


# An Alternative Particle Filter-Driven ADR for Mobile Devices in LoRaWAN Networks

Geraldo A. Sarmiento Neto   [ Universidade Federal do Piauí | [geraldosarmiento@ufpi.edu.br](mailto:geraldosarmiento@ufpi.edu.br) ]

Thiago A. Ribeiro da Silva  [ Instituto Federal do Maranhão | [thiago.allisson@ufpi.edu.br](mailto:thiago.allisson@ufpi.edu.br) ]


Artur F. da S. Veloso  [ Universidade Federal do Piauí | [arturfdasveloso@gmail.com](mailto:arturfdasveloso@gmail.com) ]

Pedro Felipe de Abreu  [ Universidade Federal do Piauí | [pedroffda@ufpi.edu.br](mailto:pedroffda@ufpi.edu.br) ]

Luis H. de O. Mendes  [ Universidade Federal do Piauí | [luishenriqueom@ufpi.edu.br](mailto:luishenriqueom@ufpi.edu.br) ]

Ricardo A. L. Rabelo  [ Universidade Federal do Piauí | [ricardoalr@ufpi.edu.br](mailto:ricardoalr@ufpi.edu.br) ]

J. Valdemir dos Reis Jr  [ Universidade Federal do Piauí | [valdemirreis@ufpi.edu.br](mailto:valdemirreis@ufpi.edu.br) ]

 Universidade Federal do Piauí, Campus Universitário Ministro Petrônio Portella, Ininga, Teresina, PI, 64049-550, Brazil.

**Received:** 04 October 2024 • **Accepted:** 10 February 2025 • **Published:** 19 May 2025

**Abstract** LoRaWAN is a leading LPWAN technology for Internet of Things applications, known for its long-range communication and low-power consumption. Its ADR mechanism optimizes performance by adjusting transmission parameters, such as spreading factor and transmission power, based on network conditions. However, ADR faces significant limitations in environments with mobile end devices, where fluctuating signal quality leads to increased packet loss, inefficient energy usage, and reduced communication reliability. To address these challenges, PF-ADR, an alternative ADR scheme, is proposed for LoRaWAN networks with mobile devices. PF-ADR employs a particle filter to estimate a representative SNR value, maintaining multiple hypotheses of the communication channel state to enable more precise parameter adjustments. Simulations conducted under various scalability conditions reveal that PF-ADR achieves up to 29.5% higher packet delivery ratio compared to M-ADR, and 52.17% more than the standard ADR, while demonstrating a 43.19% improvement in energy efficiency over MB-ADR. Additionally, the algorithm reduces packet loss due to signal degradation while maintaining scalable performance in large networks. These results highlight the potential of PF-ADR to enhance communication reliability and energy efficiency in dynamic mobile environments.

**Keywords:** ADR, Internet of Things, LoRaWAN, Mobility, Particle Filter

## 1 Introduction

The Internet of Things (IoT) is rapidly transforming various sectors, enabling the integration of physical devices into ubiquitous networks for enhanced monitoring, control, and automation [Almuhaya *et al.*, 2022]. A key aspect of IoT is the ability of devices to communicate over long distances while maintaining low power consumption, especially in remote or resource-constrained environments. This necessity led to the development of Low Power Wide Area Networks (LPWAN), a class of wireless communication networks designed to serve IoT devices with extended battery life and coverage [Milarokostas *et al.*, 2023].

One of the most prominent LPWAN technologies is Long Range Wide Area Network (LoRaWAN), a communication protocol specifically designed for IoT applications requiring long-range, low-power communication capabilities. LoRaWAN operates in unlicensed frequency bands, making it a low-cost solution for deploying IoT networks. Its architecture enables End Devices (EDs) to transmit data to Gateways (GWs), which in turn relay the information to a central Network Server (NS) for further processing [Kufakunesu *et al.*, 2020]. This structure makes LoRaWAN particularly suitable for a wide range of IoT applications, including smart metering, environmental monitoring, and asset tracking [Severino

*et al.*, 2023].

However, the spectrum efficiency and the performance of a LoRaWAN network are directly related to the proper adjustment of a set of transmission parameters, including Spreading Factor (SF), Bandwidth (BW), Transmission Power (TP), and Coding Rate (CR). The SF influences the trade-off between data rate and range, with higher SFs providing greater range but slower data rates. BW impacts channel capacity, where narrower bandwidths support longer range but lower throughput. TP ensures signal reach while balancing energy consumption and minimizing interference. CR enhances error correction, improving reliability at the cost of reduced data rate [Sallum *et al.*, 2020].

A critical resource of LoRaWAN is the Adaptive Data Rate (ADR) mechanism, which dynamically adjusts the transmission parameters to optimize data transmission based on the measured Signal-to-Noise Ratio (SNR) between the device and gateway. While the ADR mechanism effectively manages the trade-off between energy efficiency and communication range for static devices, its performance degrades significantly in mobile environments [Farhad *et al.*, 2022]. Devices that move frequently, such as in pet tracking [Francis and Mohd Shah, 2022] or vehicular monitoring [Camarillo-Escobedo *et al.*, 2022] applications, experience fluctuating signal quality due to the constantly changing distance and

environmental conditions between the device and the gateway [Benkahla *et al.*, 2021b]. In such cases, the traditional ADR algorithm struggles to adapt, leading to inefficient data transmission and packet losses.

To address this limitation, this work proposes Particle Filter-based Adaptive Data Rate (PF-ADR), an alternative solution to the LoRaWAN standard ADR mechanism, specifically tailored for mobile applications. The proposed solution employs the Sequential Importance Resampling (SIR) Particle Filter, a Monte Carlo method for state estimation in non-linear and non-Gaussian systems, to determine the representative SNR value more accurately under varying mobile conditions [Ahwiadi and Wang, 2020]. By maintaining a set of particles that represent possible states of the communication channel, PF-ADR continuously refines its estimates, allowing for accurate adjustments in spreading factor and transmission power.

To evaluate the effectiveness of the proposed solution, a simulated network environment was set up. The performance of PF-ADR was compared against three alternative ADR schemes: the standard LoRaWAN ADR, MB-ADR [Sarmiento Neto *et al.*, 2024], and M-ADR [Farhad *et al.*, 2023], all of which share relevant characteristics that align with the goals of the proposed solution. Simulations were conducted with varying numbers of mobile end devices, ranging from 200 to 1000, over a 24-hour period. The network was composed of a single gateway and mobile end devices moving randomly within a 5 km radius. The main contributions of this paper are as follows:

- **Particle Filtering for Dynamic Parameter Selection:** A novel ADR algorithm that integrates particle filtering to optimize data rate selection for mobile end devices.
- **Improved Energy Efficiency:** The proposed method reduces energy consumption by avoiding unnecessary retransmissions and minimizing the airtime of packets, particularly in mobile scenarios.
- **Enhanced Reliability:** The scheme reaches higher average Packet Delivery Ratio (PDR) by dynamically adjusting transmission parameters in response to frequent changes in the communication environment.
- **Scalability for Large IoT Networks:** The PF-ADR algorithm demonstrates scalability in simulations, showing improved performance across a range of device densities and network conditions.

The remainder of this paper is organized as follows. Section 2 reviews ADR mechanisms, highlighting their strengths and limitations. Section 3 discusses the LoRaWAN specification and its key technologies, including the standard ADR mechanism. Section 4 covers the SIR Particle Filter and its applications. Section 5 details the proposed PF-ADR scheme. Section 6 describes the research methodology, including the simulation setup and evaluation metrics. Section 7 presents the results and analysis of the proposed solution. Finally, Section 8 concludes the paper and suggests directions for future research.

## 2 Related Work

Several studies introduced solutions aimed at improving the ADR mechanism, each with a focus on distinct performance aspects. Some of these approaches focus on refining the method used to estimate the representative SNR that guides the adjustment of transmission parameters [Sarmiento Neto *et al.*, 2024; Kufakunesu *et al.*, 2022; Moysiadis *et al.*, 2021]. Others suggest a more in-depth intervention in the network resource allocation mechanism, completely redefining the core of the ADR [Wang *et al.*, 2024; Farhad *et al.*, 2023; Benkahla *et al.*, 2021b]. These modifications can be implemented on either the ED or NS side, depending on the specific requirements of the solution [Anwar *et al.*, 2021b]. A summary of the related work considering alternative ADR schemes is displayed in Table 1. It takes into account the scheme name, the method used, the scope of the implementation, and whether the work considers scenarios involving the end device mobility.

In Farhad *et al.* [2023], the authors present a novel Mobility Adaptive Data Rate (M-ADR) mechanism for LoRaWAN, aiming to improve resource allocation in mobile IoT applications. The strength of the proposal lies in the use of a Kalman Filter to estimate the SNR accurately, allowing for better adjustment of the SF and TP. This results in an improvement in the Packet Success Ratio (PSR) and reduced energy consumption through fewer retransmissions, compared to existing methods. However, the approach relies on accurate position data of the end devices, which may not always be feasible in real-world scenarios. Additionally, the Kalman Filter, while effective in relatively stable conditions, may not perform optimally in highly dynamic environments. In such cases, a particle filter-based method could be more suitable, as it can handle non-linear, non-Gaussian environments by tracking multiple hypotheses of the system state [Ullah *et al.*, 2020], potentially offering more robust performance in networks with mobile devices where signal strength fluctuates unpredictably.

Authors in Anwar *et al.* [2021a] propose Resource Management Adaptive Data Rate (RM-ADR), an adaptive approach to optimize data transmission in LoRaWAN environments. The scheme focuses on managing resources such as SF and TP based on the mobility of end devices, with the goal of improving PDR and reducing energy consumption. Despite demonstrating promising results, the scheme presents limitations in considering large-scale deployments with highly dynamic mobility scenarios, where interference and network scalability issues could significantly affect performance. Additionally, the proposal relies heavily on accurate signal strength data, which might not always be reliably available in real-world IoT applications.

Authors in Lodhi *et al.* [2022] present an innovative approach to address the issue of packet collisions caused by simultaneous transmissions, a significant challenge in dense IoT networks. By proactively monitoring destructive concurrent transmissions and assigning non-conflicting transmission windows, Nondestructive Adaptive Data Rate (ND-ADR) enhances network reliability and scalability. Results demonstrate improvements in PSR and reduced energy consumption when compared to other ADR mechanisms. This

**Table 1.** Summary of Related Work.

| <i>Reference</i>                    | <i>Scheme</i> | <i>Method</i>                      | <i>Scope</i>   | <i>ED Mobility</i> |
|-------------------------------------|---------------|------------------------------------|----------------|--------------------|
| Sarmiento Neto <i>et al.</i> [2024] | MB-ADR        | Median and Interquartile Range     | NS-side        | ✓                  |
| Kufakunesu <i>et al.</i> [2022]     | FL-ADR        | Fuzzy Logic                        | NS-side        |                    |
| Moysiadis <i>et al.</i> [2021]      | LR-ADR        | Linear Regression                  | ED and NS-side | ✓                  |
| Wang <i>et al.</i> [2024]           | TA-ADR        | Time Slot Allocation               | NS-side        |                    |
| Farhad <i>et al.</i> [2023]         | M-ADR         | Kalman Filter                      | ED and NS-side | ✓                  |
| Benkahla <i>et al.</i> [2021b]      | E-ADR         | Hidden Markov Model                | NS-side        | ✓                  |
| Anwar <i>et al.</i> [2021a]         | RM-ADR        | Sensitivity Threshold              | ED and NS-side | ✓                  |
| Lodhi <i>et al.</i> [2022]          | ND-ADR        | Concurrent Transmissions Detection | NS-side        |                    |
| Lodhi <i>et al.</i> [2024]          | TinyML        | Tiny Machine Learning              | ED and NS-side | ✓                  |
| Ilahi <i>et al.</i> [2020]          | LoRaDRL       | Deep Reinforcement Learning        | GW-side        | ✓                  |
| Farhad and Pyun [2023]              | AI-ERA        | Deep Neural Network                | ED and NS-side | ✓                  |
| Azizi <i>et al.</i> [2022]          | MIX-MAB       | Multi-armed Bandit Problem         | ED-side        |                    |
| This work                           | PF-ADR        | Particle Filter                    | NS-side        | ✓                  |

is achieved by minimizing retransmissions and better managing transmission parameters like spreading factor and power. However, the complexity introduced by the need to profile end devices and dynamically allocate transmission times may increase overhead, potentially negating some of the performance benefits.

Numerous studies have explored the implementation of machine learning techniques to enhance the precision of transmission parameter allocation [Lodhi *et al.*, 2024; Ilahi *et al.*, 2020]. Nonetheless, these methodologies can lead to significant bottlenecks in practical applications [Kufakunesu *et al.*, 2020], particularly in the context of supervised learning, which necessitates extensive phases of data collection, training, and labeling. In Farhad and Pyun [2023], a proactive resource allocation method using a Deep Neural Network (DNN) model for LoRaWAN networks is proposed. The approach dynamically adjusts SF for both static and mobile IoT applications, improving the PSR by 32% in static environments and 28% in mobile scenarios compared to standard ADR. Despite these findings, the approach also introduces limitations, such as dependence on a pre-generated dataset demanding 10 days of training, and involving a two-tier data labeling process. This complexity in data preparation could hinder the real-world applicability of machine learning-based solutions in dynamic environments. The system may also struggle with scalability, as the computational complexity of the DNN increases with the number of devices and data variations, limiting its effectiveness in large-scale scenarios.

MIX-MAB is proposed in Azizi *et al.* [2022], an algorithm that combines Exponential weights for Exploration and Exploitation (EXP3) with Successive Elimination (SE) techniques to improve resource allocation in LoRaWAN networks. MIX-MAB enables end devices to configure transmission parameters in a distributed manner, reducing network congestion, enhancing PDR, and lowering energy consumption. Through simulations, the algorithm demonstrates superior performance over existing schemes, particularly in dense network environments, while achieving faster convergence times. Nevertheless, the scheme may still face challenges in dynamic environments, and the computational overhead required for probability updates could limit its scalability

in networks with constrained end devices.

PF-ADR effectively addresses many limitations observed in existing approaches by providing a robust mechanism for managing packet collisions in dynamic IoT environments. Unlike other methods that heavily depend on accurate position data or assume static conditions, PF-ADR leverages the strengths of particle filtering to track multiple hypotheses about the network state. This adaptability enables it to excel in fluctuating signal conditions and varying device mobility, ensuring more reliable performance across diverse scenarios.

### 3 Background

This section outlines the main aspects of the LoRaWAN technology, followed by a discussion on important transmission parameters like spreading factor and coding rate. Finally, the ADR mechanism is described, focusing on its role in optimizing network performance.

#### 3.1 LoRaWAN

LoRaWAN is a communication protocol for LPWANs, optimized for energy-efficient, long-range communications between connected devices, making it ideal for the IoT. Operating in an unlicensed sub-gigahertz ISM (Industrial, Scientific, and Medical) band, LoRaWAN reduces operational costs and enables widespread deployment without the need for licensing fees [Kufakunesu *et al.*, 2020].

The LoRaWAN protocol operates at the data link layer and defines how devices communicate over the physical layer, manage their energy consumption, and exchange data with gateways and network servers. This layer is central to LoRaWAN design, as it coordinates the communication between the end devices and the gateways, ensuring that messages are transmitted efficiently and reliably [Moysiadis *et al.*, 2021].

One of the main characteristics of the LoRaWAN MAC layer is its use of ALOHA-based channel access, which allows for simple, asynchronous communication between devices. This method reduces the complexity of managing access to the shared spectrum, at the cost of potential collisions,

but it is a trade-off that ensures low-power consumption [Milarokostas *et al.*, 2023]. Furthermore, the MAC layer handles the scheduling of data transmissions, adapts the data rates, and controls the transmission power of devices based on their distance from the gateway and the quality of the communication link, which is essential for maintaining energy efficiency and network scalability.

At the physical layer, LoRaWAN relies on Long Range (LoRa), a proprietary technology developed by Semtech [Jouhari *et al.*, 2023]. Its modulation employs the Chirp Spread Spectrum (CSS) technique, which allows the signal spreading over a wide frequency band, making it robust against interference and capable of supporting long-range communication even in noisy environments [Vangelista *et al.*, 2023; Baruffa and Rugini, 2022]. CSS works by encoding data using chirp signals, which are frequency-modulated signals that increase or decrease in frequency over time. This method allows LoRa to achieve high sensitivity at the receiver, enabling communication over distances of several kilometers (km) [Li *et al.*, 2024].

The LoRaWAN architecture is built around a star-of-stars topology, where the key components are the end devices, gateways, and a network server [Magrin *et al.*, 2020], as shown in Figure 1. End devices are typically low-power sensors or actuators that gather data or perform actions based on received commands. These devices communicate directly with gateways using the LoRa modulation scheme. Gateways, which act as bridges between the end devices and the network server, do not process the data themselves but simply forward the data packets received from the end devices to the network server via IP-based backhaul technologies, such as Ethernet, cellular networks, or Wi-Fi. The gateways operate in a transparent mode, where they listen for LoRa transmissions and relay both uplink, device-to-server, and downlink, server-to-device, messages [Marais *et al.*, 2023].

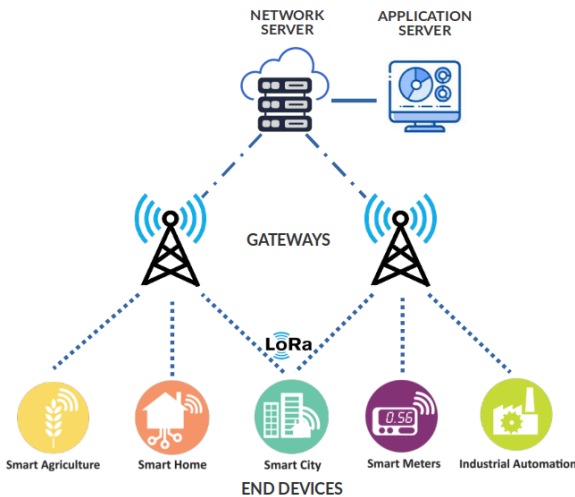


Figure 1. The architecture of LoRaWAN.

The network server is a critical component responsible for managing the entire network communication, ensuring that data from end devices is correctly processed and delivered. It handles various tasks such as packet deduplication, scheduling downlink transmissions, and implementing security measures like message integrity checks and encryption to protect data integrity and confidentiality [Milarokostas *et al.*, 2023].

In addition to the network server, the application server is responsible for processing the data received from the end devices and translating it into actionable information for specific IoT applications. For instance, in a smart metering scenario, the application server would receive energy consumption data from sensors, process it, and present it in a user-friendly interface or trigger alerts based on predefined thresholds [Kufakunesu *et al.*, 2020]. Although the application server is not part of the LoRaWAN architecture, it is essential for managing and processing application-specific data. The separation between the network server and the application server allows for flexibility and scalability in the network, enabling multiple applications to operate independently while sharing the same network infrastructure.

To accommodate different application requirements, LoRaWAN defines three classes of devices: A, B, and C, each offering different trade-offs between energy efficiency and communication latency [Loubany *et al.*, 2020]. Class A devices are the most energy-efficient, operating in an asynchronous mode where the device initiates communication and opens brief receive windows after each uplink transmission. This class is suitable for applications that require low power consumption, such as battery-operated sensors that send data periodically. Class B devices introduce scheduled downlink windows, allowing the network to communicate with the device at predetermined times. This class strikes a balance between power efficiency and responsiveness, making it useful for applications that need more frequent communication without sacrificing too much battery life. Class C devices, on the other hand, keep their receive windows open continuously, offering the lowest latency at the cost of increased power consumption. This class is ideal for applications that require real-time communication, such as smart street lighting or industrial control systems [Almuhaya *et al.*, 2022].

Consequently, LoRaWAN is a highly flexible and scalable LPWAN protocol, offering a wide range of features designed to balance energy efficiency, long-range communication, and low operational costs. Its layered approach, robust physical modulation using CSS, flexible network architecture, and differentiated device classes make it well-suited for the varied demands of IoT applications, from simple environmental monitoring to more complex, real-time industrial automation.

### 3.2 Transmission Parameters

The effectiveness and reliability of LoRaWAN transmissions depend on several transmission parameters, including the spreading factor, transmission power, bandwidth, and coding rate. Each of these parameters influences the balance among range, data rate, and energy efficiency, which is important for optimizing the performance of the network [Magrin *et al.*, 2020].

The spreading factor is a fundamental parameter that impacts both the data rate and the communication range. It determines the number of chips per symbol,  $C_n$ , thereby influencing the degree to which the signal is spread over time. Equation 1 shows how the SF value can be determined, where  $C_n = \{128, 256, 512, 1024, 2048, 4096\}$ ,  $R_c$  is the

chip rate, and  $R_s$  is the symbol rate [Jouhari *et al.*, 2023].

$$SF = \log_2(C_n) = \log_2\left(\frac{R_c}{R_s}\right) \quad (1)$$

Higher spreading factors, such as SF12, result in extended transmission times, lower data rates, and enhanced sensitivity, enabling communication over longer distances or in environments with high interference. However, these higher spreading factors also increase power consumption per transmitted bit and prolong airtime [Ilahi *et al.*, 2020], which may lead to a greater chance of collisions in densely populated networks. On the other hand, lower spreading factors, such as SF7, allow for higher data rates and shorter airtime, but at the expense of reduced range and sensitivity [Kufakunesu *et al.*, 2020]. The selection of the appropriate spreading factor is therefore a compromise between achieving long-range communication and maintaining efficient data throughput.

The transmission power of the device is another important parameter, as it determines the reach and reliability of the signal. TP refers to the energy used by the LoRa device to transmit signals, typically measured in dBm (decibel-milliwatts). Increasing transmission power enhances the signal's ability to cover longer distances or penetrate physical barriers, improving coverage in challenging environments such as urban or indoor settings [Almuhaya *et al.*, 2022]. However, higher transmission power results in increased energy consumption, which is a significant consideration for battery-powered IoT devices.

Another significant transmission parameter is the bandwidth, defining the frequency spectrum width used for transmission. In LoRa end devices, BW typically ranges from 125 kHz to 500 kHz [Almuhaya *et al.*, 2022]. A wider bandwidth allows for higher data rates, as more information can be transmitted within the same time frame. However, wider bandwidths are more prone to noise and interference, which can degrade the SNR and reduce communication reliability in noisy environments. Conversely, narrower bandwidths offer improved sensitivity and can extend the communication range, but at the cost of slower data transmission.

Finally, coding rate affects the error resilience of the transmission. LoRa technology employs Forward Error Correction (FEC) to help mitigate the effects of interference, noise, or signal degradation, and the coding rate indicates the portion of the transmitted data that is used for error correction [Sallum *et al.*, 2020]. Coding rates are typically expressed as ratios from the set  $\{\frac{4}{5}, \frac{4}{6}, \frac{4}{7}, \frac{4}{8}\}$ , where a higher denominator corresponds to more robust error correction. While higher coding rates improve the reliability of communication in difficult conditions, they also increase the amount of redundant data transmitted, thereby reducing the effective data rate. Lower coding rates, in contrast, provide less redundancy but allow for faster data transmission.

In summary, each parameter involves trade-offs that affect data rate, range, energy consumption, and reliability. As demonstrated in Equation 2, the LoRa bit rate  $R_b$  is directly impacted by SF, BW, and CR [Kufakunesu *et al.*, 2020]. Consequently, optimizing these parameters is essential to achieve the desired performance for specific IoT applications, whether the objective is to maximize coverage, con-

serve battery life, or ensure stable communication in noisy or interference-prone settings.

$$R_b = SF \cdot \frac{BW}{2^{SF}} \cdot CR \quad (2)$$

### 3.3 Adaptive Data Rate

Adaptive Data Rate is a vital mechanism in LoRaWAN networks that optimizes data transmission by dynamically adjusting transmission parameters based on network conditions and the device environment. The primary objective of ADR is to enhance the efficiency of communication while minimizing energy consumption and ensuring reliable connectivity for IoT devices [Kufakunesu *et al.*, 2020]. By adapting the spreading factor and the transmission power, ADR enables devices to maintain a balance between data rate and range, ultimately prolonging battery life and improving overall network performance.

The ADR scheme operates based on the principle of measuring the SNR at the gateway, which serves as an indicator of the link quality between the end device and the network. When a device joins the network, it begins transmitting data with default settings. As it continues to communicate, the gateway receives the data packets and calculates the SNR, which reflects the quality of the received signal. This information is critical for the ADR algorithm to assess whether adjustments to the device transmission parameters are necessary [Jouhari *et al.*, 2023].

If the SNR is found to be suitable, the ADR algorithm may decide to reduce the spreading factor, allowing for faster data rates and reducing the airtime of messages. Conversely, if the SNR is low, indicating a weaker link, the algorithm may increase the spreading factor to improve sensitivity and extend the range of the communication. Transmission power adjustments are also essential in this process; increasing the power can help improve the communication range, while reducing it conserves energy. The ADR mechanism effectively integrates these adjustments to ensure that devices can adapt to changing conditions in real time [Sarmiento Neto *et al.*, 2024].

The scheme executed on the server side involves several steps to determine the optimal settings for SF and TP based on the measured SNR, as denoted in the Algorithm 1. It begins by collecting  $M$  SNR measurements, creating an SNR list, as denoted in the lines 1-3. The algorithm identifies the maximum SNR from this list, keeping it in the variable  $SNR_m$ , and stores in the variable  $SNR_{req}$  the required SNR to demodulate a packet according to its current SF, as expressed in the lines 4-5. The difference between the maximum SNR and the required SNR for demodulation, adjusted by the device margin, establishes the variable  $SNR_{margin}$ , denoted in the line 6. This margin is then used to determine  $N_{steps}$  in the line 7. This variable is used in an iterative process for fine-tuning the SF and TP values, as shown in lines 8-16, aiming to maintain optimal performance and adapt to the varying conditions of the radio environment. Finally, in the line 17, the server sends the *LinkADRReq* MAC command to the device containing the updated SF and TP settings.

**Algorithm 1:** LoRaWAN Adaptive Data Rate.

---

**Input:**  $SF \in [7, 12]$ ,  $TP \in [2, 14]$ ,  
 $M = 20$ ,  $device_{margin} = 10$

```

1  $SNRlist \leftarrow \emptyset$ 
2 for  $i \leftarrow 0$  to  $M - 1$  do
3    $SNRlist[i] \leftarrow getSNR(i)$ 
4  $SNR_m \leftarrow getMax(SNRlist)$ 
5  $SNR_{req} \leftarrow getDemodulationFloor(SF)$ 
6  $SNR_{margin} \leftarrow SNR_m - SNR_{req} - device_{margin}$ 
7  $N_{steps} \leftarrow int(SNR_{margin}/3)$ 
8 while  $N_{steps} > 0$  and  $SF > SF_{min}$  do
9    $SF \leftarrow SF - 1$ 
10   $N_{steps} \leftarrow N_{steps} - 1$ 
11 while  $N_{steps} > 0$  and  $TP > TP_{min}$  do
12    $TP \leftarrow TP - 2$ 
13    $N_{steps} \leftarrow N_{steps} - 1$ 
14 while  $N_{steps} < 0$  and  $TP < TP_{max}$  do
15    $TP \leftarrow TP + 2$ 
16    $N_{steps} \leftarrow N_{steps} + 1$ 
17  $sendLinkADRReq(SF, TP)$ 

```

---

## 4 SIR Particle Filter

The SIR Particle Filter derives its name from two key concepts: sequential importance sampling and resampling. The importance sampling aspect comes from how the filter assigns weights to each particle, or state hypothesis, according to its likelihood given the observed data. These weights reflect how well a particle represents the current state of the system. However, as the algorithm progresses, some particles may carry more importance than others, leading to particle degradation, where the estimation becomes less accurate due to the dominance of a few particles. To address this, the resampling step is introduced, where particles with higher weights are replicated while those with lower weights are discarded. This resampling ensures that the particle set remains representative of the system state, even as the system evolves [Ahwiadi and Wang, 2020].

The particle filter approximates the posterior probability distribution of a hidden state using a set of particles, each of which represents a possible state of the system at a given time. These particles are propagated over epochs and assigned weights based on their likelihood derived from the observed data. The main advantage of the SIR Particle Filter lies in its ability to handle complex models and measurements without restrictive assumptions about the system dynamics or noise characteristics [Varsi et al., 2020].

The filter processes state estimates iteratively over time, updating the state with each new observation. It employs a series of steps that include initialization, prediction, weighting, normalization, and resampling [Gyenes and Szádeczky-Kardoss, 2021]. At the initialization stage, a set of  $N$  particles is drawn from the prior distribution of the state variable, denoted as  $p(\mathbf{x}_0)$ . Each particle is assigned an equal initial weight:

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \quad w_0^{(i)} = \frac{1}{N}, \quad i = 1, 2, \dots, N. \quad (3)$$

As time progresses, the filter performs prediction for each particle, utilizing a state transition model  $p(\mathbf{x}_t|\mathbf{x}_{t-1})$  to propagate the state of each particle:

$$\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}), \quad i = 1, 2, \dots, N. \quad (4)$$

This step incorporates the inherent dynamics of the system and the associated process noise. Following the prediction step, the filter updates the weights of each particle based on the likelihood of observing the current measurement  $\mathbf{y}_t$ . The weight update rule [Bai et al., 2023] is given by:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(\mathbf{y}_t|\mathbf{x}_t^{(i)}), \quad i = 1, 2, \dots, N \quad (5)$$

in which  $w_t^{(i)}$  represents the weight of the  $i$ -th particle at time  $t$ , which is updated according to how well the predicted state  $\mathbf{x}_t^{(i)}$  matches the observation  $\mathbf{y}_t$ . This step effectively quantifies how well each particle predicted state aligns with the actual observed data, thereby allowing the filter to refine its estimate of the state variable.

Subsequently, the weights are normalized to ensure that they sum to one, facilitating a valid probability distribution:

$$w_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^N w_t^{(j)}}, \quad i = 1, 2, \dots, N. \quad (6)$$

Normalization is an essential aspect of the SIR Particle Filter, as it enables meaningful comparisons of the weights and ensures that they represent a proper distribution over the state space.

A significant challenge in particle filtering is particle degeneracy, which occurs when a few particles receive most of the weight while others become negligible. To counteract this, the SIR Particle Filter employs a resampling step [Saleh and Momeni, 2024]. In this step, particles are resampled based on their weights, resulting in a new set of equally weighted particles:

$$\mathbf{x}_t^{(i)} \leftarrow \mathbf{x}_t^{(j)}, \quad i = 1, 2, \dots, N, \quad (7)$$

$$w_t^{(i)} = \frac{1}{N}, \quad i = 1, 2, \dots, N. \quad (8)$$

This resampling step refreshes the particle set, ensuring that particles representing more probable states are replicated, while those representing less probable states are eliminated. This process enhances the accuracy of the approximation of the posterior distribution by concentrating computational resources on the more likely regions of the state space.

To estimate the new state of the system, the SIR Particle Filter applies a weighted average of the particles after the resampling step. This approach allows the filter to aggregate the information from all particles, reflecting the system state as a probabilistic mean. The state estimate  $\hat{\mathbf{x}}_t$  at time  $t$  is obtained by computing the weighted sum of the particles:

$$\hat{\mathbf{x}}_t = \sum_{i=1}^N w_t^{(i)} \mathbf{x}_t^{(i)} \quad (9)$$

where,  $\mathbf{x}_t^{(i)}$  represents the state of the  $i$ -th particle at time  $t$ , and  $w_t^{(i)}$  is its corresponding normalized weight. The estimated state is thus a weighted average, where particles with



higher weights contribute more significantly to the final estimate. This method provides a comprehensive estimate of the system state by incorporating the most probable particle states.

The SIR Particle Filter is particularly effective in scenarios with complex, high-dimensional state dynamics, such as sensor fusion, financial forecasting, and robotic navigation [Gyenes and Szádeczky-Kardoss, 2021]. It can reliably track evolving states over time, even in the presence of uncertainties and nonlinearities, making it a versatile tool for various real-world applications. By tailoring the filter to specific state transition and observation models, and choosing suitable proposal distributions, it achieves accurate state estimates in challenging environments.

In summary, the Sequential Importance Resampling Particle Filter offers a robust approach to state estimation in dynamic systems with non-linearity and non-Gaussian noise. Unlike the Kalman Filter, which assumes linear dynamics and Gaussian noise, the SIR Particle Filter is not bound by these restrictive assumptions, allowing it to handle more complex and realistic scenarios [Grooms and Robinson, 2021].

## 5 Proposed Approach

The management of resources and quality of service in LoRaWAN networks with the presence of mobile end devices is a challenging task. In this scenario, the channel characteristics vary unpredictably. These conditions highlight the need for alternative approaches to ADR management, capable of accommodating complex noise patterns observed in real-world implementations.

To address this challenge, this work proposes PF-ADR, an Adaptive Data Rate mechanism based on a SIR Particle Filter, leveraging the robustness of particle filtering techniques to more accurately estimation of the underlying state of the communication channel. Unlike other ADR schemes, PF-ADR dynamically adapts to the mobility of the end devices, estimating the SNR through a set of weighted particles that represent different possible states of the channel. These particles evolve over time, allowing the algorithm to dynamically adjust both the data rate and transmission power according to the estimated SNR.

Particle filters are particularly suitable for this task due to their ability to model non-linear and non-Gaussian systems [Bai et al., 2023]. These characteristics can be easily observed with the analysis of the histogram shown in Figure 2 obtained from approximately 200,000 SNR samples collected in a network with mobile devices. The right-skewed nature of the collected sample indicates that the underlying distribution is not symmetric, making particle filters an appropriate tool for effectively capturing the dynamics and complexities of the state estimation in such scenarios.

Due to the inherent signal fluctuations in mobile environments, PF-ADR employs the median of the measured SNR values as the representative state information [Sarmiento Neto et al., 2024]. To estimate the actual network state, the algorithm utilizes a Sequential Importance Resampling Particle Filter, which maintains a set of particles, each

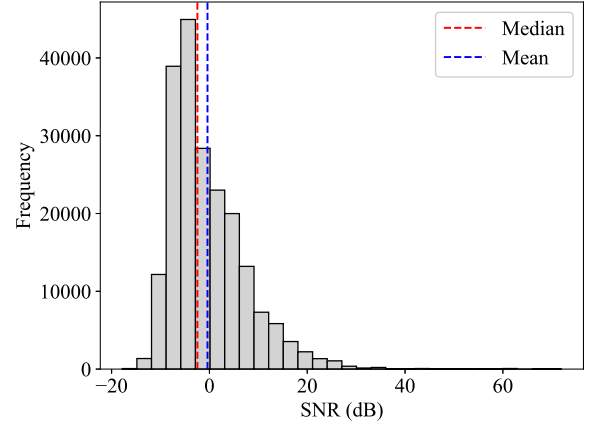


Figure 2. Histogram considering SNR samples.

representing a possible hypothesis about the current channel conditions. The estimation process is carried out through multiple iterative phases, ensuring a flexible and adaptive approach to rate control in LoRaWAN mobile scenarios. This scheme is represented in the flowchart illustrated in Figure 3.

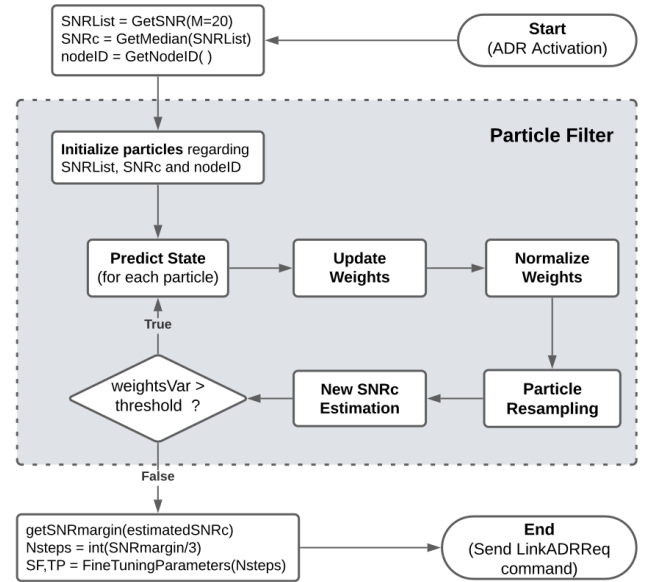


Figure 3. Flowchart illustrating the sequence of operations performed in the PF-ADR scheme.

In the *Initialize Particles* phase, the algorithm generates an initial set of particles, where each particle is assigned an initial SNR value based on the median of the measured SNR samples. These particles represent potential states of the communication channel and are initialized with equal weights to ensure an unbiased starting point for the filtering process.

During the *Predict State* phase, the particles evolve over time according to a probabilistic state transition model, incorporating process noise to account for uncertainties in the dynamic behavior of the mobile end devices. This step allows the particles to represent different possible future states of the network, accommodating variations in the wireless environment.

The *Update Weights* phase evaluates the likelihood of each particle given the newly observed SNR data. Each particle's weight is updated based on how well its predicted state aligns

with the current network conditions, incorporating measurement noise to adjust for observation inaccuracies. This step ensures that particles closely matching the actual communication conditions are given higher importance in the estimation process.

Following this, the *Normalize Weights* phase ensures that all particle weights sum to one, maintaining a valid probability distribution over the estimated states. Normalization helps in refining the influence of each particle, preventing numerical instability and ensuring meaningful comparisons between different particles.

To counteract particle degeneracy—where only a small subset of particles carry significant weight—the *Particle Resampling* phase is applied. Particles with higher weights are replicated, while those with negligible weights are discarded. This step redistributes computational resources toward more probable network states, improving estimation accuracy over time.

In the *New SNRc Estimation* phase, the algorithm computes the estimated SNR value as the weighted mean of the resampled particles. This refined estimate is then used as the basis for dynamically adjusting the spreading factor and transmission power, ensuring optimal network performance under mobile conditions.

After this stage, the *weightVariance* is updated, and the algorithm checks whether this variable is greater than a predefined *threshold*. If true, it indicates that the particles have not yet converged to a satisfactory state, and the particle filter process is restarted. If false, the particles have converged, and the algorithm proceeds similarly to line 6 onward in Algorithm 1 of the standard ADR.

This iterative process allows PF-ADR to continuously adapt to rapid environmental changes, providing a robust and energy-efficient solution for maintaining reliable communication in mobile LoRaWAN networks.

In more detail, Algorithm 2 describes the particle filter method used by PF-ADR. In lines 1-3, the scheme follows the same instructions as the standard ADR, as detailed in Algorithm 1. In line 4, the variable  $SNR_c$  stores the representative SNR of the current network state by computing the median of the values in  $SNR_{list}$ . The unique identifier of the end device,  $nodeID$ , is retrieved in line 5. The initialization phase of the filter starts in line 6 with the generation of an initial set of particles based on the  $nodeID$ , the number of particles  $N$ , and the  $SNR_c$ .

The algorithm then enters a repeat loop beginning in line 7 that continues until the variance of the particle weights,  $weightsVar$ , falls below or equals the current threshold,  $thr_{cur}$ , indicating that the particles have converged to a satisfactory state. In line 8 the  $thr_{cur}$  is updated by multiplying it by the decay factor  $thr_{dec}$ , thereby progressively tightening the convergence criterion over successive iterations and preventing endless execution. The algorithm iterates over each particle in line 9 and, in line 10, predicts the new state of each particle  $prtc$  by simulating the evolution of its SNR under the influence of process noise. In line 11 the weights of the particles,  $prtc.weights$ , are updated by comparing their predicted SNR values with the representative SNR and incorporating the measurement noise. Following this, line 12 normalizes  $prtc.weights$  to ensure they form a valid proba-

bility distribution.

In line 13 the particle resampling process is executed, which discards particles with low weights and duplicates those with higher weights to concentrate computational effort on the most likely states. In line 14 a new SNR estimate,  $SNR_{estim}$ , is computed as the weighted mean of the particles, and in line 15 the variance of the weights,  $weightsVar$ , is updated; this variance is then used as the convergence criterion for the repeat loop. Once the updated  $weightsVar$  is less than or equal to  $thr_{cur}$ , the loop terminates.

After convergence is achieved, in line 17 the algorithm computes the required SNR,  $SNR_{req}$ , for proper demodulation by invoking a function that takes the current spreading factor as input. In line 18 the  $SNR_{margin}$  is determined by subtracting the  $SNR_{req}$  from the  $SNR_{estim}$ , and in line 19 the number of adjustment steps  $N_{steps}$  is calculated. The algorithm then enters a series of while loops to fine-tune the transmission parameters in lines 20-28 similar to the standard ADR process described in Algorithm 1. Finally, in line 29 the updated settings are transmitted to the end device using the MAC command *LinkADRRReq*, which sends the new values of the spreading factor and transmission power.

In the input of the Algorithm 2, the variable  $p_{noise}$  represents the process noise, which reflects uncertainties in the dynamic model of the system itself. On the other hand, the variable  $m_{noise}$  represents measurement noise, which reflects inaccuracies in the observations made of this system. Both are critical for the correct implementation of the particle filter, which needs to account for both the uncertainty in the state and in the observations to make an accurate estimation [Ullah et al., 2020].

In terms of computational complexity, the functions *predictValue* and *updateWeights* perform simple sequential operations, having a time complexity of  $O(1)$ . On the other hand, *initializeParticles*, *normalizeWeights*, *resampleParticles*, *getEstimSNR*, and *getWeightVariance* depend on the number of particles, having a time complexity of  $O(N)$ . Therefore, assuming that the value of  $M$  is a small constant, the complexity of PF-ADR is  $O(itr \cdot N)$ , where  $itr$  is the number of iterations in the *repeat* loop. Asymptotically, as long as the number of particles is constant and predictable, the proposed scheme will not exhibit a significantly high execution time, since the threshold decay ensures that the value of  $itr$  remains low.

## 6 Methodology

This section outlines the network specifications, including the simulation environment and setup used to evaluate the proposed solution. Next, it discusses the parameterization of the particle filter, detailing the choices for process and measurement noise, as well as threshold settings. Finally, the metrics used to assess the solution's performance are detailed.



**Algorithm 2:** PF-ADR scheme algorithm.

---

**Input:**  $SF \in [7, 12]$ ,  $TP \in [2, 14]$ ,  $M = 20$ ,  
 $N, p_{noise}, m_{noise}, thr_{cur}, thr_{dec}$

```

1  $SNRlist \leftarrow \emptyset$ 
2 for  $i \leftarrow 0$  to  $M - 1$  do
3    $SNRlist[i] \leftarrow getSNR(i)$ 
4  $SNR_c \leftarrow getMedian(SNRlist)$ 
5  $nodeID \leftarrow getNodeID()$ 
6  $prtc \leftarrow initializeParticles(nodeID, N, SNR_c)$ 
7 repeat
8    $thr_{cur} \leftarrow thr_{cur} * thr_{dec}$ 
9   foreach  $prtc$  do
10     $prtc.SNR \leftarrow$   

        $predictValue(prtc.SNR, p_{noise})$ 
11     $prtc.weights \leftarrow$   

        $updateWeights(prtc.SNR, SNR_c, m_{noise})$ 
12     $prtc.weights \leftarrow$   

        $normalizeWeights(prtc.weights)$ 
13     $resampleParticles(nodeID)$ 
14     $SNR_{estim} \leftarrow$   

        $getEstimSNR(prtc.SNR, prtc.weights)$ 
15     $weightsVar \leftarrow$   

        $getWeightsVariance(prtc.weights)$ 
16 until  $weightsVar \leq thr_{cur}$ 
17  $SNR_{req} \leftarrow getDemodulationFloor(SF)$ 
18  $SNR_{margin} \leftarrow SNR_{estim} - SNR_{req}$ 
19  $N_{steps} \leftarrow int(SNR_{margin}/3)$ 
20 while  $N_{steps} > 0$  and  $SF > SF_{min}$  do
21    $SF \leftarrow SF - 1$ 
22    $N_{steps} \leftarrow N_{steps} - 1$ 
23 while  $N_{steps} > 0$  and  $TP > TP_{min}$  do
24    $TP \leftarrow TP - 2$ 
25    $N_{steps} \leftarrow N_{steps} - 1$ 
26 while  $N_{steps} < 0$  and  $TP < TP_{max}$  do
27    $TP \leftarrow TP + 2$ 
28    $N_{steps} \leftarrow N_{steps} + 1$ 
29  $sendLinkADRReq(SF, TP)$ 

```

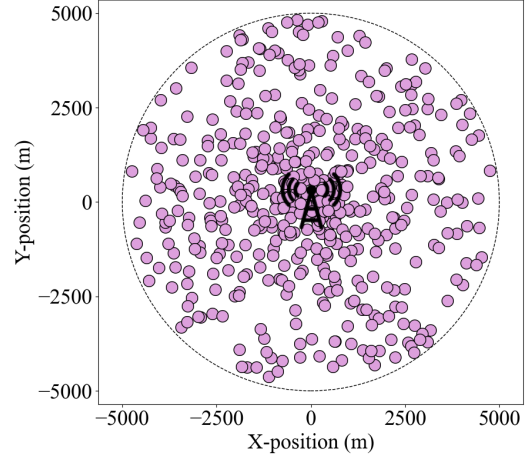
---

## 6.1 Network Specification

To evaluate the proposed solution, a simulated environment was developed to analyze the performance of PF-ADR compared to other relevant ADR solutions: MB-ADR, a median-based scheme; M-ADR, a Kalman filter-based proposal; and the standard ADR scheme from LoRaWAN protocol. All solutions were coded in C++ and executed using the Network Simulator<sup>1</sup> (NS-3), with the aid of the LoRaWAN module from SIGNET Lab<sup>2</sup>, a widely used add-on in state-of-the-art LoRaWAN network simulations [Magrin et al., 2020].

The simulated environment consists of a network with a single gateway and a network server, along with 200 to 1000 class A LoRa devices deployed in a circular area with a radius of 5 km, as adopted by Farhad et al. [2023] and illustrated in Figure 4. The EDs periodically send 144 packets over a 24-hour period, emulating a pet-tracking application

suggested by Semtech [Anwar et al., 2021a]. Each simulation instance was run 10 times with different random seeds to capture variability and ensure the robustness of the analyses. This repetition provides a more accurate estimation of performance metrics and aids in assessing result convergence. A 95% confidence interval was applied to ensure statistical validity, enabling error margin quantification and reinforcing the reliability of the conclusions.



**Figure 4.** Sample positioning of 500 EDs in the simulation area.

The EDs behavior was modeled using a 2D Random-Walk mobility model [Nouar et al., 2024; Martinez-Caro and Cano, 2019], applying random speed variations between 0.5 and 1.5 m/s and changing direction every 1000 meters (m). For more realistic attenuation effects, the wireless channel was subject to a log-distance path loss model. Additionally, shadowing effects were simulated by applying stochastic signal variations associated with obstacles and reflections in the environment [Anwar et al., 2021a]. A combined path loss and shadowing model [González-Palacio et al., 2023] is defined as:

$$L(d) = L(d_0) + 10\gamma \log_{10} \left( \frac{d}{d_0} \right) + X_\sigma \quad (10)$$

where  $L(d)$  is the path loss in decibels (dB) at a distance  $d$  from the transmitter,  $L(d_0)$  is the path loss measured at a reference distance  $d_0$ ,  $\gamma$  is the path loss exponent, and  $X_\sigma$  represents shadowing, modeled as a normally distributed variable with a mean of zero and standard deviation  $\sigma$ . A general overview of the simulation parameters is presented in Table 2.

In each simulation setup, all ADR schemes were initially configured with SF12 for all EDs. This decision was based on the technical rationale that this SF level, despite its lower data rate and longer airtime, offers the highest receiver sensitivity and maximum communication range [Benkahla et al., 2021a]. Such a conservative initialization ensures that even devices with poor channel conditions are able to establish a reliable connection at the outset, thereby reducing early packet losses and providing a robust baseline from which the adaptive mechanisms can dynamically adjust the transmission parameters.

Similar to other ADR scheme proposals for mobile scenarios [Farhad et al., 2023; Moysiadis et al., 2021], this study

<sup>1</sup><https://www.nsnam.org/releases/ns-3-42/>

<sup>2</sup><https://github.com/signetlabdei/lorawan>

**Table 2.** General parameterization adopted.

| Parameter                       | Value                    |
|---------------------------------|--------------------------|
| Simulation time                 | 24 hours                 |
| Area radius                     | 5 km                     |
| Number of EDs                   | 200, 400, 600, 800, 1000 |
| Packet rate                     | 144 packets/day          |
| Packet size                     | 30 bytes                 |
| UL packet transmission limit    | 8                        |
| Mobility model                  | 2D Random Walk           |
| ED movement speed               | [0.5, 1.5] m/s           |
| Path loss exponent ( $\gamma$ ) | 3.76                     |
| Carrier frequency               | 868 MHz (EU-868)         |
| Bandwidth                       | 125 KHz                  |

adopts the same SNR measurement window size as the standard ADR,  $M = 20$ . Although a smaller value could enable faster adaptation to channel variations, this work focuses on analyzing the benefits of the method applied in PF-ADR, ensuring that different window sizes do not influence the results.

## 6.2 Particle Filter Parameterization

The adoption of low values for process noise and measurement noise, combined with the use of the median, which is a robust statistical measure in samples with high fluctuations [Sarmiento Neto et al., 2024], enables a relatively stable system that is less sensitive to extreme noise. This ensures that particles in the particle filter are less dispersed, resulting in more accurate and reliable estimates.

As defined in Algorithm 2, the initial value set for the current threshold,  $thr_{cur}$ , ensures that, at the beginning of the filtering process, particles with very low weights are quickly discarded, focusing on the most relevant particles. This helps improve computational efficiency and the accuracy of the estimates [Ahwiadi and Wang, 2020].

The adopted value for dynamic threshold,  $thr_{dec}$ , allows the threshold to decrease gradually with each iteration, softening the exclusion criterion for particles over time. This progressive decay prevents potentially useful particles from being prematurely eliminated as the state estimate refines, maintaining good particle diversity early on and constraining the set as the estimate stabilizes. Using a dynamic threshold is also important to prevent the particle filtering process from falling into an infinite loop. The summary of particle filter parameterization is shown in Table 3.

**Table 3.** Particle filter parameterization summary.

| Parameter                         | Value |
|-----------------------------------|-------|
| Number of particles ( $nPrtc$ )   | 50    |
| Process noise ( $p_{noise}$ )     | 0.005 |
| Measurement noise ( $m_{noise}$ ) | 0.01  |
| Initial threshold ( $thr_{cur}$ ) | 0.001 |
| Threshold decay ( $thr_{dec}$ )   | 0.9   |

## 6.3 Employed metrics

In order to assess the performance of the proposed solution, several metrics were employed, which are commonly used in

the evaluation of wireless communication protocols, particularly in LoRaWAN networks. These metrics include packet delivery ratio, energy consumption, energy efficiency, and Packet Loss Ratio (PLR). Each metric provides valuable insight into different aspects of network performance, such as reliability, energy usage, and the effects of interference and link quality.

The PDR is defined as the ratio of successfully received packets at the gateway to the total number of packets transmitted by the device, as denoted in Equation 11. It is an essential measure of communication reliability, indicating how effectively data is delivered across the network [Anwar et al., 2021a].

$$PDR = \frac{N_{received}}{N_{sent}} \quad (11)$$

where  $N_{received}$  represents the number of packets successfully received by the gateway, and  $N_{sent}$  is the total number of packets sent by the end device.

The energy consumption of an end device is another important metric, especially for battery-powered IoT devices where energy efficiency is vital [Al-Gumaei et al., 2022]. For the simulated network, an energy model based on LoRa transceiver SX1272 is adopted [Kufakunesu et al., 2022; Loubany et al., 2020], taking into account four distinct states: transmit ( $tx$ ), receive ( $rx$ ), idle and sleep. During transmission, for instance, an end device exhibits energy consumption, in Joules (J), represented by:

$$E_{tx} = I_{tx} \cdot V \cdot T_{tx} \quad (12)$$

where  $I_{tx}$  is the transmission current,  $V$  is the supply voltage, and  $T_{tx}$  is the time duration of transmission [Philip and Singh, 2021].

For the other states, the same relation is used, replacing the corresponding values of  $I_{tx}$  and  $V_{tx}$  with those applicable to the other states. The current and voltage values employed in this study are detailed in Table 4.

**Table 4.** Energy model parameterization.

| Parameter               | Value     |
|-------------------------|-----------|
| ED initial energy       | 1000 J    |
| Supply voltage (V)      | 3.3 V     |
| Tx current ( $I_{tx}$ ) | 28 mA     |
| Rx current ( $I_{rx}$ ) | 11.2 mA   |
| Idle current ( $I_i$ )  | 1.4 mA    |
| Sleep current ( $I_s$ ) | 0.0015 mA |

Based on this assumption, the total energy consumption is expressed in Equation 13, where  $E_{tx}$  and  $E_{rx}$  correspond to the energy consumed when an end device is transmitting and receiving a packet, respectively, whereas  $E_i$  and  $E_s$  correspond to the energy consumed in the idle and sleep states of the ED [To and Duda, 2018].

$$E = E_{tx} + E_{rx} + E_i + E_s \quad (13)$$

Energy efficiency is another key performance indicator, reflecting the amount of data successfully transmitted, in bits, relative to the energy consumed, in J. This metric provides

an indication of how effectively the energy consumed by the device contributes to successful communication [Loubany et al., 2023]. The energy efficiency ( $EE$ ), in bits/J, can be calculated as:

$$EE = \frac{N_{bits}}{E} \quad (14)$$

where  $N_{bits}$  represents the total number of bits successfully received by the gateway, and  $E$  is the total energy consumed by the device.

The PLR is a metric that assesses network reliability by measuring the percentage of transmitted packets that are not successfully received. Several factors contribute to PLR [Anwar et al., 2021b]. PLR-T occurs when the gateway is occupied with downlink transmissions, such as sending acknowledgment packets (ACKs), which prevents it from receiving uplink packets simultaneously. PLR-S arises when packets are lost due to their signal strength falling below the gateway sensitivity threshold, resulting in weak signals that cannot be properly decoded. PLR-R refers to packet loss that happens when all the gateway reception paths are busy with other transmissions. Finally, PLR-I is caused by interference when packets overlap on the same or nearby channels, which can happen even with different spreading factors. This collision of transmissions can lead to packet loss, particularly during periods of high uplink traffic or congestion.

By analyzing these metrics, it is possible to gain a comprehensive understanding of the network performance in terms of reliability, energy consumption, and efficiency under various conditions. These insights allow for a detailed evaluation of the proposed solution effectiveness in optimizing LoRaWAN communication.

## 7 Results and Discussion

This section presents the results obtained from the simulations described in Section 6. From the reliability perspective, Figure 5 displays the average PDR of the schemes concerning the number of mobile EDs. It is evident that the schemes that use the median as a representative SNR measure demonstrate similar performance, with a 4.15% improvement of PF-ADR over MB-ADR.

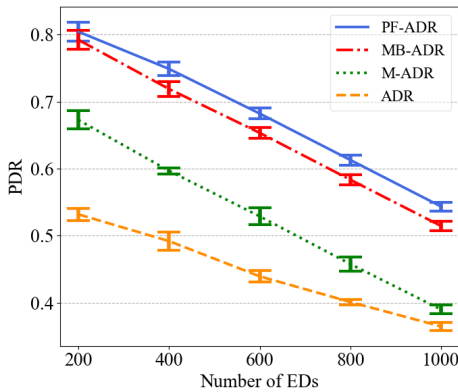


Figure 5. Average PDR in relation to the number of EDs.

The other schemes exhibited average PDR values at a lower level. The overall improvement of PDR from PF-ADR

compared to M-ADR and standard ADR was approximately 29.5% and 52.17%, respectively. This analysis also supports the assertion that standard ADR is not suitable for scenarios involving mobility.

An analysis of the average PDR over time for 1000 EDs, as shown in Figure 6, indicates that PF-ADR and MB-ADR exhibit similar performance during the initial simulation hours, with both outperforming ADR from approximately 5 hours onward. This increase in PDR is attributed to the fact that all devices initially operate at SF12; as different spreading factors are assigned over time, PLR-I tends to decrease by mitigating interference and improving spectrum efficiency. Furthermore, from around 12 hours onward, PF-ADR begins to exhibit even greater PDR, with its advantage over MB-ADR steadily increasing until the end of the simulation. These findings indicate that the proposed solution delivers superior packet performance over longer time intervals.

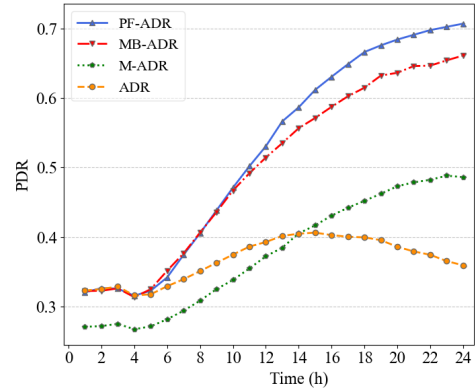


Figure 6. Evolution of average PDR over the simulation time.

In terms of average energy consumption, PF-ADR and standard ADR demonstrated similar performance. As shown in Figure 7, it cannot be concluded that there is a difference between the two solutions, as their respective error margins overlap within the adopted confidence interval. In comparison to the other schemes, M-ADR exhibits energy performance close to MB-ADR during the rounds from 200 to 600 EDs, showing an increase in energy consumption when increasing the number of end devices.

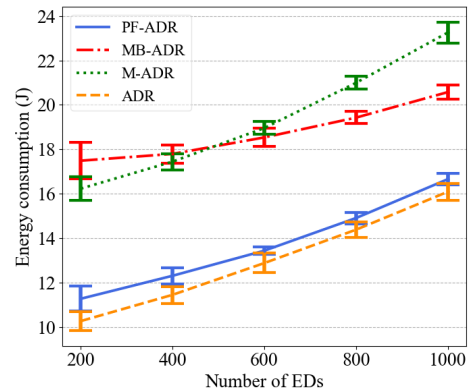


Figure 7. Average energy consumption per ED.

From the perspective of energy efficiency, a significant advantage of the proposed solution is evident, as displayed in Figure 8. PF-ADR achieved an overall improvement of

approximately 43.19%, 82.96%, and 46.76% compared to MB-ADR, M-ADR, and standard ADR, respectively. This marked advantage reflects an excellent level of utility for the proposed solution, balancing reliability and energy consumption.

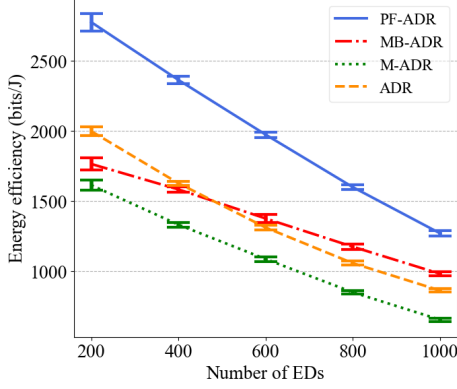


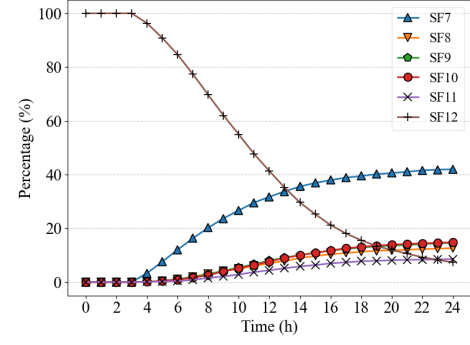
Figure 8. Average energy efficiency.

Observing the obtained spreading factor values provides insight into how PF-ADR achieved its balance between PDR and energy consumption. Figure 9a presents the SF allocation over simulation time for a network with 1000 EDs. It can be observed that, approximately after 5 hours of simulation, more end devices begin to be assigned SF7, leading to a decrease in PLR-I and, consequently, an increase in PDR, as also shown in Figure 6. This occurs because a larger number of EDs utilize a variety of spreading factors. Moreover, the increasing proportion of EDs using the lowest level, SF7, along with the sharp reduction in EDs assigned the highest level, SF12, over time results in a significant decrease in symbol durations, thereby contributing to a lower average energy consumption.

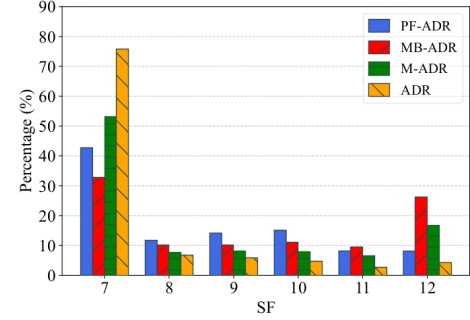
An analysis of the final spreading factor distribution, as shown in Figure 9b, indicates that the standard ADR disproportionately assigned more SF7 levels to EDs during the final hours of the simulation. Although this approach contributes to a reduction in average energy consumption, it tends to increase the PLR-S, since many end devices eventually lose connectivity with the gateway. In contrast, PF-ADR distributes the higher SF levels more evenly than the standard ADR, resulting in a lower PLR-S. The average packet loss rates for both PF-ADR and the standard ADR are presented in Figures 10a and 10b. Regarding PLR-I, PLR-R, and PLR-T, both schemes show similar values.

Despite demonstrating strong performance across the analyzed metrics, PF-ADR did not incur disadvantages in execution time. Although the analysis in Section 5 indicates a complexity with a linearly multiplicative tendency, the low number of particles used and the controlled amount of *itr* with dynamic thresholding allowed the proposed solution to maintain average execution time values equivalent to those of the other schemes, as shown in Table 5.

Overall, PF-ADR demonstrated superior performance compared to other schemes, in terms of both reliability and energy efficiency. The significant improvement in PDR compared to standard ADR highlights the effectiveness of the proposed approach in scenarios involving mobility. Additionally, the balance between PDR and energy consumption

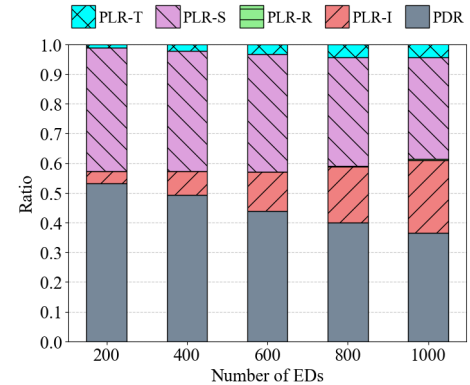


(a) Evolution of average SF assignment for PF-ADR over the simulation time.

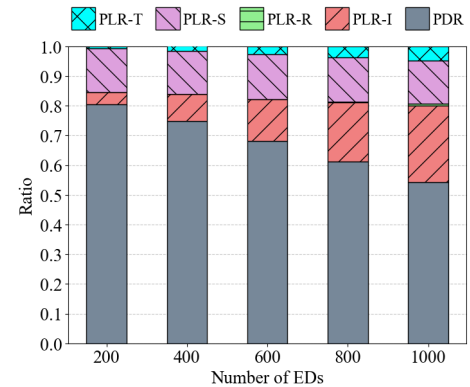


(b) Final average SF distribution across all evaluated schemes.

Figure 9. Average spreading factor assignment for 1000 EDs.



(a) Standard ADR



(b) PF-ADR

Figure 10. Proportion of PDR and the different types of PLR.

is noteworthy, with energy efficiency surpassing all analyzed schemes. Furthermore, PF-ADR ability to better assign SF levels results in a reduction of PLR-S, minimizing packet loss without compromising performance, establishing it as

**Table 5.** Mean execution time of a simulation instance (in seconds).

| # EDs | PF-ADR  | MB-ADR  | M-ADR   | ADR     |
|-------|---------|---------|---------|---------|
| 200   | 55.60   | 52.66   | 52.74   | 50.85   |
| 400   | 185.28  | 183.66  | 183.90  | 179.80  |
| 600   | 392.56  | 402.71  | 405.88  | 393.89  |
| 800   | 705.68  | 711.80  | 722.34  | 705.57  |
| 1000  | 1127.01 | 1132.61 | 1138.36 | 1126.19 |

a robust and efficient solution for applications involving mobility of EDs.

## 8 Conclusion and Future Work

This work presents PF-ADR, an alternative to LoRaWAN ADR designed to optimize communication in mobile environments. The proposed solution employs a SIR Particle Filter to handle non-linear, non-Gaussian signal variations, allowing it to more accurately estimate real-time SNR fluctuations. By continuously adjusting transmission parameters such as the spreading factor and transmission power, PF-ADR significantly improves network performance in dynamic mobile scenarios compared to existing ADR schemes.

In a simulated evaluation, PF-ADR presented a 29.5% improvement in PDR compared to M-ADR and a 52.17% increase over the standard LoRaWAN ADR. In terms of energy efficiency, PF-ADR showed a 43.19% improvement compared to MB-ADR and an 82.96% enhancement over M-ADR. These performance gains were observed across different network densities, with the algorithm demonstrating superior scalability in deployments involving up to 1000 mobile end devices.

One of the key contributions of PF-ADR lies in its ability to dynamically allocate spreading factor levels, significantly improving network resource management. The algorithm reduces packet loss by distributing spreading factors more efficiently, thereby avoiding overuse of the highest SF levels, which typically lead to increased energy consumption. This optimization not only enhances PDR, but also achieves a favorable balance between energy consumption and communication reliability. These features position PF-ADR as a robust and energy-efficient solution, especially for mobile IoT applications, where maintaining network reliability and extending device battery life are critical.

However, while PF-ADR demonstrated superior performance in simulated environments, further evaluation is required in real-world IoT deployments to fully validate its effectiveness under varying mobility patterns, interference levels, and environmental factors. Future work should focus on enhancing the algorithm adaptability to more complex and dynamic network conditions, including a deeper analysis of the impact of different  $M$  values. Further research could also investigate the integration of PF-ADR with other network optimization techniques to address challenges related to congestion and spectrum usage in high-density IoT scenarios.

## Funding

This research was funded by Brazilian National Council for Research and Development (CNPq) via Grants No. 307967/2022-0

(“Development of Intelligent Architecture for Applications on Internet of Robotic Things”) and No. 407274/2021-9 (“Strategic Solutions for Aerial-Terrestrial IoT Network Architectures”).

## Authors’ Contributions

GASN conceptualized the study and wrote the main manuscript; GASN, TARS, and PFFA developed and performed the simulations; GASN, AFSV, and LHOM handled the preparation and review of all figures and tables; RALR and JVRJ contributed to the final revision of the manuscript. All authors read and approved the final manuscript.

## Competing interests

The authors declare no competing interests.

## Availability of data and materials

No datasets were used in this study. The research was conducted entirely through a simulated network environment.

## References

- Ahwiadi, M. and Wang, W. (2020). An Adaptive Particle Filter Technique for System State Estimation and Prognosis. *IEEE Transactions on Instrumentation and Measurement*, 69(9):6756–6765. DOI: 10.1109/TIM.2020.2973850.
- Al-Gumaei, Y. A., Aslam, N., Aljaidi, M., Al-Saman, A., Alsarhan, A., and Ashyap, A. Y. (2022). A Novel Approach to Improve the Adaptive-Data-Rate Scheme for IoT LoRaWAN. *Electronics*, 11(21). DOI: 10.3390/electronics11213521.
- Almuhaya, M. A. M., Jabbar, W. A., Sulaiman, N., and Abdulmalek, S. (2022). A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions. *Electronics*, 11(1). DOI: 10.3390/electronics11010164.
- Anwar, K., Rahman, T., Zeb, A., Khan, I., Zareei, M., and Vargas-Rosales, C. (2021a). RM-ADR: Resource Management Adaptive Data Rate for Mobile Application in LoRaWAN. *Sensors*, 21(23). DOI: 10.3390/s21237980.
- Anwar, K., Rahman, T., Zeb, A., Saeed, Y., Khan, M. A., Khan, I., Ahmad, S., Abdelgawad, A. E., and Abdollahian, M. (2021b). Improving the Convergence Period of Adaptive Data Rate in a Long Range Wide Area Network for the Internet of Things Devices. *Energies*, 14(18). DOI: 10.3390/en14185614.
- Azizi, F., Teymuri, B., Aslani, R., Rasti, M., Tolvaneny, J., and Nardelli, P. H. J. (2022). MIX-MAB: Reinforcement Learning-based Resource Allocation Algorithm for LoRaWAN. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pages 1–6. DOI: 10.1109/VTC2022-Spring54318.2022.9860807.
- Bai, X., Qin, F., Ge, L., Zeng, L., and Zheng, X. (2023). Dynamic state estimation for synchronous generator with communication constraints: An improved regularized particle filter approach. *IEEE Transactions on Sustainable Computing*, 8(2):222–231. DOI: 10.1109/TSUSC.2022.3221090.



- Baruffa, G. and Rugini, L. (2022). Performance of LoRa-Based Schemes and Quadrature Chirp Index Modulation. *IEEE Internet of Things Journal*, 9(10):7759–7772. DOI: 10.1109/JIOT.2021.3114096.
- Benkahla, N., Tounsi, H., Song, Y.-Q., and Frikha, M. (2021a). Review and experimental evaluation of ADR enhancements for LoRaWAN networks. *Telecommunication Systems*, 77(1):1–22. DOI: 10.1007/s11235-020-00738-x.
- Benkahla, N., Tounsi, H., Song, Y.-Q., and Frikha, M. (2021b). VHMM-based E-ADR for LoRaWAN networks with unknown mobility patterns. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 86–91. DOI: 10.1109/IWCMC51323.2021.9498709.
- Camarillo-Escobedo, R., Flores, J. L., Marin-Montoya, P., García-Torales, G., and Camarillo-Escobedo, J. M. (2022). Smart multi-sensor system for remote air quality monitoring using unmanned aerial vehicle and lorawan. *Sensors*, 22(5). DOI: 10.3390/s22051706.
- Farhad, A., Kim, D.-H., and Pyun, J.-Y. (2022). R-ARM: Retransmission-Assisted Resource Management in LoRaWAN for the Internet of Things. *IEEE Internet of Things Journal*, 9(10):7347–7361. DOI: 10.1109/JIOT.2021.3111167.
- Farhad, A., Kwon, G.-R., and Pyun, J.-Y. (2023). Mobility Adaptive Data Rate Based on Kalman Filter for LoRa-Empowered IoT Applications. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 321–324. DOI: 10.1109/CCNC51644.2023.10060330.
- Farhad, A. and Pyun, J.-Y. (2023). AI-ERA: Artificial Intelligence-Empowered Resource Allocation for LoRa-Enabled IoT Applications. *IEEE Transactions on Industrial Informatics*, 19(12):11640–11652. DOI: 10.1109/TII.2023.3248074.
- Francis, I. and Mohd Shah, S. (2022). Cost-Effective Arduino-Based RFID Automated Cage Door and Pet Tagging with GPS Tracker using Peer-to-Peer LoRaWAN. *Journal of Electronic Voltage and Application*, 3(2):47–58. DOI: 10.30880/jeva.2022.03.02.005.
- González-Palacio, M., Tobón-Vallejo, D., Sepúlveda-Cano, L. M., Rúa, S., Pau, G., and Le, L. B. (2023). LoRaWAN Path Loss Measurements in an Urban Scenario including Environmental Effects. *Data*, 8(1). DOI: 10.3390/data8010004.
- Grooms, I. and Robinson, G. (2021). A hybrid particle-ensemble Kalman filter for problems with medium non-linearity. *PLOS ONE*, 16(3):1–20. DOI: 10.1371/journal.pone.0248266.
- Gyenes, Z. and Szádeczky-Kardoss, E. G. (2021). Particle filter-based perception method for obstacles in dynamic environment of a mobile robot. In *2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 97–102. DOI: 10.1109/MMAR49549.2021.9528442.
- Ilahi, I., Usama, M., Farooq, M. O., Umar Janjua, M., and Qadir, J. (2020). LoRaDRL: Deep Reinforcement Learning Based Adaptive PHY Layer Transmission Parameters Selection for LoRaWAN. In *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pages 457–460. DOI: 10.1109/LCN48667.2020.9314772.
- Jouhari, M., Saeed, N., Alouini, M.-S., and Amhoud, E. M. (2023). A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges. *IEEE Communications Surveys Tutorials*, 25(3):1841–1876. DOI: 10.1109/COMST.2023.3274934.
- Kufakunesu, R., Hancke, G., and Abu-Mahfouz, A. (2022). A Fuzzy-Logic Based Adaptive Data Rate Scheme for Energy-Efficient LoRaWAN Communication. *Journal of Sensor and Actuator Networks*, 11:65. DOI: 10.3390/jsan11040065.
- Kufakunesu, R., Hancke, G. P., and Abu-Mahfouz, A. M. (2020). A Survey on Adaptive Data Rate Optimization in LoRaWAN: Recent Solutions and Major Challenges. *Sensors*, 20(18). DOI: 10.3390/s20185044.
- Li, X., Xu, J., Li, R., Jia, L., and You, J. (2024). Advancing Performance in LoRaWAN Networks: The Circular Region Grouped Bit-Slot LoRa MAC Protocol. *Electronics*, 13(3). DOI: 10.3390/electronics13030621.
- Lodhi, M. A., Obaidat, M. S., Wang, L., Mahmood, K., Ibrahim Qureshi, K., Chen, J., and Hsiao, K.-F. (2024). Tiny Machine Learning for Efficient Channel Selection in LoRaWAN. *IEEE Internet of Things Journal*, 11(19):30714–30724. DOI: 10.1109/JIOT.2024.3413585.
- Lodhi, M. A., Wang, L., and Farhad, A. (2022). ND-ADR: Nondestructive adaptive data rate for LoRaWAN Internet of Things. *International Journal of Communication Systems*, 35(9):e5136. DOI: <https://doi.org/10.1002/dac.5136>.
- Loubany, A., Lahoud, S., and El Chall, R. (2020). Adaptive algorithm for spreading factor selection in LoRaWAN networks with multiple gateways. *Computer Networks*, 182:107491. DOI: <https://doi.org/10.1016/j.comnet.2020.107491>.
- Loubany, A., Lahoud, S., Samhat, A. E., and El Helou, M. (2023). Improving Energy Efficiency in LoRaWAN Networks with Multiple Gateways. *Sensors*, 23(11). DOI: 10.3390/s23115315.
- Magrin, D., Capuzzo, M., and Zanella, A. (2020). A Thorough Study of LoRaWAN Performance Under Different Parameter Settings. *IEEE Internet of Things Journal*, 7(1):116–127. DOI: 10.1109/JIOT.2019.2946487.
- Marais, J. M., Abu-Mahfouz, A. M., and Hancke, G. P. (2023). Improving the Sustainability of Confirmed Traffic in LoRaWANs Through an Adaptive Congestion Scheme. *IEEE Sensors Journal*, 23(2):1660–1670. DOI: 10.1109/JSEN.2022.3226273.
- Martinez-Caro, J.-M. and Cano, M.-D. (2019). IoT System Integrating Unmanned Aerial Vehicles and LoRa Technology: A Performance Evaluation Study. *Wireless Communications and Mobile Computing*, 2019(1):4307925. DOI: <https://doi.org/10.1155/2019/4307925>.
- Milarokostas, C., Tsolkas, D., Passas, N., and Merakos, L. (2023). A Comprehensive Study on LPWANs With a Focus on the Potential of LoRa/LoRaWAN Systems. *IEEE Communications Surveys Tutorials*, 25(1):825–867. DOI: 10.1109/COMST.2022.3229846.
- Moysiadiis, V., Lagkas, T., Argyriou, V., Sarigiannidis, A.,



- Moscholios, I. D., and Sarigiannidis, P. (2021). Extending ADR mechanism for LoRa enabled mobile end-devices. *Simulation Modelling Practice and Theory*, 113:102388. DOI: 10.1016/j.simpat.2021.102388.
- Nouar, A., Abbas, M. T., Boumerdassi, S., and Chaib, M. (2024). Impact of Mobility Model on LoRaWAN Performance. *Journal of Communications*, 19(1). DOI: 10.12720/jcm.19.1.7-18.
- Philip, M. S. and Singh, P. (2021). Energy Consumption Evaluation of LoRa Sensor Nodes in Wireless Sensor Network. In *2021 Advanced Communication Technologies and Signal Processing (ACTS)*, pages 1–4. DOI: 10.1109/ACTS53447.2021.9708341.
- Saleh, A. R. and Momeni, H. R. (2024). An improved iterative closest point algorithm based on the particle filter and K-means clustering for fine model matching. *The Visual Computer*. DOI: 10.1007/s00371-023-03195-0.
- Sallum, E., Pereira, N., Alves, M., and Santos, M. (2020). Improving Quality-Of-Service in LoRa Low-Power Wide-Area Networks through Optimized Radio Resource Management. *Journal of Sensor and Actuator Networks*, 9(1). DOI: 10.3390/jsan9010010.
- Sarmiento Neto, G. A., Da Silva, T. A. R., Abreu, P. F. F., Veloso, A. F. D. S., Mendes, L. H. D. O., and Dos Reis Junior, J. V. (2024). Addressing Mobility Challenges in LoRaWAN through Adaptive Data Rate: A Statistical Median-Based Approach. In *2024 11th International Conference on Future Internet of Things and Cloud (FiCloud)*. DOI: 10.1109/FiCloud62933.2024.00058.
- Severino, R. F., Serrador, A., Datia, N., Campos-Rebelo, R., and Machado, J. (2023). A Low-Cost Solution for Assets Location Tracking Based on LoRaWAN Networks. In *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, pages 1–8. DOI: 10.1109/WF-IoT58464.2023.10539413.
- To, T.-H. and Duda, A. (2018). Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. DOI: 10.1109/ICC.2018.8422800.
- Ullah, I., Su, X., Zhu, J., Zhang, X., Choi, D., and Hou, Z. (2020). Evaluation of Localization by Extended Kalman Filter, Unscented Kalman Filter, and Particle Filter-Based Techniques. *Wireless Communications and Mobile Computing*, 2020(1):8898672. DOI: <https://doi.org/10.1155/2020/8898672>.
- Vangelista, L., Calabrese, I., and Cattapan, A. (2023). Mobility Classification of LoRaWAN Nodes Using Machine Learning at Network Level. *Sensors*, 23(4). DOI: 10.3390/s23041806.
- Varsi, A., Taylor, J., Kekempanos, L., Pyzer Knapp, E., and Maskell, S. (2020). A Fast Parallel Particle Filter for Shared Memory Systems. *IEEE Signal Processing Letters*, 27:1570–1574. DOI: 10.1109/LSP.2020.3014035.
- Wang, K., Wang, K., and Ren, Y. (2024). Time-Allocation Adaptive Data Rate: An Innovative Time-Managed Algorithm for Enhanced Long-Range Wide-Area Network Performance. *Electronics*, 13(2). DOI: 10.3390/electronics13020434.