Efficient Large-Scale IoT Network: Integrating Asynchronous Communication and Huffman Coding in LoRa/PLC Systems

Haoyu Zhang
College of Information Engineering
Shanghai Maritime University
Shanghai, China
0009-0004-2317-4289

Weizhuo Yao
College of Information Engineering
Shanghai Maritime University
Shanghai, China
202310320126@stu.shmtu.edu.cn

Zhengyang Xu
College of Science
Shanghai Maritime University
Shanghai, China
xuzhengyang5043@stu.shmtu.edu.cn

Xianghan Hu
School of Economics and Management
Shanghai Maritime University
Shanghai, China
202210751034@stu.shmtu.edu.cn

2025 International Wireless Communications and Mobile Computing (IWCMC) | 979-8-3315-0887-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/IWCMC65282.2025.11059707

***Abstract*—As Internet of Things (IoT) deployments expand, the development of sustainable and intelligent communication architectures is critical for managing large-scale device integration and high concurrency. This paper presents an innovative approach to enhance IoT network performance in LoRa and Power Line Communication systems through the integration of asynchronous communication, a user behavior-based Huffman coding method, and edge computing. Our multilevel architecture leverages gateways to efficiently monitor and control sensors and actuators, while our dynamic Huffman coding compresses frequently used commands based on user behavior, reducing energy consumption and improving data transmission efficiency. Edge computing facilitates local data preprocessing and real-time monitoring, enabling intelligent decision making and minimizing redundant cloud data transfers. Preliminary simulations demonstrate significant improvements in network efficiency, confirming the efficacy of our integrated approach for large-scale, intelligent, and sustainable IoT deployments.**

***Index Terms*—Asynchronous Communication, Huffman Coding, LoRa, Power Line Communication, Edge Computing**

I. Introduction

The rapid advancement of IoT technology has transformed how energy consumption is monitored and managed. While IoT applications in residential settings have been extensively studied, the unique requirements of industrial and park-scale environments require tailored solutions. These environments are characterized by numerous interconnected devices, elevated energy consumption, and substantial operational complexity, highlighting the critical need for accurate monitoring and efficient communication. Enhanced IoT frameworks designed for these settings have the potential to significantly contribute to global carbon reduction efforts.

Managing IoT networks in industrial and park-scale settings presents numerous challenges, especially because of the thousands of terminal devices with varied functionalities. Traditional IoT solutions often struggle to address the requirements for robustness, scalability, and efficiency at such a scale.

In this paper, we propose a novel framework that integrates LoRa Mesh and PLC technologies, enhanced with edge computing to enable intelligent decision-making and optimize data compression. The framework is built around the following key features.

- **Asynchronous Communication**
  Reduce latency and congestion by allowing devices to function autonomously without relying on rigid synchronization protocols.
- **Data Compression**
  Implements Huffman Coding to learn user data patterns, optimizing LoRa transmission efficiency.
- **Huffman Tree Synchronization**
  Conducts updates during off-peak times, enhancing data throughput during peak periods while maintaining uninterrupted communication.
- **Edge Computing**
  Facilitates local decision-making, reduces data transmission volumes, and enhances overall network efficiency.

This framework is designed to transform IoT deployments within expansive, park-level networks, striking an ideal balance between sustainability, scalability, and intelligence.

Paper Organization. The remainder of this paper is organized as follows. Section II reviews the relevant work in the field, discussing its advantages and disadvantages. Section III presents the proposed approach in detail. Next, Section IV concludes the paper. Finally, Section V discusses potential future work and additional insights.

II. Related Works

IoT connectivity methods can be categorized into wireless and wired networks. On the wireless side, commonly used approaches include NB-IoT, 5G, Wi-Fi, Bluetooth, and LoRa. Meanwhile, wired solutions typically involve Ethernet, RS485, and Power Line Communication (PLC).

Both NB-IoT and 5G operate in the LTE spectrum. However, their subscription fees can be prohibitively expensive. Wi-Fi and Bluetooth often operate in the 2.4 GHz band, which frequently experiences interference from the heavy usage of multiple devices. Additionally, their relatively short range poses a limitation for multi-building or large-scale IoT scenarios.

In contrast, LoRa stands out by operating in license-free ISM sub-GHz bands, thereby avoiding the high costs associated with spectrum licensing. Leveraging Chirp Spread Spectrum (CSS) modulation, it can sustain communication over distances of several kilometers while maintaining low power consumption. [1]

As for wired communication, Ethernet and RS485 require additional cabling, which increases installation costs and complexity, especially in retrofitting scenarios. PLC, on the other hand, exploits existing power lines to transmit data, thereby requiring minimal additional wiring. Hence, for indoor or park settings with extensive power-line infrastructure already in place, PLC emerges as a cost-effective and practical choice. [2]

A. LoRa Communication

The key benefits of LoRa are its low power consumption and extended transmission range. Because it adopts Chirp Spread Spectrum (CSS) modulation, LoRa can decode signals at comparatively low signal-to-noise ratios (SNR) [3]. Under open-air or low-obstruction conditions, coverage can span multiple kilometers.

Despite its long-range and low-power strengths, the limited bandwidth and data rate of LoRa pose significant bottlenecks in dense or large-scale IoT deployments. Moreover, environmental obstacles such as walls, pipes, and metal enclosures can significantly reduce coverage, in some cases, to only a few hundred meters. Indoor studies (e.g., Xu et al. [4], Navarro et al. [5], and Haxhibeqiri et al. [6]) found that LoRa maintains connections at distances of approximately 100 meters in office buildings, parking garages, and warehouses.

Thus, scaling LoRa to support a massive number of devices with higher data demands continues to present significant challenges.

B. Power Line Communication

Power Line Communication (PLC) piggybacks data signals onto existing electrical wiring, removing the need for additional network cables. This characteristic makes PLC highly advantageous for scenarios where power lines are pervasive, such as large buildings or industrial campuses. In favorable conditions, PLC can achieve higher data rates, providing reliable indoor connections.

However, as a wired technology, PLC lacks mobility and is generally restricted to a building's or floor's electrical circuit. Although it performs well within a single power distribution network, crossing from one building (or floor) to another frequently necessitates additional bridges or repeaters. Meanwhile, outdoor power lines may face reliability issues due to potential damage caused by environmental or human factors. Therefore, while PLC is advantageous for indoor environments, it may not be practical for covering an entire park with multiple buildings or accommodating mobile devices. [7]

C. hybrid LoRa and Power Line Communication

To mitigate instability of LoRa in complex indoor environments and the building-bound limitations of PLC, some researchers have explored hybrid networks. In 2017, Alireza Ghiasimonfared et al. [8] proposed a dual-hop PLC/WLC network that integrates LoRa and G3-PLC technologies. In their approach, the PLC link facilitates communication through obstacles such as walls, while the wireless link aggregates multiple PLC sub-networks across physically isolated areas. This design provides heterogeneous connectivity but still encounters several critical challenges.

- **Mismatched Data Rates**
  PLC networks can reach rates of several Mbps, whereas LoRa tops out at hundreds of kbps. In a straightforward conversion mode at the gateway—simply relaying from PLC to LoRa—there is a throughput bottleneck due to the disparity in bandwidth.
- **Underutilized Gateways**
  In many existing solutions, the gateway simply forwards packets between the two technologies without performing additional processing or data compression. Consequently, the higher bandwidth capacity of PLC is underutilized, resulting in inefficiencies during periods of heavy traffic. [7]
- **Limited Scalability and Efficiency**
  The typical server–gateway–end device synchronization model can prove ineffective for large-scale network deployments with high concurrency requirements. Relying on centralized data aggregation increases latency and congestion, particularly when managing numerous endpoints.

Current fusion schemes often stick to a direct or centrally orchestrated approach, overlooking the potential of distributing intelligence at the network edges.

D. Proposed method

Our proposed solution addresses these limitations by replacing the traditional synchronous "server–gateway–end device" architecture. Instead, we adopt an asynchronous architecture that delegates edge services and localized control to gateways, significantly offloading tasks that would

TABLE I
COMPARISON BETWEEN LORA AND PLC FOR IOT APPLICATIONS

Attribute	LoRa	PLC
Transmission Range	Up to 15 km (rural), 2–5 km (urban)	In the range of existing power line
Data Rate	0.3 kbps to 50 kbps	Up to several Mbps
Scalability	Highly scalable	Moderately scalable
Interference Resistance	High (robust against RF interference)	Susceptible to electrical noise
Deployment Complexity	Simple (wireless setup)	Moderate (integration with power lines)
Latency	High	Lower
Mobility Support	Supports node mobility within range	Limited mobility

otherwise require a full round-trip to the server. We also employ Huffman coding for LoRa transmissions, leveraging the predictable patterns of downlink commands in standard IoT scenarios. This approach not only expedites batch command delivery during off-peak hours but also reduces latency and alleviates congestion. At the same time, the system delegates decision-making logic to the gateway, which only reports the final execution result back to the server. In doing so, we preserve PLC's high bandwidth for bulk data exchange within each local domain, resulting in a scalable, efficient, and robust hybrid network designed to meet the demands of park-scale IoT deployments.

III. METHODS

This section provides a detailed explanation of the proposed asynchronous communication architecture, a Huffman coding method based on user behavior and edge computing functionalities. By deploying gateways and edge computing nodes within LoRa or PLC networks, the system enables efficient monitoring and control of sensors and actuators.

A. System Architecture

To support large-scale device integration, handle high concurrency, and ensure efficient utilization of network resources, the system employs a multilevel architecture, as illustrated in Figure 1 and comprising the following layers:

- **Sensor Nodes**
  End devices deployed on-site, such as temperature sensors and motor controllers. Responsible for data collection and command execution, supporting PLC communication protocols.
- **Gateway Layer**
  Gateways distribute server-issued commands to multiple devices within local PLC segments while simultaneously collecting and processing feedback locally. They support asynchronous communication and edge computing by synchronizing Huffman coding tables and decision strategies.
- **Edge Computing Nodes**
  Physically integrated with gateways, these nodes handle data preprocessing, real-time monitoring, and local decision-making. They also dynamically update Huffman coding trees based on historical data and disseminate updates during low-load periods.

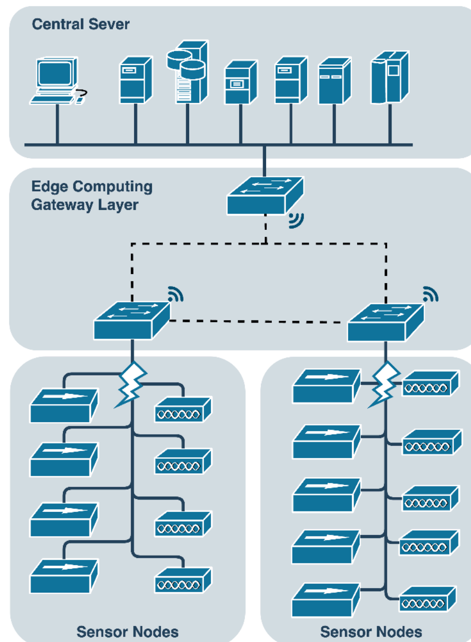

Fig. 1. System Architecture

- **Central Server**
  The sever can manage global monitoring, data aggregation, and advanced analytics, issuing command sets to various gateways in parallel. It is capable of parallel command dispatch without waiting for prior command execution, enhancing scalability and responsiveness.

This multilevel structure facilitates asynchronous communication across network segments, effectively reducing bandwidth usage and latency. Moreover, executing partial computations and decision-making at edge nodes improves overall system efficiency and reliability.

B. Asynchronous vs. Synchronous Communication

Traditional server–gateway–terminal architectures primarily depend on synchronous communication, wherein commands are executed in a sequential manner. This approach requires waiting for the execution result of each device before subsequent commands can be issued by the server. Such a model often leads to inefficiencies in large-scale

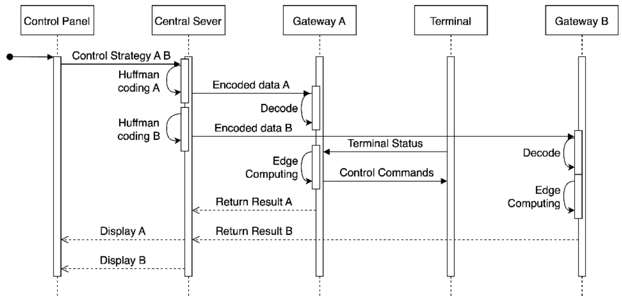

Fig. 2. Communications Timing

deployments with high concurrency, resulting in bottlenecks and heightened latency.

To overcome these challenges, the proposed system employs an asynchronous communication scheme that illustrated in Figure 2.

- **Parallel Command Dispatching**
  The server can dispatch commands to multiple gateways simultaneously, eliminating the need to wait for execution results.
- **Local Gateway Scheduling**
  Gateways autonomously operate devices within their local PLC segment, aggregating execution results without server intervention.
- **Reduced Network Congestion**
  By dispatching commands in parallel, network congestion is minimized, and command execution efficiency is significantly improved.

This asynchronous architecture decreases waiting times and congestion risks in LoRa networks, enabling faster execution and higher system throughput.

C. Huffman Coding Based on User Behavior

The system employs a user behavior-based Huffman coding method to optimize data transmission in LoRa networks by compressing frequently used operation commands or data.

Recently issued commands and corresponding execution data are recorded at edge nodes or gateways for statistical analysis. Some high-frequency commands (e.g., "all on," "all off," "status query") are identified using cumulative frequency statistics.

Real-time data statistics are used to construct Huffman trees, assigning shorter codes to higher-frequency symbols. The update of huffman coding tables are synchronized during network idle periods, ensuring minimal disruption.

Compressed data reduces the bandwidth required for LoRa transmissions, increasing network throughput and lowering power consumption. For common commands like "all on" or "all off," Huffman coding can reduce data length to 1–2 bits, significantly improving transmission efficiency.

Assume a system with a total of N possible "command combinations." Among these, 20% of the symbols collectively occupy 80% of the total probability, whereas the remaining 80% of the symbols occupy the remaining 20% of the probability. For simplicity of analysis, a uniform distribution is often assumed for these two groups. Specifically, for the "hot" (popular) 20%:

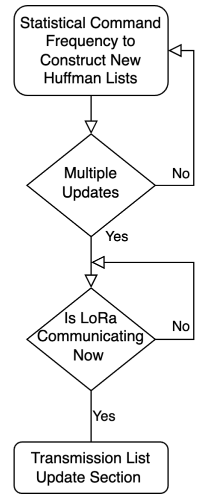

Fig. 3. Huffman List Update Process

$$p_{\text{hot}} = \frac{0.8}{0.2N} = \frac{4}{N}, \tag{1}$$

and for the "cold" (less popular) 80%:

$$p_{\text{cold}} = \frac{0.2}{0.8N} = \frac{0.25}{N}. \tag{2}$$

Under the probability distribution above, the Shannon entropy is

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i. \tag{3}$$

Then, breaking down the entropy into the contributions of the hot and cold sets. The hot set contribution is

$$H_{\text{hot}} = 0.2N \times \frac{4}{N} \times \left[-\log_2\left(\tfrac{4}{N}\right)\right] = 0.8 \times \left[-\log_2\left(\tfrac{4}{N}\right)\right]. \tag{4}$$

Simplifying,

$$H_{\text{hot}} = 0.8 \log_2(N) - 1.6. \tag{5}$$

The cold set contribution is

$$H_{\text{cold}} = 0.8N \times \frac{0.25}{N} \times \left[-\log_2\left(\tfrac{0.25}{N}\right)\right] = 0.4 + 0.2 \log_2(N). \tag{6}$$

Therefore, the total entropy will be

$$H = H_{\text{hot}} + H_{\text{cold}} = \log_2(N) - 1.2. \tag{7}$$

Given this distribution, Huffman coding can theoretically achieve an average code length close to the entropy H. Thus,

$$L_{\text{Huffman}} \approx H = \log_2(N) - 1.2. \tag{8}$$

In contrast, a fixed-length approach assigns the same code length to all symbols regardless of their frequencies. If there are N total symbols, the required codeword length is:

$$L_{\text{fixed}} = \lceil \log_2(N) \rceil. \tag{9}$$

Hence, Huffman coding can reduce the average code length by approximately 1.2 bits compared to fixed-length coding. For a single gateway, which typically connects to fewer than one hundred devices, this translates to an average code length reduction of more than 29%.

Specifically:

$$\frac{L_{\text{fixed}}}{L_{\text{Huffman}}} \approx \frac{\log_2(N)}{\log_2(N) - 1.2}. \tag{10}$$

The coding method reduces communication duration and data volume while maintaining data integrity, significantly enhancing system performance in high-concurrency scenarios.

D. Edge Computing Functions

Traditional IoT systems centralize monitoring and decision-making on servers, requiring all data to be uploaded to the cloud. This approach consumes bandwidth and introduces latency, especially for large-scale deployments. The proposed system incorporates edge computing functionalities to mitigate these limitations.

Gateways preprocess and monitor sensor data locally based on server-configured strategies. Normal data is reported as a simple "normal" status, while only abnormal data is compressed and uploaded, reducing network traffic.

What's more, alarms or automated operations are handled locally at edge nodes, enabling rapid responses without cloud involvement. LoRa segments primarily transmit status updates or execution results, drastically reducing uplink data volume.

By leveraging edge computing, the system minimizes resource overhead and latency while enhancing responsiveness and scalability, making it well-suited for park-level IoT deployments.

E. Update Overhead Analysis

Although adaptive Huffman coding can shrink the average LoRa payload by 29%, the code-book itself must be rebuilt and synchronised periodically. To validate the maintenance cost of the adaptive Huffman code table in real IoT services, we use the MQTT-IoT-IDS 2020 public dataset [9] and extract its Layer 7 payload for experiments. The dataset contains 1,056k MQTT messages, and the collection time is about 2 h, which is sufficient to cover multiple peak and idle cycles.

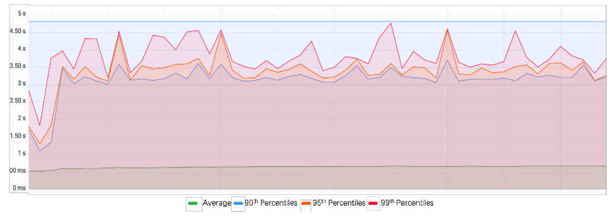

Fig. 4. Responding Time

Gateways monitor the symbol-frequency histogram in a sliding window of W = 1000 down-link commands. A full tree reconstruction is executed only when the Kullback–Leibler divergence between the current and the stored distribution exceeds 0.5; otherwise the previous tree is reused. In this trace comprising commands this criterion fired 5 times, i.e. 2.5 times per hour on average, which is well below the "once-per-minute" upper bound recommended for memory-constrained edge nodes in [10].

Across the whole trace the mean payload shrank from 30.7 B to 7.58 B (compressed/original = 0.247), i.e. a 75.3% traffic reduction. These measurements confirm that, when governed by a divergence-based trigger, Huffman-tree maintenance is cheap.

IV. Results

To evaluate the proposed system under realistic conditions, we ran experiments in a simulated park-like setting equipped with typical electrical appliances—fans, power outlets, and lamps. The primary goals were to measure the system's reliability and responsiveness.

Across 1,056 sliding-window samples, the lowest compression ratios (compressed size ÷ original size; lower is better) were:

- Arithmetic coding: 0.218
- Proposed adaptive Huffman: 0.247
- LZW: 0.264

While arithmetic coding is theoretically optimal, its on-device implementation consumed more CPU cycles per symbol and required an additional RAM buffer for probability tables—resources unavailable on the gateways targeted in this work. The adaptive Huffman scheme therefore offers the best compression-versus-resource trade-off, shrinking the Layer-7 payload to 24.7% of its original size while remaining MCU-friendly.

The system achieved a 98.8% success rate across all test scenarios, with an average response time (ART) of 670ms. Figure 4 presents the response-time distribution at the 90th, 95th, and 99th percentiles.

These findings indicate that the proposed framework effectively supports multiple simultaneous requests while maintaining low latency and high reliability. Such performance makes the system particularly suitable for campus-wide IoT deployments, where scalability and responsiveness are critical.

V. Conclusion and Discussion

This paper has introduced an asynchronous LoRa–PLC fusion architecture that couples edge computing with adaptive Huffman coding to overcome the bandwidth and latency bottlenecks that plague large-scale, park-level IoT deployments. By migrating data preprocessing and partial control logic to the gateway layer, the scheme relieves the narrow LoRa uplink, while PLC back-haul capacity is exploited for bulk traffic. Simulations on a two-building test-bed and trace-driven evaluations with the MQTT-IoT-IDS 2020 dataset confirmed that the proposed system sustains a 98.8% command-success rate, cuts Layer-7 payloads to 24.7% of their original size, and adds only five code-book updates in a million-packet, two-hour session.

Limitations. The current prototype targets low- and medium-power circuits. Heavy industrial feeders inject broadband impulsive noise and harmonic distortion that can swamp PLC carriers. In addition, although adaptive Huffman updates are lightweight, they still incur short CPU bursts that must be budgeted on ultra-low-power gateways.

Future work. We outline three immediate research directions:

1) *Field deployment at scale.* A six-month pilot will be rolled out across a 12-building university campus. The study will compare indoor and outdoor cable segments, quantify long-term compression gains, and capture interference patterns under real load diversity.
2) *Robustness on high-power circuits.* To tame the spectral pollution of dynamic loads, we will integrate adaptive OFDM sub-carrier blanking and notch filtering, and evaluate low-density parity-check (LDPC) and polar codes for forward-error protection.
3) *Algorithmic diversification.* Beyond Huffman coding, we will investigate context-mixing arithmetic coders and lightweight neural compressors whose hyper-parameters are tuned online via traffic-surge prediction models. A multi-armed bandit controller will select the least-energy, highest-throughput codec in real time, subject to MCU memory caps.

In conclusion, the proposed asynchronous, edge-centric and compression-enhanced architecture offers a pragmatic route to resilient, scalable and sustainable IoT networking. The planned field trials and interference-aware extensions are expected to push its applicability from park-level campuses to electrically harsher, heavy-industry scenarios, thereby broadening the impact of this work on next-generation smart infrastructures.

References

[1] Li, Chenning, and Zhichao Cao. ”Lora networking techniques for large-scale and long-term iot: A down-to-top survey.” ACM Computing Surveys (CSUR) 55.3 (2022): 1-36.

[2] Coutinho, Yan F., et al. ”Two Decades of Research Progress in Resource Allocation for PLC Systems: From Core Concepts to Frontiers.” IEEE Communications Surveys & Tutorials (2024).

[3] Liu, Li, et al. ”Deeplora: Learning accurate path loss model for long distance links in lpwan.” IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE, 2021.

[4] Xu, Weitao, et al. ”Measurement, characterization, and modeling of lora technology in multifloor buildings.” IEEE Internet of Things Journal 7.1 (2019): 298-310.

[5] Navarro-Ortiz, Jorge, et al. ”Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things.” IEEE Communications Magazine 56.2 (2018): 60-67.

[6] Haxhibeqiri, Jetmir, et al. ”LoRa indoor coverage and performance in an industrial environment: Case study.” 2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA). IEEE, 2017.

[7] Ribeiro, Moisés V., et al. ”Seamless connectivity: The power of integrating power line and wireless communications.” IEEE Communications Surveys & Tutorials 26.1 (2023): 1-40.

[8] Ghiasimonfared, Alireza, et al. ”Development of a hybrid LoRa/G3-PLC IoT sensing network: An application oriented approach.” 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm). IEEE, 2017.

[9] H. Hindy, C. Tachtatzis, R. Atkinson, et al., “MQTT-IoT-IDS 2020: MQTT Internet of Things Intrusion Detection Dataset,” IEEE Dataport, doi: 10.21227/bhxy-ep04, 2020.

[10] A.S. Nasif et al., “Huffman Deep Compression of Edge Node Data for Reducing IoT Network Traffic,” IEEE Access, vol.12, pp.122988-122997, Sep.2024.