(Online) ISSN 2712-0562
Original Research

Sustainable Engineering and Innovation
Vol. 7, No. 2, 2025, pp.647-662
https://doi.org/10.37868/sei.v7i2.id665

Improving IoT support in Smart Cities through LoRa technology upgrading

Marwan S. M. Al-Dabbagh[1*], Carlos T. Calafate[2], Pietro Manzoni[3]
[1] Department of Computer Science, College of Education for Pure Science, Mosul University, Mosul, Iraq
[2,3] Department of Computer Engineering (DISCA), Universitat Politècnica de València, Valencia, Spain

*Corresponding author E-mail: Marwan.aldabbagh@uomosul.edu.iq

Received Sep. 24, 2025
Revised Dec. 2, 2025
Accepted Dec. 8, 2025
Online Dec. 15, 2025

Abstract

The Internet of Things (IoT) has advanced Smart City services through extensive device connectivity. LoRa, a leading LPWAN technology, provides long-range communication with low power consumption but suffers from scalability, latency, and energy-efficiency challenges in dense urban settings. To address these issues, this study introduces an integrated optimization framework that combines adaptive data rate (ADR) control, multi-channel communication, and dynamic resource allocation. The framework aims to reduce transmission delays, minimize packet collisions, and improve overall energy performance. It leverages multi-channel communication to distribute traffic, resource scheduling to prioritize critical data, and ADR to adjust transmission power and data rate based on real-time network conditions. Large-scale simulations conducted in OMNET++ demonstrate significant improvements over standard LoRa configurations, including baseline, ADR-only, and multi-channel setups. In a representative urban environment, the proposed framework achieved packet delivery rates of approximately 92.3% at 300 nodes and 85.7% at 900 nodes, while maintaining low latency and energy consumption. Overall, the integrated approach delivers robust performance across varying node densities, making it a strong candidate for future large-scale IoT deployments in Smart City architectures.

Keywords: LoRa Network, Smart Cities, Adaptive data rate, LPWAN technology, IoT

1. Introduction

The development of smart cities can benefit greatly from the Internet of Things (IoT). The IoT allows devices to communicate and share data for better urban management. This technological revolution step serves a variety of public needs, including real-time traffic and environmental control systems, smart public safety monitors, and smart meters [1]. All of these IoT functions can perform in a 24/7 active urban resource optimization setting. However, the effectiveness of these IoT networks heavily depends on the availability of powerful communication protocols that can be applied to widespread IoT deployments with a constrained energy and latency budget, while maintaining heightened dependability [2]. LoRa long-range low-power wide-area networks (LPWAN) have gained widespread adoption for IoT solutions due to their extended range capabilities and minimal power usage. This network protocol operates using frequency bands below one GHz, instead different from the conventional wireless channels used by Wi-Fi and Zigbee. This allows LoRa to achieve superior coverage and obstacle penetration compared to conventional systems [3]. LoRa networks are perfect for Smart City applications because they support wide-area coverage while maintaining low power requirements for IoT devices. Although LoRa networks support IoT systems through long-range communication, their performance declines notably in dense urban locations. A lot of IoT devices may cause a lot of network traffic, which can cause latency issues, lower packet delivery ratios (PDR), and too many packet collisions [4]. LoRa networks provide low bandwidth, and this makes effects very difficult, particularly when trying to scale to large-scale IoT systems. When dealing with battery-powered IoT, which needs extended battery life without having to be changed or charged continuously, efficiency is even harder to achieve [3]. Different optimization techniques were used to fix the issues with LoRa networks and make them perform optimally.

The adaptive data rate (ADR) optimization method adjusts transmission parameters like power and data Kbps depending on network conditions to make communication more reliable and save energy [5]. Several studies have examined multi-channel communication systems, which send network data across several frequency channels to enhance performance and reduce traffic [6]. Resource scheduling techniques use time division multiple access (TDMA) and class-based precedence scheduling to limit the number of packet collisions and make data transfer more dependable [7]. However, many of these findings are relevant to certain problems; they don't completely satisfy the requirements for the correction of a series of faults in a LoRa network system. ADR optimization by itself does not work efficiently in areas with plenty of traffic since there is no network traffic operation. To provide for varying network conditions, adaptive resources are necessary for communication operation and resource scheduling on multiple channels. This study offers a complete framework that combines ADR optimization with multi-channel communication and dynamic resource scheduling to address the problems with the existing LoRa network. This framework aims to make networks run better by reducing packet collisions, lowering latency, and making better use of energy. The proposed method combines various strategies to provide a complete solution that works well for Smart City demands.

The main contributions of this paper are summarized as follows:

- ✓ Integration of multiple optimization techniques: A new technique has been introduced, which integrates adaptive data rate (ADR) optimization, multi-channel communication, and dynamic resource scheduling for effective utilization of LoRa networks in smart cities. The approach integrates the various gaps of methods that work in isolation with an integrated implementation.
- ✓ Enhanced network scalability and efficiency: The designed framework is intended for application in the relative scalability of the networks in an IoT environment of thick fog, thereby again reducing packet collisions and congestion without making demands on energy.
- ✓ Dynamic and adaptive mechanisms: The designed framework also reacts dynamically to changing transmission requirements, including timeslot and channel allocation, together with that for the other resources, to the end that the specific time-dependent varying demands of performance of the network behaviors of the different network scenarios can be obtained.
- ✓ Extensive Simulation-based Validation: The validity of the performance of the methods has been extensively validated with a large number of detailed OMNeT++ simulations using varying network scenarios and the density of traffic nodes. Various key performance indicators, including packet delivery, latency, and utilization of energy, have been examined with the aim of establishing their suitability for the methods proposed.
- ✓ Applicability to Smart City: The intended applicability of the technology developed in this work is in smart cities for actual applications such as traffic management, environment monitoring, and smart metering systems, etc., thereby illustrating the merits of the method as developed.

By addressing scalability, reliability, and energy efficiency challenges, this study provides a significant contribution to advancing LoRa networks for IoT applications in urban environments.

This paper is organized into the following sections: Section 2 presents the related work in LoRa network enhancements. Section 3 introduces LoRa technology and applications in Smart Cities. Section 4 explains the proposed framework. Section 5 then presents the simulation setup, including the simulation environment, simulation parameters, scenarios, and metrics. Section 6 analyses the results obtained by comparing the performance of the proposed framework with baseline scenarios and existing techniques. Finally, Section 7 concludes the paper, summarizing the main findings and listing future lines.

2. Related work

The accelerated advancement of IoT solutions and their application in Smart Cities has stimulated substantial research into the optimization of LoRa networks. Various improvements to handle scalability, energy efficiency, and latency issues in dense IoT installations have been the subject of current research [8]. In particular, this section seeks to summarize related work on the topics of ADR optimization, multi-channel communication, and resource allocation.

2.1. Adaptive data rate (ADR) optimization

One of LoRaWAN's main characteristics, ADR dynamically changes the transmission power of devices—such as power and data rate—based on signal-to-noise ratio (SNR) and network circumstances. While earlier research works have shown that ADR may lower energy use and collisions, its efficacy in high-density installations (without further improvements) tends to drop [9].

Slabicki et al. [10] presented a study on adaptive configuration of communication parameters for dense IoT scenarios using LoRa networks. They implemented FLoRa, an open-source framework for end-to-end LoRa simulations built on OMNeT++. Their proposed solution increases reliability and power efficiency in noisy channels, independently of network size. The results show that ADR improves the delivery ratio under stable conditions while maintaining low energy consumption. However, because dynamic transmission-parameter configuration is not incorporated into the ADR mechanism's decision process, ADR performance is affected—especially in highly variable wireless channels—indicating that additional mechanisms are required.

Ilarizky et al. [11] show the importance of the optimization of the adaptive data rate because of the possible enhancement of the network's performance. They study the performance of the ADR scheme realized by the network-server side in LoRaWAN networks. The results show that the optimized version of ADR yields energy savings of up to 25.07 % and an increase of 6.86 % in packet success rate (PSR) over the standard ADR.

Ksiazek K. and Grochla K. [12] presented an extensive simulation study of the behavior of energy usage and packet delivery probability for different ADR parameters. They discuss the performance of both basic ADR and an enhanced ADR+ under different network conditions. Their results point out that generally ADR+ outperforms basic ADR, especially in high variance signal conditions.

He J. [13] put forward a novel ADR algorithm for LoRa networks based on the exponential weighted moving average (EWMA) system. The paper addresses the limitations of the standard ADR under unstable channel conditions. The findings indicated that the proposed algorithm significantly outperformed the standard ADR and ADR algorithms.

2.2. Multi-channel communication

Multi-channel communication is one of the most explored methods to disseminate network traffic, which leads to relieving traffic in LoRa networks. studies show that multi-channel communication improves results, but it usually lacks coordination with other optimization methods [6].

Liu et al. [14] introduced ChirpBox, a cost-effective LPWAN testbed for cognitive IoT research and development by using an all-to-all multi-channel protocol leveraging concurrent transmissions for efficient communication. The study proposed the LoRaDisC protocol, which addresses LoRa's data rate and duty-cycle constraints for reliable data exchange.

Yu et al. [15] focused on improving performance for nodes located at the boundaries between spreading-factor regions and proposed an adaptive multi-channel allocation policy. Their research highlighted the long-range communication capabilities of LoRa networks—achieving 2–5 km in urban environments and 10–30 km in rural areas—and described an IoT network built on LoRa technology operating in the unlicensed 868 MHz band with a total bandwidth of 1 MHz.

Petrariu et al. [16] presented a low-cost, high-performance multi-channel gateway for LoRaWAN technology in rural and urban environments, enhancing IoT communication capabilities. The study has validated gateway performance using the TTN cloud data network, where measurements have indicated better performance in rural areas due to fewer signal obstructions, ensuring effective data transmission.

2.3. Resource scheduling

To enhance the performance of LoRa networks, resource scheduling techniques such as time division multiple access (TDMA) and priority scheduling have been investigated; however, they have not been studied in conjunction with adaptive data rate (ADR) and multi-channel configurations. Thus, more enhancements are possible with a fully optimized solution [17].

Resource management in long-range (LoRa) wireless networks is researched by Hamdi and Qarage [18], specifically on the focus of energy-efficient management for low-powered connecting devices, which are powered by independent energy harvesting sources. The study presents an optimal SF assignment algorithm to maximize energy bonded and channel condition energy for scheduled LoRa devices (LDs). The use of SF assignment as a basis of simulation proved the system's performance to be impressive, and of the proposed systems to outperform a randomly assigned SF.

In the study presented by [19] an analysis of the problems involved in IoT study was made with a concentration on resource management in networks that are dense and with limited capability of devices. The intelligent use of radio resource allocation in LoRaWAN was analyzed, with the resource allocation efficiency improved using different reinforcement learning techniques. The effect of capture and of inter-SF collisions on the performance of the network is stressed. Even though in their conclusions they make use of a performance analysis of the EXP3.S algorithm, again measured against centralized solutions and simple heuristics under realistic conditions, the scalability of the network is limited by increased inter-SF collisions and the effect of capture as the density of devices employed is increased.

Gao et al. [20] provide an account of a dynamic network resource allocation system. This study focuses on the AdapLoRa system with the objective of maximizing the network lifetime of LoRa networks using resource allocation changes depending on the quality of the links. It uses a fine-granulated network model to estimate the impacts of adjustments in links and networks. It also accounts for the impacts of the error correction schemes of the network. There are a few sins in the AdapLoRa system that limit its stiffness to network environment changes. Most notably, the mobility of the end devices and distances change in relation to the gateways. This can have a remarkable impact on the persistence and performance of the network.

Although the above studies are a valuable contribution, they have several limitations in common. Studies using ADR typically study data rate adaptation independently, with no reference to either channel load imbalance or scheduling policy. Multi-channel approaches improve traffic load balance but do not support dynamic adaptation of transmission parameters. On the other hand, resource-scheduling works try to make a better exploit of assigned slots, but they do not often take into account ADR or multi-channel coordination in their design. These piecemeal techniques leave an apparent gap in the literature. There is currently no work that offers a unifying solution that incorporates ADR, channel diversity, and resource scheduling to deal simultaneously with congestion control, energy efficiency, and scalability in dense Smart City scenarios. Thus, this study aims to address the lack of studies that combine these three goals under one study design.

3. Introduction to LoRa/LoRa WAN network

Long-range radio, or LoRa for short, is a physical layer technology that Semtech created and marketed for long-range connectivity and low-power communications. It is used in many Internet of Things (IoT) networks worldwide, including for energy management, natural resource reduction, pollution control, infrastructure efficiency, disaster prevention, etc. [21]. LoRa has the ability to allow reliable connections over distances of 2–5 km in urban areas, and up to 15 km in suburban areas [22], [23]. The LoRaWAN network's architecture, as shown in Figure 1, consists of four key components interconnected through a star topology: end devices, gateways, network servers, and application servers. EDs communicate with GWs using single-hop LoRa communication. The GW simply relays received messages to a central network server via an IP backbone. The central network server manages the network access, including security functions, communication, and device management, which forwards them to the respective application servers [24], [25].

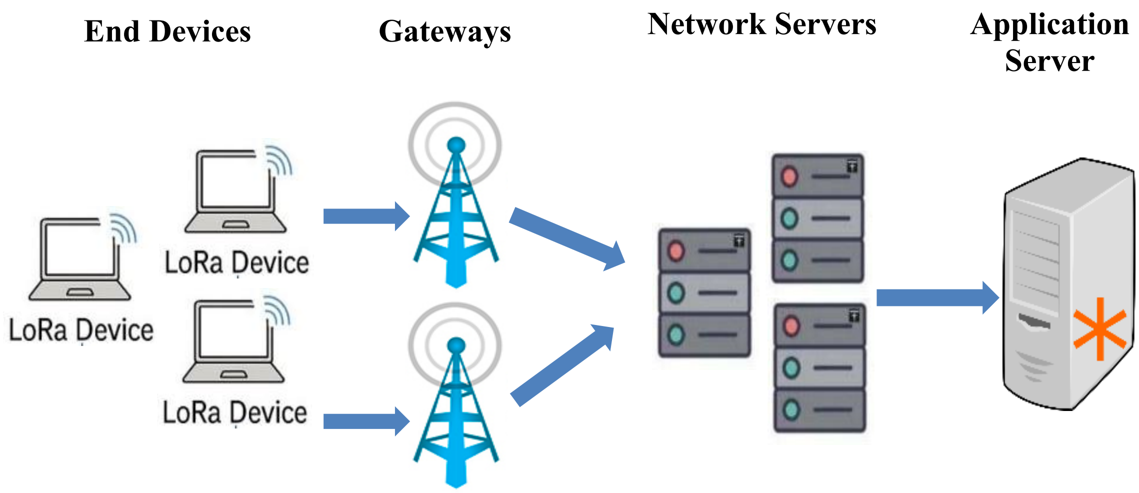

Figure 1. LoRaWAN network architecture

LoRa network relies on two components, namely, LoRa and LoRaWAN, each corresponding to a different layer of the protocol stack, as shown in Figure 2 [26].

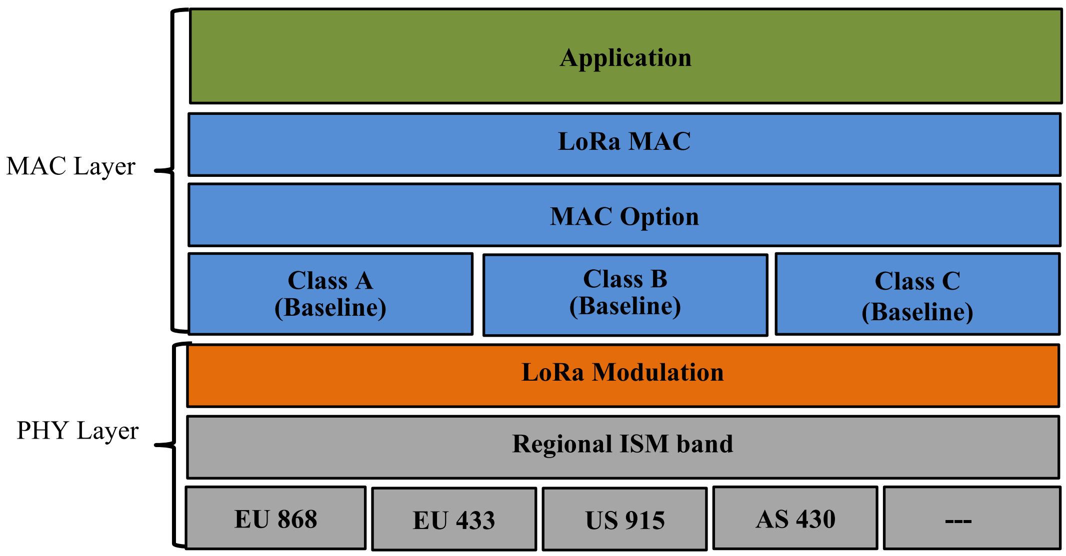

Figure 2. LoRaWAN layers

The LoRa physical layer operates in the unlicensed sub-GHz ISM band. LoRa utilizes chirp spread spectrum modulation to encode an input signal into chirp pulses spread over a wide spectrum. With LoRa's technique, long-distance communication is attainable even with low data rates. Any LoRa transmission, and its uniqueness, can be distinguished by the five parameters of the spreading factor, transmission power, code rate, center frequency, and bandwidth. Each of these parameters plays a role in adjusting communication range and data rate, determining robustness towards interference and noise, and the receiver's capability to decipher the information. The available values for each parameter depend on the region where LoRa devices are deployed. The spreading factor is the ratio between the data symbol rate and chirp rate. The configuration of the spreading factor allows tuning the data rate and the reachable distance. In fact, LoRa provides a maximum data rate of 27 kbps using spreading factor 7 and 500 kHz bandwidth. Higher spreading factors reduce the data rate but extend the range. Choosing different spreading factors also enables orthogonal signals, implying that a receiver can successfully receive distinct signals sent over a given channel at the same time. Transmission power is typically limited by regional regulations, which may depend on frequency band and duty cycle constraints. The code rate is the forward error correction rate, and it affects the airtime of packet transmissions [27], [28]. The center frequency depends on the ISM band used in a particular region.

4. Proposed framework

This paper takes a holistic position of how to functionality of LoRa networks in a Smart City environment. There are three core elements that compose the proposed solution these are intelligent resource allocation, multi-channel communication, and adaptive data rate (ADR) optimization. The entire aLorac Smart City framework for the selected city of a suggested LoRa architecture is shown in Figure 3. It portrays the framework of the suggested layout and the interactions of the resource scheduler with the multi-channel handler and the ADR. The system consists of a centralized network server that executes the proposed optimization logic in conjunction with several LoRa gateways and end devices. The ADR module intelligently adapts the transmission parameters of each device according to the real-time state of the network, and the multi-channel handler allocates the traffic through the various available frequencies. At the same time, a hybrid scheduling unit uses a combination of priority-based scheduling and TDMA to ensure efficient time-slot allocation. These components address the persistent issues of latency, congestion, and energy inefficiency in dense urban IoT environments. They are theoretically based on wireless communication and network optimization concepts.

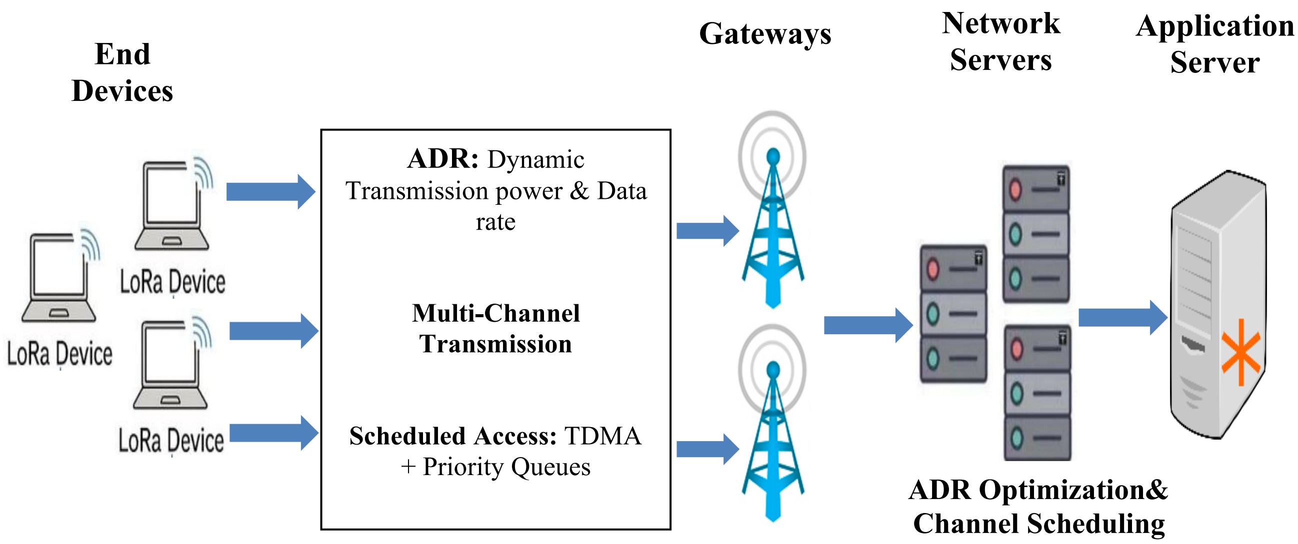

Figure 3. The Proposed framework architecture

The pseudocode below summarizes the initial and main functions for the proposed framework.

Algorithm 1: Initialization Function

```
Input:
   N  → Number of end nodes
   G  → Number of gateways
   A  → Simulation area
   F  → Set of available frequency channels
   SF → Set of LoRa spreading factors
   T  → Set of TDMA time slots
   P  → Set of application priority levels
   maxTime → Total simulation duration
   DR_min, DR_max → Minimum and maximum data rates
   DR → is a number of additional selected rows
   PWR_min, PWR_max → Transmission power limits
   SNR_HighThreshold, SNR_LowThreshold → SNR decision bounds

Function InitializeNetwork(N, G, A, F, SF):
   Randomly position N nodes and G gateways within area A
   Initialize each node with default transmission parameters:
      - Data rate
      - Transmission power
      - Frequency channel
   Return configured network topology
```

Algorithm 2: Main Function

```
Function LoRaOptimizationFramework():
   Network ← InitializeNetwork(N, G, A, F, SF)
   While simulationTime < maxTime:
      For each node ∈ Network:
            AdaptiveDataRate(node, gateway, HighThreshold, LowThreshold, DR_max,
                             DR_min, PWR_max, PWR_min)

            ChannelAllocations ← AssignChannels(Network.nodes, F)

            Schedule ← ResourceScheduling(Network.nodes, T, P)

      For each time slot t ∈ Schedule:
            ExecuteTransmission(Schedule[t])

            UpdateNetworkMetrics(N)

   Return PerformanceMetrics(Network)
```

4.1. Adaptive data rate (ADR) optimization

The ADR optimization algorithm changes the communication parameters of network nodes in real time based on the quality of the gateway signal. It employs the signal-to-noise ratio (SNR) as a specific metric to check the quality of the connection. The approach increases the node's data rate while lowering its transmission power at the same time when the SNR is over a certain high threshold, which means that the transmission conditions are favorable. The goal of this adjustment is to save energy and minimize channel occupancy without making communication less reliable. On the other hand, if the SNR falls below a certain low level, which shows the connection is becoming poorer, the algorithm increases the transmission power and lowers the data rate to keep the connection robust. The ADR algorithm is highly useful for low-power wide-area networks like LoRaWAN because it continually changes the amount of energy used, the amount of data sent, and the stability of the connection to keep them all in balance. The pseudocode below shows how the ADR optimization technique works.

Algorithm 3: ADR Optimization Function

```
Function ADR_Optimization (node, gateway, HighThreshold, LowThreshold, DR_max, DR_min,
                           PWR_max, PWR_min):
  SNR ← Measure_SNR (node, gateway)

  If SNR ≥ HighThreshold:
    node.dataRate ← Increase_DataRate(node.dataRate, DR_max)
    node.txPower  ← Decrease_TxPower(node.txPower, PWR_min)

  Else if SNR ≤ LowThreshold then
    node.dataRate ← Decrease_DataRate(node.dataRate, DR_min)
    node.txPower  ← Increase_TxPower(node.txPower, PWR_max)

  Return updated node settings

Function Measure_SNR(node, gateway):
  distance ← Calculate_Distance (node. Position, gateway.position)
  signal ← node.txPower - PathLoss(distance)
  noise ← Get_Noise_Level()
  Return signal - noise

Function Increase_DataRate(current, max):
  If current < max then Return current + 1
  Else Return current

Function Decrease_DataRate(current, min):
  If current > min then Return current - 1
  Else Return current

Function Increase_TxPower(current, max):
  If current < max then Return current + 1
  Else Return current

Function Decrease_TxPower(current, min):
  If current > min then Return current - 1
  Else Return current
```

4.2. Multi-channel communication

The multi-channel function's goal is to improve how channels are assigned to nodes in a wireless communication network, particularly when there aren't enough frequency resources, and traffic needs change. The algorithm goes over each node in the network and assigns it to a communication channel from a list of options. The current traffic load in each channel and the proximity of the distance from its nearest gateway are the two significant parameters that govern the assignment. The system's goal is to balance network applications and minimize traffic using traffic load. This will increase overall results and lower packet collisions. Adding the distance parameter also guarantees that nodes with worse links — generally those farthest from gateways are allocated to channels that may offer less impediment or more trustworthiness. This approach uses a binary-criterion technique to guarantee communication resources are shared equally and adaptively. This makes the network perform and be adaptive. The next pseudocode provides a brief summary of the multi-channel algorithm.

Algorithm 4: Multi-Channel Function

```
Function AssignChannels(nodes, channels)
        channelLoad ← Dictionary where each channel in channels → 0
        channelAssignments ← Empty dictionary for each node

   For each node ∈ nodes:
        nearestGateway ← FindNearestGateway(node)
        bestChannel ← null
        bestScore ← ∞

        For each ch ∈ channels:

                // Compute traffic and quality metrics
                 load ← channelLoad[ch]
                  interference ← EstimateInterference(ch)
                 distance ← Calculate_Distance(node.position, nearestGateway.position)

                // Compute score: lower is better
                 score ← (weightLoad × load) + (weightDist × distance) + (weightInterf × interference)
                 If score < bestScore then
                         bestScore ← score
                         bestChannel ← ch

        node.channel ← bestChannel
        channelAssignments[node.id] ← bestChannel
        channelLoad[bestChannel] ← channelLoad[bestChannel] + 1

    Return channelAssignments

Function FindNearestGateway(node):
   minDist ← ∞
   nearest ← null
   For each gw ∈ Gateways do:
      dist ← Calculate_Distance (node.position, gw.position)
      If dist < minDist then
         minDist ← dist
         nearest ← gw
   Return nearest

Function EstimateInterference(channel):
   interferenceSources ← GetExternalInterference(channel)
   collisions ← GetRecentCollisions(channel)
   Return α × interferenceSources + β × collisions
```

4.3. Resource scheduling

In time-slotted wireless networks, the resource scheduling algorithm uses a precedence-based system to allocate communication resources. It's used by a two-subcaste system. Time division multiple access (TDMA) guarantees that each device receives a quantum of time to send data. This minimizes packet collision by preventing devices from transmitting data at the same time. This improves the overall performance of the network by making sure that each knot receives its own time to utilize the communication channel. Priority-based scheduling is actually more efficient by placing the most vital applications at the head of the line. For example, nodes that provide exigency alerts or time-sensitive information are assigned lower priority, which allows them to circumvent delays caused by traffic that is less critical. This channel not only reduces quiescence for critical operations, but also keeps effects fair by making sure that all nodes receive an opportunity to utilize the channel in their assigned time windows. The resource scheduling algorithm is concisely described in the subsequent pseudocode.

Algorithm 5: Resource Scheduling

```
Function ResourceScheduling(nodes, timeSlots, priorityLevels)
   priorityGroups ← Group_Nodes_By_Priority(nodes, priorityLevels)
  For each group ∈ priorityGroups do:
    Sort group.nodes by PendingQueueSize descending
  schedule ← Dictionary where each timeSlot ∈ timeSlots → null
  usedSlots ← 0
  maxSlots ← Length(timeSlots)
  For priority in SortedDescending(priorityLevels) do:
    For node ∈ priorityGroups[priority] do:
      If node.HasPendingData() and usedSlots < maxSlots then
        schedule[timeSlots[usedSlots]] ← node
        usedSlots ← usedSlots + 1
  Return schedule
Function Group_Nodes_By_Priority(nodes, priorityLevels)
  groups ← Dictionary where key = priority level, value = empty list
  For each node ∈ nodes do:
    level ← node.priorityLevel
    If level ∈ priorityLevels then
      groups[level].append(node)
  Return groups
Function node.HasPendingData()
  Return Length(node.packetQueue) > 0
Function PendingQueueSize(node)
  Return Length(node.packetQueue)
```

5. The simulation setup

The developed framework offers a practical evaluation due to the addition of an urban setup comprising precisely designed and urban obstacle modeled inclusion, as well as smart-city specific application performance evaluation metrics in varied network environments. Buildings, trees, and walls develop scenarios of signal detrainment and consequently create realistic attenuation conditions to be encountered in actual implementations. To understand proposed extensions, we look at several node net densities in regards to their scalability and flexibility for the proposed enhancements. The assessment of the proposed study fully captures the breadth of the optimization strategies across a richly varied network setup that included baseline configuration, ADR optimization, multi-channel communication, and fully integrated structured parameters and a comprehensive solution. To shed light on the communication network's reliability and efficiency, the evaluative metrics of PDR, latency, and energy consumption were purposefully chosen to provide comprehensive insights into the communication network.

5.1. Simulation environment

To evaluate the efficacy and applicability of the proposed framework, we simulate a realistic urban scenario with multiple obstacles and node distributions. This paper simulates the OMNeT++ simulator with the LoRaWAN module for 86400 seconds (24 hours), as shown in Figure 4. The case study is an urban grid (5km×5km) where nodes are randomly distributed in the scenario, being that the actual number of nodes tested ranging from 300 to 900 to study network scalability. Five gateways have been deployed to ensure full coverage and very high data speeds.

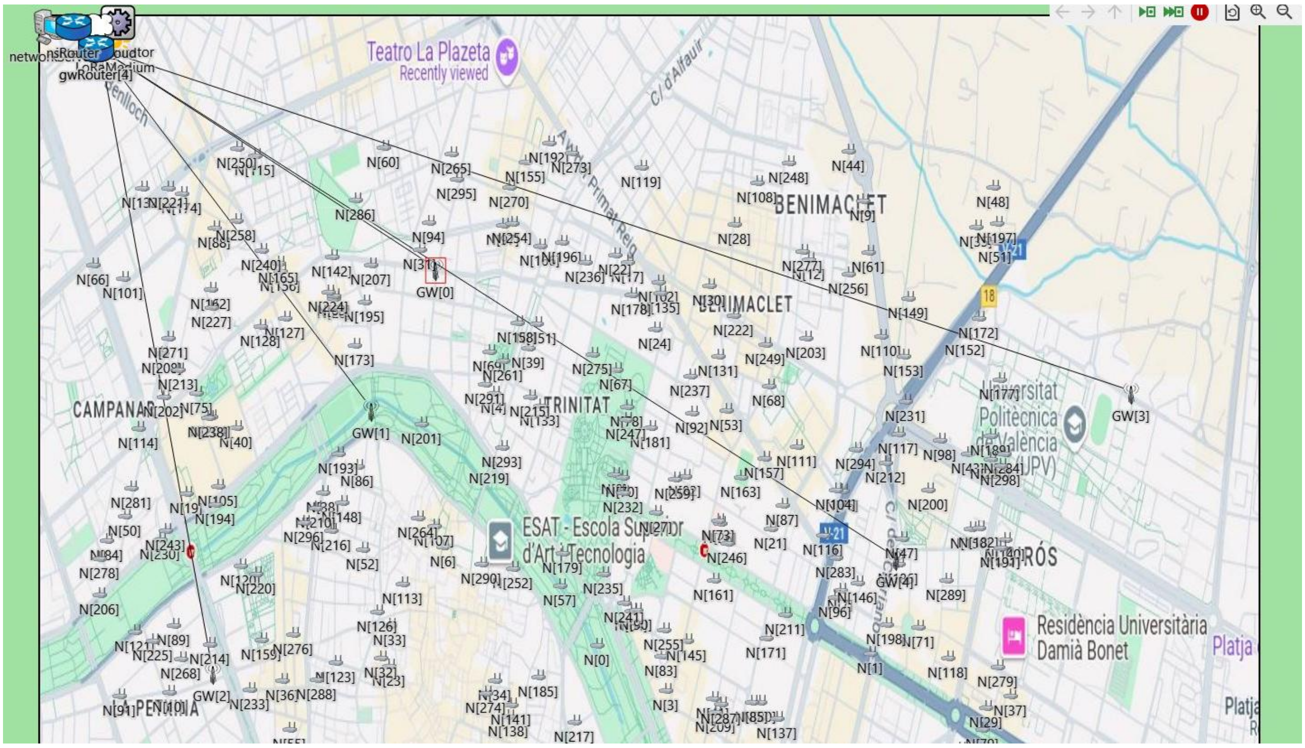

Figure 4. The deployment of gateways and nodes of the LoRa network

The simulation parameter settings are shown in Table 1. This setup lets the recommended network enhancements be thoroughly evaluated.

Table 1. Simulation parameters

Parameter	Value
Simulation Tool	OMNeT++ simulator (v. 6.0) with the LoRaWAN module (v. 2024-Q1 release)
Simulation Time	86400 seconds (24 hours)
Number of Nodes	300, 500, 700, and 900 nodes
Number of Gateways	5 gateways providing full coverage (137 dBm)
Node Distribution	Randomly
Area Size	5 km x 5 km urban grid
Spreading Factors	SF7 to SF12
Power level	2 dBm and 14 dBm
MAC class	Class A
Mobility	Static nodes
Packet Size	20 to 100 bytes
Propagation Model	Log-normal shadowing

5.2. Scenarios and evaluation metrics

Four simulation scenarios are defined to assess the benefit of the proposed framework: (1) A baseline operation scenario using standard LoRa features; (2) a scenario where only the adaptive data rate (ADR) operates; (3) a multi-channel setup based on channels used to spread network traffic; and finally (4) the proposed framework for this paper including ADR, multi-channel support, and resource scheduling. Packet delivery ratio (PDR), delay, and power consumption will be used to assess metrics. The outcomes of the comparisons in these criteria throughout the scenarios assist in providing an overall performance analysis for the suggested improvements in dense urban IoT implementations.

6. Results and discussion

The section reviews simulation results for different scenarios consisting of baseline, ADR only, multi-channel, and the proposed framework across different node densities of 300, 500, 700, and 900 nodes. The results are analyzed based on performance metrics of packet delivery ratio (PDR), latency, and energy consumption.

6.1. Packet delivery ratio (PDR)

With regard to our four unique cases, Figure 5 shows the interpretation of PDR (packet delivery ratio). As more nodes were added, the baseline scenario had the lowest PDR; therefore, it showcased the worst case of packets being lost, and the greater the congestion in the network. The PDR of the baseline scenario continues to worsen while highlighting the case of traditional LoRa networks' PDR and its limitations in large-density cases. PDR dropped from 72.5% to 300 nodes, then further dropped to 60.0% as 900 nodes were added. On the contrary, during the baseline scenario, especially with lower node densities, the ADR only scenario significantly advanced, achieving a PDR of 80.3% with 300 nodes. With a PDR of 72.8% with 900 nodes, ADR’s advantages lessen as network density increases; thus, alone, ADR only performs poorly on sustaining good performance on peaks. The multi-channel scenario surpasses both baseline and ADR-only setups by a remarkable margin. By dramatically reducing PDR congestion at 300 nodes to 85.6% and maintaining 78.5% at 900 nodes, the distributed network of nodes exceeded the expected performance level. Therefore, the improved PDR facilitated by frequency diversity and lower network congestion is lacking under heavy network pressure. By integrating ADR optimization with multi-channel communication and dynamic resource scheduling, the proposed framework reaches the highest PDR for all the different node densities evaluated. With 300 nodes, it achieves a PDR of 92.3%, and with 900 nodes, 85.7%, evidencing the system's capability to handle the complexities of highly dense IoT environments. This reflects the use of multiple optimization techniques in an integrated manner, which translates to greater network reliability and optimization in effectiveness compared to the highly traditional measures or with any of the optimizations used in isolation.

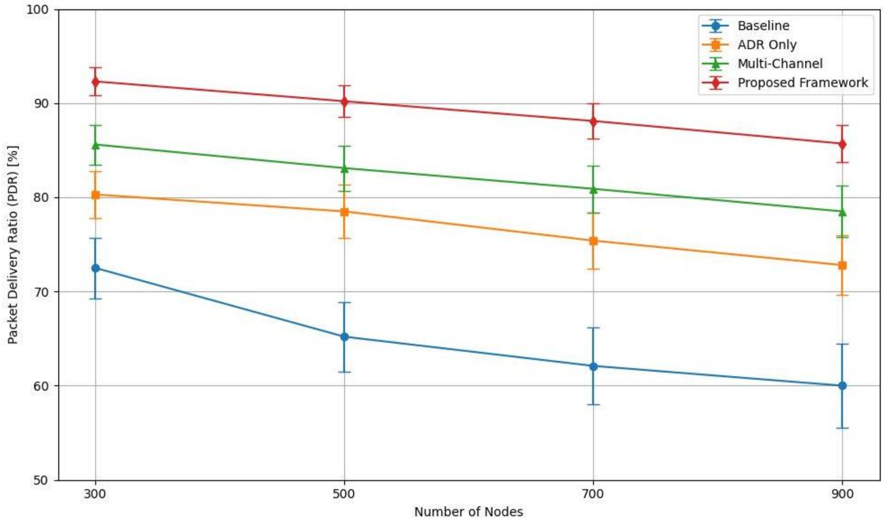

Figure 5. PDR values for the different optimization methods tested when varying the number of nodes

6.2. Latency

Figure 6 depicts the results on latency after testing scenarios. The baseline scenario has the longest latency, with values reaching almost 260 ms at 900 nodes, suggesting that delays get longer with more nodes. This indicates that standard LoRa networks struggle with high-density environments. The ADR reduces some level of congestion, but the ADR only condition shows a slight increase in latency in comparison to the baseline, staying around 200 ms at 900 nodes. At high densities, ADR alone is insufficient to manage the situation. The Multi-Channel scenario is even better, with latency dropping to approximately 165 ms at 900 nodes. This is because additional channels reduce transmission delays and congestion. However, the increase in latency as more nodes are added is still a significant problem that needs addressing. The proposed framework with 900 nodes is the highest density, and combines ADR, multi-channel communication, and resource scheduling to keep the values of latency at 120 ms. This means that it always has the lowest latency. This remarkable result shows that using more than one optimization strategy to reduce transmission delays works well. This means that the proposed framework is scalable and durable for application in smart cities. The big reduction in latency compared to other scenarios suggests that it can support low-latency and real-time applications in IoT settings with a lot of people.

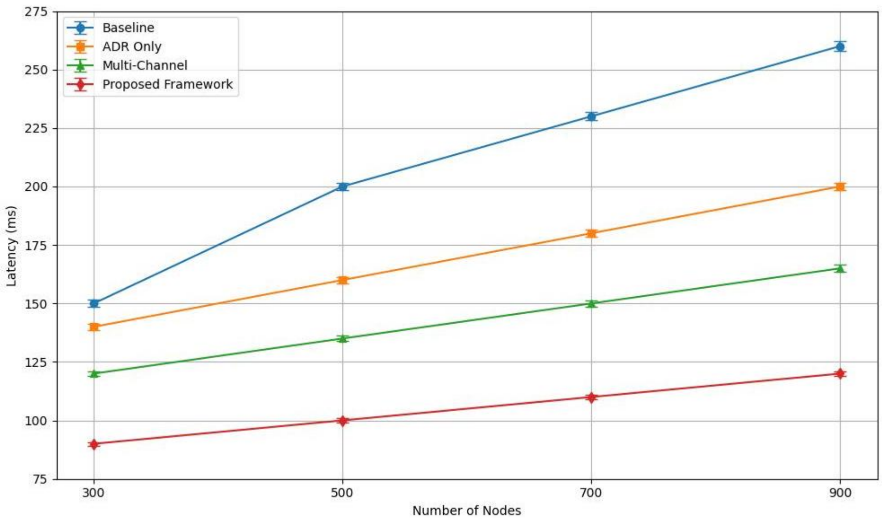

Figure 6. Latency values for the different optimization methods tested when varying the number of nodes.

6.3 Energy consumption

Figure 7 presents energy consumption data for node densities of 300, 500, 700, and 900 nodes across four scenarios: baseline, ADR only, multi-channel, and the proposed framework. The baseline scenario demonstrates the highest energy consumption when increasing from 18 mJ at 300 nodes to 25 mJ at 900 nodes, which indicates substantial inefficiencies for high-density systems management. Traditional LoRa networks struggle to sustain power efficiency in congested environments, as demonstrated by the noticeable increase in energy consumption. The ADR only configuration demonstrates a substantial reduction in energy consumption using 15.5 mJ for 300 nodes and 21.0 mJ for 900 nodes, as compared to baseline metrics. This increase stems from the optimization of transmission power through adaptive data rate adjustments. The unresolved energy efficiency issues on large-scale networks, ADR has improved upon but not resolved. The multi-channel condition improves energy efficiency further by mitigating collisions and retransmissions, conserving energy, and saving 13.5 mJ for 300 nodes and 18.0 mJ for 900 nodes. The proposed system demonstrates minimum energy consumption across all node levels when compared with other scenarios. That shows the integration of ADR with multi-channel communication and resource scheduling is useful, given that the system achieves 11.0 mJ at 300 nodes and 15.0 mJ at 900 nodes. Thus, the proposed solution indicates strong suitability for energy-constrained IoT use cases in Smart City applications, as this integrated approach optimizes resource availability, reduces transmission power, and clears blocks. The improved energy efficiency of the proposed system indicates its scalability and longevity, hence proving its potential to support efficient and sustainable Smart City deployments.

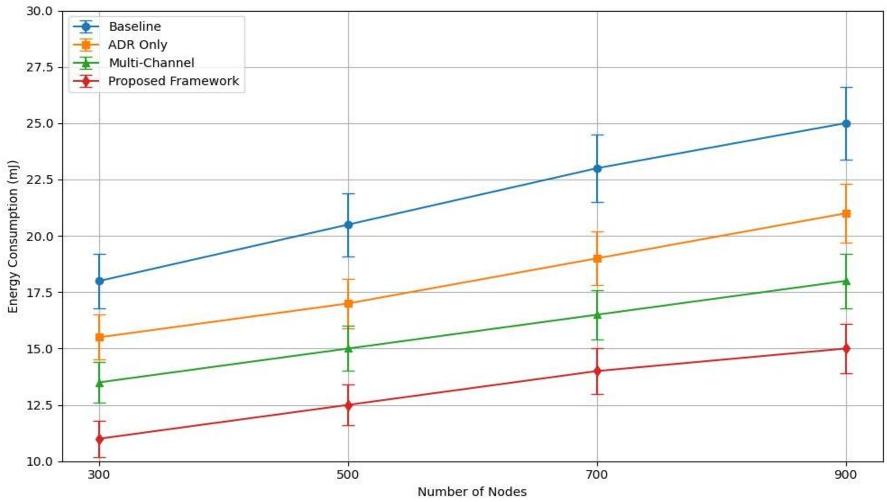

Figure 7. Energy consumption values for the different optimization methods tested when varying the number of nodes.

7. Conclusion

This study advocates configuring the LoRa network architecture for performance improvement in smart megacities. Considering the system's Adaptive Data Rate (ADR) optimization, multi-channel communication, and dynamic resource scheduling, the system approaches scalability, latency, and energy efficiency challenges. The system designed has been compared, and improvements recognized, to traditional LoRa configurations as baseline, ADR-only, and multi-channel scripts, which all demonstrate the system's advantages via comprehensive simulations in OMNeT. The proposed advanced version of the LoRa system demonstrated multiple configurations by the system's higher PDR, low latency, and low energy consumption on different node densities in many configurations, validating the proposed system over the existing versions. The proposed framework achieved the most notable success of maintaining very high PDR compared to ADR-only and baseline configurations in lower PDRs, even in the higher node densities, of 92.3 PDR in 300 nodes and 85.7 PDR in 900 nodes. These findings underline the strength and scalability of the suggested design, which makes it rather appropriate for large-scale Smart City uses. The limitation of this study is the absence of hardware validation. For future work, the proposed framework will be tested on a real LoRaWAN testbed using commercial gateways and end devices.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors express gratitude to the University of Mosul and the Universitat Politècnica de València for their cooperation in conducting this work.

Author contribution

Marwan S. M. Al-Dabbagh: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis. Carlos T. Calafate: Writing – review & editing, Visualization, Validation, Supervision, Investigation. Pietro Manzoni: Writing – original draft, Visualization, Conceptualization.

References

[1] A. Griva, A. D. Boursianis, S. Wan, P. Sarigiannidis, G. Karagiannidis, and S. K. Goudos, "Performance Evaluation of LoRa Networks in an Open Field Cultivation Scenario," *2021 10th Int. Conf. Mod. Circuits Syst. Technol. MOCAST 2021*, pp. 1–5, 2021, https://doi.org/10.1109/MOCAST52088.2021.9493416.

[2] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad, and L. Khoukhi, "Internet of Things (IoT) Technologies for Smart Cities," *IET Networks*, vol. 7, no. 1, pp. 1–13, 2018, https://doi.org/10.1049/iet-net.2017.0163.

[3] Y. Guo, J. Niu, X. Zhou, T. Gu, Y. Li, and D. Fang, "Power-Efficient Transmissions in LoRa Uplink Systems," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 11224–11236, 2024, https://doi.org/10.1109/TVT.2024.3371157.

[4] S. Dasiga, A. A. R. Bhatia, A. Bhirangi, and A. Siddiqua, "LoRa for the last mile connectivity in IoT," *Proc. 2020 9th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2020*, pp. 195–200, 2020, https://doi.org/10.1109/SMART50582.2020.9337114.

[5] H. Wang, H. Chen, J. Lu, B. Zhang, S. Pang, and R. Pan, "Design and Implementation of a LoRa Adaptive Rate Algorithm Based on Power Optimization," *Proc. - 2023 Int. Conf. Networks, Commun. Intell. Comput. NCIC 2023*, pp. 279–285, 2023, https://doi.org/10.1109/NCIC61838.2023.00053.

[6] J. H. Huh, D. Tanjung, D. H. Kim, S. Byeon, and J. D. Kim, "Improvement of Multichannel LoRa Networks Based on Distributed Joint Queueing," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4343–4355, 2022, https://doi.org/10.1109/JIOT.2021.3105660.

[7] M. Saravanan, M. V. Kumar, S. Dhineshkumar, C. Manikandan, P. Prabhakaran, and a. Aparna, "IoT and LoRa based Under Mining Surveillance System for Worker Health and Environment Monitoring," *Int. Conf. Sustain. Commun. Networks Appl. ICSCNA 2023 - Proc.*, no. Icscna, pp. 354–359, 2023, https://doi.org/10.1109/ICSCNA58489.2023.10370308.

[8] M. Alkhayyal and A. M. Mostafa, "Enhancing LoRaWAN Sensor Networks: A Deep Learning Approach for Performance Optimizing and Energy Efficiency," *Comput. Mater. Contin.*, vol. 83, no. 1, pp. 1079–1100, 2025, https://doi.org/10.32604/cmc.2025.061836.

[9] V. Fialho, "A Comprehensive Strategy for Implementing Adaptive Data Rate (ADR) in LoRaWAN Technology: Optimizing IoT Networks," *Int. J. Innov. Technol. Explor. Eng.*, vol. 3075, no. X, pp. 1–6, 2023, https://doi.org/10.35940/ijitee.

[10] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," *IEEE/IFIP Netw. Oper. Manag. Symp. Cogn. Manag. a Cyber World, NOMS 2018*, pp. 1–9, 2018, https://doi.org/10.1109/NOMS.2018.8406255.

[11] A. Ilarizky, A. Kurniawan, E. P. Subagyo, R. Harwahyu, and R. F. Sari, "Performance Analysis of Adaptive Data Rate Scheme at Network-Server Side in LoRaWAN Network," *Proceeding - 2021 2nd Int. Conf. ICT Rural Dev. IC-ICTRuDev 2021*, pp. 1–5, 2021, https://doi.org/10.1109/IC-ICTRuDev50538.2021.9656523.

[12] K. Ksiazek and K. Grochla, "Flexibility Analysis of Adaptive Data Rate Algorithm in LoRa Networks," *2021 Int. Wirel. Commun. Mob. Comput. IWCMC 2021*, pp. 1393–1398, 2021, https://doi.org/10.1109/IWCMC51323.2021.9498664.

[13] J. He, "An Optimized ADR Algorithm for LoRa Network Based on EWMA," *2024 3rd Int. Conf. Artif. Intell. Internet Things Cloud Comput. Technol. AIoTC 2024*, pp. 235–239, 2024, https://doi.org/10.1109/AIoTC63215.2024.10748331.

[14] Y. Liu, P. Tian, C. A. Boano, X. Ma, Q. Yang, and H. Wang, "A Low-Cost and Infrastructure-Less LoRa Wireless Network Testbed for Cognitive Internet of Things," *IEEE Trans. Cogn. Commun. Netw.*, vol. PP, p. 1, 2024, https://doi.org/10.1109/TCCN.2024.3461671.

[15] Y. Yu *et al.*, "Adaptive Multi-Channels Allocation in LoRa Networks," *IEEE Access*, vol. 8, pp. 214177–214189, 2020, https://doi.org/10.1109/ACCESS.2020.3040765.

[16] A. I. Petrariu, A. Lavric, and E. Coca, "LoRaWAN Gateway: Design, Implementation and Testing in Real Environment," *SIITME 2019 - 2019 IEEE 25th Int. Symp. Des. Technol. Electron. Packag. Proc.*, vol. 1, no. October, pp. 49–53, 2019, https://doi.org/10.1109/SIITME47687.2019.8990791.

[17] P. Gkotsiopoulos, D. Zorbas, and C. Douligeris, "Performance determinants in LoRa networks: A literature review," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1721–1758, 2021, https://doi.org/10.1109/COMST.2021.3090409.

[18] R. Hamdi and M. Qaraqe, "Resource Management in Energy Harvesting Powered LoRa Wireless Networks," *IEEE Int. Conf. Commun.*, pp. 1–6, 2021, https://doi.org/10.1109/ICC42927.2021.9500638.

[19] D. T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: Toward an intelligent resource allocation approach for LoRaWAN," *Proc. - IEEE Glob. Commun. Conf. GLOBECOM*, 2019, https://doi.org/10.1109/GLOBECOM38437.2019.9013345.

[20] W. Gao, Z. Zhao, and G. Min, "AdapLoRa: Resource Adaptation for Maximizing Network Lifetime in LoRa networks," *Proc. - Int. Conf. Netw. Protoc. ICNP*, vol. 2020-Octob, 2020, https://doi.org/10.1109/ICNP49622.2020.9259398.

[21] R. O. Andrade and S. G. Yoo, "A comprehensive study of the use of LoRa in the development of smart cities," *Appl. Sci.*, vol. 9, no. 22, 2019, https://doi.org/10.3390/app9224753.

[22] S. Debdas, A. Chakraborty, H. K. Verma, A. Kushwaha, D. V. P. Varma, and A. Karmakar, "Long Range (LoRa) Fire Fighter In Dense Forest," *2022 1st Int. Conf. Comput. Power Commun. ICCPC 2022 - Proc.*, pp. 286–293, 2022, https://doi.org/10.1109/ICCPC55978.2022.10072080.

[23] D. Manoj Kumar, R. Arthi, K. Dhanveer, V. P. Kumar, and S. Mahidhar, "A Renewable Energy Smart Metering System Using LoRa Network," *ESIC 2024 - 4th Int. Conf. Emerg. Syst. Intell. Comput. Proc.*, pp. 626–630, 2024, https://doi.org/10.1109/ESIC60604.2024.10481548.

[24] M. Marchese, A. Moheddine, and F. Patrone, "Towards Increasing the LoRa Network Coverage: A Flying Gateway," *2019 Int. Symp. Adv. Electr. Commun. Technol. ISAECT 2019*, pp. 1–4, 2019, https://doi.org/10.1109/ISAECT47714.2019.9069697.

[25] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of Lora: Long range & low power networks for the internet of things," *Sensors (Switzerland)*, vol. 16, no. 9, pp. 1–18, 2016, https://doi.org/10.3390/s16091466.

[26] H. Bougaddou, N. Cherkaoui, and O. Mouhib, "Long-range, low-power for IoT devices: The LoRa Network a review," *Proc. 2024 1st Ed. Mediterr. Smart Cities Conf. MSCC 2024*, no. Mscc, pp. 1–6, 2024, https://doi.org/10.1109/MSCC62288.2024.10697015.

[27] L. Davoli, E. Pagliari, and G. Ferrari, "Hybrid lora-ieee 802.11s opportunistic mesh networking for flexible uav swarming," *Drones*, vol. 5, no. 2, 2021, https://doi.org/10.3390/drones5020026.

[28] A. Lavric and A. I. Petrariu, "LoRaWAN communication protocol: The new era of IoT," *2018 14th Int. Conf. Dev. Appl. Syst. DAS 2018 - Proc.*, pp. 74–77, 2018, https://doi.org/10.1109/DAAS.2018.8396074.