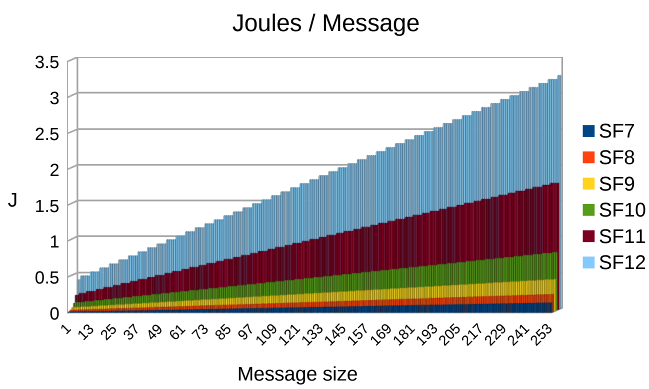

Figure 4. Energy consumption for sending a message.

In summary, the observation that shorter, smaller-message transmissions with lower spreading factor settings incur lower energy consumption emphasizes the importance of optimizing spreading factor selection for energy-efficient LoRa parameterization.

4.3. Preamble Length

Communication between LoRa nodes relies on mutual agreement on configuration parameters such as bandwidth, channel, and spreading factor for message transmission. Following this agreement, the sender can initiate message transmission at any time. The receiver continuously monitors the medium for activity in the specified configuration to initiate message decoding. Message transmission begins with the transmission of a preamble at the onset, allowing for activity validation and synchronization of message transmission and reception. The preamble's length can vary and is adjustable to ensure adequate time for receiver detection, typically ranging between 8 and 16 symbols. Upon detecting activity in the medium, the receiver initiates the decoding protocol, attempting to decode the preamble and subsequently the transmitted message. Thus, ensuring sufficient preamble duration in the medium is crucial to allow receiver detection and subsequent message decoding. This subsection explores the probability of detecting and receiving a message under varying preamble and message lengths, considering bandwidth, channel, and spreading factor configurations.

Figure 5 illustrates the probability of successful message decoding and reception following medium activity detection, considering different bandwidth and spreading factor settings. This probability is assessed by altering the preamble and message lengths, resulting in varying time on air for each transmission. Notice that preamble length, measured in symbols, is adjusted across values of 6, 8, 12, 16, 24, and 32. Similarly, message length, measured in bytes, is varied across values of 1, 2, 5, 10, 20, 50, 100, and 200 bytes.

In general, configurations with a preamble that is too short or a data portion with excessive time on air show poor performance. Both cases reduce the probability of successfully decoding a message once activity is detected in the medium. This phenomenon occurs because the protocol relies on accurately decoding the preamble to correctly process the start and end of the message data. In this way, when the preamble duration in the medium is insufficiently brief, the receiver lacks the necessary time to detect it accurately, leading to message loss and few decoding attempts. Conversely, if the data portion of the message is excessively large, detection of activity in the medium occurs, but it cannot be interpreted as a preamble for decoding purposes. Consequently, there are numerous detections with failed decoding attempts. For instance, in SF7, where the time on air for a preamble of size 6 is low, more failures are observed compared to higher SFs such as 11 and 12, where the time on air is greater. Additionally, it is noticeable that for large messages of 200 bytes, many detections occur without successful decoding of the preamble or message. This is particularly evident for configurations such as SF7-6/200, SF8-8/200, and SF9-12/200.

In conclusion, the duration of the preamble and the size of the message play pivotal roles in determining the success rate of message decoding following medium activity detection. Configurations with excessively short preamble durations or excessively long message lengths tend to result in significant drops in the probability of successful message decoding. To mitigate such challenges and enhance the efficiency of LoRa systems, it is crucial to strike a balance between preamble duration and message size. This involves selecting appropriate spreading factors and adjusting time on air parameters to optimize transmission efficiency while minimizing energy consumption. Moreover, considering the trade-offs between spreading factor settings and message size can help tailor LoRa parameterization to specific application requirements, thereby maximizing communication reliability and network performance.

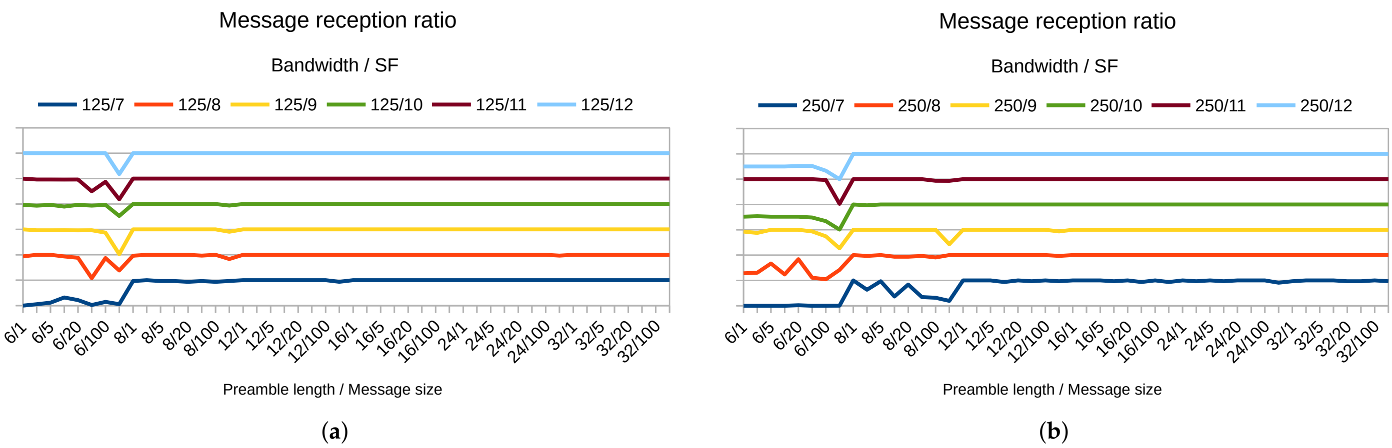

Figure 5. Probability of detection and reception of a message varying the size of the preamble and the size of the message. (**a**) 125 kHz. (**b**) 250 kHz.

4.4. Latency of Changing Parameters

Understanding the time implications associated with modifying configuration parameters in LoRa communications is essential to ensure efficient and effective management of this technology. This knowledge not only allows for optimizing network performance but also enables informed decision-making regarding the implementation and adjustment of transmission parameters, thereby contributing to the success and effectiveness of LoRa systems. In our case, it is crucial to know weather the time required to modify a couple of SF and channel at the reception is significant in the search for the specific SF and channel assumed in the transmission. In order to achieve this, we executed a code that modifies the parameters employed in our proposal. Specifically, the bandwidth, channel, spreading factor, and transmission power were modified. For each change, the modification of that parameter was measured a thousand times, thus avoiding precision errors and thus obtaining an average that represents the cost of an individual change. The values in Table 1 represent the average execution time (in microseconds) required by Arduino functions to modify transmission parameters on a TTGO LoRa32 development board using the Semtech SX127x chipset.

Table 1. Cost in microseconds of changing LoRa parameters.

Parameter	Microseconds	Stdev
Bandwidth	211.535	0.0070
Spreading Factor	375.798	0.0035
Channel	244.881	0.0033
Transmission Power	243.897	0.0058

It can be seen that when changing the four parameters, the one that takes the most is the SF. Nevertheless, if we want to change SF and channel, both times are added signifying that the total time is 620.679 microseconds.

Table 2 presents the time on air (ToA) in microseconds for preambles ranging from 6 to 32 symbols, using SF7 (which offers the minimum ToA) and varying the bandwidth between 125 kHz and 500 kHz. These values were obtained through an Arduino-based experiment in which we measured the transmission time of packets while varying the message size, allowing us to isolate and estimate the cost of transmitting a single preamble symbol. The results were validated on the same previous hardware.

Table 2. Cost in microseconds for sending preamble.

Bandwidth–SF	Preamble Symbol Size						
	6	8	12	16	24	32	Mean [1]
125 KHz–SF7	9250	12,290	18,320	24,280	36,000	47,480	1510
250 KHz–SF7	4650	6180	9210	12,210	18,110	23,870	760
500 KHz–SF7	2280	3130	4660	6180	9160	12,070	380

[1] Mean for one symbol of the preamble.

In this way, a relative comparison can be made between the cost of transmitting this preamble with respect to the cost of modifying the transmission parameters. Thus, we can observe that the real cost of changing parameters is about 10 times faster than sending the preamble in the best of cases, becoming about 100 times faster for typical values of SF7-125KHz-16.

In conclusion, SF modification consumes the most time, and when changing multiple parameters simultaneously, the time for each modification is cumulative. However, despite these time costs, the overall time required for parameter modification is significantly faster compared to transmitting preambles, particularly for typical LoRa configurations. This highlights the importance of optimizing parameter modification processes to enhance the efficiency and effectiveness of LoRa systems.

5. Our Proposal

5.1. Overview

Based on the analysis and insights from Section 4, we present a novel strategy to improve LoRa network efficiency. It enables adaptable communication configurations to respond to changing network conditions. This ensures reliable and efficient communication across diverse environments and scenarios. Unlike traditional methodologies that address the issue through centralized means within the LoRaWAN layer, our approach embraces a decentralized methodology within the LoRa layer itself. Our innovative approach entails the simultaneous utilization of various channels (CHs) and spreading factors (SFs) on client chips, effectively reducing node-to-node contention. Unlike existing proposals that focus primarily on SF adjustments, our method also includes channel modifications. This enables comprehensive management of orthogonal channels and SF, ensuring seamless, collision-free operations. Furthermore, our method for selecting the SF and channel during transmission and detecting them during reception is fully adaptable. It can adjust to the SF, the channel, or both.

Choosing the right SF and CH in a LoRa system is essential for effective communication. The appropriate selection of SF and CH improves network coverage and ensures reliable data transmission over long distances. This decision also mitigates interference by minimizing collisions and packet loss. Lower SF values increase data rates but require higher signal-to-noise ratios, while higher SF values improve noise resilience at lower data rates. Balancing these factors optimizes energy consumption, which is crucial for battery-powered devices. Additionally, dynamic adaptation to environmental changes (terrain, weather, electromagnetic sources) is necessary to maintain optimal performance. In summary, a well-informed selection of SF and CH maximizes coverage, minimizes interference, balances data rate and energy consumption, and adapts to environmental changes to enable efficient LoRa communication.

Detecting SF and CH during reception in standalone LoRa nodes is challenging because of their inability to independently determine these parameters in real time. Traditional LoRa nodes are statically configured for specific channels, lacking dynamic detection capabilities without gateway assistance. Our method utilizes LoRa radio chips with Channel Activity Detection (CAD) mode, designed to detect incoming transmissions with efficiency. The process involves brief receiver activation, rapid channel scan, and deactivation, comparing the received signal to the ideal packet preamble. This method is significantly more energy-efficient than continuous reception, complete in one symbol period. CAD is beneficial for single-channel LoRa transceivers, allowing automatic detection and reception across various SFs. An algorithm continuously prompts the transceiver to detect transmissions, triggering a heuristic-based process to identify SF and CH upon detection.

Our proposal, adapted for LoRa networks without gateways, focuses on two key aspects: dynamic selection of both SF and CH at the transmitting node and dynamic detection at the receiving node to identify the SF and CH utilized during transmission. Figure 6 illustrates the general concept of our proposal, featuring a transmitting (sending) node and a receiving node as essential elements in LoRa network data transmission.

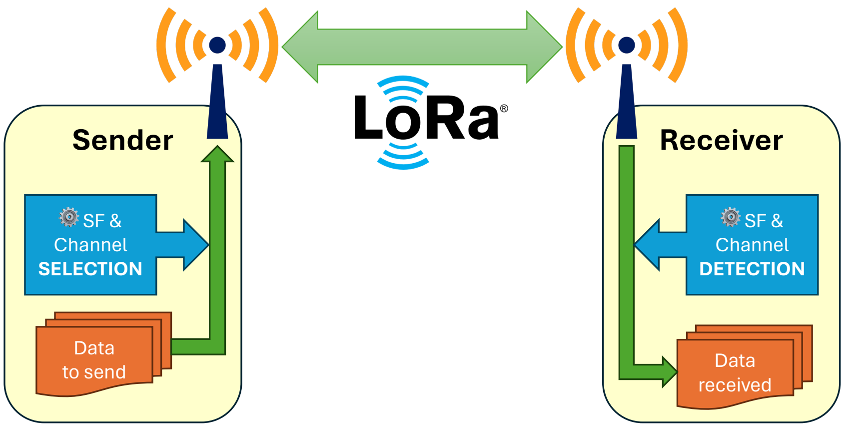

Figure 6. Network architecture of LoRa multi-SF (our proposal).

In both cases, whether for selection during transmission or detection during reception, a two-step mechanism is employed that combines values generation through heuristics and selection among these values using policies. In the first step, heuristics are used to explore and generate a list of potential values for the SFs and CHs by considering a variety of factors that can be preset, random, performance-based, environmental, or pattern-based. These generated values can range from a single value to the maximum possible, and they are stored in a vector where each element may even have an associated weight determining its usage. In the second step, policies are applied to optimally select a specific value from those previously generated, either for transmission or reception. This two-step method, depicted in Figure 7, facilitates flexible adjustment of communication configurations in response to evolving network conditions, thus guaranteeing efficient and reliable communication across diverse environments and scenarios.

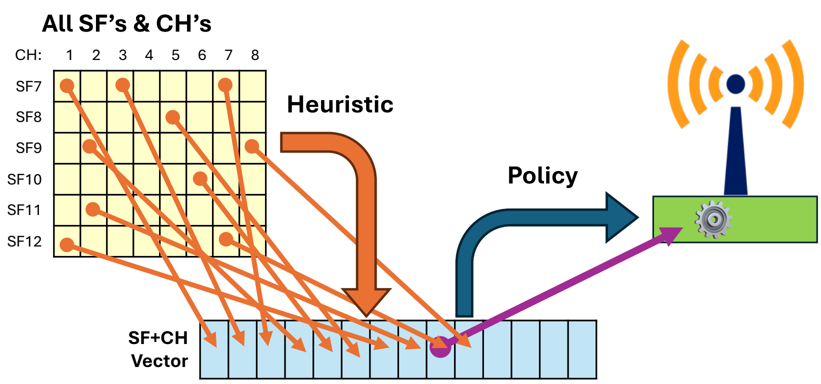

Figure 7. Heuristic and policy to select/detect SF and channel.

Both processes in LoRa networks rely on the use of heuristics, which are general rules or principles, to generate a range of potential values that could be utilized. These values are assessed by a specific policy that selects a definitive value for implementation, either during data transmission (Algorithm 1) or during reception detection (Algorithm 2). It is important to note that these same heuristics and policies can be applied interchangeably in both transmission and reception contexts, ensuring consistency and coherence in the selection and detection processes. The primary difference between selection during transmission and detection upon reception in LoRa networks is in the management of selected values. During transmission, the values usually remain static but can be changed depending on the conditions of the network. In contrast, reception involves continuous selection until the transmitted value is identified. This process is vital to adapt to environmental changes and ensure reliable data reception.

Algorithm 1 Two-step LoRa Communication–Setup Algorithm (Sender)

```
Vector ← CHOOSE SF + CH values using selected Heuristics          ▷ Apply Heuristic
loop
    CommSetup ← SELECT SF + CH from Vector using chosen Policy     ▷ Apply Policy
    APPLY CommSetup to configure LoRa transmission parameters      ▷ Selection
    repeat
        TRANSMIT message using current SF+CH configuration
        Optionally EVALUATE if CommSetup is still suitable
    until CommSetup no longer ensures efficient communication conditions
end loop
```

Algorithm 2 Two-step LoRa Communication–Setup Algorithm (Receiver)

```
Vector ← CHOOSE SF + CH values using selected Heuristics          ▷ Apply Heuristic
loop
    CommSetup ← SELECT SF + CH from Vector using chosen Policy     ▷ Apply Policy
    if CommSetup is DETECTED via preamble and reception succeeds then ▷ Detection
        repeat
            LISTEN for messages using current SF + CH configuration
            Optionally EVALUATE if CommSetup is still suitable
        until CommSetup no longer ensures efficient communication conditions
    end if
end loop
```

It is important to note that this work does not aim to propose a specific heuristic, nor does it aim to outperform any particular strategy. Instead, our contribution is the design and validation of a general mechanism for dynamic SF and CH selection and detection in LoRa networks, applicable in scenarios without infrastructure or gateways. This mechanism assumes a communication context with constrained end-nodes, energy limitations, static or quasi-static topologies, and single-hop transmissions. Within this scope, the mechanism enables autonomous and adaptive configuration decisions, independent of the specific strategy implemented.

5.2. Heuristics for Generating SF and CH Vector Values

The proposed framework is intentionally designed to be modular and extensible, allowing for the integration of any kind of heuristic—ranging from simple rules to advanced, learning-based strategies. The goal of this work is not to propose or evaluate a specific heuristic, but to establish a general-purpose architecture that decouples the decision logic from the communication mechanism. This separation allows developers and researchers to focus solely on designing new heuristics or policies without modifying the core system.

The examples presented in this section—based on static rules, performance metrics, or environmental characteristics—are intentionally simple and lightweight. Their role is to validate that the mechanism is able to dynamically select and detect SF and CH combinations without prior coordination and to confirm that this process can be performed efficiently, with minimal overhead, and under constrained hardware conditions. We emphasize that these heuristics are not intended to be optimal or competitive in terms of performance, but rather to demonstrate that the mechanism itself operates correctly.

This distinction is important in response to the concern of Reviewer 2 about the significance of performance improvement. In our case, the main result is not the gain achieved by a specific strategy, but the empirical confirmation that the system is capable of detecting and adapting to transmission parameters (among 48 possible SF+CH combinations) autonomously, in a fully decentralized setting. Performance gains are a byproduct of this mechanism working as intended, and future work will include comparisons with other approaches to quantify such improvements more systematically.

In both transmission and reception, our framework employs heuristics to generate and detect possible SF and CH values. These heuristics are versatile and can be applied to both contexts, although their effectiveness may vary depending on the specific scenario.

The heuristics for generating SF and/or CH values can be classified into four distinct categories based on unique methodologies and principles. These heuristics can be used individually or in combination, facilitating flexibility in optimization processes. Importantly, the system was designed from the ground up to support the inclusion of new and more sophisticated heuristics as needed, ensuring adaptability to evolving requirements and technological advances. This promotes the integration of novel approaches—including learning-based or environment-aware strategies—to further improve the efficiency and effectiveness of SF and CH value generation.

1. Statically Predetermined: This category consists of using a predefined set of spreading factor (SF) and/or channel (CH) values that remain fixed, regardless of network conditions or environmental changes. Although straightforward and easy to implement, this approach lacks adaptability and may lead to suboptimal performance in dynamic or congested environments.
    - Heuristic 1.1 (Subset of Values): The vector of candidate SF-CH combinations may include a limited subset, ranging from a single pair to multiple configurations. These values can be statically defined at deployment time or generated randomly, depending on the application's requirements.
    - Heuristic 1.2 (All Values): This heuristic includes the full set of 48 possible SF-CH combinations (6 spreading factors $\times$ 8 channels), offering the maximum exploration range without any runtime filtering or prioritization.
2. Performance-Based: These heuristics generate candidate SF and/or CH values by analyzing runtime performance metrics observed in previous transmissions. The goal is to dynamically adapt to current network and environmental conditions in order to improve reliability, throughput, energy efficiency, and overall communication performance. These heuristics are particularly useful in scenarios with variable interference, mobility, or heterogeneous traffic patterns.
    - Heuristic 2.1 (Packet Delivery Ratio—PDR): Prioritizes SF-CH combinations that have shown high delivery success in past transmissions. By leveraging historical PDR data, it adapts to dynamic conditions such as interference or collisions, favoring configurations that consistently yield reliable communication. It is especially useful in fluctuating environments, promoting lower SFs in low-

contention, short-range scenarios, and higher SFs when robustness or range is needed.

- Heuristic 2.2 (Data Rate): Focuses on selecting configurations that maximize the number of bits successfully transmitted per second. Since the LoRa data rate depends on SF, bandwidth, and coding rate, this heuristic favors combinations with low SFs and wider bandwidths, as long as signal conditions allow. It is especially relevant in applications that generate frequent or time-sensitive data (e.g., real-time telemetry), where reducing latency and maximizing throughput is a priority.
- Heuristic 2.3 (Time on Air): Aims to minimize the airtime required to transmit each packet, which helps reduce both energy consumption and the likelihood of packet collisions in dense networks. This heuristic implicitly favors configurations with lower SFs and smaller payload sizes. It also benefits networks subject to duty-cycle regulations, where reducing time on air increases the number of allowable transmissions per device.
- Heuristic 2.4 (Received Signal Strength Indicator—RSSI): Selects SF-CH combinations that yield higher RSSI values at the receiver, assuming that stronger signals generally correspond to better propagation and more reliable links. However, since RSSI measures total received power—including interference and noise—it does not always reflect true signal quality. Therefore, this heuristic is most effective when combined with complementary metrics such as SNR or historical PDR. For example, RSSI can be used as a preliminary filter to discard weak links, while final selection is refined based on more robust indicators. Its simplicity and availability on most LoRa transceivers make it particularly suitable for low-cost or resource-constrained devices.
- Heuristic 2.5 (Signal-to-Noise Ratio—SNR): Selects SF-CH combinations that maximize SNR, as higher values indicate cleaner signals with lower background noise or interference. Unlike RSSI, which measures total received power, SNR reflects the actual quality of the signal relative to noise, making it a more reliable indicator of link performance. Configurations with better SNR can support higher data rates and reduce packet errors. This heuristic is especially useful in environments where RSSI remains stable across channels but decoding performance varies due to fluctuating interference.
- Heuristic 2.6 (Energy Consumption): Prioritizes configurations that minimize the energy cost per message, either by reducing transmission time (e.g., low SF) or avoiding retransmissions due to failed packets. It considers not only SF and CH, but also the message size, hardware power profile, and recent transmission outcomes. For battery-powered IoT devices, this heuristic is critical to prolonging device lifetime and reducing maintenance costs.

3. Environment-Based: These heuristics adapt SF and/or CH values based on environmental factors that may affect signal propagation, reception quality, or transmission reliability. They consider real-world physical and geographical elements such as node distance, obstacles, building materials, electromagnetic interference (EMI), and atmospheric conditions. The objective is to improve robustness and coverage by tailoring configuration parameters to specific environmental scenarios.

   - Heuristic 3.1 (Distance): This heuristic adjusts SF and CH based on the estimated or measured physical distance between sender and receiver. For short-range communication (e.g., within a room or small area), lower SFs are preferred due to their higher data rates and reduced airtime. For long-range links (e.g., across fields or buildings), higher SFs are favored because they provide increased sensitivity and longer time on air, improving the likelihood of successful reception. Channel selection also considers propagation loss across frequencies; lower frequencies may be selected for long-range due to better penetration capabilities.

- Heuristic 3.2 (Data Interference): This heuristic monitors congestion and cross-interference from other nodes or devices operating in the same band (e.g., other LoRa devices, Wi-Fi, ZigBee). In low-interference environments, low SFs and broader bandwidths can be safely used to optimize throughput. In high-interference scenarios, higher SFs and less congested channels are preferred to ensure transmission reliability. This heuristic may leverage local spectrum sensing or past transmission success rates to dynamically adapt to channel conditions.
- Heuristic 3.3 (Electromagnetic Interference—EMI): This heuristic focuses on external noise sources such as industrial machinery, electric motors, or high-voltage lines. It selects SF and CH combinations that maximize immunity to EMI. In clean environments, fast configurations (low SFs) are suitable. In contrast, in EMI-heavy zones, higher SFs are used for their robustness, and channels are selected to avoid known interference bands. This heuristic may benefit from integration with external EMI monitoring or pre-characterization of the deployment area.
- Heuristic 3.4 (Weather Conditions): This heuristic adapts to changes in atmospheric conditions (e.g., rain, humidity, fog), which can affect signal attenuation and propagation, especially over long distances. Under clear weather, low SFs suffice for efficient transmission. During adverse weather events, higher SFs may be used to improve reliability, as they tolerate higher levels of attenuation and fading. If available, real-time weather data from sensors or APIs can be integrated to guide the heuristic dynamically.

4. Pattern Recognition: These heuristics generate candidate SF and/or CH values by identifying temporal or statistical patterns in historical communication data. The rationale is that communication behavior—either from individual nodes or the surrounding environment—often exhibits repetitive or predictable patterns that can be exploited to improve performance, reduce scanning effort, and accelerate convergence in both transmission and reception phases.
   - Heuristic 4.1 (Past History): This heuristic tracks the historical usage and success rates of previously applied SF and CH combinations. The system maintains a log of recent communication attempts and their outcomes (e.g., successful delivery, RSSI, SNR, energy cost) and uses this log to prioritize configurations that have performed well in similar contexts. For example, if SF9 and CH3 have consistently resulted in successful transmissions at a given time of day or in a certain location, they may be selected again. This approach is lightweight and effective in relatively stable or cyclical environments (e.g., periodic sensing applications, static deployments).
   - Heuristic 4.2 (Machine Learning—ML): This heuristic applies data-driven models to predict the most suitable SF and/or CH configuration for upcoming transmissions. It uses historical communication records to train a model (e.g., decision trees, reinforcement learning, or neural networks), where input features may include message size, time of day, battery level, recent signal quality, or node location. The model outputs a probability or ranking of SF-CH pairs, which are then evaluated before use. As new data accumulate, the model is periodically retrained to adapt to changing conditions. ML-based heuristics are particularly beneficial in dynamic, large-scale, or heterogeneous networks, but may require

additional computational and memory resources depending on the complexity of the model.

Finally, associated with each value in the SF+CH vector, a weight or priority can be assigned, reflected in its significance or suitability for specific transmission or reception conditions. These weights or priorities can vary on the basis of the characteristics of each heuristic. Subsequently, these weighted values can be utilized by policies responsible for selecting the most appropriate SF and/or CH for transmission or for detecting the transmission parameters during reception. This weighting mechanism enables the system to dynamically adapt to changing network conditions, optimizing communication performance and reliability based on the prioritization of relevant criteria.

5.3. Policies for Selecting SF and CH Vector Values

Similar to the heuristics, the selection policies described in this section serve as basic examples to validate the behavior of the proposed mechanism. They are intentionally simple and lightweight, allowing us to demonstrate that the system can execute the decision-making process (selection and detection of SF and CH) independently of the specific strategy used. The framework supports arbitrary policy implementations, enabling developers to define more sophisticated or application-specific policies in future work. This design allows the communication layer to remain agnostic of the decision logic, which is essential for achieving modularity, extensibility, and experimental flexibility.

Once the SF and/or CH values have been generated by a given heuristic—either for transmission or reception—a policy is applied to select which specific configuration to use at each step. Each of these candidate values may optionally carry an associated weight, which the policy can use to prioritize or probabilistically bias the selection. The policies described below exemplify different selection approaches, including uniform, weighted, and history-based strategies.

- Policy 1 (Equitable Random Selection With Repetition): This policy selects one SF-CH pair at random from the candidate list, assigning equal probability to all elements in each iteration. The same value may be chosen repeatedly. It is a simple and stateless approach, suitable for scenarios where diversity is less critical or where the list of candidates is frequently updated. Its low computational cost makes it appropriate for resource-constrained nodes or early exploration phases.
- Policy 2 (Equitable Random Selection Without Repetition): Like Policy 1, this policy applies uniform random selection, but ensures that each SF-CH pair is selected exactly once per cycle. Once the entire set is used, the list is reset and selection continues. This promotes exhaustive exploration, which is valuable in environments where conditions vary over time and it is important to periodically test all available configurations to avoid overlooking optimal ones.
- Policy 3 (Weighted Random Selection With Repetition): Each candidate SF-CH pair is assigned a weight based on past performance, estimated reliability, or external heuristics. Elements are selected randomly, but the probability of selection is proportional to their weight. This allows the system to bias toward high-performing configurations while maintaining some randomness. It is particularly effective in dynamic environments where conditions evolve and a balance between exploitation and exploration is needed.
- Policy 4 (Weighted Random Selection Without Repetition): This policy extends Policy 3 by excluding selected elements from future draws within the same cycle, ensuring no repetitions until the list is exhausted. It combines performance-driven prioritization with controlled diversity, allowing the system to favor better candidates early while

still evaluating the full spectrum of options. It is well suited to reception scenarios where rapid detection is needed, but full SF-CH coverage is also required.

- Policy 5 (Deterministic Weight-Based Selection): In this policy, the element with the highest assigned weight is always selected first, followed by the next highest, in descending order. After all values are used, the selection process restarts—potentially with updated weights reflecting recent network behavior. This deterministic approach is ideal for stable environments where strong historical patterns exist, or in applications requiring predictable, repeatable behavior with minimal overhead.

Regardless of the policy chosen to determine a specific value from a range of options, prioritizing the last detected value (SF and/or CH) is essential. This approach ensures that the previously detected configuration is thoroughly evaluated before considering new selections, enhancing efficiency and stability in the decision-making process and minimizing disruptions. Validating the last detected pair before selecting a new one maintains continuity and reliability, contributing to the network's overall robustness.

5.4. One-Time or Continous Mode

The proposed mechanism for SF/CH selection and detection supports two main implementation modes: a one-time setup or continuous adaptation. In the one-time mode, SF and/or CH values are selected and applied once—typically during deployment—to initialize communication settings. This approach minimizes resource usage and network overhead, making it well-suited for static environments with predictable conditions.

In contrast, the continuous mode enables real-time re-evaluation and adaptation of SF and CH parameters during operation. This allows the system to respond to changes in network conditions, such as interference, traffic patterns, or node availability. While this mode offers higher robustness and adaptability, it comes at the cost of increased computational effort and potential overhead. The decision between one-time and continuous operation depends on the specific needs and constraints of the application, and hybrid approaches may be applied depending on context.

The mechanism was specifically designed to support both modes in practice. It can operate in static deployments (e.g., environmental sensing) as well as in dynamic or mobile scenarios where runtime adaptation is required. This flexibility is one of the core strengths of the architecture, ensuring its applicability across a wide range of IoT use cases.

Importantly, the entire framework was developed with resource-constrained IoT nodes in mind. All parameter-switching operations (such as SF, CH, and transmission power updates) are executed using standard transceiver driver commands and complete in less than 1 millisecond, as shown in Table 1 (Section 4.4). The computational and memory overhead of the heuristic generation and policy selection components is minimal, making the solution viable for embedded systems with limited capabilities, such as Cortex-M0/M3-class microcontrollers. This ensures compatibility with real-world LoRa nodes, maintaining energy efficiency and responsiveness in both configuration modes.

6. Simulation and Analysis of Results

6.1. Experimental Framework

To evaluate the effectiveness of the decentralized method described in Section 5, we conducted a series of experiments under controlled conditions. The experiments in this section were conducted with two LoRa nodes inside a Faraday cage; one node functioned as the sender and the other as the receiver. The sending node continuously transmitted messages, adhering to the duty cycle time. Message sizes varied, and multiple messages were transmitted over a specific period of time. The emission parameters were determined by the heuristics applied. In summary, the sender selected the content and transmission frequency (message size and interval between messages). It also chose the emission parameters (SF, channel) and repeatedly sent the message within a defined timeframe. In contrast, the receiver waited for incoming messages. The receiver had no prior knowledge of the broadcast parameters, timing, or duration. The receiver employed SF and channel scanning strategies and attempted to receive the message upon detecting its preamble. Upon successful reception, it recorded the message content, timing, and communication parameters.

As discussed previously, the 868-MHz frequency band in Europe is allocated for low-power applications, including LoRa communications. Regulatory standards allow up to eight communication channels within this band, each spaced 125 kHz apart. LoRa systems use six common spread factors (SFs) ranging from SF7 to SF12, enabling adaptation of transmission speed and resilience to noise based on communication requirements. Thus, the system provides 48 potential combinations from the interaction of six SFs and eight channels.

The experiments in this study were intentionally conducted under highly controlled conditions using a two-node setup enclosed in a Faraday cage. This eliminated external interference, collisions, and multipath fading, enabling us to validate the correctness and efficiency of the proposed mechanism in isolation. As a result, the presented results represent best-case performance. Future work will extend this evaluation to multi-node environments with interference and contention to assess scalability and robustness under realistic deployment conditions.

6.2. Experiment 1: All Channel and SF Combinations. All Values. Equitable Random Selection Without Repetition

The experiments performed in this section follow the guidelines of the previous section. Therefore, we have a sender node and a receiver node.

The specific behavior of the sender is as follows. First, it chooses a duty cycle among 50%, 20%, 10%, 5%, 2% and 1%. Then, it selects a message size of 5, 10, 20, 50, 100, or 200 bytes. Finally, it sends that message for one minute at the selected periodicity using one of the 48 possible SF and channel combinations, chosen at random.

Meanwhile, the behavior of the receiver is as follows. It tries each of the 48 possible SF and channel combinations. If the preamble is not detected, it randomly selects another combination. If detected, it tries to read the full message and records the reception parameters.

This broadcast experiment follows the value generation heuristic of SF and CH number 1.2: "Statically predetermined: All Values". Follow the selection policy of SF and channel number 2: "Equitable random selection without repetition". Concerning the periodicity of the testing: "Continuous-mode polling". All of them are described in the previous section.

This experiment is performed by testing all duty cycle values described and all message sizes specified. In all these experiments, the size of the preamble is 16 symbols.

It is important to note that the receiver tries all 48 configurations with the same probability. In this case, the occupation time of the physical medium differs for each configuration. Slower configurations that occupy the medium longer are more likely to be detected than faster ones occupying it for shorter periods. In the following, we discuss the behavior of this experiment.

Figure 8 shows the receiver detection ratio of the messages sent by the sender. Thus, each of these bars indicates the percentage of success in receiving a message given an emission periodicity that meets the specified duty cycle, given a message size, and given a certain SF, regardless of the channel used. For example, the last column of Figure 8f shows that when the sender transmits a 200-byte message, occupying the medium 50% of the time, and uses any of the eight possible channels with SF 12, the receiver has a 37% probability of correctly reading the message.

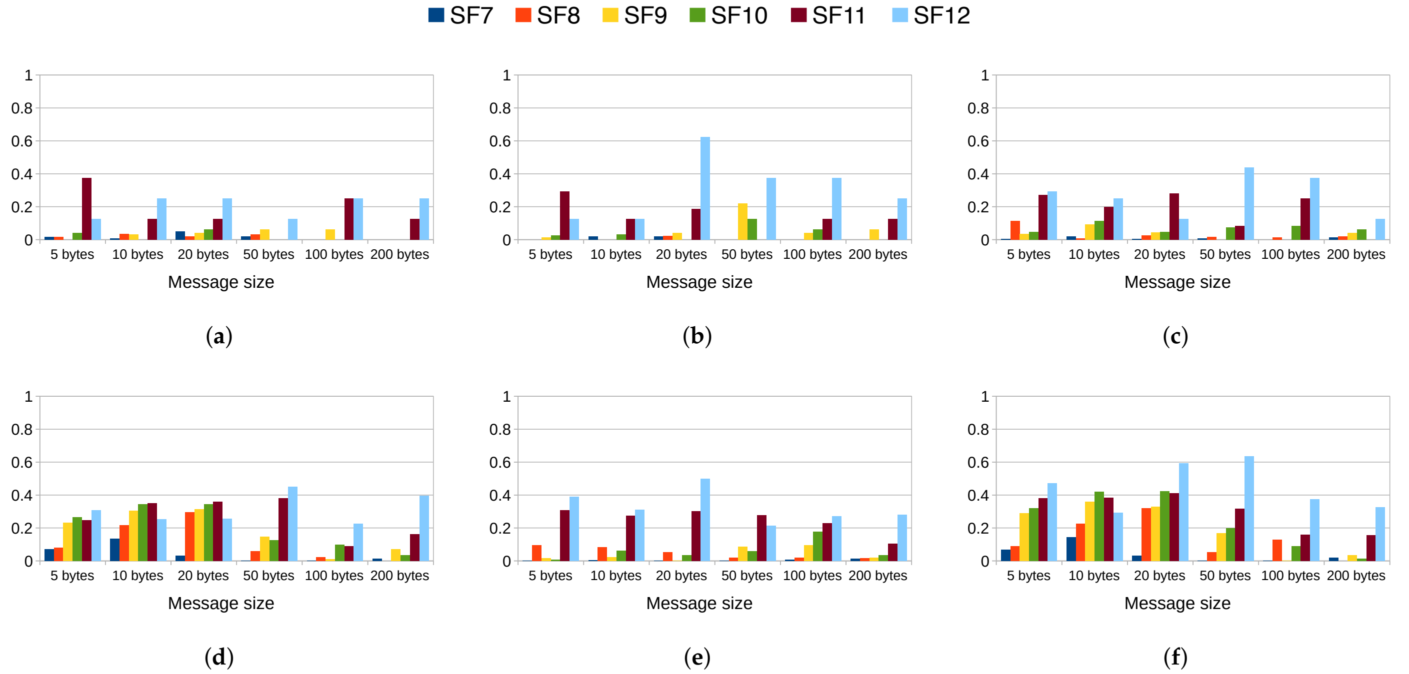

Figure 8. Reception rate of random heuristic for different duty cycles, message sizes, and SFs. (**a**) 1% Duty Cycle. (**b**) 2% Duty Cycle. (**c**) 5% Duty Cycle. (**d**) 10% Duty Cycle. (**e**) 20% Duty Cycle. (**f**) 50% Duty Cycle.

The experiment hinges on the likelihood that the receiver is actively scanning for the SF and channel values the sender uses at a specific moment. Moreover, the preamble of the message must be transmitting through the physical medium at that instant. The receiver needs less than two symbols to notice the broadcast and attempt to read the message; however, at any other time, the receiver will miss the message. Therefore, detection/reception success relies on the preamble's duration in the medium and the number of retries the receiver makes with a particular SF and channel configuration, ensuring both align simultaneously.

Transmissions with low SF values of 7 and 8 have been shown to exhibit a very low detection probability, approximately 2%, and invariably less than 10%. In contrast, transmissions with higher SF values of 11 and 12 show a detection probability around 20%, often exceeding 50% in certain scenarios.

Conversely, it can be inferred that the medium's occupancy affects the likelihood of detecting transmissions. Observations indicate that for low occupancy levels, ranging from 1% to 5%, the detection probability is relatively low, approximately 10%. Meanwhile, for medium to high occupancy levels, between 20% and 50%, the detection probability increases to about 30%.

6.3. Experiment 2: All Channel and SF Combinations. All Values. Weighted Random Selection Without Repetition

This experiment builds on the previous one. The sender's behavior remains consistent, meaning that it randomly picks a specific SF and CH emission configuration for each duty cycle and message size tested, broadcasting the message for one minute. However, the behavior of the receiver is altered. In a previous experiment, randomly choosing 1 of the 48 possible configurations resulted in higher SFs (11, 12) being more likely to be detected compared to lower SFs (7, 8). In this experiment, our objective is to equalize the detection time for various configurations. To achieve this, the probability of testing configurations with lower SFs relative to higher SFs is adjusted. Therefore, tied to the time on air of each transmission, configurations with SF 7 are tested 32 times more than those with SF 12. Specifically, the test frequency for each SF is increased as follows: SF7 32 times, SF8 16 times, SF9 8 times, SF10 4 times, SF11 2 times, and SF12 1 time. As a result, this experiment involves a total of 504 potential combinations in each loop, arising from the interaction between 63 SFs (32 + 16 + 8 + 4 + 2 + 1) and 8 channels.

The broadcasting test adheres to the value generation methodology of SF and Channel number 1.2, which states "Values are statically predetermined". It utilizes the selection strategy of SF and channel number 4, described as "Weighted Random Selection Without Repetition". Regarding the frequency of the tests, it operates in a "Continuous-mode polling" manner. All of these methods are detailed in the preceding section.

As shown in Figure 9, now the configurations with low SF (7, 8) greatly increased the detection ratio. This was one of the objectives of this experiment, and it was achieved. Obviously it is due to the increase in sampling of these SFs, and this improvement is consequently seen. However, now transmissions with high SF (11, 12) greatly decreased the detection rate and are penalized by this sampling policy. Therefore, these types of policies are appropriate in environments where there are transmissions mainly with low SF.

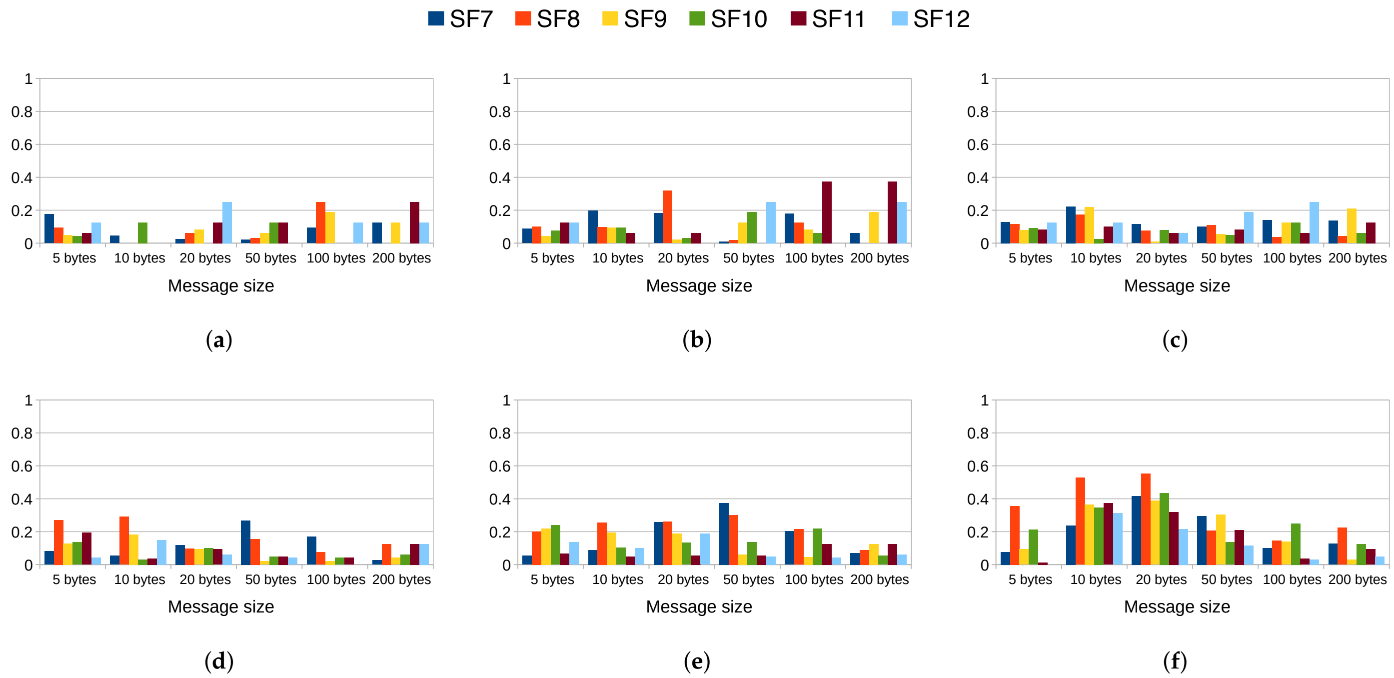

Figure 9. Reception rate of weighted random selection for different duty cycles, message sizes, and SFs. (**a**) 1% Duty Cycle. (**b**) 2% Duty Cycle. (**c**) 5% Duty Cycle. (**d**) 10% Duty Cycle. (**e**) 20% Duty Cycle. (**f**) 50% Duty Cycle.

6.4. Experiment 3: All Channel and SF Combinations. All Values. Past History. Weighted Random Selection with Repeatition of Last Values

This third experiment is based on simple heuristics that take into account the behavioral history of the sending nodes. The simplest mechanism would be that when the receiver detects a transmission with certain configuration parameters, it repeats that same configuration for a certain time. This benefits those environments where a transmitter does not change the configuration parameters at all or little and has a high transmission frequency.

The broadcasting test adheres to the value generation methodology of SF and Channel number 1.2, which states "Values are statically predetermined". It utilizes the selection strategy of SF and channel number 3, described as "Wighted Random Selection with Repetition". And it also applies the simplest version of heuristic 4.1 based on historical patterns that repeats the last value. Regarding the frequency of the tests, it operates in a "Continuous-mode polling" manner. All of these methods are detailed in the preceding section.

This experiment uses the same emission and reception pattern as in the previous experiment, ultimately prioritizing low SF configurations. But the important difference is that after correctly detecting and reading a message, it continues to test the same configuration throughout the sampling cycle; in our case, it tries the same configuration 504 more times. If it detects a message again, the cycle begins again. If during this repetition cycle it does not detect a new transmission, sampling will continue in the same way as before with the next random weighted combination.

It is evident from Figure 10 that this change allows frequent transmissions with a duty cycle of 10% to 50% to achieve a very high detection rate. Once the sampling of the most weighted configurations detects the first message, they continue to detect it until the transmission parameters are changed.

However, for those transmissions with a low duty cycle periodicity of 1% to 5%, the time distance of the transmissions is greater than the repetition cycle time and therefore does not detect a new transmission with those parameters. Since time was wasted in this repetition, no progress was made, and no other emissions were detected with other configurations, and therefore the detection rate decreased. Thus, it can be deduced that this simple mechanism is efficient only when the frequency of transmissions is high.

From these three experiments, it is shown that the probability of detection of a message in a possible transmission configuration depends mainly on the time that its preamble is in the physical medium and the frequency with which the receiver tests that configuration. Thus, the success of the detection depends on the hand-sacking synchronization between sender and receiver.

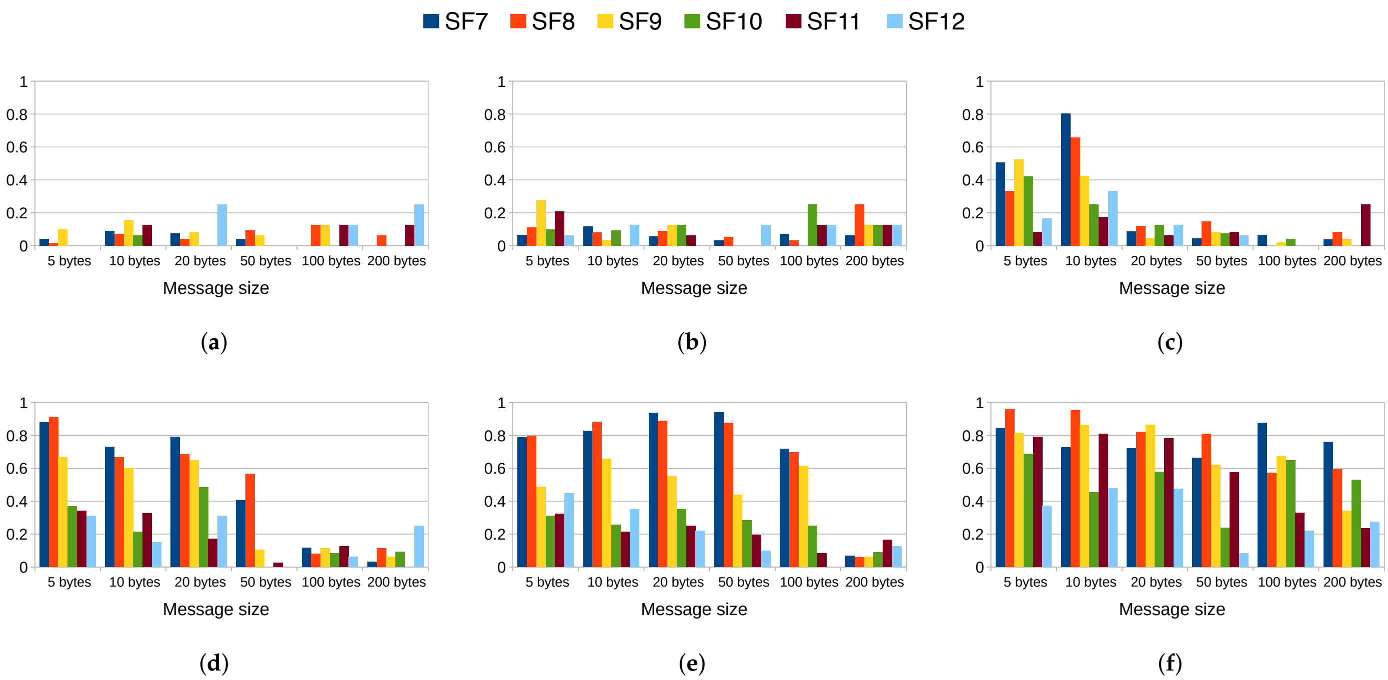

Figure 10. Reception rate of weighted random selection with Last Value repetition for different duty cycles, message sizes, and SFs. (**a**) 1% Duty Cycle. (**b**) 2% Duty Cycle. (**c**) 5% Duty Cycle. (**d**) 10% Duty Cycle. (**e**) 20% Duty Cycle. (**f**) 50% Duty Cycle.

In the first experiment, transmissions with high SF (11, 12) were detected with higher probability than those with low SF (7, 8). This is because all configurations were tested with equal probability, but high SF values occupy the physical medium for longer periods.

Thus, in the second experiment, by changing the test frequency and prioritizing transmissions with low SF over those with high SF, the probability of reception of those with low SF compared to those with high SF was drastically increased. Therefore, transmissions with low SF (7, 8) are now the ones most likely to be detected.

And finally, in the third experiment, priority was given to detecting emissions that do not frequently change their configuration parameters and have a high sending frequency.

From these results, we see that future work should focus on improving the message detection and reception ratio. These improvements can be made from different points of view. On the one hand, results could be improved by fine-tuning the detection frequency and pattern repetition. The search field of the configurations could also be restricted to reduce the scanning time. Finally, techniques for learning the behavior of emitting nodes could also be applied. In any case, the search would be directed to a subset of values, increasing the probability of success in the emission parameters.

7. Conclusions and Future Work

Based on the experimental results obtained, we summarize our key contributions and outline directions for future work. The research explores the potential performance of LoRa networks and introduces a method to enhance their efficiency. The study involves an in-depth performance analysis of LoRa and proposes a novel approach to SF and channel selection at the transmitter using various heuristics and policies, as well as SF and channel detection at the receiver. The findings indicate that LoRa networks can operate efficiently without traditional gateways, improving network architecture, reducing latency, and potentially lowering implementation costs.

First, we conducted a comprehensive analysis of various aspects that influence the performance and efficiency of LoRa communication systems. Then, we categorized the heuristics for optimizing the SF and CH values, providing flexibility in the selection processes. Subsequently, we examined various LoRa parameters including message size, preamble size, transmission power, SF, channel, and bandwidth. We highlighted the importance of balancing these factors to maximize performance metrics such as data rate, RSSI, and SNR, while minimizing energy consumption. In addition, we explored the impact of preamble length and message size on successful message decoding, emphasizing the need for a careful balance between the two. Furthermore, we evaluated the time implications of modifying LoRa parameters, noting SF modification as the most time-consuming aspect but highlighting the overall efficiency of parameter modification compared to preamble transmission. These findings underscore the importance of optimizing LoRa system configurations and parameter modification processes to improve performance, energy efficiency, and overall effectiveness in various communication contexts.

Second, our research presents a groundbreaking strategy aimed at improving the efficiency and reliability of LoRa network communications. By introducing a decentralized methodology within the LoRa layer itself, our approach enables adaptable communication configurations to address evolving network conditions. Through the simultaneous utilization of various channels and SFs on client chips, we effectively reduced node-to-node contention, ensuring seamless and collision-free operations. Unlike existing methodologies, our approach includes both SF adjustments and channel modifications, facilitating comprehensive management of communication parameters. Our proposal enables dynamic selection and detection of spreading factors and channels, crucial to achieving effective and reliable communication in diverse environments and scenarios. Using a two-step mechanism that combines heuristics for value generation and selection policies, our method ensures flexibility and responsiveness to changing network conditions. This research significantly advances the field of LoRa communications, offering a practical and efficient solution to optimize network performance and adaptability.

It is important to highlight that the primary goal of this study was not to maximize performance compared to existing methods, but rather to validate that the proposed mechanism can operate correctly in a decentralized and uncoordinated environment. In addition, we evaluated the effectiveness of our proposed mechanism through a series of experiments employing simple heuristics and policies. The results of these experiments are highly promising and demonstrate significant improvements in network performance and adaptability. By dynamically adjusting SFs and CHs based on changing network conditions, our mechanism effectively mitigates interference, improves data transmission reliability, and conserves energy. These findings underscore the practical applicability and efficacy of our approach in real-world LoRa network deployments. Furthermore, the simplicity and scalability of our heuristics and policies make our mechanism easily implementable across a wide range of LoRa network configurations, promising widespread adoption and impact in the field.

This work presented and validated the feasibility of a complete decentralized mechanism for SF and CH selection and detection in LoRa networks, evaluated in a controlled environment. Our objective was to define a global, infrastructure-free framework capable of adapting to changing network conditions, covering both transmission and reception. The controlled setup allowed us to confirm the correctness, efficiency, and responsiveness of the mechanism in isolation. Although large-scale tests are beyond the scope of this paper, the framework was designed to be modular and scalable by construction, and its scalability under high-density or interference-prone environments remains to be validated.

Future work will focus on evaluating the proposed mechanism in larger deployments with multiple nodes, where synchronization, contention resolution, and collision management become more complex due to increased traffic and heterogeneous node configurations. We also plan to explore more advanced heuristics and selection policies, potentially incorporating machine learning techniques to enhance adaptability to dynamic environments and to enable predictive adjustments based on historical communication patterns. In addition, we aim to assess the impact of external factors—such as weather conditions, terrain variation, and network load—on system performance, considering both short-term fluctuations and long-term operational stability. Furthermore, we plan to benchmark the proposed decentralized SF/CH optimization method across diverse environments, including rural areas with long-range links and sparse device distribution, smart city scenarios with high device density and significant interference from urban infrastructure, and industrial settings characterized by challenging electromagnetic conditions and physical obstructions. These extended evaluations will not only help validate the robustness, scalability, and adaptability of our approach in realistic IoT scenarios but also provide actionable insights for tailoring deployment strategies to specific application domains.

Author Contributions: Conceptualization, R.P.C., R.M., C.M., P.M. and C.A.; methodology, C.M. and C.A.; software, C.A.; validation, C.A.; investigation, C.A., C.M. and R.P.C.; modelling, R.M. and C.M.; resources, C.A.; field data collection, C.A.; data curation, C.A.; writing—original draft preparation, C.A. and P.M.; writing—review and editing, P.M., R.M. and C.M.; supervision, R.M., P.M. and C.M.; project administration, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially funded by the EU Horizon program under grant agreements No. 833057, No. 825184 and No. 101092644. It has also received support from the Ministry of Economic Affairs and Digital Transformation, together with the European Union-NextGenerationEU (within the PRTR and MRR frameworks), through the CLOUDLESS UNICO I+D CLOUD 2022. Additional funding was provided by the Government of Spain under contracts PID2019-106774RB-C21,

PID2019-106774RB-C22, PID2020-113172RB-I00 and PID2024-155476OB-I00, and by the Government of Catalonia as Consolidated Research Groups 2021-SGR-383, 2021-SGR-109 and 2017-SGR-990.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors gratefully acknowledge the financial support received from the European Union, the Government of Spain, and the Government of Catalonia through various research and innovation programs.

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. Pueyo Centelles, R.; Freitag, F.; Meseguer, R.; Navarro, L. Beyond the Star of Stars: An Introduction to Multihop and Mesh for LoRa and LoRaWAN. *IEEE Pervasive Comput.* **2021**, *20*, 63–72. [CrossRef]
2. Goursaud, C.; Gorce, J.M. Dedicated networks for IoT: PHY/MAC state of the art and challenges. *EAI Endorsed Trans. Internet Things* **2015**, *1*, e3. [CrossRef]
3. Shanmuga Sundaram, J.P.; Du, W.; Zhao, Z. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 371–388. [CrossRef]
4. Farhad, A.; Kim, D.H.; Kim, B.H.; Mohammed, A.F.Y.; Pyun, J.Y. Mobility-Aware Resource Assignment to IoT Applications in Long-Range Wide Area Networks. *IEEE Access* **2020**, *8*, 186111–186124. [CrossRef]
5. Liando, J.C.; Gamage, A.; Tengourtius, A.W.; Li, M. Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study. *ACM Trans. Sen. Netw.* **2019**, *15*, 1–35. [CrossRef]
6. Heeger, D.; Garigan, M.; Tsiropoulou, E.E.; Plusquellic, J. Secure LoRa Firmware Update with Adaptive Data Rate Techniques. *Sensors* **2021**, *21*, 2384. [CrossRef]
7. Paredes, M.; Bertoldo, S.; Carosso, L.; Lucianaz, C.; Marchetta, E.; Allegretti, M.; Savi, P. Propagation measurements for a LoRa network in an urban environment. *J. Electromagn. Waves Appl.* **2019**, *33*, 2022–2036. [CrossRef]
8. Elselini, A.S.; Eleribi, H.R.; Sanaani, M.; Alwerfalli, A. A Performance Study of an IoT System Using LoRa Access Network Technology. In Proceedings of the 6th International Conference on Engineering & MIS 2020, ICEMIS'20, Almaty, Kazakhstan, 14–16 September 2020; Association for Computing Machinery: New York, NY, USA, 2020. [CrossRef]
9. Pietrosemoli, E.; Rainone, M.; Zennaro, M. On Extending the Wireless Communications Range of Weather Stations using LoRaWAN. In Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good, GoodTechs '19, Valencia, Spain, 25–27 September 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 78–83. [CrossRef]
10. Dayana, A.A.; Karthika, P.; Kavitha, R.; Priyanka, M.P.; Boomadevi, D.P. Smart Electric Meter using LoRA Protocols. *Int. J. Eng. Res. Technol.* **2019**, *7*. Available online: https://www.ijert.org/smart-electric-meter-using-lora-protocols (accessed on 19 August 2025).
11. Nakamura, K.; Manzoni, P.; Zennaro, M.; Cano, J.C.; Calafate, C.T.; Cecilia, J.M. FUDGE: A frugal edge node for advanced IoT solutions in contexts with limited resources. In Proceedings of the 1st Workshop on Experiences with the Design and Implementation of Frugal Smart Objects, FRUGALTHINGS'20, London, UK, 21 September 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 30–35. [CrossRef]
12. Haxhibeqiri, J.; Van den Abeele, F.; Moerman, I.; Hoebeke, J. LoRa Scalability: A Simulation Model Based on Interference Measurements. *Sensors* **2017**, *17*, 1193. [CrossRef]
13. Jouhari, M.; Saeed, N.; Alouini, M.S.; Amhoud, E.M. A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1841–1876. [CrossRef]
14. Van den Abeele, F.; Haxhibeqiri, J.; Moerman, I.; Hoebeke, J. Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3. *IEEE Internet Things J.* **2017**, *4*, 2186–2198. [CrossRef]
15. Georgiou, O.; Raza, U. Low Power Wide Area Network Analysis: Can LoRa Scale? *IEEE Wirel. Commun. Lett.* **2017**, *6*, 162–165. [CrossRef]
16. Haif, H.; Arous, A.; Arslan, H. A Decentralized Dynamic Relaying-Based Framework for Enhancing LoRa Networks Performance. *IEEE Internet Things J.* **2024**, *11*, 22183–22196. [CrossRef]
17. Liao, W.S.; Zhao, O.; Ishizu, K.; Kojima, F. Adaptive Parameter Adjustment for Uplink Transmission for Multi-gateway LoRa Systems. In Proceedings of the 2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC), Lisbon, Portugal, 24–27 November 2019; pp. 1–5. [CrossRef]

18. Alahmadi, H.; Bouabdallah, F.; Al-Dubai, A.; Ghaleb, B. A Novel Autonomous Adaptive Frame Size for Time-Slotted LoRa MAC Protocol. *IEEE Trans. Ind. Inform.* **2024**, *20*, 12284–12293. [CrossRef]
19. Zhu, G.; Liao, C.H.; Sakdejayont, T.; Lai, I.W.; Narusue, Y.; Morikawa, H. Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering. *IEEE Access* **2019**, *7*, 21584–21596. [CrossRef]
20. Gkotsiopoulos, P.; Zorbas, D.; Douligeris, C. Performance Determinants in LoRa Networks: A Literature Review. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1721–1758. [CrossRef]
21. Bor, M.; Roedig, U. LoRa Transmission Parameter Selection. In Proceedings of the 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 5–7 June 2017; pp. 27–34. [CrossRef]
22. Li, C.; Cao, Z. LoRa Networking Techniques for Large-scale and Long-term IoT: A Down-to-top Survey. *ACM Comput. Surv.* **2022**, *55*, 1–36. [CrossRef]
23. Dix-Matthews, B.; Cardell-Oliver, R.; Hübner, C. LoRa Parameter Choice for Minimal Energy Usage. In Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks, RealWSN'18, Shenzhen, China, 4 November 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 37–42. [CrossRef]
24. Pirayesh, H.; Zhang, S.; Sangdeh, P.K.; Zeng, H. MaLoRaGW: Multi-user MIMO transmission for LoRa. In Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, SenSys '22, Boston, MA, USA, 6–9 November 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 179–192. [CrossRef]
25. Kim, S.; Lee, H.; Jeon, S. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. *Sensors* **2020**, *20*, 1008. [CrossRef]
26. Zorbas, D.; Papadopoulos, G.Z.; Maille, P.; Montavont, N.; Douligeris, C. Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations. In Proceedings of the 2018 25th International Conference on Telecommunications (ICT), Saint-Malo, France, 26–28 June 2018; pp. 516–520. [CrossRef]
27. Reynders, B.; Meert, W.; Pollin, S. Power and spreading factor control in low power wide area networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]
28. Shayo, E.I.; Abdalla, A.T.; Mwambela, A.J. Dynamic multi-frame multi-spreading factor scheduling algorithm for LoRaWAN. *J. Electr. Syst. Inf. Technol.* **2023**, *10*, 11. [CrossRef]
29. Kang, J.M. LoRa Preamble Detection with Optimized Thresholds. *IEEE Internet Things J.* **2023**, *10*, 6525–6526. [CrossRef]
30. Tian, P.; Boano, C.A.; Schuß, M.; Wei, J. RSSF: Towards Real-Time Decoding of LoRa Packets without Prior Knowledge of their Spreading Factor. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Kuala Lumpur, Malaysia, 4–8 December 2023. [CrossRef]
31. Sanchez-Iborra, R.; Sanchez-Gomez, J.; Ballesta-Viñas, J.; Cano, M.D.; Skarmeta, A.F. Performance Evaluation of LoRa Considering Scenario Conditions. *Sensors* **2018**, *18*, 772. [CrossRef]
32. Sisinni, E.; Fernandes Carvalho, D.; Depari, A.; Bellagente, P.; Flammini, A.; Pasetti, M.; Rinaldi, S.; Ferrari, P. Assessing a Methodology for Evaluating the Latency of IPv6 with SCHC Compression in LoRaWAN Deployments. *Sensors* **2023**, *23*, 2407. [CrossRef]
33. Fragkopoulos, M.; Panagiotakis, S.; Kostakis, M.; Markakis, E.K.; Astyrakakis, N.; Malamos, A. Experimental Assessment of Common Crucial Factors That Affect LoRaWAN Performance on Suburban and Rural Area Deployments. *Sensors* **2023**, *23*, 1316. [CrossRef]
34. The Things Network. Adaptive Data Rate. 2022. Available online: https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/ (accessed on 25 April 2025).
35. Kim, D.Y.; Kim, S.; Hassan, H.; Park, J.H. Adaptive data rate control in low power wide area networks for long range IoT services. *J. Comput. Sci.* **2017**, *22*, 171–178. [CrossRef]
36. Babaki, J.; Rasti, M.; Aslani, R. Dynamic Spreading Factor and Power Allocation of LoRa Networks for Dense IoT Deployments. In Proceedings of the 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, 31 August–3 September 2020; pp. 1–6. [CrossRef]
37. Cuomo, F.; Campo, M.; Caponi, A.; Bianchi, G.; Rossini, G.; Pisani, P. EXPLoRa: Extending the Performance of LoRa by Suitable Spreading Factor Allocations. In Proceedings of the 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, Italy, 9–11 October 2017; pp. 1–8. [CrossRef]
38. Semtech Corporation. LoRaWAN Mobile Applications: Blind ADR. 2019. Available online: https://lora-developers.semtech.com/uploads/documents/files/LoRaWAN_Mobile_Apps-Blin_ADR_Downloadable.pdf (accessed on 3 July 2023).
39. Anwar, K.; Rahman, T.; Zeb, A.; Khan, I.; Zareei, M.; Vargas-Rosales, C. RM-ADR: Resource Management Adaptive Data Rate for Mobile Application in LoRaWAN. *Sensors* **2021**, *21*, 7980. [CrossRef] [PubMed]
40. Lehong, C.; Isong, B.; Lugayizi, F.; Abu-Mahfouz, A.M. A Survey of LoRaWAN Adaptive Data Rate Algorithms for Possible Optimization. In Proceedings of the 2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Kimberley, South Africa, 25–27 November 2020; pp. 1–9. [CrossRef]

41. Koch, D.J.; Shahid, M.O.; Krishnaswamy, B. Spreading Factor Detection for low-cost Adaptive Data Rate in LoRaWAN Gateways. In Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, SenSys' 2022, Boston, MA, USA, 6–9 November 2022; Association for Computing Machinery: New York, NY, USA, 2023; pp. 918–924. [CrossRef]
42. Minhaj, S.U.; Mahmood, A.; Abedin, S.F.; Hassan, S.A.; Bhatti, M.T.; Ali, S.H.; Gidlund, M. Intelligent Resource Allocation in LoRaWAN using Machine Learning Techniques. *IEEE Access* **2023**, *11*, 10092–10106. [CrossRef]
43. Chen, M.; Mokdad, L.; Ben-Othman, J.; Fourneau, J.M. Dynamic Parameter Allocation with Reinforcement Learning for LoRaWAN. *IEEE Internet Things J.* **2023**, *10*, 10250–10265. [CrossRef]
44. Hong, S.; Yao, F.; Zhang, F.; Ding, Y.; Yang, S.H. Reinforcement Learning Approach for SF Allocation in LoRa Network. *IEEE Internet Things J.* **2023**, *10*, 18259–18272. [CrossRef]
45. Scarvaglieri, A.; Palazzo, S.; Busacca, F. A lightweight, fully-distributed AI framework for energy-efficient resource allocation in LoRa networks. In Proceedings of the 16th IEEE/ACM International Conference on Utility and Cloud Computing (UCC '23), Taormina, Italy, 4–7 December 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 37–42. [CrossRef]
46. Ochoa, M.N.; Maman, M.; Duda, A. Spreading Factor Allocation for LoRa Nodes Progressively Joining a Multi-Gateway Adaptive Network. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
47. Ivoghlian, A.; Wang, K.I.K.; Salcic, Z. Application-Aware Adaptive Parameter Control for LoRaWAN. *J. Parallel Distrib. Comput.* **2022**, *166*, 166–177. [CrossRef]
48. Marais, J.M.; Abu-Mahfouz, A.M.; Hancke, G.P. Improving the Sustainability of Confirmed Traffic in LoRaWANs Through an Adaptive Congestion Scheme. *IEEE Sens. J.* **2023**, *23*, 1660–1670. [CrossRef]
49. Leonardi, L.; Lo Bello, L.; Patti, G. MRT-LoRa: A multi-hop real-time communication protocol for industrial IoT applications over LoRa networks. *Comput. Commun.* **2023**, *199*, 72–86. [CrossRef]
50. Manzoni, P.; Merzougui, S.E.; Palazzi, C.E.; Pozzan, P. A Resilient LoRa-Based Solution to Support Pervasive Sensing. *Electronics* **2023**, *12*, 2952. [CrossRef]
51. Nowak, M.; Różycki, R.; Waligóra, G.; Szewczyk, J.; Sobiesierski, A.; Sot, G. Data Processing with Predictions in LoRaWAN. *Energies* **2023**, *16*, 411. [CrossRef]
52. 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]
53. Singh, R.K.; Berkvens, R.; Weyn, M. Synchronization and efficient channel hopping for power efficiency in LoRa networks: A comprehensive study. *Internet Things* **2020**, *11*, 100233. [CrossRef]
54. Figueiredo, A.; Luís, M.; Zúquete, A. A Fair Channel Hopping Scheme for LoRa Networks with Multiple Single-Channel Gateways. *Sensors* **2022**, *22*, 5260. [CrossRef]
55. Leenders, G.; Callebaut, G.; Ottoy, G.; Van der Perre, L.; De Strycker, L. An Energy-Efficient LoRa Multi-Hop Protocol through Preamble Sampling for Remote Sensing. *Sensors* **2023**, *23*, 4994. [CrossRef]
56. Prade, L.; Moraes, J.; de Albuquerque, E.; Rosário, D.; Both, C.B. Multi-radio and multi-hop LoRa communication architecture for large scale IoT deployment. *Comput. Electr. Eng.* **2022**, *102*, 108242. [CrossRef]
57. Eric B. LoRa Documentation. 2018. Available online: https://lora.readthedocs.io/ (accessed on 22 April 2025).
58. Semtech Corporation. Errata Note. SX1276/77/78–137 to 1020 MHz Low Power Long Range Transceiver. 2013. Available online: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000HSPv/sqi9xX0gs6hgzl2LoPwCK0TS9GDPlMwsXmcNzJCMHjw (accessed on 19 August 2025).