International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 02 | Feb - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Comparative Study and Implementation of Efficient Bandwidth Utilization Technique for LoRaWAN

Sandhya Rani1, Harish V Mekali2, M.N. Suma3

123 Department of Electronics and Communication Engineering,
BMS College of Engineering, Bangalore, 560019, Karnataka, India

---***---

Abstract - LoRaWAN (Long Range Wide Area Network) is one of the most widely adopted Low-Power Wide Area Network (LPWAN) technologies, enabling long-range, energy-efficient communication for Internet of Things (IoT) applications. A key mechanism in LoRaWAN is Adaptive Data Rate (ADR), which dynamically configures each end device's Spreading Factor (SF) and transmission power (Tx Power) to balance energy consumption, coverage, and reliability. The efficient allocation of SFs is particularly critical, as it directly influences bandwidth utilization, throughput, latency, and packet collision rates. In this paper, we present a comparative study and implementation of three approaches for ADR-based bandwidth optimization schemes: Static ADR, Centralized ADR, and a proposed Decentralized ADR scheme. Python-based simulations are conducted under varying traffic conditions, incorporating realistic wireless channel models such as Free Space, Okumura-Hata, and Nakagami-m fading.Performance metrics including throughput, packet success rate, collision rate, and SF distribution are evaluated to provide a clear and comprehensive comparison of the different ADR approaches. The results demonstrate that the proposed Decentralized ADR significantly improves overall performance, achieving up to 18.6% higher throughput compared to the conventional Centralized ADR scheme. These findings highlight the importance of distributed decision-making in LoRaWAN and provide useful insights into scalable, adaptive, and efficient ADR designs for next-generation large-scale IoT deployments.

Key Words: LoRaWAN, IoT, Adaptive Data Rate, Bandwidth Optimization, Decentralized Networks

1.INTRODUCTION

LoRWAN is a MAC-layer protocol that operates above the physical layer of LoRa. The network adopts a star-ofstars topology and functions in the unlicensed ISM band (865–867 MHz in India). It uses chirp spread spectrum modulation, enabling reliable long-distance communication, typically 2–3 km in urban areas and up to 15km in rural regions [1]. Compared to other existing LPWAN technologies such as NB-IoT and Sigfox, LoRaWAN offers a better balance between communication range, cost-effectiveness and spectrum efficiency [2].

The data rate in a LoRaWAN network is primarily determined by three key physical layer parameters: the Spreading Factor (SF), Bandwidth (BW), and Coding Rate (CR). SF, which ranges from 7 to 12, determines how many symbols are used to encode each bit, i.e., how long a symbol is spread in time. Bandwidth refers to the width of the frequency spectrum over which the chirp signal is transmitted, typically 125 kHz, 250 kHz, or 500 kHz, with 125 kHz being the most commonly used for uplink transmissions. The CR specifies the amount of forward error correction added and is expressed as $\frac{4}{4+n}$, where $n \in \{1, 2, 3, 4\}$ [3], [4]. LoRaWAN supports a range of data rates from 0.3 kbps to 50 kbps, enabling a trade-off between communication range and throughput. Increasing the spreading factor enhances the communication range but reduces the data rate [5].

In dense LoRaWAN networks, increasing the data rate can lead to reduced throughput due to a higher likelihood of collisions. Since network congestion in LoRaWAN is closely related to SF allocation, efficient SF assignment is crucial for optimal bandwidth utilization and overall network performance. In this study, an Adaptive Data Rate (ADR) algorithm is initially implemented to improve Spreading Factor (SF) allocation compared to static ADR, thus enhancing network airtime efficiency. However, the conventional ADR mechanism operates in a centralized manner, where the end devices depend on the network server to receive optimized transmission parameters. In dynamic environments where channel conditions vary frequently, this reliance can lead to suboptimal performance due to delayed or missing feedback. To address this limitation, a Decentralized ADR approach is proposed, wherein each node independently adjusts its transmission parameters based on locally observed signal conditions. Subsequently, a comparative evaluation is conducted in three approaches: static ADR, centralized ADR, and the proposed decentralized ADR, under varying network traffic loads.

2. Related Work

Mekki et al. [1] carried out an extensive comparison of various LPWAN technologies, highlighting LoRaWAN's advantages in large-scale IoT deployments. More and Patel [4] analyzed LoRaWAN performance with varying data rates, demonstrating the critical role of proper SF allocation. While these studies establish LoRaWAN's potential, they lack detailed analysis of adaptive mechanisms under varying network conditions.

Lehong et al. [5] surveyed LoRaWAN ADR algorithms, identifying potential optimization areas but focusing primarily on centralized approaches. Serati et al. [6] proposed ADR-Lite, a low-complexity ADR scheme that reduces computational overhead while maintaining centralized control. These works demonstrate the need for ADR optimization but do not adequately address the limitations of centralized schemes in large-scale or distributed deployments.

Alipio et al. [7] classified existing testing methodologies for LoRa and LoRaWAN, revealing inconsistencies in performance evaluation and the absence of standardized benchmarking for ADR algorithms. Campos et al. [8] similarly emphasize the need for standardized and realistic parameter settings in LoRaWAN simulations to ensure reproducibility and accuracy. Harinda et al. [9] compared empirical propagation models (Okumura-Hata, Log-distance, COST231) for urban LoRaWAN scenarios, identifying Okumura-Hata as the most reliable in city environments.

However, most ADR research employs simplified propagation models that fail to capture realistic fading and interference effects. Boudjellal and Bounceur [10] proposed methods to optimize LoRa network performance for IoT applications, while Malik et al. [11] examined LoRa scalability in congested environments. Yet, these studies do not sufficiently address how different ADR mechanisms perform under varying traffic loads and collision scenarios. Polonelli et al. [12] investigated Slotted ALOHA on LoRaWAN, providing insights into MAC layer optimization, but without exploring interactions with different ADR strategies.

While several studies focus on optimizing ADR strategies, most are either centralized or lack evaluation under realistic fading models and dynamic traffic conditions. Channel modeling efforts exist but are often disconnected from ADR evaluations. Furthermore, decentralized ADR approaches remain underexplored, particularly in dynamic environments with fading. To address these gaps, this paper proposes a decentralized ADR scheme and evaluates it using Free Space Path Loss, Okumura-Hata, and Nakagami-m models under varying traffic loads.

3. Methodology

Figure 1 illustrates the simulated LoRaWAN network model. The network consists of 100 end devices (nodes) that are randomly distributed over a 10 × 10 km2 area. The gateway is deployed at the center to ensure even communication coverage. The simulation environment is developed using Python. Each node generates uplink traffic at random intervals, with rates ranging from 5 to 300 packets per hour. The transmissions follow the behavior of Class A devices under the pure ALOHA protocol, meaning that nodes send data without channel sensing. The network operates within 865 MHz to 867 MHz. Within this frequency band, eight orthogonal uplink channels are allocated, each having a bandwidth of 125 kHz. The simulation setup reflects a rural deployment scenario. Table 1 lists the key simulation parameters used

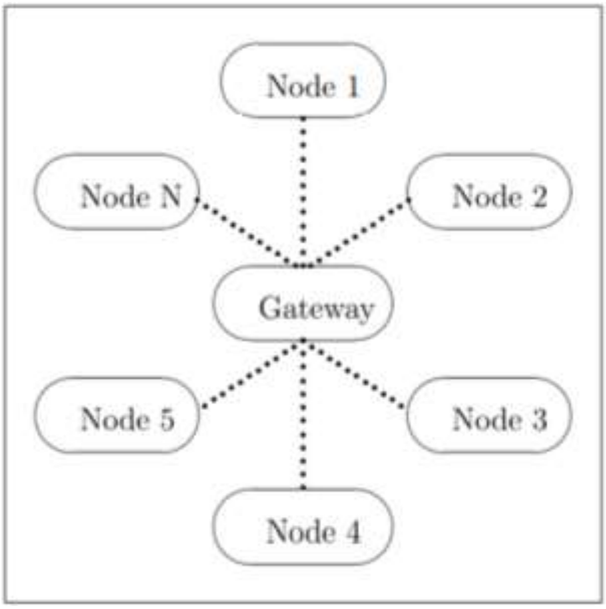

Fig -1: LoRaWAN Network Setup

Table -1: Simulation Parameters

Parameter	**Value**
Total nodes	**100**
Simulation area	**10 km × 10 km**
Available channels	**8**
Payload size	**51 bytes**
Channel bandwidth	**125 kHz**
Operating frequency	**865–867 MHz**
Gateway height	**15 m**
Node height	**1.5 m**
Coding rate	**4/5**

3.1. Channel Models

To simulate a realistic rural deployment scenario, a hybrid path loss model is combined with a fading channel model. The models used in the simulation are described in the following subsections.

3.1.1. Free Space Path Loss (FSPL)

The Free Space Path Loss (FSPL) model assumes an ideal line-of-sight propagation scenario between the transmitter and receiver, i.e., no obstacles in between. In the simulation, for distances less than or equal to 1 km, the 100 FSPL model is applied [13].

3.1.2. Okumura-Hata Model

The Okumura-Hata model is an empirical path loss model widely used to predict signal attenuation in urban, suburban, and rural environments. This model is valid for frequencies between 150 MHz and 1500 MHz, making it suitable for LoRaWAN networks operating in the sub-GHz band. In the simulation, the Okumura-Hata model is applied for distances beyond 1 km to account for large-scale signal attenuation caused by terrain features and vegetation.

3.1.3. Nakagami-m Fading

To account for small-scale fading, which is essential for realistic channel modeling, the Nakagami-m fading model is employed due to its flexibility in adapting to various fading conditions through its adjustable fading severity parameter m.

3.2. Time-on-Air (ToA) Calculation

Time-on-Air determines how long the channel is occupied by a single packet. It directly affects the collision rate and energy consumption. The total ToA includes the preamble and payload transmission durations.

3.3. Adaptive Data Rate (ADR)

In this project, the core ADR optimization logic (Algorithm 1) is adopted for the adjustment of transmission parameters (SF and Tx Power) based on SNR and RSSI. The optimization logic is common to both centralized and decentralized approaches; its execution differs as discussed in the next sections.

Algorithm 1 ADR Algorithm

Require:

Average SNR (SNR_avg), Minimum SNR (SNR_min), margin

Ensure:

Optimized SF and TP

1. Compute SNR Margin:
   $\text{SNR\_margin} = \text{SNR\_avg} - \text{SNR\_min} - \text{margin}$
2. Compute adjustment steps:
   $\text{N\_step} \leftarrow \text{SNR\_margin} / 3$
3. If $\text{N\_step} > 0$ then (Good signal quality)
4. If $\text{SF\_i} \leq \text{SF\_min}$ then
5. Reduce SF (increase data rate)
6. Else
7. Reduce TX Power (save energy)
8. End If
9. Else If $\text{N\_step} < 0$ then (Poor signal quality)
10. If $\text{SF\_i} \geq \text{SF\_max}$ then
11. Increase SF (improve sensitivity)
12. Else
13. Increase TX Power (if allowed)
14. End If
15. Else ($\text{N\_step} = 0$)
16. Optimized Parameters
17. End

3.3.1. Centralized ADR Algorithm

In the centralized ADR approach, transmission parameter optimization is performed at the gateway, which acts as a proxy for the Network Server (NS) to simplify the simulation without requiring a full server–gateway infrastructure. End devices periodically report SNR and RSSI through uplink packets, and the gateway collects this feedback over an observation window of the last 20 uplinks, chosen as a balanced trade-off between stability and adaptation speed. After computing the average SNR for each node, the gateway applies the ADR algorithm to determine the optimal Spreading Factor (SF) and transmission power, and communicates these settings back via a downlink message containing a LinkADRReq command. This centralized control enables efficient network-wide management, assigning lower SFs to closer nodes and higher SFs to distant nodes, thereby reducing airtime, minimizing collisions, and improving overall channel utilization.

3.3.2. Decentralized ADR Algorithm

In the decentralized ADR approach, transmission parameter optimization is performed locally at each end device. Each node independently monitors its SNR and RSSI values from gateway acknowledgments (ACKs) and maintains a sliding window of the most recent 20 ACKs to estimate the average SNR. Based on this estimate, the node applies the ADR logic to adjust its Spreading Factor (SF) and transmission power autonomously. If ACKs are not received for a predefined number of transmissions (timeout threshold), the node assumes poor link quality and incrementally increases its SF or transmission power until connectivity is restored, though this may increase airtime and energy consumption. Unlike the centralized method, no LinkADRReq downlink command is required, reducing dependency on network infrastructure. This self-adaptive mechanism enables faster response to environmental changes, improves scalability, and is better suited for scenarios with limited downlink capacity or infrequent gateway feedback.

4. Results and Discussion

4.1. SF Allocation

Initially, all nodes were configured using Static ADR, where each node was assigned the highest Spreading Factor (SF12). This conservative approach maximizes coverage and link reliability but leads to inefficient airtime utilization due to the prolonged transmission time associated with SF12. As shown in Figure 2, the total network airtime consumed under Static ADR was 246.57 seconds.

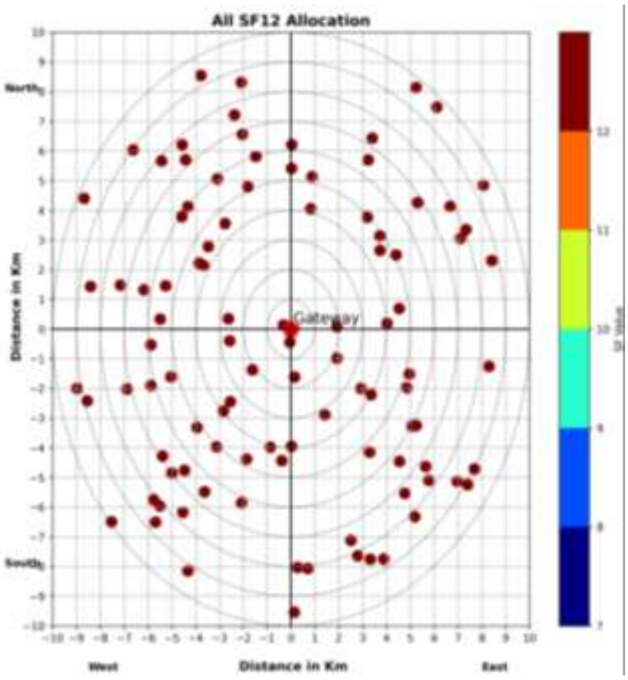

Fig 2-SF 12 Allocation

Initially, all nodes were configured using Static ADR, where each node was assigned the highest Spreading Factor (SF12). This conservative approach maximizes coverage and link reliability but leads to inefficient airtime utilization due to the prolonged transmission time associated with SF12. As shown in Figure 2, the total network airtime consumed under Static ADR was 246.57 seconds.

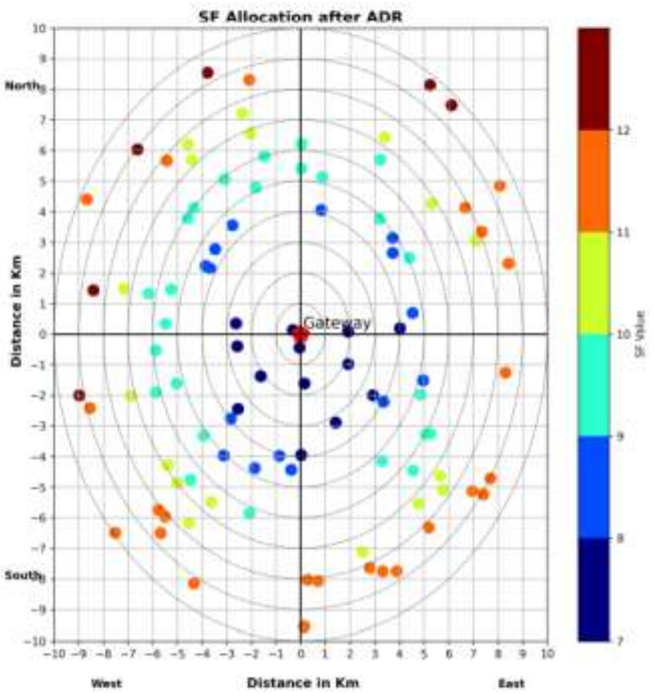

Fig 3- SF allocation using Centralized ADR

Subsequently, the Centralized ADR algorithm was applied. Based on the SNR values derived from channel conditions, the network server reassigned appropriate SFs to each node. This reallocation significantly reduced total airtime to 73.20 seconds, as shown in Figure 3.

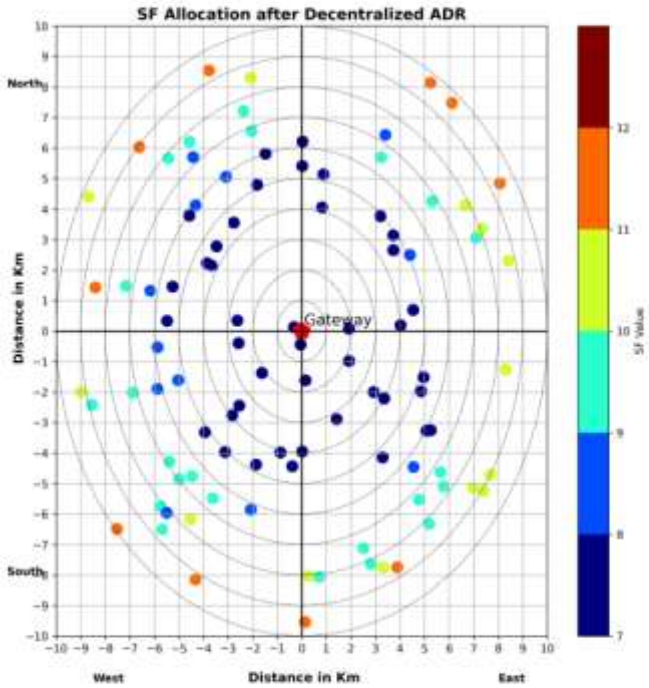

Fig 4- SF allocation using Decentralized ADR

Finally, the Decentralized ADR algorithm was applied, further optimizing the SF distribution and reducing the total airtime to 39 seconds. The SF allocation under this approach is depicted in Figure 4.

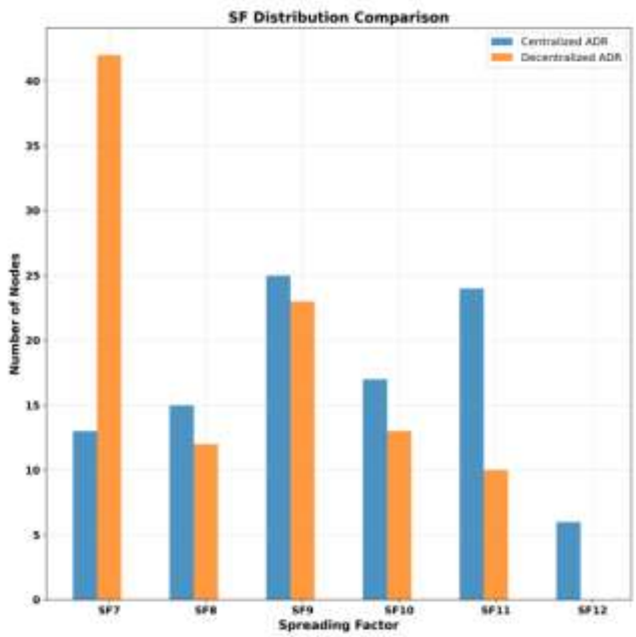

Fig 5- Comparison of SF distribution

A comparative view of SF distributions under Centralized and Decentralized ADR schemes is presented in Figure 5. he SF distribution for Decentralized ADR reveals that a substantial proportion of nodes are assigned the lowest spreading factor (SF7), thereby minimizing time-on-air and reducing overall channel occupancy. Conversely, Centralized ADR allocates a significant number of nodes to higher spreading factors (SF9–SF11), which increases transmission durations and, consequently, channel utilization.

4.2 Comparison and Evaluation

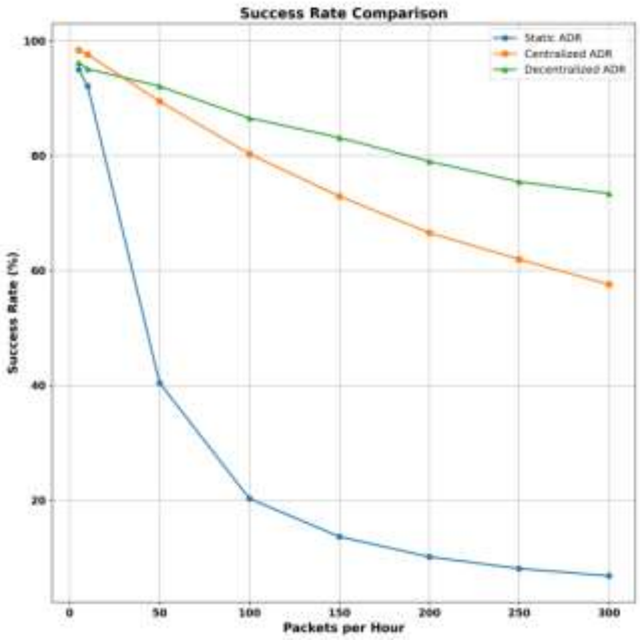

Fig 6- Comparison of Successful Packet Rate

The three approaches Static ADR, Centralized ADR, and Decentralized ADR are compared using key performance metrics: throughput, successful packet rate, and collision rate. Figures 6, 7, and 8 illustrate performance under varying traffic loads, where each node transmits data at random intervals with rates ranging from 5 to 300 packets per hour.

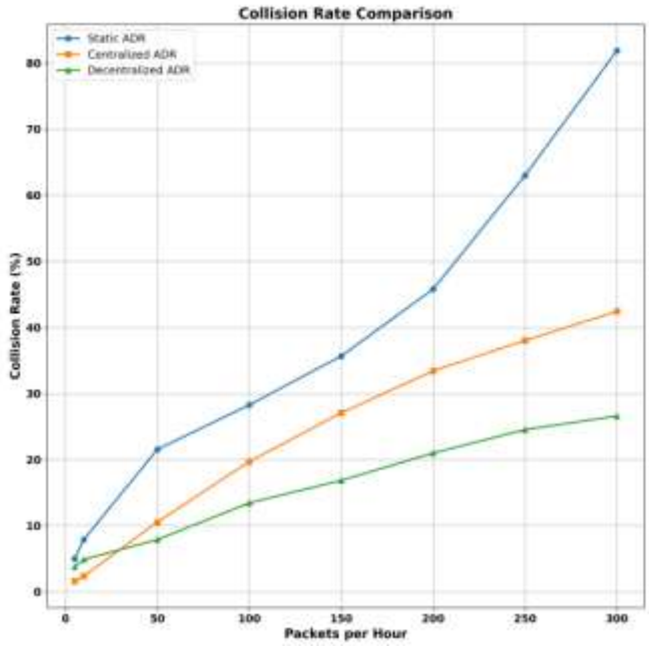

Fig 7- Comparison of Collision Rate

With increasing network congestion, the collision rate in static ADR rises significantly, as all nodes are assigned SF12 and increased probability of packet collisions. Centralized ADR mitigates this issue by dynamically reallocating spreading factors among nodes, thereby reducing the collision rate. Decentralized ADR further optimizes SF distribution through localized adaptation mechanisms, resulting in the lowest collision rate,as shown in Figure 7.

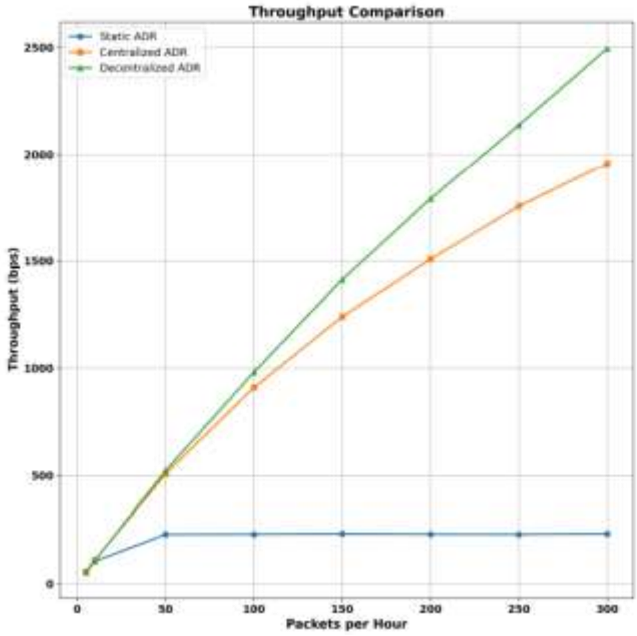

Fig 8- Comparison of Throughput

Decentralized ADR consistently outperforms the other methods, While it initially suffers from a cold start, performance improves rapidly. Notably, Decentralized ADR achieves an 18.6% improvement in throughput over the centralized approach, as shown in Figure 8.

5. Conclusion

This work has presented a comparative performance analysis of three Adaptive Data Rate (ADR) mechanisms in LoRaWAN: Static ADR, Centralized ADR, and the proposed Decentralized ADR. The evaluation focused on key metrics such as throughput, collision rate, and packet success rate under varying network loads. The results clearly indicate that the Static ADR approach, The Static ADR approach, with fixed SF12 allocation, results in excessive airtime consumption averaging 246.57 seconds—and high collision rates, especially under dense network traffic. While Centralized ADR improves performance by dynamically adjusting transmission parameters, reduces average Time-on-Air to 73.20 seconds.However, this method still faces issues in fast-changing environments, because it relies on the server's feedback. In contrast, the proposed Decentralized ADR method—where each node autonomously adjusts its transmission parameters based on locally observed SNR values demonstrates superior performance. It reduces average airtime to 39 seconds, improves throughput by 18.6% over Centralized ADR, and ensures more efficient SF distribution. This leads to enhanced scalability, reduced collisions, and better energy efficiency in dense LoRaWAN deployments.

REFERENCES

1. LoRa Alliance: LoRaWAN®Specification v1.0.4. [Online]. Available: https:// lora-alliance.org/resource hub/lorawan-specification-v1-0-4/ (2020)
2. Mekki, K., Bajic, E., Chaxel, F., Meyer, F.: A comparative study of lpwan technologies for large-scale iot deployment. ICT Express 5(1), 1–7 (2019) https: //doi.org/10.1016/j.icte.2017.12.005
3. More, P.A., Patel, Z.M.: Lorawan performance analysis with data rate. In: Proc. 1st Int. Conf. Cognit. Comput. Eng. Educ. (ICCCEE), pp. 1–6 (2023). https: //doi.org/10.1109/ICCCEE55951.2023.10424626
4. Semtech Corporation: LoRa Modulation Basics. AN1200.22. [Online]. Available: https://www.semtech.com/uploads/documents/an1200.22.pdf (2015)
5. Lehong, C., Isong, B., Lugayizi, F., Abu-Mahfouz, A.M.: A survey of lorawan adaptive data rate algorithms for possible optimization. In: Proc. 2nd Int. Multidiscip. Inf. Technol. Eng. Conf. (IMITEC), pp. 1–6 (2020). https://doi.org/10. 1109/IMITEC50163.2020.9334144
6. Serati, R., Teymuri, B., Anagnostopoulos, N.A., Rasti, M.: Adr-lite: A lowcomplexity adaptive data rate scheme for the lora network. In: Proc. 18th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob), Thessaloniki, Greece, pp. 1–6 (2022). https://doi.org/10.1109/WiMob55322.2022.9941614
7. Alipio, M., Bures, M.: Current testing and performance evaluation methodologies of lora and lorawan in iot applications: Classification, issues, and future directives. Internet Things 25 (2024) https://doi.org/10.1016/j.iot.2023.101053
8. Campos, J., Ortega, P., Pedroza, A., Santos, R.: Standardized parameter settings for lorawan simulations: ensuring reproducibility and accuracy in iot network evaluation. Computer Networks 198, 108356 (2021) https://doi.org/10.1016/j. comnet.2021.108356
9. Harinda, E., Hosseinzadeh, S., Larijani, H., Gibson, R.M.: Comparative performance analysis of empirical propagation models for lorawan 868 mhz in an urban scenario. In: Proc. IEEE 5th World Forum Internet Things (WF-IoT), Limerick, Ireland, pp. 102–107 (2019). https://doi.org/10.1109/WF-IoT.2019.8767245
10. Boudjellal, S., Bounceur, A.: An approach to optimize lora network performance for efficient iot applications. Wireless Pers. Commun. 128, 209–229 (2023) https: //doi.org/10.1007/s11277-022-09756-6
11. Malik, M., Kothari, A., Pandhare, R.: Scalability analysis of lora and sigfox in congested environment and calculation of optimum number of nodes. Sensors 24(20) (2024) https://doi.org/10.3390/s24206673
12. Polonelli, T., Brunelli, D., Marzocchi, A., Benini, L.: Slotted aloha on lorawan— design, analysis, and deployment. Sensors 19(4) (2019) https://doi.org/10.3390/ s19040838
13. Rappaport, T.S.: Wireless Communications: Principles and Practice, 2nd edn. Prentice Hall PTR, Upper Saddle River, NJ, USA (2002)