

RESEARCH ARTICLE

FAST-LoRa: An Efficient Simulation Framework for Evaluating LoRaWAN Networks and Transmission Parameter Strategies

LAURA ACOSTA-GARCÍA¹, JUAN AZNAR-POVEDA², FABIAN MARGREITER²,
ANTONIO-JAVIER GARCÍA-SÁNCHEZ¹, JOAN GARCÍA-HARO¹, (Member, IEEE),
THOMAS FAHRINGER², (Member, IEEE), JOSÉ LORENTE LÓPEZ³,
AND JOSÉ-VÍCTOR RODRÍGUEZ³

¹Telematics Engineering Group, Technical University of Cartagena, 30202 Cartagena, Spain

²Distributed and Parallel Systems Group, University of Innsbruck, 6020 Innsbruck, Austria

³Mobile Communications Systems Group, Technical University of Cartagena, 30202 Cartagena, Spain

Corresponding author: Juan Aznar-Poveda (juan.aznar-poveda@uibk.ac.at)

This work was supported in part by Ministerio de Ciencia e Innovación (MCIN)/Agencia Estatal de Investigación (AEI)/10.13039/501100011033 and European Union NextGeneration EU/Plan de Recuperación, Transformación y Resiliencia (PRTR) under Grant TED2021-129336B-I00, in part by Ministerio de Ciencia, Innovación y Universidades (MICIU)/AEI/10.13039/501100011033 and Fondo Europeo de Desarrollo Regional (FEDER)/EU under Grant PID2023-148214OB-C21, and in part by MICIU/AEI/10.13039/501100011033 and European Union under Grant PCI2024-153485.

ABSTRACT The Internet of Things (IoT) has transformed many industries, and LoRaWAN (Long Range Wide Area Network), built on LoRa (Long Range) technology, has become a crucial solution for enabling scalable, low-cost, and energy-efficient communication in wide-area networks. Simulation tools are essential for optimizing the transmission parameters and, therefore, the energy efficiency and performance of LoRaWAN networks, where energy efficiency is widely recognized as a primary performance objective in distributed and parallel systems. However, although existing simulation frameworks can accurately represent LoRaWAN behavior by modeling multiple protocol layers or by focusing on specific network aspects, they often suffer from high computational cost and limited flexibility when fast and systematic exploration of transmission parameter configurations is required. To address this issue, we introduce FAST-LoRa, a novel simulation framework designed to enable fast and efficient evaluation of LoRaWAN networks and selection of transmission parameters. FAST-LoRa streamlines computation by relying on analytical models without complex packet-level simulations and implementing gateway reception using efficient matrix operations. We evaluate FAST-LoRa by employing a well-established simulator using multiple network configurations with varying numbers of end devices and gateways. The results show that FAST-LoRa achieves similar accuracy in estimating key network metrics, even in complex scenarios with interference and multi-gateway reception, with a Mean Absolute Error (MAE) of 0.940×10^{-2} for the Packet Delivery Ratio (PDR) and 0.040 bits/mJ for Energy Efficiency (EE), while significantly reducing computational time by up to three orders of magnitude.

INDEX TERMS Internet of Things, long-range communications, wireless communications, simulation, transmission parameters, energy efficiency.

I. INTRODUCTION

The Internet of Things (IoT) has transformed how devices communicate, enabling advancements in sectors such as

The associate editor coordinating the review of this manuscript and approving it for publication was Angelo Trotta¹.

healthcare, agriculture, and smart cities. One of the technologies driving this change is LoRa (Long Range), a low-power wide-area network (LPWAN) that allows for long-range communication with minimal energy consumption [28], [54]. LoRaWAN, the protocol built on LoRa, further enhances this capability, making it well-suited for

battery-powered IoT devices. Due to its scalability and cost-effectiveness, LoRaWAN is becoming an essential part of modern IoT systems, supporting a wide range of applications from environmental monitoring to industrial automation.

An essential mechanism for optimizing LoRaWAN network performance is the Adaptive Data Rate (ADR). The ADR adjusts the transmission parameters of devices based on network conditions, such as signal strength, interference, and network load. By dynamically modifying transmission power and the spreading factor, the ADR ensures that devices maintain efficient communication while balancing critical factors such as range, data rate, and energy consumption. As LoRaWAN networks expand and become more complex, optimizing the ADR becomes more challenging. However, it remains crucial for maintaining appropriate network performance and energy consumption [9], [16], [17], [24], [33], [41], [42], [43], [58]. Optimizing LoRaWAN networks in real-world environments often requires significant financial investments in hardware, such as gateways and end devices, and infrastructure, including installation and maintenance [53]. Additionally, the inherent variability of environmental conditions, interference, and the spatial distribution of nodes complicates a comprehensive evaluation of network performance.

In this context, simulation tools offer a cost-effective solution for testing and optimizing LoRaWAN network configurations without the need for extensive real-world deployments. These tools help evaluate various network configurations and data transmission strategies. Several simulation tools have been developed to support LoRaWAN networks by modeling aspects such as network topology, protocol layers, signal propagation, interference, mobility, traffic patterns, and energy consumption [8], [18], [19], [37], [38], [40], [50], [51], [52], [55]. While these simulators play a key role in protocol validation and detailed performance analysis, their applicability to transmission parameter selection is often limited. Many tools either rely on high-fidelity packet-level modeling, which entails significant computational overhead in large or dense networks, or are tailored to specific aspects or scenarios, restricting their flexibility for systematic and iterative exploration of transmission parameter configurations and ADR mechanisms.

To address this limitation, we introduce FAST-LoRa, a novel simulation framework designed to enable efficient and scalable evaluation of transmission parameters for their optimization in LoRaWAN networks. FAST-LoRa streamlines the simulation process by optimizing core transmission parameters, such as the transmission rate, power, and spreading factor. This approach permits rapid and iterative testing of ADR settings and network configurations without the computational burden associated with simulating complex network behaviors that are not directly relevant to transmission optimization. To achieve this efficiency, FAST-LoRa simplifies the simulation process by abstracting packet-level behavior

using analytical models and computing gateway receptions through matrix-based operations. While FAST-LoRa is not designed to replace detailed packet-level discrete-event simulators, it provides a lightweight and accurate alternative for evaluating transmission strategies in scenarios where traffic patterns are stable, and communication is predominantly uplink-focused. FAST-LoRa enables the simulation of large-scale LoRaWAN networks with numerous devices and gateways, making it ideal for fast evaluations of transmission parameter selection strategies and network layouts, e.g., planning of gateway placement. Beyond efficient simulation, FAST-LoRa enables fast and large-scale evaluation of transmission parameter configurations and ADR strategies. This allows systematic exploration of design choices that would be computationally prohibitive with packet-level simulators, supporting the assessment of advanced mechanisms such as adaptive transmission strategies, dynamic parameter tuning, and network layout planning (e.g., gateway placement), with direct impact on key network performance metrics, such as packet delivery ratio, throughput, and energy efficiency. The contributions of this paper are summarized as follows:

- We introduce FAST-LoRa, a novel open-source simulation framework that enables fast, scalable, and flexible evaluation of LoRaWAN networks, specifically targeting transmission parameter selection and ADR optimization in large-scale deployments.
- Unlike existing packet-level simulators, FAST-LoRa leverages analytical models and matrix-based computations to efficiently capture interference, multi-gateway reception, and propagation effects, enabling rapid evaluation of complex network scenarios without incurring high computational overhead.
- FAST-LoRa accurately estimates critical network performance metrics, including Packet Delivery Ratio (PDR) and Energy Efficiency (EE), providing actionable insights for dynamic transmission parameter tuning, ADR strategies, and network layout planning.
- We validate FAST-LoRa against the well-established FloRa simulator across a wide range of network configurations. Results demonstrate that FAST-LoRa achieves comparable accuracy while reducing execution time by up to three orders of magnitude in dense multi-gateway networks. This highlights FAST-LoRa's ability to support rapid, systematic, and large-scale exploration of transmission strategies.

The rest of this paper is organized as follows. Section II reviews related work. Section III presents the analytical model of FAST-LoRa, including network description, transmission time, reception criteria, and implementation details. Section IV details the evaluation setup, metrics, and baselines. In Section V, we validate FAST-LoRa against a well-established simulator. Finally, Section VI summarizes the key findings and outlines future work.

II. RELATED WORK

In this section, we review the state of the art in selecting transmission parameters in LoRaWAN networks and how this is achieved by commonly used LoRaWAN simulators.

A. TRANSMISSION PARAMETER SELECTION

Numerous research efforts have been made to select transmission parameters in LoRaWAN networks [9], [10], [11], [16], [17], [24], [33], [41], [42], [43], [58]. For example, the authors of [16] introduced RM-ADR, a novel mechanism that operates on the network server and end devices. The end device algorithm reduces retransmission attempts, while the network server leverages previous frame reception power to dynamically select the spreading factor (f) and transmission parameters. Other techniques, such as Gaussian-based filtering (G-ADR) and the Exponential Moving Average (EMA-ADR), have been explored to smooth out rapid Signal-to-Noise Ratio (SNR) fluctuations [24], [25]. Many of these works leverage artificial intelligence to dynamically learn new transmission policies over time. For instance, [58] employs reinforcement learning to select energy efficiency and ensure reliable packet delivery. The work in [43] proposes an intelligent routing policy to find suitable transmission parameters and move beyond the traditional star topology to improve network performance. In [9], the authors employ Deep Reinforcement Learning (DRL) to determine the optimal spreading factor and transmission power combination, reducing energy consumption while maintaining an adaptive Bit Error Rate (BER) threshold. Similarly, [11], [41] uses DRL to improve energy efficiency by dynamically adjusting transmission settings. However, the vast majority of these works either propose their custom simulator [9], [10], [11], [33], [58], which often lack rigorous validation, or employ well-known simulators, which often imply notable computational overhead and hamper the selection process.

B. STATE-OF-THE-ART LORAWAN SIMULATORS

Simulation frameworks are frequently employed to find suitable transmission parameters and energy consumption in LoRa networks. One of the most widely used simulation tools for LoRa networks is LoRaSim [18], [19], a simulation tool that focuses on key aspects such as collisions, spreading factors, and gateway coverage. Recent improvements to LoRaSim, such as those proposed in [26], address some of its limitations by incorporating more realistic propagation models and the capture-effect phenomenon. These enhancements help improve the accuracy of network behavior simulations, particularly in dense environments. LoRaWANSIM [45], an extension of LoRaSim, adds features such as bidirectional communication and medium access control (MAC) layer improvements. However, although LoRaSim and LoRaWANSIM are effective for evaluating ADR mechanisms, their focus on the physical and MAC layers restricts their ability to fully capture application-layer dynamics. References [51],

[52], built on the OMNeT++ simulation framework [35] and the INET framework [36], provides a more comprehensive simulation environment, offering detailed models for communication delays, energy consumption, and network dynamics. Its modular architecture allows for the simulation of multiple layers of the LoRaWAN protocol stack—including the physical, MAC, network, and application layers—enabling cross-layer analysis and the integration of custom ADR mechanisms. Similarly, the ns-3 simulator [44], an open-source platform that supports a wide range of wireless communication technologies, including LoRaWAN, offers extensive flexibility for simulating complex, multi-technology scenarios [37], [38]. Its ability to model physical and higher-layer interactions makes it a powerful tool for evaluating Adaptive Data Rate (ADR) mechanisms in integrated environments. However, both FloRa and ns-3 face similar limitations due to the significant computational resources required for simulating individual transmissions. This makes them impractical for tasks requiring rapid iterations, such as reinforcement learning training [9], [33], [43], or for evaluating critical network performance metrics like energy efficiency, communication reliability, and the packet delivery rate.

LWN-Simulator [8] provides a LoRaWAN node simulator with a web interface that interacts with real or simulated network infrastructures, such as ChirpStack. It supports multiple device classes, MAC-layer procedures, and ADR mechanisms, making it suitable for protocol-level validation and functional experimentation. However, its reliance on a full LoRaWAN backend and interactive execution limits its suitability for fast, large-scale evaluation of transmission parameter configurations. Similarly, LoRa-MAB [55] employs Multi-Armed Bandit (MAB) algorithms for decentralized resource allocation, enabling end devices to make autonomous decisions. However, its design is tightly focused on the evaluation of specific learning mechanisms and resource allocation strategies, rather than on fast and systematic exploration of transmission parameter configurations or large-scale analytical performance evaluation. Other tools focus on very specific aspects of LoRaWAN performance. For instance, the MATLAB-based simulator presented in [20] evaluates learning-based spreading factor selection under realistic physical-layer conditions. Although it provides valuable insights for this specific problem, it is tailored to a particular scenario and is not intended for fast, large-scale evaluation of transmission parameter configurations or overall network performance. Similarly, the LoRa/Sigfox simulator presented in [4] focuses on physical layer characteristics such as coverage, transmission power constraints, and propagation effects in localized environments, but does not address network-level behavior or support scalable analysis of transmission strategies. Overall, existing LoRaWAN simulators span a wide spectrum, ranging from detailed packet-level frameworks and protocol-oriented tools to simulators tailored to specific mechanisms or layers of the stack. While these approaches provide valuable

insights for their respective purposes, they either incur high computational cost or lack the flexibility required for fast, large-scale exploration of transmission parameter configurations and joint evaluation of reliability, interference, and energy efficiency. FAST-LoRa is designed to fill this gap by offering a lightweight analytical framework that enables efficient and scalable evaluation of transmission parameters in LoRaWAN networks.

III. FAST-LoRa SIMULATION FRAMEWORK

FAST-LoRa is a simulation framework designed to provide fast LoRaWAN network evaluations and find suitable transmission parameters efficiently. Unlike traditional simulators that require high computational power to consider numerous aspects and multiple layers of communication protocols, FAST-LoRa adopts a statistical approach [23], [60] that efficiently obtains network performance metrics, such as the Packet Delivery Ratio (PDR) and Energy Efficiency (EE). By exclusively focusing on the essential aspects of transmission parameter selection, our approach enables significantly faster simulations, making it particularly suitable for adaptive transmission strategies, dynamic parameter tuning, and network layout planning. Furthermore, mechanisms that inherently require a large number of iterations—such as multi-objective auto-tuning approaches [29] or machine-learning techniques—would also benefit from the reduced computational complexity and accelerated convergence afforded by our methodology. This efficiency is due to focusing on transmission-related metrics, replacing packet-level interactions with analytical models. For instance, FAST-LoRa supports a broad range of MAC-layer ADR mechanisms that operate by adjusting the currently supported transmission parameters and performance metrics. Due to its modular design, users (e.g., researchers and developers) can easily extend these tunable parameters and metrics to integrate their own ADR algorithms and evaluate their behavior in large-scale scenarios. This flexibility allows for the exploration of a wide range of network configurations and optimization of transmission strategies in a computationally efficient manner. Furthermore, although FAST-LoRa has been designed and validated for LoRaWAN networks, its underlying statistical modeling principles could be extended to other long-range IoT technologies with similar communication structures, provided that their key performance metrics can be analytically modeled.

A. NETWORK MODEL

FAST-LoRa leverages a realistic analytical network model to efficiently estimate key performance metrics. We consider a LoRa network consisting of N end devices (EDs) and K gateways (GWs). Each ED i is assigned a transmission power $p_i \in \mathcal{P}$, a spreading factor $f_i \in \mathcal{F}$, and a coding rate (CR) $r_i \in \mathcal{R}$. Regarding the network topology, we consider the star topology typically used in LoRaWAN networks, where all the EDs communicate with the gateways in coverage, which are in turn interconnected through a central Network Server

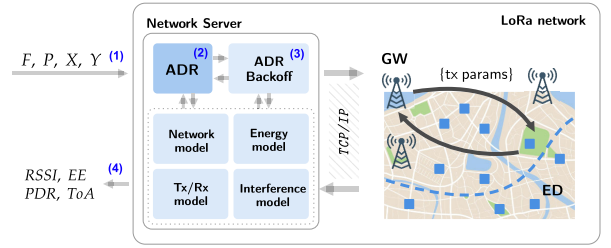


FIGURE 1. Overview of FAST-LoRa architecture, including its main components and dataflow. (1) Initialize network; (2) ADR algorithm, generating transmissions considering interference and determining transmission reliability and reception outcomes; (3) Failed receptions trigger ADR backoff; (4) key performance metrics are recorded during and at the end of the simulation.

(NS). The NS controls and optimizes the communication between the end devices and the gateways. Although LoRa networks support the simultaneous use of multiple channels, cross-channel interference is generally negligible when the frequency ranges of the channels do not overlap [59]. This condition holds for most LoRaWAN channel plans defined by the LoRa Alliance [12]. Consequently, FAST-LoRa focuses on intra-channel interference. This results in a more efficient simulation while still providing a realistic representation of the network's performance. To better illustrate how the network components and the various mechanisms interact within FAST-LoRa, an overview is presented in Figure 1.

B. TIME-ON-AIR MODEL

Time-on-air (T_i) is a key metric for evaluating performance, particularly concerning interference and energy efficiency. Time-on-air refers to the duration that a wireless transmission occupies the communication medium, from the start of the preamble until the end of the payload transmission. The time-on-air required by ED i on a channel with bandwidth B using a spreading factor of $f_i \in \mathcal{F}$ and CR $r_i \in \mathcal{R}$ for transmitting a single LoRa packet with L bytes of payload can, according to [48], be calculated as:

$$T_i = T_i^{pr} + T_i^{pl} \quad (1)$$

where T_i is the total time-on-air for ED i , consisting of preamble time (T_i^{pr}) and payload transmission time (T_i^{pl}). The time to transmit a single symbol (T_i^{sym}) can be calculated as [48]:

$$T_i^{sym} = \frac{2f_i}{B} \quad (2)$$

where f_i is the spreading factor of ED i , and B is the bandwidth of the channel. The time to transmit the LoRa preamble (T_i^{pr}), where the number of preamble symbols n_{pr} is a fixed value, can be calculated as follows [48]:

$$T_i^{pr} = (n_{pr} + 4.25) T_i^{sym} \quad (3)$$

The time to transmit the payload (T_i^{pl}) is calculated as [48]:

$$T_i^{pl} = n_{pl} T_i^{sym} \quad (4)$$

where n_{pl} is the number of symbols needed to encode the payload information and is calculated as [48]:

$$n_{pl} = 8 + \max \left(\left\lceil \frac{8L - 4f_i + 28 + 16CRC - 20H}{4(f_i - 2DE)} \right\rceil (r_i + 4), 0 \right) \quad (5)$$

where L is the payload size in bytes, CRC is 1 if the packet contains a checksum and 0 otherwise, H is 0 if the packet header is enabled and 1 otherwise, and DE is 1 if low data rate optimization is enabled and 0 otherwise. In our approach, the time-on-air calculations are specifically focused on uplink messages where we model the transmission of packets from EDs to GWs. Downlink messages are not explicitly simulated, as LoRaWAN operation is typically dominated by uplink traffic, particularly in best-effort IoT applications relying on unconfirmed transmissions, a design choice that improves scalability and limits downlink congestion [18], [22], [30]. Moreover, downlink payloads are generally small and the corresponding energy consumption in reception mode can be considered negligible [11]. This consideration is typically sufficient to evaluate the performance of different ADR algorithms, which primarily rely on uplink behavior [3], [18], [31], [39], [60]. Regarding the decoding probability of downlink messages, we assume that it is similar to their predecessor uplink messages [21], as channel conditions remain essentially unchanged within the time interval between them. Consequently, if an uplink transmission is not decoded, the associated downlink is also considered lost. This modeling choice allows us to maintain scalability while still capturing the dominant effects of downlink reliability.

C. TRANSMISSION RELIABILITY MODEL

In a LoRa network, transmission reliability can be determined by two key factors: (i) the Received Signal Strength (RSS) at the gateway must exceed its receiver sensitivity, and (ii) the packet must not be corrupted by interference from other end device transmissions [27]. The PDR represents the probability that a packet from ED i is successfully received by GW k , and it can be calculated as the product of two probabilities: that the RSS is sufficient for successful reception (ψ_{ik}) and that the packet is not corrupted by interference (ζ_{ik}) [60]:

$$PDR_{ik} = \psi_{ik} \cdot \zeta_{ik} \quad (6)$$

To calculate the probability that the RSS is sufficient for reception (ψ_{ik}), we express the RSS as the difference between a deterministic component z_{ik} and a random noise component N_{ik} , where $RSS_{ik} = z_{ik} - N_{ik}$, with $z_{ik} = p_i - \overline{PL}(d_0) - 10\gamma \log_{10} \left(\frac{d_{ik}}{d_0} \right)$ and $N_{ik} \sim \mathcal{N}(0, \sigma)$, a Gaussian random variable, representing the noise. In this paper, we consider the log-distance path loss model since it is suitable for rural and urban environments. It considers the transmission power, distance between the end device and the gateway, and the effects of signal attenuation due to fading. Using the log-distance model, the RSS of a transmission can be calculated

as follows:

$$RSS_{ik} = p_i - \overline{PL}(d_0) - 10\gamma \log_{10} \left(\frac{d_{ik}}{d_0} \right) - N_{\sigma} \quad (7)$$

where $p_i \in \mathcal{P}$ is the transmission power setting of ED i ; $\overline{PL}(d_0)$ is the mean path loss at a reference distance d_0 , usually selected under line-of-sight (LoS) conditions; γ is the path loss exponent, indicating how quickly the signal attenuates in a particular environment; d_{ik} is the euclidean distance between ED i and GW k ; and N_{σ} is a Gaussian random variable with a zero mean and standard deviation of σ , reflecting the signal variation caused by shadow fading. The probability that the RSS exceeds receiver sensitivity η_{fi} for a given spreading factor f_i is then calculated using the error function (erf) as follows:

$$\begin{aligned} \psi_{ik} &= \mathbb{P}(RSS_{ik} \geq \eta_{fi}) \\ &= \mathbb{P}(N_{ik} \leq z_{ik} - \eta_{fi}) = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{z_{ik} - \eta_{fi}}{\sqrt{2}\sigma} \right) \end{aligned} \quad (8)$$

where η_{fi} describes the sensitivity of a receiving GW at spreading factor f_i , and erf denotes the Gauss error function.

D. INTERFERENCE MODEL

To accurately model the interference across transmissions, we first calculate the likelihood of overlapping transmissions. Given two EDs, i and j , the probability that ED j will interfere with ED i is based on the time overlap of their transmissions. We assume that the transmission from an ED can be successfully recognized if at least the last five preamble symbols are received without interference [18], as shown in Figure 2.

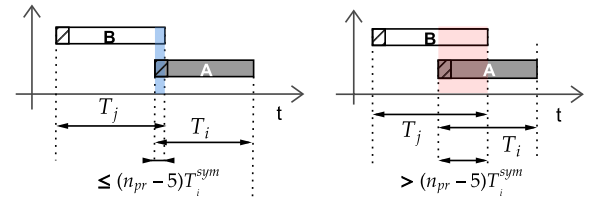


FIGURE 2. Illustration of overlap and interference between two transmissions A and B.

Therefore, the time interval during which ED j must not transmit to avoid interfering with a transmission from ED i can be calculated as:

$$T'_{ij} = T_j + T_i - (n_{pr} - 5)T_i^{\text{sym}} \quad (9)$$

where T_j and T_i are the transmission durations of EDs j and i , respectively. With this, we calculate the probability that ED j interferes with ED i , considering the exponential nature of packet generation in the network [51], [60]:

$$h_{ij} = 1 - e^{-\lambda T'_{ij}} \quad (10)$$

where λ is the mean packet generation rate.

When calculating the total interference probability, we also need to take into account the modulation characteristics

of LoRa. The robustness of LoRa modulation allows for successful decoding even if the RSS is below the noise floor, provided that the signal-to-interference ratio (SIR) is above a certain threshold, which depends on the spreading factor employed by transmissions from the affected ED i and the interfering ED j [15], [56], [57]. Then, the probability that the transmission from ED i will be corrupted by interference from ED j is calculated as:

$$\begin{aligned} & \mathbb{P}(RSS_{ik} - RSS_{jk} < \omega_{fifj}) \\ &= \mathbb{P}(N_{jk} - N_{ik} < \omega_{fifj} - (z_{ik} - z_{jk})) \\ &= \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\omega_{fifj} - (z_{ik} - z_{jk})}{2\sqrt{2}\sigma}\right) \end{aligned} \quad (11)$$

where ω_{fifj} is the signal-to-interference ratio threshold, and erf is the Gauss error function, which can be applied since both N_{jk} and N_{ik} are Gaussian random variables $\mathcal{N}(0, \sigma)$, combinable with another Gaussian random variable $\mathcal{N}(0, 2\sigma)$.

Considering all the end devices in a network, the probability that a packet from ED i received at GW k will not be corrupted by transmissions from other end devices can be calculated as:

$$\zeta_{ik} = \prod_{j \in \{1..N\}, i \neq j} (1 - h_{ij} \mathbb{P}(RSS_{ik} - RSS_{jk} < \omega_{fifj})) \quad (12)$$

The modeling components described above play complementary roles in determining network performance during the simulations. As illustrated in Fig. 1, the simulation procedure follows four main steps. (1) The simulation starts with the input parameters, including the transmission settings (F , P) and the node positions (X , Y). (2) These inputs are used by the ADR mechanism to determine suitable transmission parameters for each device. (3) In case of connectivity losses, the ADR backoff mechanism is triggered to adapt the parameters accordingly. (4) The resulting configuration is then passed to the propagation, interference, and reception models. Specifically, the propagation model captures how signal power decreases with distance and is affected by fading, shaping the probability that a transmission is successfully received at each gateway. The interference model accounts for temporal overlaps between transmissions and the spreading factor, dependent SIR thresholds, determining whether a collision results in packet loss. Finally, the reception model aggregates the decoding probabilities across gateways, leveraging spatial diversity to estimate network reliability. Together, these steps enable FAST-LoRa to provide accurate performance estimates while maintaining low computational complexity.

E. PERFORMANCE METRICS

Finally, the overall PDR of ED i can be calculated by considering all the available gateways GW, and combining the individual PDRs for each GW. As defined in Eq. (6), the PDR between ED i and each GW k is given by the product of the probabilities that the RSS will be sufficient for reception

(ψ_{ik}) and that the packet will not be corrupted by interference (ζ_{ik}). Once these components are calculated for each pair of EDs and GWs, the overall PDR for ED i is obtained by combining the individual PDRs from each gateway using the following equation:

$$PDR_i = 1 - \prod_{k \in \{1..K\}} (1 - PDR_{ik}) \quad (13)$$

This equation aggregates the individual PDRs from each gateway, providing the overall probability that a packet from ED i will be successfully received by at least one GW in the network, considering the interference and signal quality at each GW. It is possible to sample the RSS of an individual packet transmission from end device i arriving at gateway k as:

$$RSS_{ik} = \begin{cases} RSS_{ik} & \text{if } RSS_{ik} \geq \eta_{fi}, \text{ with probability } \zeta_{ik} \\ -\infty & \text{otherwise} \end{cases} \quad (14)$$

where RSS_{ik} represents the sampled RSS, which is calculated as the RSS (including random signal attenuation by flat fading) if it exceeds receiver sensitivity η_{fi} for the used spreading factor and also considering possible corruption by interference with transmissions from other end devices ζ_{ik} . An RSS of $-\infty$ indicates that the packet has not been decoded successfully in the chosen sample.

Similarly, the Signal-to-Noise Ratio (SNR) and Energy Efficiency (EE) are two crucial metrics that impact transmission performance. The SNR determines the quality of the received signal at the GW, which influences the likelihood of successful demodulation, while EE reflects the efficiency of energy consumed during data transmission. The SNR of a transmission from ED i arriving at GW k can generally be calculated as:

$$SNR_{ik} = RSS_{ik} - P_{noise} \quad (15)$$

where P_{noise} represents the noise floor in dBm. LoRa's modulation allows for successful demodulation even when the RSS is below the noise floor, depending on the spreading factor used [24], [25], [42], [52]. The minimum SNR required for successful demodulation for each spreading factor (f) is listed in Table 1.

TABLE 1. Minimum SNR for successful demodulation.

f	SNR_f^{min} (dB)
7	-7.5
8	-10.0
9	-12.5
10	-15.0
11	-17.5
12	-20.0

Many algorithms designed for selecting transmission parameters rely on the SNR of packet transmissions to determine optimal settings [24], [25], [42], [52]. Given that the

presented simulation model uses RSS to assess transmission success or failure, it is important for the calculated SNR to exceed the minimum threshold for successful transmissions and fall below this threshold for failed transmissions if such SNR-based algorithms are to be evaluated using the simulation framework. Consequently the SNR of a sampled transmission from ED i arriving at GW k is approximated as:

$$SNR_{ik} = R\hat{S}_{ik} - \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} (\eta_f + SNR_f^{min}) \quad (16)$$

Finally, we define the Energy Efficiency (EE) of ED i as the number of payload bits successfully transmitted per amount of energy consumed by the ED. EE can be calculated as follows:

$$EE_i = \frac{8L \cdot PDR_i}{e_{p_i} T_i} \quad (17)$$

where L is the number of payload bytes, T_i is the time required for transmitting a single packet, and e_{p_i} is the power consumption expressed in mW using a transmission power setting $p_i \in \mathcal{P}$.

F. NETWORK CONFIGURATION AND RESULTS

The analytical model described above forms the core of FAST-LoRa, enabling us to accurately model network behavior. To enhance FAST-LoRa's efficiency, we extend these equations to matrix operations, permitting faster computations, especially in large-scale network simulations. Based on these operations, we define the required inputs for FAST-LoRa, such as network configuration, and determine the simulation outputs, including key performance metrics. The simulation model, which includes N end devices and K gateways, operates with the following input parameters:

- $X \subset \mathbb{R}^{N \times 2}$: Positions of end devices as 2-dimensional coordinates.
- $Y \subset \mathbb{R}^{K \times 2}$: Positions of gateways as 2-dimensional coordinates.
- $F \subset \mathcal{F}^N$: Assigned spreading factors for end devices.
- $P \subset \mathcal{P}^N$: Assigned transmission power settings for end devices.

Other network configuration settings, such as the coding rate, channel bandwidth, and packet generation rate, are considered scalar values shared across all the end devices within the simulated network. However, they could be readily tuned to optimize the time-on-air computation in future works to broaden the range of supported configurations. Moreover, frequency channel allocation could also be modeled by extending the interference model to account for channel-dependent collisions and transmission distribution across multiple frequencies.

FAST-LoRa produces the following metrics and simulation results:

- $RSS \subset \mathbb{R}^{N \times K}$: The expected RSS of transmissions from every end device at each gateway measured in dBm (ignoring signal attenuation due to flat fading).

- $\hat{RSS} \subset (\mathbb{R} \cup \{-\infty\})^{N \times K}$: The sampled RSS of transmissions from every end device at each gateway measured in dBm (considering signal attenuation, receiver sensitivity, and potential transmission corruption due to interference).
- $\hat{SNR} \subset (\mathbb{R} \cup \{-\infty\})^{N \times K}$: The sampled SNR of transmissions from every end device at each gateway measured in dB.
- $PDR^{GW} \subset [0, 1]^{N \times K}$: The Packet Delivery Ratio (PDR) of each end device to each gateway.
- $PDR \subset [0, 1]^N$: The overall PDR of each end device considering all the available gateways.
- $EE \subset \mathbb{R}^N$: The energy efficiency of each end device measured in bits/mJ.

G. ADR AND BACKOFF MECHANISMS

FAST-LoRa includes the ADR and backoff mechanisms proposed by Semtech [46], [49], standard LoRaWAN procedures, and commonly used in scientific research as a comparison baseline. We have reproduced the logic of the Semtech ADR in our implementation, as detailed in Algorithm 1, and integrated the backoff variant described in Algorithm 2. Note that we explicitly present these algorithms in the manuscript to clearly state the exact procedures used in FAST-LoRa. This choice is motivated by the observation that implementations of these mechanisms often vary across papers and datasheets, leading to potential ambiguities. By including the algorithms explicitly, we ensure clarity, reproducibility, and consistency with the specification.

1) ADR

In the default ADR mechanism, a message buffer stores the maximum of the packet transmissions' SNRs across all gateways. Once the buffer is full, a link margin depending on the maximum SNR in the buffer and the currently assigned spreading factor is determined and a number of steps to execute are calculated (where the truncate function retains the integer part of a floating point value). Depending on the number of calculated steps to execute, the spreading factor and transmission power settings are adjusted.

2) ADR BACKOFF

We have also implemented the ADR backoff mechanism to handle communication losses, e.g., caused by the selection of inadequate transmission parameters. Upon initialization, a counter $n_{ADR_ACK_CNT}$ is set to 0. This counter is increased whenever the end device attempts an uplink transmission (lines 7-9). Alternatively, after $n_{ADR_ACK_LIMIT}$ uplink transmissions have been attempted without receiving new transmission parameters, the end device actively requests transmission parameters (lines 10-14). If the network server receives an uplink transmission including such a request, it will reconfirm the current transmission settings with a downlink transmission (assumed to arrive at the end device), resetting the counter to 0. If transmissions still do not

Algorithm 1 ADR**Require:** $\mathcal{F}, \mathcal{P}, \eta$

```

1: procedure init
2:   initialize measurement buffer  $M$  of size 20
3: end procedure

4: procedure add_measurement( $\hat{SNR}, f, p$ )
5:    $M \xleftarrow{\text{push}} \max \hat{SNR}$ 
6:   if  $M$  is full then
7:      $SNR_{max} \leftarrow \max M$ 
8:      $SNR_{margin} \leftarrow SNR_{max} - \eta_f - 10$ 
9:      $n \leftarrow \text{truncate}(\frac{SNR_{margin}}{3})$ 
10:    while  $n > 0$  and  $f > \min \mathcal{F}$  do
11:       $f \leftarrow f - 1$ 
12:       $n \leftarrow n - 1$ 
13:    end while
14:    while  $n > 0$  and  $p > \min \mathcal{P}$  do
15:       $p \leftarrow p - 2$ 
16:       $n \leftarrow n - 1$ 
17:    end while
18:    while  $n < 0$  and  $p < \max \mathcal{P}$  do
19:       $p \leftarrow p + 2$ 
20:       $n \leftarrow n + 1$ 
21:    end while
22:    yield ( $f, p$ )
23:     $M \xleftarrow{\text{clear}} \emptyset$ 
24:  end if
25: end procedure

26: procedure reset
27:    $M \xleftarrow{\text{clear}} \emptyset$ 
28: end procedure

```

arrive at any gateway, after further $n_{ADR_ACK_DELAY}$ uplink transmissions, the transmission power will be set to the allowed maximum (lines 15-25). Finally, after every further $n_{ADR_ACK_DELAY}$ transmission, the spreading factor will be increased by one step, up to the possible maximum.

H. IMPLEMENTATION

FAST-LoRa has been implemented leveraging matrix operations using the well-known NumPy¹ library, efficiently applying caching techniques to avoid redundant calculations. The source code of FAST-LoRa and the default ADR and backoff mechanisms have been made publicly available [2] as a fully open-source Python package. This not only ensures transparency and reproducibility but also permits seamless integration with common libraries that can be used to optimize LoRa networks [1], [5], [6], [7].

¹<https://numpy.org/>

Algorithm 2 ADR Backoff**Require:** $\theta = (n_{ADR_ACK_LIMIT}, n_{ADR_ACK_DELAY}, \mathcal{F}, \mathcal{P})$

```

1: procedure init( $\theta$ ) ▷ Initialization
2:    $n_{ACK\_CNT} \leftarrow 0$ 
3: end procedure

4: procedure update ▷ No response received
5:    $n_{ACK\_CNT} \leftarrow n_{ACK\_CNT} + 1$ 
6: end procedure

7: procedure reset_on_receive ▷ Reset, upon receiving ADR
8:    $n_{ACK\_CNT} \leftarrow 0$ 
9: end procedure

10: procedure reset_on_request ▷ Reset, with requesting ADR
11:   if  $n_{ACK\_CNT} > n_{ADR\_ACK\_LIMIT}$  then
12:      $n_{ACK\_CNT} \leftarrow 0$ 
13:   end if
14: end procedure

15: procedure recover( $f_{current}, p_{current}$ ) ▷ Adjust transmission parameters
16:    $f_{new} \leftarrow f_{current}$ 
17:    $p_{new} \leftarrow p_{current}$ 
18:   if  $n_{ADR\_ACK\_CNT} = n_{ADR\_ACK\_LIMIT} + n_{ADR\_ACK\_DELAY}$  then
19:      $p_{new} \leftarrow \max(\mathcal{P})$ 
20:   else if  $n_{ADR\_ACK\_CNT} > n_{ADR\_ACK\_LIMIT} + n_{ADR\_ACK\_DELAY}$  then
21:     if  $n_{ADR\_ACK\_CNT} - n_{ADR\_ACK\_LIMIT} \equiv 0 \pmod{n_{ADR\_ACK\_DELAY}}$  then
22:        $f_{new} \leftarrow \min(f_{current} + 1, \max(\mathcal{F}))$ 
23:     end if
24:   end if
25:   return  $f_{new}, p_{new}$ 
26: end procedure

```

IV. EVALUATION SETUP

This section describes the configuration setup to be evaluated, detailing the baselines used, network configuration, performance metrics, and evaluation scenarios. During this evaluation, we focus on congested multi-gateway networks to cover all the aspects of the analytical model behind FAST-LoRa (e.g., interference among EDs). We simulate different distances between EDs and GWs to verify the correct calculation of signal strength combined with receiver sensitivity. In this evaluation, we assume homogeneous receiver sensitivities across all the devices in the network, meaning all EDs have the same sensitivity, power consumption, and transmission settings, as is common in

other simulators [18], [37], [51]. While FAST-LoRa allows these parameters to be configured on a per-device basis, a homogeneous configuration is adopted here to improve clarity, reproducibility, and interpretability of the results. Furthermore, by simulating a large number of EDs, we ensure correct interference calculations. Including multiple active GWs within a network allows us to verify the accuracy of reception calculations at the network server level. The network parameters selected for the validation process are listed in Table 2. All simulations have been conducted on an Intel Core i5-8265U processor with 8 GB of RAM.

A. METRICS

To evaluate FAST-LoRa, we employ key performance metrics, such as the Packet Delivery Ratio (PDR) and Energy Efficiency (EE). The PDR is defined as the ratio of successfully received packets to the total number of packets transmitted by ED i :

$$PDR_i = \frac{n_i^{rx}}{n_i^{tx}} \quad (18)$$

where n_i^{rx} is the number of packets received by any gateway in the network, and n_i^{tx} is the total number of packets transmitted by ED i . Similarly, EE is defined as the number of payload bits successfully transmitted per unit of energy consumed by the ED, as shown in Eq. (17).

B. BASELINES

FAST-LoRa correctness and accuracy are validated using FloRa, a well-known discrete event-based simulation framework for LoRa networks [51]. We selected FLoRa as our baseline because it is one of the most widely adopted packet-level simulators for LoRaWAN, providing a verified and strong reference for validating the accuracy of our simulator [13], [14], [32]. It should be highlighted that our goal is not to perform a systematic performance comparison, but rather to prove that FAST-LoRa produces correct and consistent results. While simulator performance is indeed important, it depends heavily on the fidelity of the modeled communication layers and the abstraction level adopted (e.g., physical, link, transport, or application). As such, performance comparisons are inherently tied to specific scenarios and research goals, making a general benchmarking study beyond the scope of this paper. Due to the heterogeneity of modeling assumptions and optimization strategies across simulators, such a comparison represents a substantial research effort of its own. To keep the scope focused, we restricted our validation to FLoRa and showed that FAST-LoRa achieves comparable accuracy.

To achieve a fair comparison, we carefully reviewed the implementation of FloRa and appropriately configured the random packet generation and energy consumption calculations. FloRa models the time between ED packet generation using an exponential distribution. To maintain compatibility with the underlying simulation framework, constraints were

TABLE 2. Network parameters employed during the evaluation [18], [47], [51].

Parameter	Value
$\mathcal{P}$	$\{2, 4, 6, 8, 10, 12, 14, 16\}$ dBm
$\mathcal{F}$	$\{7, 8, 9, 10, 11, 12\}$
r	1
B	125 kHz
n_{pr}	8
CRC	1
H	1
DE	0 ($f=7-10$), 1 ($f=11-12$)
$\overline{PL}(d_0)$	127.41 dBm
d_0	40 m
γ	2.08
λ	0.001 s^{-1}
σ	3.57
$\omega_{f_i f_j}$	See Table 3
η_{f_i}	See references [47], [51]
e_{p_i}	See reference [34]

introduced for different spreading factors to prevent potential runtime errors [51]. As a result, these constraints cause the time between packet generations to deviate from a strictly exponential distribution. These constraints are only required in edge cases, and the simulation framework generally allows shorter delays without issues. Since the assumption of an exponential inter-arrival time is essential to accurately model interference probability, we opted to disable the original constraints while implementing an error-handling mechanism to manage the exceptional cases where constraints were required.

In this work, FAST-LoRa focuses on energy consumption analysis and calculation of energy consumption in uplink packets. FloRa estimates energy consumption using predefined table values without a specified data source. Upon review, we observed that some transmission power levels were assigned identical consumption values (e.g., 24 mA for 2–4 dBm, 25 mA for 5–8 dBm), which deviates from experimentally measured values [34]. To improve the accuracy of the validation process, we set the power consumption values to better reflect the measured data.

V. PERFORMANCE EVALUATION RESULTS

This section describes the experimental scenarios used for validating FAST-LoRa and the methodology applied to carry out the experiments. The evaluation is designed to assess both the accuracy and the computational efficiency of FAST-LoRa through a set of complementary scenarios that progressively increase in complexity. First, a simple end device–gateway scenario is considered to directly validate the reliability model under controlled conditions without interference. Second, the evaluation is extended to dense

TABLE 3. Signal-to-interference ratio thresholds $\omega_{f_i f_j}$ in dB for successful transmission decoding [15], [51], [56], [57].

$f_i \backslash f_j$	7	8	9	10	11	12
7	6	-8	-9	-9	-9	-9
8	-11	6	-11	-12	-13	-13
9	-15	-13	6	-13	-14	-15
10	-19	-18	-17	6	-17	-18
11	-22	-22	-21	-20	6	-20
12	-25	-25	-25	-24	-23	6

and more realistic network configurations with multiple end devices and gateways, where interference effects and multi-gateway reception are explicitly exercised. Finally, a computational performance analysis is conducted to quantify the scalability of FAST-LoRa as the network size increases.

A. END DEVICE TO GATEWAY COMMUNICATION

In the first experiment, we validated the basic functionality of FAST-LoRa by comparing the PDR and EE calculations for a network with only one end device and one gateway. This extremely simplified setup allows for a direct comparison of FAST-LoRa and FLoRa performance in a controlled environment. The experiment was conducted by varying the ED-GW distance, considering values of 20, 50, and 100 m. The spreading factors and transmission power values were configured according to the values specified in Table 2. This setup allowed us to test the impact of these parameters on the PDR and EE performance metrics. The simulations were run for 120 hours using FLoRa, while FAST-LoRa computed the metrics without the need to simulate individual packets due to the statistical nature of its underlying mathematical model. In FLoRa, ADR mechanisms were disabled to maintain the initially configured spreading factor and transmission power settings throughout the simulation. The results were then compared to assess the accuracy and efficiency of FAST-LoRa, directly comparing it with FLoRa.

The results of this comparison are illustrated in Fig. 3 and Fig. 4. The behavior of the PDR and EE for different values of spreading factor (f) and transmission power (p) are shown in Fig. 3. In the first plot, where the PDR is shown as a function of f , both FAST-LoRa and FLoRa exhibit similar performance, and the PDR increases with higher spreading factors. This behavior is expected, as a higher spreading factor leads to better reception, albeit at the cost of longer transmission times. The accuracy of FAST-LoRa's PDR calculations is confirmed, with results closely matching those of FLoRa. In the second plot, where the PDR is plotted as a function of p , both models show that increasing transmission power improves the PDR. However, FAST-LoRa computes these values significantly faster, providing near-instantaneous results compared to

FLoRa's more computationally expensive approach, which requires simulating each packet. This will be further and quantitatively analyzed in Section V-C. The third plot, which depicts EE as a function of f , shows that EE decreases with higher spreading factors, as expected due to longer transmission times. FAST-LoRa produces results comparable to FLoRa but much more efficiently, confirming the model's ability to calculate energy efficiency quickly and accurately. Finally, in the fourth plot, where EE is shown as a function of p , we observe that EE decreases as transmission power increases, which aligns with the expected behavior of a real-world system. FAST-LoRa handles this calculation in a fraction of the time that FLoRa requires, further highlighting its suitability for rapid simulations and network evaluations. The results shown in Fig. 4 provide the same analysis as Fig. 3 but consider the average values across different transmission powers as a function of the distance. The first plot illustrates how the PDR decreases with increasing distance, with FAST-LoRa and FLoRa following similar trends. The second plot presents EE behavior across different distances, confirming the expected decline as distance increases. As observed previously, FAST-LoRa provides results comparable to FLoRa but with significantly reduced computation time, making it highly suitable for large-scale simulations.

To assess the simulation and accuracy quality of FAST-LoRa, two key error metrics were employed: the Mean Absolute Error (MAE) and the Standard Deviation of Error (SDE). These metrics help quantify how well FAST-LoRa approximates the performance metrics obtained from FLoRa simulations. First, we present the results from Table 4, showing the MAE and SDE values for the PDR and EE under different spreading factor (f) settings. This table calculates the MAE and SDE by averaging the results over the different transmission power values considered in this work. In this table, we observe how the performance of FAST-LoRa compares to FLoRa as the spreading factor is varied for different ED-GW distances. For example, at a distance of 20 meters, the MAE for the PDR significantly decreases as the spreading factor increases, reaching zero values at $f=12$. This suggests that the PDR of FAST-LoRa becomes more accurate as the spreading factor increases. Similarly, the MAE for EE follows the same trend, with fewer errors at higher spreading factor values. The SDE values show similar behavior, with low values indicating that the error is consistent and stable across different spreading factor settings. However, when the ED-GW distance increases, we notice a corresponding increase in the MAE values for the PDR and EE. This is expected because as the distance increases, the signal strength diminishes, leading to greater variability in the performance metrics. Despite this, the MAE and SDE still show a clear trend where higher spreading factor values continue to improve the accuracy of the calculations, albeit with slightly larger errors at longer distances. In summary, Table 4 demonstrates that as the ED-GW distance increases,

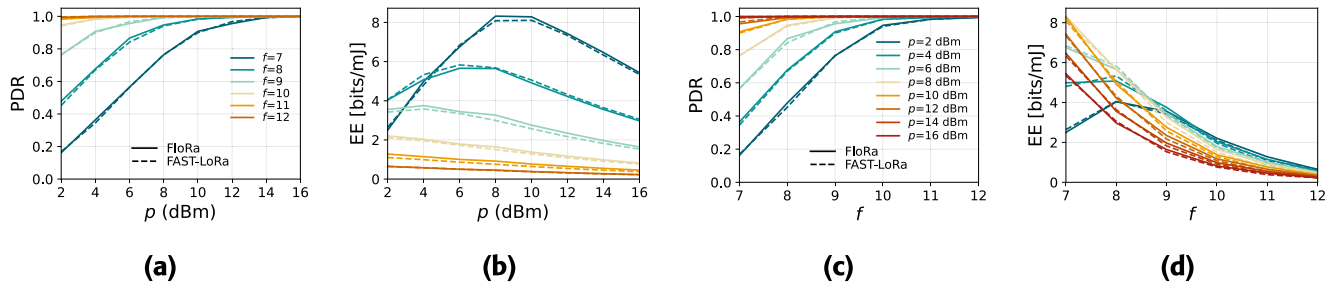


FIGURE 3. Comparison of the PDR and EE obtained with FloRa and FAST-LoRa considering different values of f (a, b) and different values of p (c, d) for an ED-GW distance of 50 m.

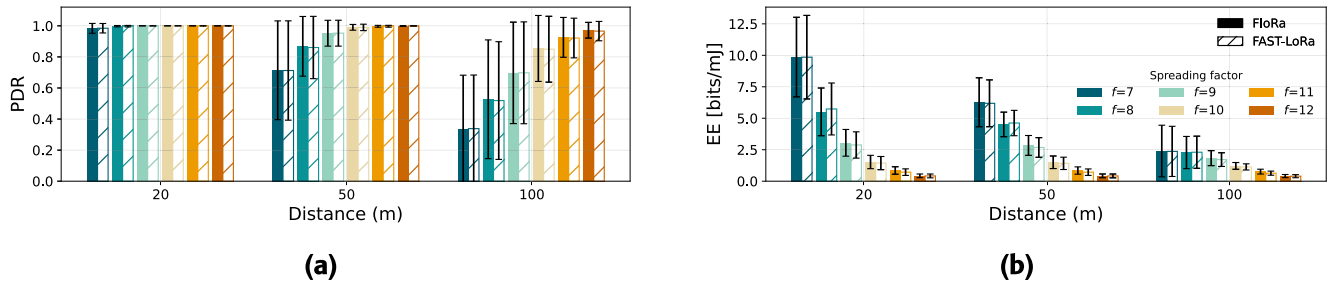


FIGURE 4. Comparison of the PDR (a) and EE (b) of FloRa and FAST-LoRa as a function of ED-GW distance for different values of f .

FAST-LoRa becomes slightly less accurate, but the general trend remains that higher spreading factor values improve the MAE and SDE, indicating that FAST-LoRa is still able to provide consistent and reliable results under varying conditions.

Table 5 presents the MAE and SDE for the PDR and EE with varying transmission power (p) settings. Like the previous table, these values are averaged over different spreading factor settings. This table provides insights into how transmission power affects FAST-LoRa accuracy and consistency of FAST-LoRa performance. For the PDR metric, the MAE decreases as p increases, which is expected, as higher transmission power typically leads to better signal reception and, consequently, more accurate predictions. At a distance of 20 meters, for instance, the MAE for the PDR approaches zero as the p increases, showing that the model is increasingly accurate with higher transmission power settings. The SDE values for the PDR also decrease, further indicating that FAST-LoRa provides stable and consistent results across different transmission power levels. However, as the ED-GW distance increases (e.g., at 50 or 100 meters), we notice that the MAE for the PDR begins to rise. This is expected because, as distance increases, signal strength diminishes, leading to varying network performance and less accurate values. Even so, the MAE for the PDR still decreases with higher transmission power values, indicating that increasing transmission power helps mitigate the effects of increased distance and signal degradation. Similarly, the SDE values increase slightly with distance, reflecting a greater variability in the computed PDR values as distance grows.

However, they remain relatively low for higher transmission power values. For EE, a similar behavior is observed. The MAE decreases as the transmission power increases, and the SDE values remain relatively low, reflecting the model's reliability in predicting energy efficiency across different transmission power settings. As with the PDR, at greater distances (e.g., at 100 meters), the MAE for EE is slightly higher, but the increase is not as pronounced as for the PDR. At higher transmission power values (e.g., $p=16$), both the PDR and EE values from FAST-LoRa are very close to the FloRa results, confirming the robustness of FAST-LoRa in accurately estimating network performance, even as both transmission power and distance vary.

B. REALISTIC DENSE NETWORKS

In the second validation step, we extended the evaluation to a more complex network scenario involving multiple end devices (EDs) and gateways (GWs) to assess FAST-LoRa's performance in a real-world-like network with interference and reception calculations. In this setup, different network configurations were simulated, considering 10, 50, 100, and 500 EDs, along with 1 to 4 GWs, to evaluate how well FAST-LoRa handles interference and reception when combining data from multiple gateways. For each combination of end devices and gateways, a total of 5 network configurations were generated to ensure randomness without introducing unintentional bias. The spreading factor and transmission power settings were randomly assigned to the end devices, sampled from all possible configurations (as

TABLE 4. MAE and SDE values for PDR and EE across different spreading factor (f) settings. The results are averaged over different transmission power (p) configurations and ED-GW distances between the ED and GW.

Distance (m)	f	PDR [$\times 10^{-2}$]		EE (bits/mJ)	
		MAE	SDE	MAE	SDE
20	7	0.122	0.214	0.201	0.219
	8	0.089	0.168	0.230	0.177
	9	0.017	0.037	0.172	0.045
	10	0.001	0.002	0.085	0.022
	11	0.000	0.000	0.131	0.034
	12	0.000	0.000	0.008	0.007
50	7	0.630	0.855	0.143	0.126
	8	0.846	1.059	0.111	0.074
	9	0.295	0.493	0.162	0.052
	10	0.132	0.236	0.086	0.024
	11	0.057	0.114	0.130	0.033
	12	0.048	0.109	0.008	0.008
100	7	0.574	0.740	0.064	0.087
	8	1.347	1.631	0.079	0.087
	9	0.595	0.735	0.117	0.049
	10	0.820	1.226	0.081	0.034
	11	0.751	1.237	0.122	0.031
	12	0.614	1.119	0.010	0.007
All	All	0.385	0.811	0.108	0.129

shown in Table 2), ensuring broad coverage of different network configurations. Additionally, the positions of the end devices were randomly assigned by sampling x- and y-coordinates from a uniform distribution, limiting both coordinates between 0 m and 1000 m. This resulted in diverse network topologies, as seen in Fig. 5, where the gateways are represented in orange and the end devices in black.

In this scenario, both FAST-LoRa and FLoRa were used to calculate the PDR and EE for each end device. The FLoRa simulations ran for 5 days, simulating approximately 432 packet transmissions per ED using the parameters specified in Table 2, resulting in minor statistical fluctuations that were considered acceptable for validation purposes. ADR mechanisms were disabled to ensure that the initially assigned spreading factor and transmission power settings were maintained throughout the simulations. Due to the statistical nature of FAST-LoRa, it calculated the PDR and EE through faster simulations, avoiding the need for packet-based simulations. The MAE and SDE of the PDR and EE were calculated across all EDs and simulations for each network configuration, and the results are presented in Table 6 and Table 7, respectively. From Table 6, we can observe that as the number of gateways and end devices increases, the MAE and SDE tend to rise slightly. As we increase the number of end devices and gateways, the MAE for the PDR shows slight increases, reaching a value of 1.044×10^{-2} for the MAE and 1.579×10^{-2} for the SDE in networks

TABLE 5. MAE and SDE values for the PDR and EE with varying transmission power (p) settings. The results are averaged over different spreading factor (f) configurations and ED-GW distances.

Distance (m)	p	PDR [$\times 10^{-2}$]		EE (bits/mJ)	
		MAE	SDE	MAE	SDE
20	2	0.101	0.145	0.201	0.246
	4	0.169	0.245	0.199	0.245
	6	0.011	0.019	0.156	0.194
	8	0.021	0.044	0.167	0.140
	10	0.003	0.006	0.113	0.109
	12	0.000	0.001	0.109	0.093
50	14	0.000	0.000	0.087	0.077
	16	0.000	0.000	0.069	0.064
50	2	0.728	1.069	0.095	0.108
	4	0.634	0.721	0.142	0.154
	6	0.685	1.095	0.094	0.109
	8	0.156	0.232	0.144	0.110
	10	0.170	0.280	0.123	0.115
	12	0.244	0.415	0.096	0.087
100	14	0.033	0.052	0.088	0.078
	16	0.027	0.056	0.070	0.065
100	2	0.997	1.466	0.044	0.051
	4	1.034	1.433	0.059	0.076
	6	1.741	1.596	0.107	0.097
	8	0.569	0.648	0.084	0.073
	10	0.402	0.496	0.100	0.101
	12	0.988	1.402	0.108	0.062
100	14	0.386	0.579	0.065	0.064
	16	0.150	0.212	0.063	0.059
All	All	0.385	0.811	0.108	0.129

with 500 EDs and 4 GWs. Despite this increase, both error metrics remain within an acceptable range, which reflects that FAST-LoRa's PDR values are accurate in complex network configurations. Overall, FAST-LoRa's PDR results have an average MAE of 0.940×10^{-2} and an average SDE of 1.498×10^{-2} across all network configurations. Similarly, Table 7 shows the MAE and SDE values for EE, with the trend being quite similar to that observed for the PDR. Both the MAE and SDE for EE are small across all network configurations, with an average MAE of 0.038 bits/mJ and an average SDE of 0.068 bits/mJ, demonstrating the consistency of FAST-LoRa's EE calculations. Even with higher network sizes, the error remains low, with slight increases in the MAE and SDE as network complexity grows. For instance, in networks with 500 EDs and 4 GWs, the MAE increases slightly to 0.055 bits/mJ and the SDE to 0.078 bits/mJ. In conclusion, Table 6 and Table 7 show the accuracy of FAST-LoRa and the consistency of the results across different network configurations, with MAE and SDE values remaining low enough to be considered reliable for practical use in transmission parameter optimization. These results

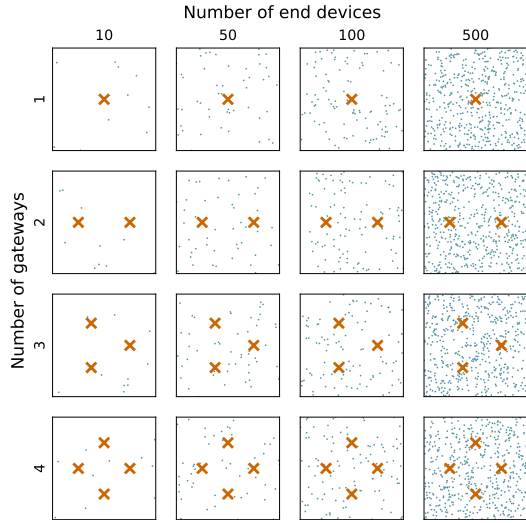


FIGURE 5. Different network configurations including end devices (black points) and gateways (orange crosses) within the 1×1 km area used for evaluation.

TABLE 6. MAE ($\times 10^{-2}$) and SDE (in brackets) values for the PDR calculated across all EDs and network configurations, with varying numbers of EDs and GWs.

EDs \ GWs	1	2	3	4
10	0.670 (1.182)	0.960 (1.509)	1.067 (1.730)	0.765 (1.390)
50	0.718 (1.207)	0.671 (1.109)	0.852 (1.426)	0.857 (1.404)
100	0.645 (1.174)	0.759 (1.237)	0.820 (1.367)	0.935 (1.451)
500	0.868 (1.464)	1.040 (1.590)	1.002 (1.526)	1.044 (1.579)

TABLE 7. MAE and SDE (in brackets) values for EE [bits/mJ] calculated across all EDs and network configurations, with varying numbers of EDs and GWs.

EDs \ GWs	1	2	3	4
10	0.022 (0.035)	0.034 (0.051)	0.040 (0.065)	0.042 (0.072)
50	0.021 (0.036)	0.039 (0.070)	0.050 (0.071)	0.049 (0.073)
100	0.018 (0.037)	0.030 (0.050)	0.042 (0.064)	0.055 (0.074)
500	0.022 (0.042)	0.038 (0.061)	0.048 (0.071)	0.055 (0.078)

confirm that FAST-LoRa is capable of providing efficient and precise PDR and EE values, even in complex and large-scale networks.

C. COMPUTATIONAL PERFORMANCE

After evaluating the performance of both approaches in terms of network-related metrics, this section focuses on the computational performance of FloRa and FAST-LoRa. Specifically, we compare their times when simulating networks with increasing numbers of end devices and gateways, to assess the computational cost and scalability of each framework. As Figure 6 shows, FAST-LoRa consistently achieves significantly lower simulation times than FloRa across all network sizes. For instance, with only 10 EDs, FAST-LoRa completes a simulation in approximately 0.6 milliseconds, while FloRa takes over 15 seconds. This gap becomes even more pronounced as the network grows: with 200 EDs, the scenario with the highest end device density evaluated, FloRa

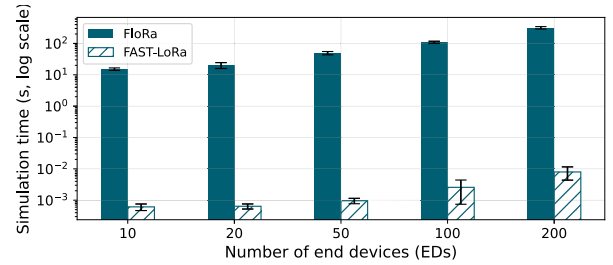


FIGURE 6. Comparison of the average execution time of FloRa and FAST-LoRa as the number of end devices increases. Results are averaged over simulations with 1 to 5 gateways and shown on a logarithmic scale.

exceeds 5 minutes, whereas FAST-LoRa remains under 8 milliseconds. This results in an acceleration of up to three orders of magnitude in favor of FAST-LoRa. This significant improvement is mainly due to FAST-LoRa's core design, which relies on an analytical model that obtains network performance metrics without the computational burden associated with simulating complex network behaviors that are not directly relevant to transmission parameter selection.

VI. CONCLUSION

In this work, we introduce FAST-LoRa, an efficient simulation framework designed to enable the fast and efficient optimization of transmission parameters in LoRaWAN networks. This is achieved through a statistical approach that focuses on computing core performance metrics rather than explicitly modeling multiple layers of communication protocols. To this end, packet-level interactions are replaced with analytical models, and gateway receptions are computed through matrix-based operations. By abstracting low-level protocol details, FAST-LoRa significantly reduces computational complexity, making it well suited for efficient exploration of transmission parameter configurations in large-scale LoRaWAN networks. Moreover, the FAST-LoRa framework, together with all experiments and results presented in this work, is publicly available, ensuring full reproducibility. Our evaluation demonstrates that FAST-LoRa achieves comparable accuracy to the well-established FloRa simulator, with a Mean Absolute Error (MAE) of 0.940×10^{-2} for the Packet Delivery Ratio (PDR) and 0.040 bits/mJ for Energy Efficiency (EE) in complex scenarios with interference and multi-gateway reception. Moreover, in terms of computational performance, FAST-LoRa completes simulations in a fraction of the time required by FloRa. In future work, we will focus on extending the framework to support a more comprehensive analysis of LoRaWAN networks. In particular, we plan to incorporate additional transmission parameters (e.g., bandwidth, coding rate, and frequency channels) and to exploit FAST-LoRa for large-scale studies such as gateway placement and network layout optimization. We will also include complementary validation with other simulation tools or real traces as a natural extension to further assess the generality and robustness of the framework.

ACKNOWLEDGMENT

Minor edits to this manuscript for clarity and correctness were supported using OpenAI's ChatGPT.

REFERENCES

- [1] Gym Documentation. Accessed: Apr. 8, 2025. [Online]. Available: <https://gymnasium.farama.org/#>
- [2] Juanaznarp94/fast-LoRa: Fast-LoRa Simulation Framework for LoRa Networks. Accessed: Mar. 11, 2025. [Online]. Available: <https://github.com/juanaznarp94/fast-lora>
- [3] LoRasim. Accessed: Dec. 20, 2023. [Online]. Available: <https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html>
- [4] Maartenweyn/Lpwansimulation. Accessed: Jan. 9, 2025. [Online]. Available: <https://github.com/maartenweyn/lpwansimulation/>
- [5] Pettingzoo Documentation. Accessed: Apr. 8, 2025. [Online]. Available: <https://pettingzoo.farama.org/index.html>
- [6] Rllib: Industry-Grade, Scalable Reinforcement Learning-Ray 2.44.1. Accessed: Apr. 8, 2025. [Online]. Available: <https://docs.ray.io/en/latest/rllib/index.html>
- [7] Stable-Baselines3 Docs-Reliable Reinforcement Learning Implementations-Stable Baselines3 2.6.1a0 Documentation. Accessed: Apr. 8, 2025. [Online]. Available: <https://stable-baselines3.readthedocs.io/en/master/>
- [8] Unict-Arslab/LWN-Simulator: A LoRaWAN Nodes' and Network Simulator That Works With a Real LoRaWAN Environment (such as Chirpstack) and Equipped With a Web Interface for Real-time Interaction. Accessed: Dec. 30, 2024. [Online]. Available: <https://github.com/UniCT-Arslab/LWN-Simulator>
- [9] L. Acosta-García, J. Aznar-Poveda, A. J. García-Sánchez, J. García-Haro, and T. Fahringer, "Dynamic transmission policy for enhancing LoRa network performance: A deep reinforcement learning approach," *Internet Things*, vol. 24, Dec. 2023, Art. no. 100974.
- [10] L. Acosta-García, J. Aznar-Poveda, A.-J. García-Sánchez, J. García-Haro, and T. Fahringer, "Proactive adaptation of data rate in mobile LoRa-based IoT devices using machine learning," in *Proc. IEEE 99th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2024, pp. 1–5.
- [11] L. Acosta-García, J. Aznar-Poveda, A. J. García-Sánchez, J. Holenstein, J. García-Haro, and T. Fahringer, "ADRL: A reconfigurable energy-efficient transmission policy for mobile LoRa devices based on reinforcement learning," *Internet Things*, vol. 31, May 2025, Art. no. 101577.
- [12] L. Alliance. (2022). *RP002-1.0.4 Regional Parameters*. [Online]. Available: <https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters>
- [13] M. A. M. Almuḥaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on LoRaWAN technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, p. 164, Jan. 2022.
- [14] M. A. M. Almuḥaya, W. A. Jabbar, N. Sulaiman, and A. H. A. Sulaiman, "An overview on LoRaWAN technology simulation tools," in *Proc. Int. Conf. reliable Inf. Commun. Technol.*, 2022, pp. 345–358.
- [15] L. Amichi, M. Kaneko, E. H. Fukuda, N. El Rachkidy, and A. Guittion, "Joint allocation strategies of power and spreading factors with imperfect orthogonality in LoRa networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3750–3765, Jun. 2020.
- [16] K. Anwar, T. Rahman, A. Zeb, I. Khan, M. Zareei, and C. Vargas-Rosales, "RM-ADR: Resource management adaptive data rate for mobile application in LoRaWAN," *Sensors*, vol. 21, no. 23, p. 7980, Nov. 2021.
- [17] N. Benkahla, H. Tounsi, S. Ye-Qiong, and M. Frikha, "Enhanced ADR for LoRaWAN networks with mobility," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1–6.
- [18] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proc. 19th ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, 2016, pp. 59–67.
- [19] M. C. Bor and T. Voigt. (2024). [Online]. Available: <https://mcbor.github.io/lorasim/>
- [20] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.
- [21] U. Coutaud, M. Heusse, and B. Tourancheau, "LoRa channel characterization for flexible and high reliability adaptive data rate in multiple gateways networks," *Computers*, vol. 10, no. 4, p. 44, Apr. 2021.
- [22] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in LoRaWAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.
- [23] H. Eigner, "Interference analysis of LoRaWAN systems," Master's thesis, Inst. Electrodynamics, Microw. Circuit Eng., TU Wien, Vienna, Austria, 2021.
- [24] A. Farhad, D.-H. Kim, S. Subedi, and J.-Y. Pyun, "Enhanced LoRaWAN adaptive data rate for mobile Internet of Things devices," *Sensors*, vol. 20, no. 22, p. 6466, Nov. 2020.
- [25] A. Farhad and J.-Y. Pyun, "HADR: A hybrid adaptive data rate in LoRaWAN for Internet of Things," *ICT Exp.*, vol. 8, no. 2, pp. 283–289, Jun. 2022.
- [26] S. Francisco, P. Pinho, and M. Luís, "Improving LoRa network simulator for a more realistic approach on LoRaWAN," in *Proc. Telecoms Conf. (ConfTELE)*, Feb. 2021, pp. 1–6.
- [27] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in LoRa networks," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 788–798.
- [28] P. Gkotsiopoulos, D. Zorbas, and C. Douligeris, "Performance determinants in LoRa networks: A literature review," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1721–1758, 3rd Quart., 2021.
- [29] P. Gschwandtner, J. J. Durillo, and T. Fahringer, "Multi-objective auto-tuning with insieme: Optimization and trade-off analysis for time, energy and resource usage," in *Euro-Par 2014 Parallel Processing*, F. Silva, I. Dutra, and V. Santos Costa, Eds., Cham, Switzerland: Springer, 2014, pp. 87–98.
- [30] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT traffic and its impact on LoRaWAN," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
- [31] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018.
- [32] S. Idris, T. Karunathilake, and A. Förster, "Survey and comparative study of LoRa-enabled simulators for Internet of Things and wireless sensor networks," *Sensors*, vol. 22, no. 15, p. 5546, Jul. 2022.
- [33] M. Jouhari, K. Ibrahim, J. B. Othman, and E. M. Amhoud, "Deep reinforcement learning-based energy efficiency optimization for flying LoRa gateways," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 6157–6162.
- [34] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," *ACM Trans. Sensor Netw.*, vol. 15, no. 2, pp. 1–35, May 2019.
- [35] (2019). *Omnnet++ Discrete Event Simulator*. [Online]. Available: <https://omnetpp.org/>
- [36] (2023). *Inet Framework*. [Online]. Available: <https://inet.omnetpp.org/>
- [37] D. Magrin, "Network level performances of a LoRa system," Master's thesis, Dept. Inf. Eng., University of Padua, Padua, Italy, 2016.
- [38] D. Magrin, M. Capuzzo, S. Romagnolo, and M. Luvisotto. (2024). *LoRaWAN NS-3 Module*. [Online]. Available: <https://github.com/signetlabdei/lorawan>
- [39] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.
- [40] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "A review of LoRaWAN simulators: Design requirements and limitations," in *Proc. Int. Multidisciplinary Inf. Technol. Eng. Conf. (IMITEC)*, Nov. 2019, pp. 1–6.
- [41] J. Mhatre and A. Lee, "Dynamic reinforcement learning based scheduling for energy-efficient edge-enabled LoRaWAN," in *Proc. IEEE Int. Perform., Comput., Commun. Conf. (IPCCC)*, Nov. 2022, pp. 412–413.
- [42] V. Moysiadias, T. Lagkas, V. Argyriou, A. Sarigiannidis, I. D. Moscholios, and P. Sarigiannidis, "Extending ADR mechanism for LoRa enabled mobile end-devices," *Simul. Model. Pract. Theory*, vol. 113, Dec. 2021, Art. no. 102388.
- [43] M. S. A. Muthanna, A. Muthanna, A. Rafiq, M. Hammoudeh, R. Alkanhel, S. Lynch, and A. A. Abd El-Latif, "Deep reinforcement learning based transmission policy enforcement and multi-hop routing in QoS aware LoRa IoT networks," *Comput. Commun.*, vol. 183, pp. 33–50, Feb. 2022.
- [44] (2024). *Ns-3 Network Simulator*. [Online]. Available: <https://www.nsnam.org/>

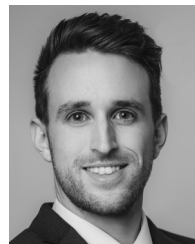
- [45] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?" in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
- [46] (2024). *LoRaWAN-Simple Rate Adaptation Recommended Algorithm*. [Online]. Available: <https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f515d14a1.pdf>
- [47] (2019). *Sx1272*. [Online]. Available: <https://www.semtech.com/products/wireless-RF/lorconnect/sx1272#documentation>
- [48] (2020). *Sx1276/77/78/79 Datasheet*. [Online]. Available: <https://www.semtech.com/products/wireless-RF/lorconnect/sx1276#documentation>
- [49] (2023). *Implementing Adaptive Data Rate*. [Online]. Available: <https://learn.semtech.com/mod/book/view.php?id=174&chapterid=162>
- [50] R. Serati, B. Teymuri, N. A. Anagnostopoulos, and M. Rasti, "ADR-lite: A low-complexity adaptive data rate scheme for the LoRa network," in *Proc. 18th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2022, pp. 296–301.
- [51] M. Slabicki. (2022). *Flora—A Framework for LoRa Simulations With Omnet++*. Accessed: 18. [Online]. Available: <https://flora.aalto.fi/>
- [52] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of LoRa networks for dense IoT deployments," in *Proc. NOMS - IEEE/IFIP Netw. Operations Manage. Symp.*, Apr. 2018, pp. 1–9.
- [53] S. Sotiriadis, N. Bessis, E. Asimakopoulou, and N. Mustafee, "Towards simulating the Internet of Things," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 444–448.
- [54] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 371–388, 1st Quart., Jan. 2020.
- [55] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: Toward an intelligent resource allocation approach for LoRaWAN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
- [56] V. Toro-Betancur, G. Premsankar, M. Slabicki, and M. Di Francesco, "Modeling communication reliability in LoRa networks with device-level accuracy," in *Proc. IEEE Conf. Comput. Commun.*, France, May 2021, pp. 1–10.
- [57] Z. Xu, J. Luo, Z. Yin, S. Wang, C. Chen, J. Lin, R. Xiong, and T. He, "X-MAC: Achieving high scalability via imperfect-orthogonality aware scheduling in LPWAN," in *Proc. IEEE 30th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2022, pp. 1–11.
- [58] Y. Yazid, A. Guerrero-González, I. Ez-Zazi, A. E. Oualkadi, and M. Arioua, "A reinforcement learning based transmission parameter selection and energy management for long range Internet of Things," *Sensors*, vol. 22, no. 15, p. 5662, Jul. 2022.
- [59] F. Yu, X. Zheng, Y. Ma, L. Liu, and H. Ma, "Resolve cross-channel interference for LoRa," in *Proc. IEEE 44th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2024, pp. 1027–1038.
- [60] X. Zhang, Z. Lin, S. Gong, B. Gu, and D. Niyato, "Multiagent reinforcement learning with an attention mechanism for improving energy efficiency in LoRa networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 4152–4157.



LAURA ACOSTA-GARCÍA received the B.Sc. and M.Sc. degrees in telecommunications engineering from the Technical University of Cartagena, Spain, in 2021 and 2023, respectively. She is currently pursuing the Ph.D. degree with the Department of Information and Communication Technologies, UPCT. Her research interests include wireless networks, the Internet of Things (IoT), machine learning, and reinforcement learning. She was awarded the Research Activity Initiation Scholarship by Santander Bank.



JUAN AZNAR-POVEDA received the Ph.D. degree in telecommunications engineering from the Technical University of Cartagena, Spain, in 2022. He is currently a Postdoctoral Researcher with the Distributed and Parallel Systems Group, University of Innsbruck, Austria. His research interests include distributed systems, distributed databases, artificial intelligence, reinforcement learning, and wireless communications. He was awarded the "Liberalization of Telecommunications" Prize (National Level) for the Best B.Sc. Thesis, in 2016.



FABIAN MARGREITER received the B.Sc. degree in computer science from the University of Innsbruck, Austria, in 2025. Since 2013, he led the Software and Electronics Engineering Department, Rothlehner Arbeitsbühnen GmbH, a German Construction Machinery Manufacturer. In 2022, he joined the BOC Group, where he is currently a Cloud and DevOps Engineer. His research interests include distributed systems, cloud computing, and artificial intelligence.



ANTONIO-JAVIER GARCÍA-SÁNCHEZ received the Ph.D. degree in industrial engineering from the Technical University of Cartagena, Spain, in 2005. He is currently a Full Professor of telecommunications engineering with the Technical University of Cartagena. He has been a co-author of more than 100 conference papers and journal articles, the head of several EU research projects in the field of communication networks and optimization, and an inventor/co-inventor of 12 patents or utility models. He is also a reviewer of several journals listed in ISI-JCR. His main research interests include wireless sensor networks (WSNs), streaming services, artificial intelligence, the Internet of Things (IoT), and nanonetworks. He has been a TPC member or the chair in about 40 international congresses or workshops.



JOAN GARCÍA-HARO (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Universitat Politècnica de Catalunya, Spain, in 1989 and 1995, respectively. He was a Visiting Scholar with Queen's University, Kingston, ON, Canada, from 1991 to 1992, and Cornell University, Ithaca, NY, USA, from 2010 to 2011. He is currently a Full Professor with the Technical University of Cartagena (UPCT), Spain. He is the author or co-author of more than 70 journal articles, mainly in the fields of switching, wireless networking, and performance evaluation. He received the Honorable Mention for the IEEE Communications Society Best Tutorial Paper Award, in 1995. He served as the Editor-in-Chief for the IEEE Global Communications Newsletter, including *IEEE Communications Magazine*, from April 2002 to December 2004. He has been a Technical Editor of *IEEE Communications Magazine*, from March 2001 to December 2011.



THOMAS FAHRINGER (Member, IEEE) received the Ph.D. degree from Vienna University of Technology, in 1993. Since 2003, he has been a Full Professor of computer science with the Institute of Computer Science, University of Innsbruck, Austria. His main research interests include software architectures, programming paradigms, compiler technology, performance analysis, and prediction for parallel and distributed systems.



JOSÉ LORENTE LÓPEZ was born in Murcia, Spain, in 2000. He received the degree in telecommunications systems engineering from the Technical University of Cartagena (UPCT), Spain, in 2022, and the M.Sc. degree in applied artificial intelligence from Universidad Carlos III de Madrid, Madrid, Spain, in 2023. He is currently pursuing the Ph.D. degree in applied artificial intelligence with the Telecommunications Sector, UPCT. He is the Data Analyst Lead with Capgemini,

focusing on big data and generative AI. His research interests include the modeling of radio wave propagation in urban environments and machine learning applications.



JOSÉ-VÍCTOR RODRÍGUEZ was born in Murcia, Spain, in 1975. He received the degree in telecommunications engineering from the Universidad Politécnica de Valencia (UPV), Spain, in 2001, after doing his graduate thesis at Lund Institute of Technology, Lund University, Sweden, in collaboration with Ericsson Mobile Communications AB, and the Ph.D. degree in communications engineering from the Technical University of Cartagena (UPCT), Spain, in 2006. In 2002, he joined the Department of Information Technologies and Communications, UPCT, where he is currently a Full Professor. His research interests include the modeling of radio-wave propagation in the mobile environment, with an emphasis on multiple-building diffraction.

...