ECRLoRa: LoRa Packet Recovery under Low SNR via Edge–Cloud Collaboration

LUOYU MEI, Southeast University, China and City University of Hong Kong, China
ZHIMENG YIN, City University of Hong Kong, China
SHUAI WANG, Southeast University, China
XIAOLEI ZHOU, National University of Defense Technology, China
TAIWEI LING and TIAN HE, Southeast University, China

Low-Power Wide-Area Networks (LPWANs), extensively utilized for connecting billions of IoT devices, encounter wireless interference challenges in unlicensed frequency bands. Cutting-edge research suggests employing Received Signal Strength Indication (RSSI) sequences for error detection to mitigate interference-related issues. Nevertheless, the effectiveness of this method significantly declines under low signal-to-noise ratios (SNRs). Additionally, long-range communication often results in low SNR received signals, sometimes even below the noise floor. Targeting this fundamental issue, this article proposes the LPWAN packet technique, broadly applicable across diverse scenarios through edge–cloud collaboration. On the edge side, we propose an innovative architecture that fully exploits spatial distribution and interference independence in the field. Rather than utilizing resource-intensive RSSI-based error detection, we leverage a lightweight coding scheme for error detection at the Long Range (LoRa) edge, forwarding correct frames to the cloud. On the cloud side, packet recovery is achieved utilizing group-weighted voting. We design and implement ECRLoRa with commercially available devices (SemTech's SX1278 and SX1302 LoRa chipsets) and assess its performance in low SNR environments. Our thorough evaluation demonstrates that our approach attains a Packet Recovery Ratio of 96% with low SNR (i.e., below −10 dB), resulting in 1.8× throughput, 7.5× faster recovery time, and 4.92× average accuracy compared to state-of-the-art cloud-optimized application layer solutions.

CCS Concepts: • **Networks** → **Wireless access points, base stations and infrastructure;**

Additional Key Words and Phrases: LPWANs, interference mitigation, edge–cloud collaboration, packet recovery, low signal-to-noise ratio (SNR), coding

This article was presented in part at the 16th International Conference on Wireless Algorithms, Systems, and Applications (WASA21) [21]. https://doi.org/10.1007/978-3-030-85928-2_39.

This work was supported in part by Science and Technology Innovation 2030 - Major Project 2021ZD0114202, National Natural Science Foundation of China under Grant No. 62272098.

Authors' addresses: L. Mei, Southeast University, China and City University of Hong Kong, Hong Kong; e-mail: lymei@seu.edu.cn; Z. Yin (Corresponding author), City University of Hong Kong, Hong Kong and City University of Hong Kong Shenzhen Research Institute, China; e-mail: zhimeyin@cityu.edu.hk; S. Wang (corresponding author), T. Ling, and T. He, Southeast University, Nanjing, China; e-mails: shuaiwang@seu.edu.cn, 213170098@seu.edu.cn, tianhe@seu.edu.cn; X. Zhou, National University of Defense Technology, Nanjing, China; e-mail: zhouxiaolei@nudt.edu.cn.

1550-4859/2024/01-ART40 $15.00
https://doi.org/10.1145/3604936

ACM Reference format:
Luoyu Mei, Zhimeng Yin, Shuai Wang, Xiaolei Zhou, Taiwei Ling, and Tian He. 2024. ECRLoRa: LoRa Packet Recovery under Low SNR via Edge–Cloud Collaboration. *ACM Trans. Sensor Netw.* 20, 2, Article 40 (January 2024), 25 pages.
https://doi.org/10.1145/3604936

1 INTRODUCTION

Low-Power Wide-Area Networks (LPWANs) are gaining increasing attention in both industry and academia. Due to their long-range coverage, low energy consumption, and cost-effective deployment, **Long Range (LoRa)** has emerged as one of the leading LPWAN technologies in unlicensed sub-1-GHz/2.4-GHz bands [22]. A multitude of wireless devices utilizing various communication protocols, such as LoRa [16, 26], Sigfox [27], Zigbee [9], Bluetooth [12, 13, 17], and Wi-Fi [18, 37, 38, 40], are densely deployed under the coverage of LoRa edge (e.g., LoRa base station). These wireless devices share the same unlicensed spectrum with LoRa, causing interference in LoRa communications. This issue becomes more severe in large-scale LoRa deployments. Moreover, low **signal-to-noise ratios (SNR)** deployments are common in LoRaWAN, making it crucial to enhance the robustness of LoRaWAN under low SNR conditions [15, 29, 36].

Traditional methods for addressing interference issues in LPWANs involve re-transmitting corrupted packets. These approaches, though easy to deploy, face challenges due to increased energy consumption, protocol complexity, and channel congestion. Additionally, some studies propose redesigning the PHY and MAC layers [5, 6, 10, 32, 34] for interference mitigation. While these methods significantly improve the throughput and reliability of LPWANs, they necessitate additional hardware or firmware modifications, making them incompatible with existing LPWAN systems.

The emergence of cloud computing has led to the design of many modern IoT systems, connect to the cloud [4, 7, 33], such as LoRaWAN. The sender, edge, and cloud network architecture provide new opportunities to optimize interference mitigation utilizing cloud computing power. Recent approaches demonstrate that cloud-optimized application layer interference mitigation, which involves offloading information and recovering data packets in the cloud, offers numerous benefits. For example, DaRe [20] combines convolution and fountain codes and utilizes redundant data in payloads from other receivers for packet recovery. OPR [1] reconstructs corrupted packets by transmitting the packets and their **Received Signal Strength Indication (RSSI)** samples to the cloud, where the cloud iterates through alternative fragments matched in the error-detection fields. These approaches significantly reduce the failure rate of LoRa packets when a cloud optimizes packet processing upon receiving the RSSI sequence. However, they result in excessive RSSI transmissions or computational overhead, considerably limiting their practical feasibility.

This article aims to develop a fast, cost-effective packet recovery approach for mitigating severe interference in **Long Range (LoRa)** systems. To accomplish this, we investigate the potential of edge–cloud collaboration and introduce a novel edge–cloud collaborative packet recovery approach for LoRa, termed ECRLoRa. In the ECRLoRa architecture, LoRa base stations possess computing and networking capabilities, functioning as edge devices. The LoRa senders, edge devices, and the cloud naturally form a typical edge–cloud collaboration scenario. On the one hand, commercial LoRa edge devices have signal perception advantages, enabling error position recognition. On the other hand, the cloud connects to numerous LoRa edge devices, enhancing packet recovery by leveraging global information. To achieve the proposed edge–cloud collaborative packet recovery, ECRLoRa faces several challenges: (i) For the LoRa edge, how do we rapidly detect error segments under low SNR and report necessary frames timely to support packet recovery in the cloud? (ii) For the cloud, how do we recover the correct packet with disjoint interfered frames from multiple LoRa edges within the time limitation of the LoRa communication ACK threshold?

To conquer those challenges, we design a lightweight error detection method for quick error position detection under low SNR conditions at the ECRLoRa edge. We employ the LoRa edge for error detection, thereby avoiding the transmission of the RSSI sequence. Furthermore, we create a group-weighted voting algorithm for packet recovery in the cloud, which effectively gathers correct frames from multiple LoRa edges. In summary, the contributions of this work include the following:

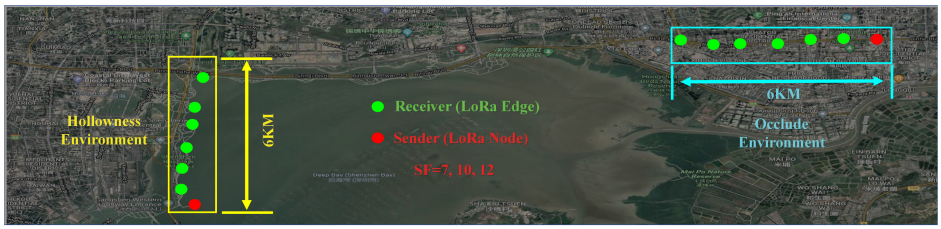
- We design ECRLoRa, a fast edge–cloud packet recovery technique that identifies corruption at the edges and recovers packets in the cloud. ECRLoRa features an edge–cloud collaborative design without transmitting the RSSI sequence of the packet, making it deployable in existing LoRa infrastructures with widespread applicability.
- To expedite the packet recovery process, we address several unique challenges, such as (i) optimized corruption indication utilizing error detection codes, (ii) efficient data transmission, (iii) collaborative packet recovery, (iv) parallel symbol computation, and (v) reliability under low SNR conditions. These techniques offer general guidance for extending the range of edge–cloud packet recovery designs.
- We implement and evaluate ECRLoRa on commercial devices (SemTech's SX1278 LoRa sender [24] and SX1302 LoRa gateway [24]). Our comprehensive evaluation shows that ECRLoRa achieves fast, robust, SNR-tolerant packet recovery across a full range of wireless configuration settings, including hollow, occluded, long-distance, and noisy environments. In these settings, the **packet recovery ratio (PRR)** reaches 96% with three edges. The throughput and recovery speed surpass existing cloud-optimized application layer approaches [1, 20, 21].

The remainder of the article is structured as follows. Section 2 elucidates the motivation through empirical studies. Section 3 provides an overview of ECRLoRa. Sections 4 and 5 describe the crucial components of our design, namely error detection and packet recovery, respectively. Section 6 assesses ECRLoRa's performance utilizing extensive test-bed implementations in various scenarios. Section 7 summarizes related work. Section 8 concludes this article.

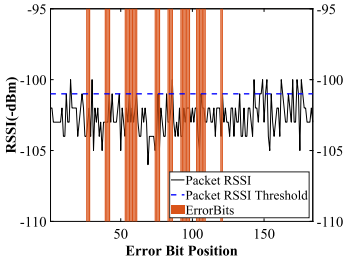
2 MOTIVATION

The ability to mitigate interference, particularly under severe conditions, has become a crucial factor limiting the advancement of LoRa technology. In this section, we first examine the constraints of cloud-optimized approaches. Subsequently, we demonstrate the benefits of adopting an edge–cloud collaborative strategy for packet recovery.

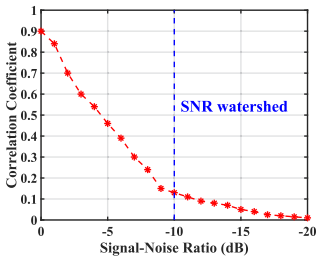
Limitation of Cloud-optimized Application Layer Approaches: Recent interference mitigation strategies concentrate on recovering corrupted information from a vast collection of packets utilizing cloud-optimized methods [1, 20]. Cloud-optimized application layer approaches employ redundancy coding at the application layer and leverage cloud computing resources for packet recovery. State-of-the-art cloud-optimized application layer methods include DaRe [20] and OPR [1]. DaRe integrates convolution and fountain codes to explore spatial and temporal information in LoRa communication channels. However, these redundancy coding techniques are limited to specific LoRa edges and lack full collaboration for distributed error detection. In long-range communication, multiple edges simultaneously receive valuable information. Exploiting this information allows LoRa to enhance data accuracy while reducing coding redundancy, leading to improved throughput and reliability. The LoRa sender, edge, and cloud architecture also offer opportunities for collaborative information processing from multiple edges in the cloud. Conversely, OPR extracts the RSSI samples of corrupted packets from various edges and recovers the corrupted packet utilizing these samples on the cloud side. The RSSI length reaches nearly



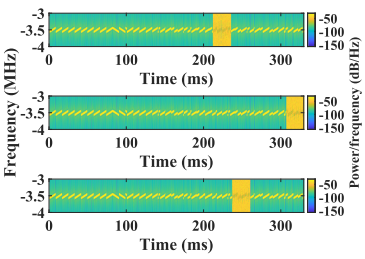
(a) Experimental Settings.



(b) RSSI Sampling.



(c) Correlation Coefficient.



(d) Received I/Q Sequence.

Fig. 1. (a) The environmental settings. (b) The RSSI sample of a LoRa packet that is not strongly correlated with Error Bits position. (c) The Correlation Coefficient of RSSI sequence and Error Bits Position. (d) The in Phase/in Quadrature (I/Q) of three LoRa packets with collision captured with a software-defined Radio.

6,000 bits for a 100-bit payload when the spreading factor is 12. Consequently, the main drawback of cloud-optimized application layer approaches is the high communication cost incurred while transmitting massive RSSI samples to the cloud.

Moreover, the performance of cloud-optimized application layer approaches significantly declines under low SNR conditions. To validate this, we carry out extensive real-world experiments, with typical experimental settings displayed in Figure 1(a). We observe that *RSSI values do not exhibit correlation with erroneous bit positions in low SNR scenarios.* As depicted in Figure 1(b), the orange bar represents actual error bits, while the black line illustrates the unrelated RSSI values. This phenomenon considerably impairs error detection performance or even renders it incapable of identifying error bits. The statistical results concerning the correlation between RSSI and errors are presented in Figure 1(c). Here, the X axis represents SNR, and the Y axis corresponds to the correlation coefficient of RSSI and errors. The figure indicates that the RSSI sequence is nearly uncorrelated with error bits under low SNR conditions. Consequently, existing cloud-optimized application layer approaches that rely on RSSI to pinpoint corruption have limitations in low SNR environments.

Advantages of Edge–Cloud Collaborative Packet Recovery: LoRa exhibits an extensive transmission range owing to its unique modulation technique. A single LoRa sender's packet is received by multiple LoRa edges situated in various locations. These edges experience distinct interference conditions, resulting in spatially disjoint error bit positions. Figure 1(d) demonstrates the lack of correlation between error bit positions in packets received by different edges. Consequently, it becomes feasible to recover a packet in the cloud utilizing multiple parallel corrupted packets.

Moreover, LoRaWAN's sender (LoRa node), receiver (LoRa edge), and cloud architecture enable error detection at the edges without relying on imprecise RSSI values, thus eliminating the need to transmit large RSSI sequences to the cloud. This leads to a lower data transmission volume for edge–cloud collaborative designs. ECRLoRa leverages the opportunities presented by disjoint

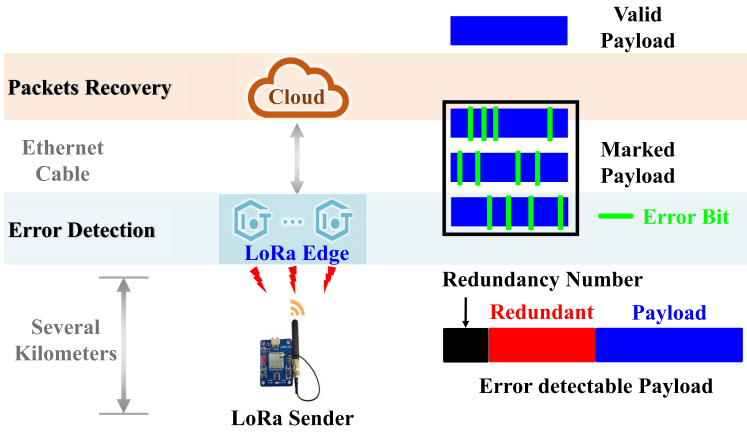


Fig. 2. Architecture of ECRLoRa.

interference to perform error detection across multiple LoRa edges while utilizing the cloud's global management capabilities for packet recovery. Sharing valuable information among multiple LoRa edges through the cloud capitalizes on the benefits of disjoint interference.

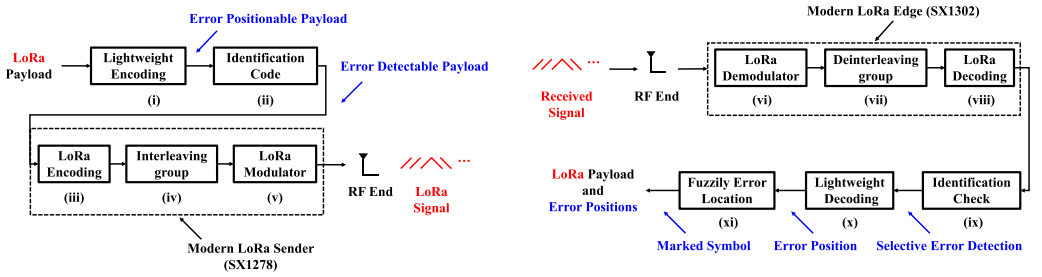
The edge–cloud collaborative architecture addresses the limitations of cloud-optimized application layer approaches by enabling error detection at the LoRa edge and performing packet recovery in the cloud. In comparison to other methods like adjusting the **spreading factor (SF)**, reducing link distance, or changing communication channels, the edge–cloud collaborative architecture offers enhanced robustness by incorporating error detection at the LoRa edge and facilitating packet recovery within the cloud. This approach delivers superior adaptability and efficiency in mitigating interference and restoring data packets, especially in noisy environments with low SNR conditions.

3 OVERVIEW OF ECRLORA

This section provides an overview of ECRLoRa, which consists of two primary components: *error detection* and *packet recovery*. As illustrated in Figure 2, the error detection module operates at the edge, while the packet recovery module functions in the cloud. ECRLoRa effectively harnesses both the signal awareness capabilities of each LoRa edge and the global management ability of the cloud.

Error Detection: ECRLoRa features a lightweight and flexible error detection module that functions under various interference conditions. To accomplish this, we examine LoRa's error conditions in the presence of interference and meticulously design the redundancy method based on the observed interference. The redundancy's hash value is computed and added as a prefix to the payload. Subsequently, the payload containing redundancy is input into the LoRa encoding model, where the payload undergoes encoding and transmission utilizing commercial LoRa devices, as depicted in the lower part of Figure 2. Consequently, commercial LoRa edges are able to detect the payload's error conditions (utilizing the prefix) and position (employing redundancy) for fast error detection, as demonstrated in the middle section of Figure 2. Notably, ECRLoRa's error detection design does modify alter the hardware architecture of the LoRa encoder and modulator. As a result, the redundancy distribution and hash algorithm are compatible with multiple hash functions and redundancy constructions, enabling flexible adjustments to the encoding process. We elaborate on the specifics of error detection in Section 4.

Packet Recovery: Another critical component of ECRLoRa is packet recovery. Due to the geographic independence of edges, the errors in packets received by multiple LoRa edges differ. As



(a) The Workflow of ECRLoRa at Senders.

(b) The Workflow of ECRLoRa at Edges.

Fig. 3. Workflows of ECRLoRa.

shown in the middle part of Figure 2, when a LoRa edge identifies a corrupted packet and pinpoints the exact error positions, it transmits the marked payload to the cloud. If the packet is error-free, then a LoRa edge proceeds with the transmission. On the cloud side, it collects multiple frames consisting of both correct and corrupted payloads. It then leverages the recovery component of these frames to determine the accurate values and provides feedback to the corresponding edges, as illustrated in the upper part of Figure 2. The cloud employs group-weighted voting to recover corrupted transmissions, as elaborated in Section 5.

Feasibility with Low SNR: Conventionally, cloud-optimized application layer packet recovery employs RSSI to indicate error positions [1]. Utilizing RSSI for error detection offers certain advantages, such as the ease of obtaining the RSSI sequence for a LoRa packet, since LoRa devices feature an addressable register to record it. However, when environmental noise surpasses and obscures the received signal at a specific level, the received signal strength indicator becomes inadequate for distinguishing actual signal interference. Direct error detection utilizing RSSI is infeasible under low SNR conditions. We observe that the coding/decoding and modulation/demodulation functions of LoRa devices are vulnerable under low SNR. Consequently, we design an error detection module that employs lightweight coding instead of RSSI sequences. This approach overcomes the limitations of cloud-optimized application layer SoA under low SNR.

4 ERROR DETECTION

In this section, we introduce the error detection component of ECRLoRa. We first describe the workflows of error detection at both the sender and edge sides. As illustrated in Figure 3(a), at the sender side, (i) the payload is initially encoded utilizing a lightweight coding module, generating a positionable error payload. (ii) Next, we process the encoded payload to a certain degree to obtain its identification code. This identification code allows for the fast detection of errors in the received payload. Following this, we prepend the identification code and combine it with the encoded payload to create a new payload. This new payload is then subjected to encoding, interleaving grouping, and modulation in steps (iii), (iv), and (v), respectively. Finally, the packet is transmitted by the RF end.

As shown in Figure 3(b), at the edge side, after receiving the signal from the transmitter, the LoRa edges process demodulation, deinterleaving, and decoding with steps (vi), (vii), and (viii). The received I/Q signal is subsequently converted into an error-detectable payload. This payload is the new encoded payload from the sender, albeit with errors. (ix) Next, the LoRa edge verifies if the recorded identification value matches the payload's identification value for selective error detection. (x) The payload undergoes processing by the lightweight decoding module, which decodes the original payload. (xi) We identify the common characteristics of interference-induced errors

and develop a fuzzy error detection method utilizing these characteristics, enhancing error detection accuracy and minimizing overall time overhead. Finally, error bits are pinpointed through the fuzzy error position module.

4.1 ECRLoRa at LoRa Sender

This subsection introduces the design of ECRLoRa at the LoRa sender. In practical scenarios, unsuitable encoding schemes do not enhance performance but instead result in throughput degradation due to additional redundant encoding bits caused by dynamic interference changes. To circumvent this issue, we initially analyze reception performance and error under interference to determine an appropriate coding design. Subsequently, we introduce the ECRLoRa coding method, which dynamically adjusts the checking code settings in response to varying interference levels. Furthermore, we provide an example to facilitate a more comprehensive understanding of the ECRLoRa code's functionality.

Coding Redundancy vs. Symbol Error Ratio. This subsection presents the model employed to determine the coding design. ECRLoRa incorporates dynamic coding for varying interference levels, as LoRa devices are susceptible to environmental noise. We introduce Lemma 4.1 to analyze the **symbol error ratio (SER)** of LoRa. The symbol error ratio offers a theoretical foundation for ECRLoRa's dynamic coding approach.

Lemma 4.1. *Given LoRa spreading factor SF and the noise power spectral density σ^2. For transmitting symbol "k," the SER of LoRa is as follows:*

$$P(\widehat{k} \neq k|k) = \int_0^{\infty} \left[1 - \left[1 - exp\left[-\frac{\mu^2}{2\sigma^2}\right]\right]^{2^{SF}-1}\right] f_{\mu_k}(\mu)d\mu. \tag{1}$$

Here "k" is the transmitted symbol, μ follows Rician distribution, SF is the spreading factor at LoRa edge, and $\sigma^2 = \frac{N_0}{2}$ is the noise power spectral density. The proof details of Lemma 4.1 are available in Appendix.

To substantiate our analysis, we delve deeper into the investigation of the SER of LoRa by conducting experiments with various SFs while maintaining a fixed **bandwidth (BW)** of 125 kHz. In our experimental setup, we utilize the SemTech SX1278 as the sender and the SX1302 as the receiver. We manipulate the distance between the transmitter and receiver to control the SNR for our tests. For each SNR value, ranging from −24 to −8 dB with increments of 2 dB, we conduct multiple trials to minimize the impact of environmental noise and fluctuations on our measurements. We systematically test various combinations of valid bits and redundancy bits to evaluate the error detection ability of ECRLoRa checking code.

As depicted in Figure 4, the X axis represents SNR, while the Y axis denotes the symbol error ratio. As interference intensifies, the symbol error ratio escalates to an intolerable level. In LoRa transmissions, it is typical for the interference strength to be markedly higher than the received LoRa signal strength [15], thereby significantly increasing the symbol error ratio at the LoRa edge.

Table 1 shows the error detection ability of the checking code when using different numbers of redundant bits for different numbers of valid bits. ECRLoRa takes the symbol error ratio as input and dynamically adjusts the number of valid bits and the number of redundancy bits for error detection. Since parity-checking with interleaving groups has been proven suitable and compatible for LoRa communication [24], we adopt the idea of Hamming code and provide our dynamic ECRLoRa checking code.

Coding Design of ECRLoRa. ECRLoRa introduces a dynamic parity-checking method that incorporates interleaving groups and a hash algorithm at the LoRa sender. This method involves the dynamic insertion of redundant bits into the payload based on the symbol error ratio. These

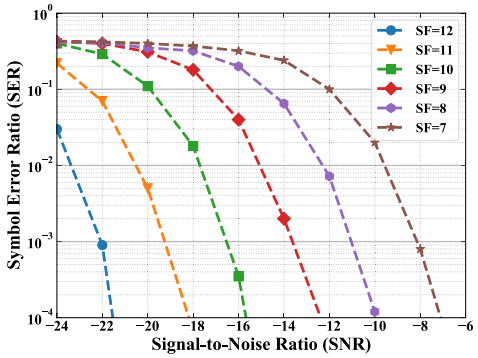


Fig. 4. Symbol error ratio under low SNR.

Table 1. Error Detection Ability of ECRLoRa Checking Code

Valid bits	Redundancy	Error detection ability
7	4	0.118
6	4	0.111
10	4	0.062
9	4	0.061
8	4	0.059
16	5	0.031

redundant bits are generated to facilitate error detection under various interference conditions. To fully index error conditions in an m-bit payload LoRa packet, at least r redundant bits are needed, with r following the condition: $2^r \geq m + r + 1$. Assuming a packet to be transmitted has an m-bit payload and r is the number of redundant bits added, the LoRa sender indicates a minimum of $m + r + 1$ distinct states. Since r bits represent 2^r states, to fully index error conditions, we derive the equation $2^r \geq m + r + 1$ for calculating the minimum redundant bits.

In addition, ECRLoRa employs a hash algorithm to facilitate fast error detection. On the sender side, the payload is mapped to a bit array at the sender utilizing a hash function. The hash value is recalculated at the edge side to verify its consistency with the aforementioned bit array. If they match perfectly, then the received packet is considered error-free. Otherwise, the edge proceeds to identify the actual error position. Traditional hash functions utilize fixed lengths for redundancy construction (e.g., MD5: 32-bits, SHA-3: 256-bits), which are not space-efficient for small-size packets. To strike a balance between redundancy and transmission efficiency, ECRLoRa applies redundant bits in low SNR scenarios where transmission efficiency is enhanced. This redundancy indexes both correct and incorrect cases of the payload. LoRa edges employ the same encoding algorithm as the hash function for consistency.

Lemma 4.2. *Given an m-bit LoRa payload and r redundant bits, the coding rate of the ECRLoRa encoding process is given by* $\frac{P_s}{B} = \frac{m}{m+r+1}$, *where* P_s *represents the number of bits in a message and B denotes the block length.*

Proof. The ECRLoRa error checking code consists of 2^r check bits and 2^{m-r} valid bits in each group. These groups have $2^{m-k} - 1$ transition methods. Consequently, the theoretical undetectable bit distortions in the ECRLoRa error checking code amount to $N_m = 2^m(2^{m-r} - 1)$. The detection capability is defined as the ratio of undetectable distortions to the total number of possible distortions of valid bits, i.e., $P_m = \frac{N_m}{N}$. The error detection capability of ECRLoRa checking code with varying levels of redundancy is illustrated in Table 1. □

Theoretically, in the ideal case with no interference, the bit rate of LoRa is given by $R_b = SF * \frac{BW}{2^{SF}} * \frac{4}{4+CR}$, where SF represents the spreading factor, BW denotes the bandwidth, and CR signifies the coding rate. Under low SNR conditions and in the presence of high interference, the LoRa bit rate decreases to $R_l = P(\widehat{k} \neq k|k) * R_b$, where $P(\widehat{k} \neq k|k)$ is the bit error ratio calculated in Lemma 4.1. Nonetheless, ECRLoRa is able to recover corrupted packets and achieves a bit rate of $R_e = R_l + P_r * (1 - P(\widehat{k} \neq k|k)) * R_b$, where P_r is the recovery ratio of ECRLoRa, as verified in

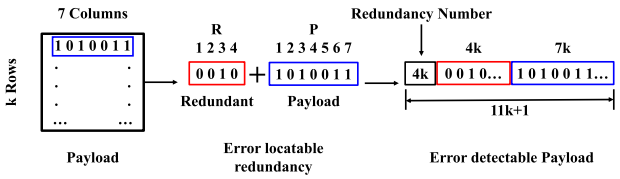


Fig. 5. Inserting the redundant bits.

Table 2. Interleaving Groups

	Index bits
G_1	$P_1, P_2, P_4, P_5, P_7, R_1$
G_2	$P_1, P_3, P_4, P_6, P_7, R_2$
G_3	P_2, P_3, P_4, R_3
G_4	P_5, P_6, P_7, R_4

Section 6.2. Under low SNR conditions with high interference, ECRLoRa yields a higher bit rate compared to LoRa.

Example of ECRLoRa at LoRa Sender. ECRLoRa employs an identification code calculation algorithm to expedite error detection by examining the payload's condition. To reduce redundancy, ECRLoRa repurposes the encoding algorithm for the identification code calculation. Consider an example with $m = 7$ and $r = 4$: The LoRa sender divides the payload into several groups, as illustrated in the left part of Figure 5. Each group comprises m bits. Subsequently, the LoRa sender utilizes the encoding algorithm to compute r redundant bits for each group, which indicate the correctness of each group, as depicted in the middle part of Figure 5. ECRLoRa calculates the identification code of the encoded payload, as detailed in Section 4.2. Finally, it appends the identification code as prefixes to the payload before forwarding it to the LoRa encoding module, as demonstrated in the right part of Figure 5.

Each redundant bit is an additional bit that renders the number of "1"s in a group either even or odd. The LoRa sender divides each m-bit group into four interleaving groups and calculates one parity bit as redundancy for each interleaving group. ECRLoRa employs even parity for redundancy calculation. An illustrative example for each interleaving group is presented in Table 2. If the total number of "1"s in a group is even, the redundant bit is set to "0"; otherwise, it is set to "1."

As per Lemma 4.2, the coding rate of ECRLoRa checking code is $\frac{m}{m+r+1}$. Although redundancy decreases the throughput of the LoRa sender, the packet recovery ratio for both LoRa and SoA is under 20% in low SNR conditions [14]. Unrecoverable packets must be re-transmitted utilizing higher SF, which negatively impacts throughput more significantly than our lightweight coding. Under low SNR with high interference, LoRa switches from SF = 6 to SF = 12 to maintain stable communication, causing the theoretical bit rate to drop from 37.5 kbps to 18 bps, according to SemTech's LoRa deployment guide [25]. In contrast, ECRLoRa can recover corrupted packets and maintain a bit rate of 24 kbps.

4.2 ECRLoRa at LoRa Edges

ECRLoRa swiftly identifies and rectifies errors at LoRa edge nodes, such as LoRa base stations. Initially, the LoRa edge nodes receive signals transmitted by the LoRa sender. They then perform LoRa demodulation, deinterleaving, and decoding processes to obtain the received payload. The received payload, originating from the sender as an error-detectable payload, contains erroneous bits due to interference. Subsequently, the LoRa edges nodes conduct an identification check to determine if the received payload contains errors. By utilizing lightweight decoding and fuzzy error localization techniques, these edges identify and mark error bits within the payload. In the final step, the LoRa edges transmit the corrected payload to the cloud for packet recovery.

Identification Checking. Upon receiving the encoded payload, the edge device extracts the first element from the payload as the identification code. It then compares this identification code with a newly recalculated code derived from the received payload. The relationship between the first element value, R, and the new identification code, P_r, is expressed by the equation

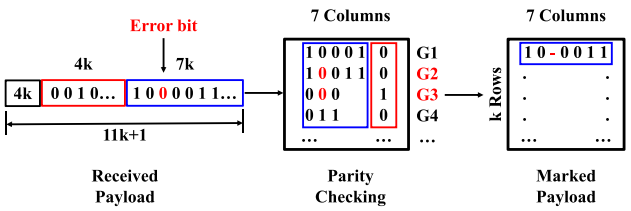


Fig. 6. Parity-checking and error position.

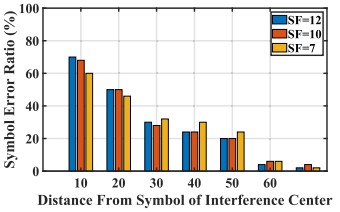


Fig. 7. Symbol error ratio.

$P_r = R * \frac{r*\frac{B}{P_s}+rk+1}{rk}$, where P_r represents the new identification code and R denotes the value of the first element in the received payload.

The aforementioned equation is not valid when the received packet contains erroneous bits. Under this condition, the edge device performs error detection. If no errors are found, then the LoRa edge proceeds with hash calculation, utilizing the encoding function as the hash algorithm and the received payload as input. If the hash value of the received payload matches the recorded one, then the LoRa edge considers the payload correct and uploads it to the cloud. Otherwise, the edge device initiates the parity-checking process to identify the correct frames within the received payload. It is important to note that reported bit values may not be entirely accurate due to hash collisions. Therefore, a group-weighted voting algorithm is employed to select the most plausible value for each bit position during error recovery.

Parity-checking. Parity bits are computed based on the payload bits and redundancy, utilizing the same interleaving groups as the encoding module, as illustrated in Table 2. The LoRa edge device calculates the decimal equivalent of the binary values of the parity bits to identify errors. If the result is "0," then there are no errors. Otherwise, the decimal value indicates the error bit position. More specifically, Figure 6 presents an example of parity-checking. When the received redundancy $R_1, R_2, \ldots, R_r$ are $0, 1, \ldots, 0$, and the received payload $P_1, P_2, P_3, \ldots, P_m$ are $1, 0, \underline{0}, \ldots, 1$, the LoRa sender detects that the parity checks for interleaving groups G_2 and G_3 have failed. This implies that interleaving groups G_2 and G_3 contain erroneous bits, while bits in G_1 and G_4 are correct. By employing cross-checking, the LoRa edge determines that the bit at P_3 is incorrect. It then extracts the error bit and marks the other bits as correct before uploading them to the cloud.

Fine-grained Fuzzy Error Position. ECRLoRa enables error detection at the LoRa edge through lightweight code. However, LoRa suffers from interference signals that are significantly shorter in duration than LoRa packets, resulting in distinct SER under various SFs. This interference tends to concentrate error bits, making symbols closer to the interference more prone to errors, regardless of the SF setting. To demonstrate this phenomenon, we set up an experimental environment utilizing a SemTech SX1278 as the sender and an SX1302 as the receiver. We perform multiple tests with different SFs to analyze the probability of SER depending on the symbol's proximity to the the symbol location of the interference center, as depicted in Figure 7. The figure reveals that as symbols approach the interference center, the likelihood of errors occurring increases, irrespective of the SFs used. Based on these observations, we propose a fuzzy algorithm to be implemented at the LoRa edge to boost error detection precision, ultimately enhancing the system's robustness against interference-induced errors. This improved resilience will provide more reliable communication in environments with varying interference levels.

We propose the pseudocode for the algorithm in Algorithm 1. The error detection result from parity-checking is employed to determine the approximate error position. Subsequently, we leverage interference features, allowing error penetration into symbols in nearby interleaving groups. ECRLoRa achieves the best outcomes when penetration begins until the number of error symbols

ALGORITHM 1: Fuzzy error positing algorithm

Input: P, Error Detected Payload
Output: DP_i, Accurate Error positing

```
for all P_i ∈ P do
  if P_i == "-" then
    if interleaving group is marked as error then
      mark nearby interleaving group as error
    end if
    mark interleaving group as error
  end if
end for
return marked error positing as DP_i
```

in the interleaving group reaches two. Specifically, we mark the adjacent interleaving group as incorrect when concentrated errors occur in a single interleaving group. Finally, the marked error position is returned as the error detection result.

Error Detection Complexity. We further analyze the computational overhead associated with localization. As demonstrated earlier, the coding rate for our payload construction is $\frac{m}{m+r+1}$. The computational overhead consists of three components: bit number calculation, redundancy value calculation, and redundancy insertion, with complexities of $O(1)$, $O(m*N)$, and $O(N)$, respectively. Consequently, the computational complexity for the LoRa sender, $T_s(N)$, is $O(N)$, where N denotes the payload length. At the LoRa edge, the computational cost of error detection comprises three parts: hash-checking, parity-checking, and cross-error posting, with costs of $O(m * N)$, $O(r * N)$, and $O(N)$, respectively. Therefore, the computational cost of error detection at the LoRa edge, $T_r(N)$, is $O(N)$, where N represents the received payload length.

We have two key advantages for fast error detection at the LoRa edge: (i) the delicate insertion of marking bits, preventing wasteful usage of redundancy, and (ii) minimizing the computation of LoRa devices. We propose a lightweight approach for error checking that detects and locates errors at the LoRa edge. More specifically, we leverage the benefits of the LoRaWAN architecture. The LoRa edge processes segmentation error detection and uploads appropriate frames, disregarding error recovery. The cloud, connected to numerous edges, receives the correct frames from multiple packets for packet recovery. Since ECRLoRa reconstructs the LoRa payload, it necessitates neither hardware nor firmware modifications in existing commercial technologies. Consequently, ECRLoRa offers a convenient architecture for deployment within modern LoRaWAN infrastructures.

5 PACKET RECOVERY

ECRLoRa identifies error bits and transmits correct frames to the cloud at the LoRa edge. The LoRa edges extract the correct frames and forward them to the cloud, accompanied by the packet mark. The cloud associates frames originating from the same packet, collecting a sufficient number of frames to assemble packets that pass error checking. Subsequently, these packets are distributed to the required LoRa edges and recorded by the cloud with a predetermined lifecycle. Fast packet recovery is achieved by minimizing the amount of transmitted data from the LoRa edge to the cloud and expediting the frame combination process.

ECRLoRa leverages the global management competencies and computational capabilities of the cloud to facilitate fast packet recovery. However, ECRLoRa also encounters challenges due to hardware constraints, such as bandwidth capacity, computational capability, and ACK thresholds.

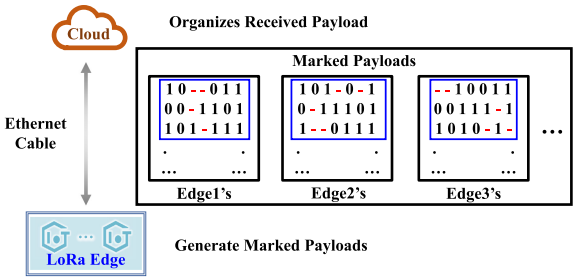


Fig. 8. Correct frame transmission from multiple LoRa edges.

This section discusses the error recovery processes and presents strategies to address these challenges effectively.

5.1 Correct Frame Transmission

In the LoRaWAN architecture, each LoRa edge connects to the cloud via an ethernet cable or a cellular network. In ECRLoRa, LoRa edges detect error bits upon receiving a packet from the LoRa sender. Subsequently, the LoRa edge transmits the correct frame to the cloud, as illustrated in Figure 8. The LoRa edge sends the marked payload to the cloud to facilitate fast packet recovery, circumventing the need to transmit the RSSI sequence. This approach optimizes bandwidth utilization at the LoRa edges. Our second micro-benchmark in the motivation section demonstrates that the payload received by LoRa edges exhibits disjoint corrupted positions due to the irrelevant interference affecting each edge. As a result, the cloud receives distinct segments of the same packet, which are then collaboratively employed for efficient packet recovery.

5.2 Group-weighted Voting

The cloud processes packet recovery upon receiving the correct frames from the LoRa edge. Despite the LoRa edge transmitting frames that have passed both hash-checking and parity-checking, these frames may still harbor errors. Consequently, the cloud does not grant them complete trust. To address this issue, the cloud employs each frame in a fast packet recovery process and adeptly combines the correct segments to create a packet that passes error checking.

More specifically, ECRLoRa employs a group-weighted voting algorithm within the cloud to meticulously extract accurate segments from received frames and reconstruct packets. The weight assigned to each symbol represents the probability of the symbol value, which is influenced by the packet's error condition, the length of the received accurate frame, and the parity-checking outcome. The following equation represents the error condition and parity-checking results:

$$E_i = \wedge_{n=1}^{\cup(PinG_i)} P_n. \tag{2}$$

In this equation, E_i symbolizes the error condition of the ith interleaving group, G_i represents the ith interleaving group, and P refers to the payload within the ith interleaving group.

The frame length serves as the prefix of the marked payload, and the following equation computes the length of the ith interleaving group:

$$L_i = \sum_{k=1}^{p} \exists i_k \wedge \neg \exists i_{k+1}. \tag{3}$$

In this equation, L_i denotes the length of the ith interleaving group, and i_k represents the kth symbol within the ith interleaving group.

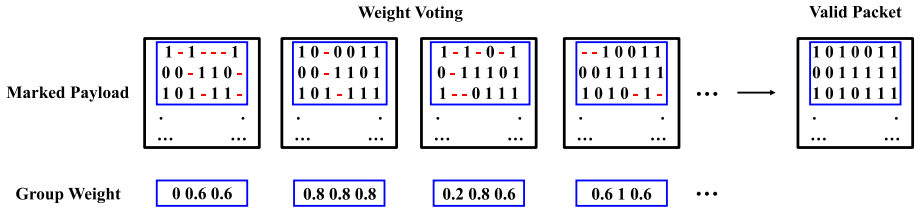


Fig. 9. Group-weighted voting process for packet recovery.

The weight of the kth symbol is represented by the following equation:

$$W_k = \frac{\sum_{i=1}^{L_i} i_t - \sum_{i=1}^{L_i} (i_t == 1 \wedge G_i)}{\sum_{i=1}^{L_i} i_t}. \tag{4}$$

In this equation, W_k signifies the weight of the kth symbol, while i_t represents the tth bit in the binary sequence of i.

As depicted in Figure 9, ECRLoRa employs the group-weighted voting algorithm to assign weights to each symbol, effectively assessing the reliability of packets during the group-weighted voting process. A higher weight indicates a more acceptable symbol value. This approach enables the correct parts of the received frame to be combined to reconstruct a valid packet. Furthermore, the cloud organizes received frames corresponding to the same packet and executes the weighted voting algorithm among frames within a single group, facilitating fast packet recovery. Once a sufficient number of frames are received to constitute a valid packet, the cloud halts the group-weighted voting, records the packet, and transmits it back to the LoRa edge.

5.3 Recovered Packet Feedback

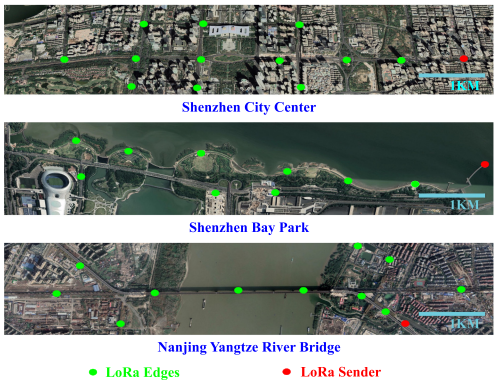
The cloud stores the valid packet with a specified lifetime following the group-weighted voting computation, expediting packet recovery and enabling consecutive LoRa edges to utilize these packets without requiring re-calculation. As mentioned in Section 2, the corrupted segments of the frames are disjoint. This characteristic enhances the cloud's packet recovery capability when multiple frames from distinct edges are accessible. ECRLoRa converts the challenge of isomorphism interference into an opportunity for packet recovery via multiple edges within the LoRaWAN architecture. The lifetime of each valid packet is established at six rounds. The cloud preserves the recovered packet until six LoRa edges request it or until it surpasses the ACK threshold, which is one second in LoRaWAN.

6 EVALUATION

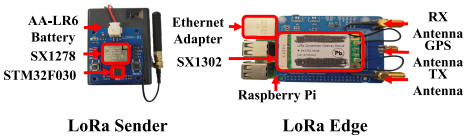
In this section, we commence by outlining the experimental configurations, scenarios, and data acquisition methods employed for assessing the performance of ECRLoRa. Subsequently, we evaluate the effectiveness of ECRLoRa in comparison to **state-of-the-art (SoA)** techniques [1, 20] under various conditions.

6.1 Experiment Setup

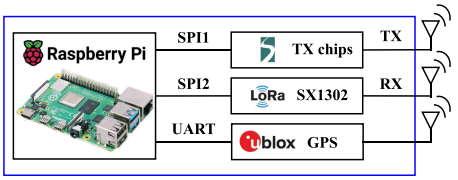
LoRaWAN senders, edge, and cloud deployments: We evaluate the effectiveness of ECRLoRa on several 10 km^2 test beds in two major Chinese cities, as depicted in Figure 10(a). The test bed in the Shenzhen city center (Figure 10(a), top figure) features an occluded, high-interference, and on-the-land environment. The test bed in Shenzhen bay park (Figure 10(a), central figure) presents a hollowness, low-interference, and an on-the-water environment. Furthermore, the test



(a) Evaluation Experiments Setting (SF = 7, 10, 12 Distance = 100m, 1km, 2km, 3km, 4km, 5km Tx Power = 10dBm Rx Gain = 10dB).



(b) LoRa Device.



(c) LoRa Edge's Architecture.

Fig. 10. Experimental settings and hardware architecture.

bed along the Nanjing Yangtze River bridge (Figure 10(a), bottom figure) presents a hollowness, low-interference, and cross-bridge environment.

In each test bed, we establish one LoRa sender and several groups of LoRa edges. The LoRa sender consists of a transceiver SX1278 and a controller STM32F030 (Figure 10(b), LoRa Sender). For each group of LoRa edges, we deploy an SX1302 digital baseband chip, along with several SX1250, SX1255, and SX1257 transceiver chips, and a Raspberry Pi (Figure 10(c)) manages these chips to form the LoRa edge. The subsequent experiments are conducted within the CN470-510MHz frequency band.

In our experiments, we introduce additional interference signal sources along each LoRa edge to generate varying levels of interference. These same-frequency, random interference signal sources are implemented utilizing USRP N310 devices. To maximize the interference impact, we equip each USRP N310 device with a 20 dB gain antenna and continuously transmit signals at maximum power. We employ three different configurations to represent varying interference levels: 0×, 2×, and 4×. The 0× configuration represents a LoRa edge exposed only to environmental noise, with no additional interference signal sources. In this scenario, we expect a relatively low channel occupancy of approximately 10%. The 2× configuration includes environmental noise and two additional USRP N310 interference signal sources, resulting in a higher channel occupancy of around 30%. The 4× configuration encompasses environmental noise and four additional USRP N310 interference signal sources. In this case, we anticipate an even higher channel occupancy of about 60%, as the increased number of interference sources contributes to packet collisions and delays.

To fully exploit the potential of disjoint interference, we need to identify each LoRa edge [30]. We employ a GPS module for LoRa edge identification. Specifically, the LoRa sender is situated at a fixed, pre-recorded location. For each LoRa edge in the experiment groups, we log the location value from the GPS module as illustrated in Figure 10(c), enabling the calculation of the distance and identifying the LoRa edge. Moreover, the distance serves as a crucial factor in determining whether the experiments are conducted under low SNR conditions. Thus, we primarily change the distance between the LoRa edges and the LoRa sender in each group of experiments. To control the distance more precisely and minimize the impact of occlusions on the SNR, we predominantly deploy each group of LoRa edges along the main road. However, we also place some groups of LoRa edges off the road—as indicated by the green nodes away from the main road in Figure 10(a). Data recording and identical experiments are performed on these groups as well.

In each group of experiments, SNR, SF, edge number, and interference conditions serve as variable settings. SNR is defined as the ratio of signal power to background noise power: $SNR = \frac{P_{signal}}{P_{noise}}$. As signal power diminishes with increasing distance, SNR is chiefly controlled by the distance between the LoRa sender and the LoRa edge. We set the sender-to-edge distance to range from 100 m to 5 km. In each group of experiments, we designate SF values between 7 and 12 and record data packets. Specifically, we employ an SF value of seven for distances between 100 m and 1 km, 10 for distances between 1 and 3 km, and 12 for distances between 3 and 5 km. We utilize six LoRa edges to collect data packets in each experiment group. Simultaneously, the edge number is controlled later during the packet recovery module, incorporating data packets from specific LoRa edges based on the required edge number. To minimize redundancy and ensure error localization performance, ECRLoRa elaborately selects four redundancies for seven bits (the reserved parameters in Section 4). We find that, with this configuration, the LoRa edge accurately identifies error fragments, preventing the waste of correct bits.

It is crucial to note that the −10 dB SNR value mentioned in the experiments is not fixed but serves as an example under our experimental conditions. Our methodology dynamically selects the SF at the LoRa edge based on the current SNR, SER conditions, and the overall communication environment. The LoRa edge makes the decision regarding the coding rate of redundancy, taking these factors into account.

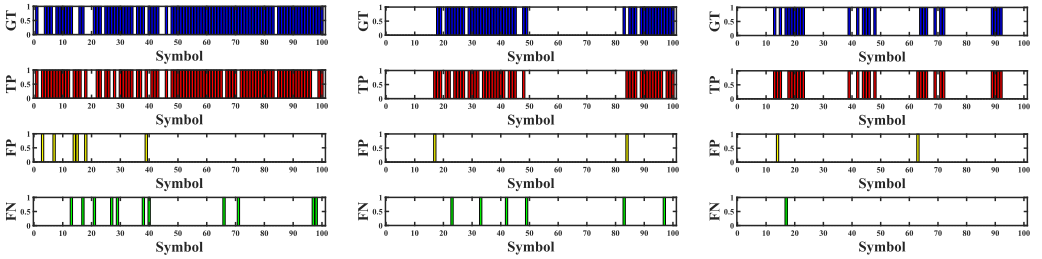
To implement dynamic SF selection and synchronization, the LoRa edge periodically sends a specific broadcast frame containing the chosen SF value or triggers it when significant changes in the communication environment occur. This enables other devices within the network to update their SF values accordingly. By incorporating these dynamic adjustments and synchronization mechanisms into the experimental setup, we intend to demonstrate the adaptability and responsiveness of our approach in various real-world communication scenarios.

Experimental Setup for Comparison: In our study, we conduct comparative experiments involving two state-of-the-art techniques, namely OPR and DaRe. As the source code for OPR is not accessible, we re-implement the algorithm following the descriptions and methodology outlined in the pertinent literature [1]. We diligently replicate deployment details, encompassing parameter settings, node configurations, and channel characteristics, to maintain congruence with the original work. However, we utilize the open source code for DaRe [20] and introduce several enhancements to ensure a fair comparison between DaRe and ECRLoRa. The most significant enhancement to the DaRe source code is the incorporation of the multi-receiver function. We thoroughly analyze the DaRe algorithm and adapt it to conform with our experimental setup, guaranteeing that both OPR and DaRe are assessed under identical conditions.

Concerning low-level settings such as BW, Power, and RF-frequency, we adhere to the consensus parameter settings within the LoRa community, specifically 500 KHz, 20 dBm, and unlicensed sub-GHz bands, respectively. Our experimental settings align with those detailed in the OPR publication. As a result, we achieve exceptional performance and reliability in our experiments, demonstrating the effectiveness of our approach across a variety of conditions, including different SNR, SF, edge numbers, and interference scenarios. This comprehensive comparison of OPR and DaRe enables us to gain insights into their individual strengths and weaknesses while highlighting the advantages of our proposed ECRLoRa approach.

6.2 Experiment Results

In this subsection, we compare ECRLoRa with state-of-the-art approaches under various settings. Our experiments evaluate the following performance metrics: (a) Error detection ability, (b) Packet recovery ability, (c) Time consumption, and (d) Throughput. Each experiment encompasses diverse



(a) SF=7, Range=1km. (b) SF=10, Range=2km. (c) SF=12, Range=3km.

Fig. 11. Examples of error detection.

settings of SNR, SF, edge number, and interference conditions. The experimental settings cover most scenarios that LoRa may encounter. The subsequent section provides detailed experiment settings and results.

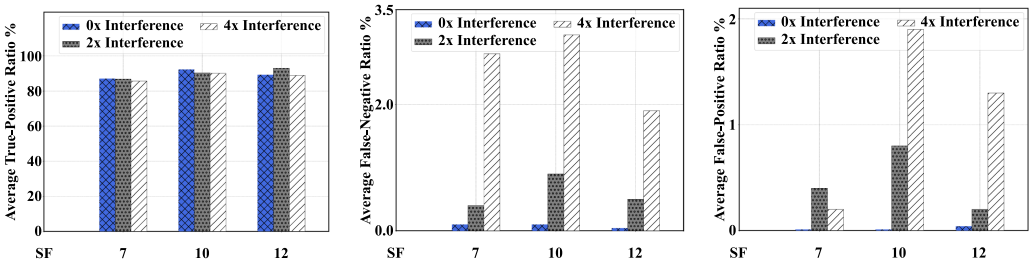
Error Detection Ability. To evaluate the accuracy of error detection, we deactivate the interleaving feature of the original LoRa to further evaluate ECRLoRa's error detection capability. We measure the true-positive, false-positive, and false-negative conditions for each symbol within the LoRa packet.

Figure 11 illustrates examples of ECRLoRa's error detection ability. Label GT represents the ground truth of the actual error position, while TP, FP, and FN denote the true-positive, false-positive, and false-negative conditions of ECRLoRa error detection. ECRLoRa accurately detects most errors with a few false-negative values (less than 3.5%) and a minimal number of false-positive values (less than 2%). We observe that the false-positive values are in close proximity to the true-positive values. Therefore, ECRLoRa naturally employs group recovery to minimize the influence of false-positive values.

Figure 12 presents the statistical error detection capability of ECRLoRa under different SF and various interference conditions. We employ the LoRa SX1278 chip to generate interference. Each interfering node transmits at the same frequency as the LoRa sender and edge. *Furthermore, interference nodes utilize full transmit power at +20 dBm (maximum transmit power of the SX1278 chip) and continuously transmit LoRa packets without any blank frames.*

Figure 12(a) demonstrates the average true-positive ratio of ECRLoRa error detection when encountering zero, two, and four interfering nodes with SF values of 7, 10, and 12, respectively. ECRLoRa successfully detects over 90% of errors when facing environmental noises. Figure 12(b) illustrates the average false-negative ratio under the same experimental conditions. ECRLoRa maintains a false-negative ratio of less than 3.5% across all experimental settings. Although interference impacts error detection performance (as interference introduces errors), ECRLoRa reliably identifies error positions with an acceptable false-negative ratio. Figure 12(c) indicates the average false-positive ratio under different SFs and varying interference conditions. ECRLoRa achieves a false-positive ratio of less than 2% during the error detection process, thereby improving the packet's recovery foundation.

Figure 13 illustrates the probabilities of successful packet reception by a varying number of nodes in three different scenarios. Specifically, for Shenzhen City Center, the probability of a packet being received by one node is highest at 98%, and it decreases as the number of nodes increases, dropping to 50% for six nodes. In Shenzhen Bay Park, the probability starts at 98% for one node, and it similarly declines with an increasing number of nodes, reaching 65% for six nodes. Lastly, for the Nanjing Yangtze River Bridge scenario, the probability is highest for one node at 99% and decreases to 70% for six nodes. The figure effectively demonstrates how the probability of



(a) Average True-Positive Ratio. (b) Average False-Negative Ratio. (c) Average False-Positive Ratio.

Fig. 12. Error detection ratio.

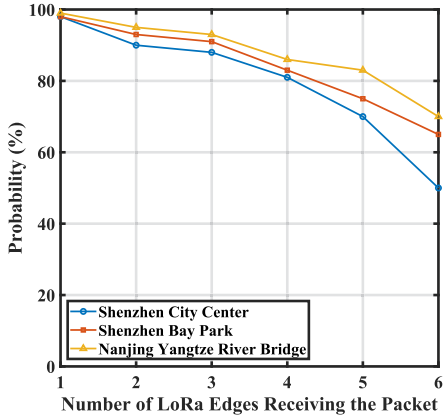


Fig. 13. Probability of a packet received by multiple LoRa edges.

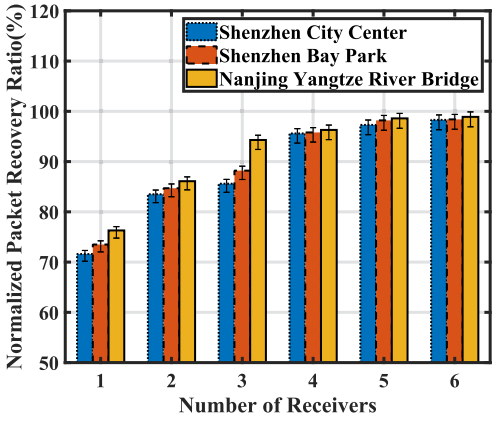
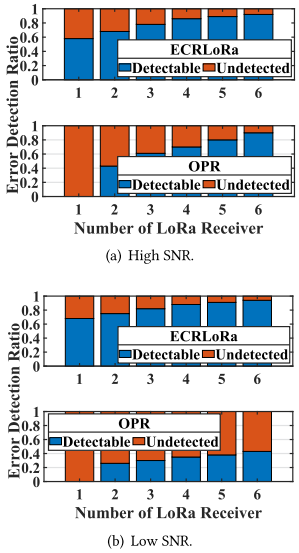


Fig. 14. Comparison of packet recovery ratio under three scenarios.

successful packet reception varies across different environments and with the number of nodes receiving the packet.

Figure 14 presents the packet recovery performance in three distinct deployment scenarios. Owing to environmental differences, the performances are slightly different. Shenzhen city center, with its occluded and high-interference environment, exhibits the poorest performance. Shenzhen Bay Park, characterized by more open spaces and reduced interference, demonstrates better average performance. Meanwhile, the Nanjing Yangtze River Bridge scenario, featuring the most hollowness and low-interference environment, delivers the best performance among the three scenarios. In summary, while packet recovery performance varies slightly due to environmental differences, the overall packet recovery capability and trends remain consistent across all three scenarios. Furthermore, detailed performance results for the remainder of this section are presented utilizing the deployment scenario of the Nanjing Yangtze River Bridge.

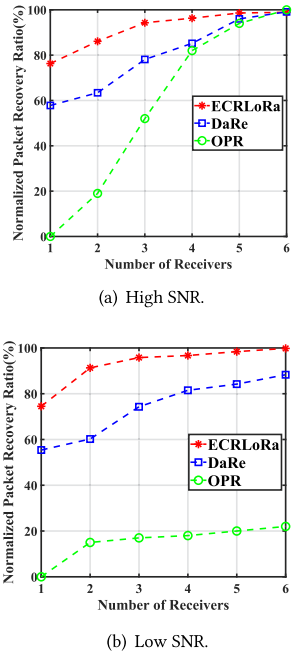
To further evaluate ECRLoRa's performance, we conduct experiments under various SNR conditions utilizing multiple LoRa edges. Figure 15(a) demonstrates that in high SNR scenarios, both ECRLoRa and the state of the art achieve an error detection ratio exceeding 90% when utilizing more than five LoRa edges. Conversely, under low SNR conditions, as shown in Figure 15(b), ECRLoRa attains a 75% error detection ratio with only two edges, while the state of the art reaches a 22% error detection ratio under same settings. Furthermore, ECRLoRa's error detection



(a) High SNR.

(b) Low SNR.

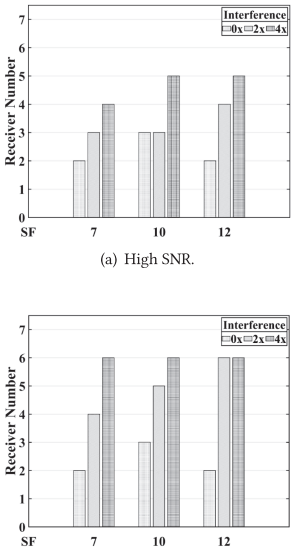
Fig. 15. Error detection ability vs. SoA.



(a) High SNR.

(b) Low SNR.

Fig. 16. Packet recovery ratio.



(a) High SNR.

(b) Low SNR.

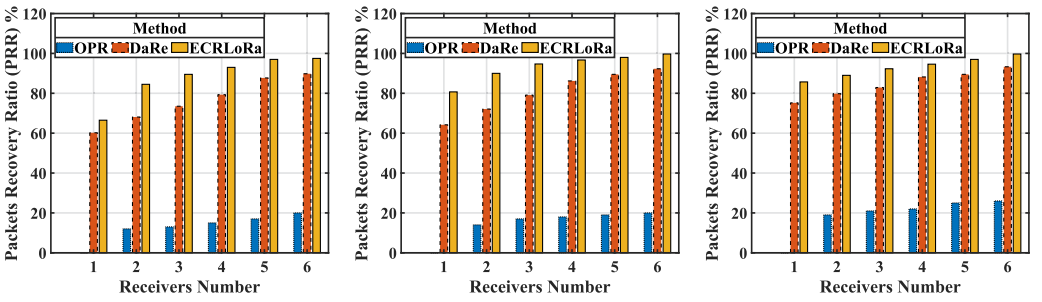
Fig. 17. Number of edges.

ratio approaches 90% when five or more edges receive LoRa packets, whereas the state-of-the-art achieves an accuracy of no more than 40%.

To further demonstrate ECRLoRa's capabilities, we conduct a series of experiments under a broader range of conditions. We compare ECRLoRa with state-of-the-art techniques [1, 20] in various SNR conditions, ranging from high to low. We examine the experimental results based on data collected in both low and high SNR conditions. Figure 16 showcases the statistical results of our experiments. By exploiting disjoint interference opportunities, as discussed in Section 2, ECRLoRa attains a higher PRR as the number of edges increases. The packet recovery ratio surpasses 90% when there are three edges. ECRLoRa outperforms DaRe under high SNR conditions when the edge number is less than six, as illustrated in Figure 16(a). Furthermore, by utilizing accurate error detection in low SNR conditions, ECRLoRa achieves an average performance improvement of 1.25× compared to DaRe [20] under ultra-low SNR scenarios (e.g., SNR<−10 dB), as depicted in Figure 16(b).

Packet Recovery Ability. To investigate ECRLoRa's packet recovery capabilities by leveraging the disjoint correct frames from multiple edges, we first determine the maximum number of edges required for ECRLoRa to recover packets under various experimental conditions. We employ the same experimental setup as in the error detection ability experiments. Figure 17 displays the number of edges ECRLoRa necessitates for packet recovery. ECRLoRa successfully recovers packets with fewer than five edges under high SNR conditions. Moreover, ECRLoRa utilizes no more than six edges for packet recovery when encountering low SNR situations. By implementing the group-weighted voting algorithm, ECRLoRa effectively combines the disjoint correct frames from multiple edges and minimizes the number of edges needed for packet recovery.

We evaluate the performance of ECRLoRa under low SNR conditions utilizing various experimental configurations of spreading factors and edge numbers. Figure 18 demonstrates the



(a) PRR when SF=7. (b) PRR when SF=10. (c) PRR when SF=12.

Fig. 18. Packet recovery ratio under low SNR.

differences in Packet Recovery Ratio between ECRLoRa and state-of-the-art cloud-optimized application layer approaches [1, 20] under low SNR with multiple experimental settings. DaRe exhibits a higher packet recovery ratio due to redundant coding, but it does not fully exploit the benefits of the LoRa edge–cloud architecture, resulting in lower performance than ECRLoRa. OPR has a packet recovery ratio below 30% because of inaccurate RSSI-based error detection under low SNR conditions. Furthermore, by leveraging error detection at the edge, ECRLoRa attains over 60% packet recovery ratio utilizing a single edge. Additionally, the group-weighted voting algorithm enables ECRLoRa to capitalize on the disjointness of correct frames from multiple edges. Consequently, ECRLoRa's packet recovery performance tends to increase with a growing number of edges.

Throughput. We compare the throughput of ECRLoRa with cloud-optimized application layer approaches [1, 20] and LoRa under a signal-to-noise ratio ranging from 0 to −20, utilizing SF = 7, as depicted in Figure 19. ECRLoRa's throughput is the bit rate after removing redundancy. It is important to note that ECRLoRa features dynamic redundancy under varying SNRs. For example, ECRLoRa employs a dynamic coding methodology based on SNR. Under low SNR conditions, ECRLoRa chooses the coding rate for redundancy according to the SER, as detailed in Section 4.1. ECRLoRa deactivates redundancy when SNR increases to the point where the symbol error ratio is too low to necessitate redundancy. However, the received packet still contains errors. As a result, ECRLoRa conducts error detection and employs the group-weighted voting algorithm for packet recovery. Under the above conditions, ECRLoRa's throughput is, on average, 1.36× higher than OPR and 1.05× higher than DaRe. Last, under high SNR conditions, the quality of the received signal is so exceptional that some LoRa edges correctly receive the packet. ECRLoRa selects the first correct packet that arrives.

We further compare ECRLoRa's throughput with DaRe, OPR, and LoRa under low SNR conditions with severe interference (4× interference nodes). The throughput is calculated utilizing the packet received ratio, packet recovery ratio, and valid payloads. As illustrated in Figure 20, ECRLoRa achieves the highest throughput when SF = 7, 8, 9, 10, 11. On average, ECRLoRa's throughput is 1.88× and 1.23× higher than OPR and DaRe, respectively. The throughput of DaRe, OPR, and LoRa decreases when SF changes from 7 to 9 due to the longer packet time, which intensifies interference in each packet, degrading the packet and reducing the throughput. However, ECRLoRa attains more than 4.4× throughput compared to OPR and LoRa when SF = 9 by recovering corrupted packets. As SF changes from 9 to 11, the throughput of both OPR and LoRa experiences a slight increase, because the higher SF reduces the symbol error ratio of the received signal. When SF = 12, ECRLoRa's throughput is 0.7445 kbps, marginally lower than OPR's 0.8 kbps.

Time Complexity. ECRLoRa consists of three steps: encoding, error detection, and packet recovery. We conduct numerous experiments to assess the average time consumption of ECRLoRa's

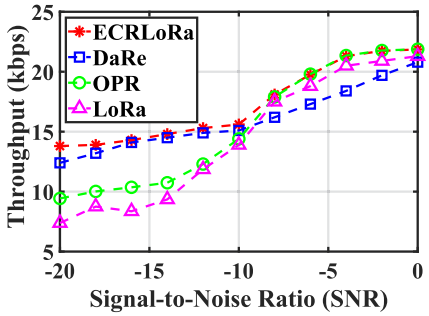


Fig. 19. Throughput under different SNR when SF = 7.

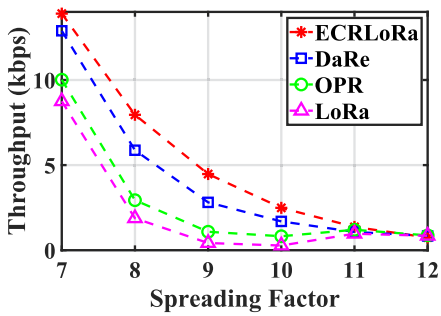
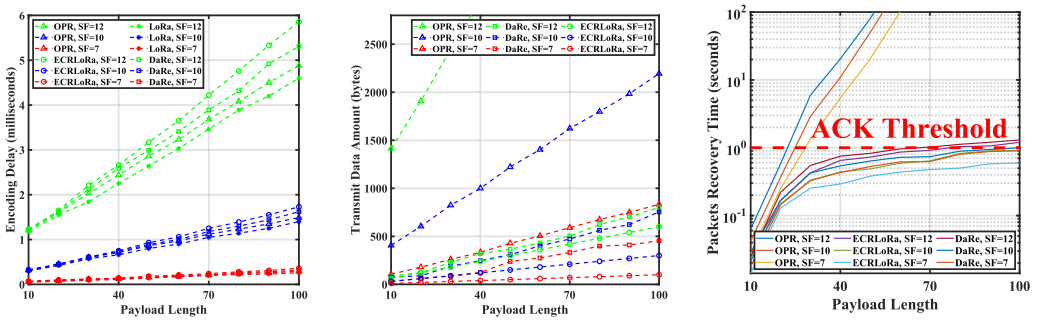


Fig. 20. Throughput with SF from 7 to 12 under Low SNR.



(a) Encoding Delay. (b) Transmit Data Amount. (c) Packet Recovery Time (seconds).

Fig. 21. Average time consumption.

steps. The encoding step brings an encoding delay at the LoRa sender, resulting in a drawback in LoRa performance. To evaluate ECRLoRa's feasibility at the LoRa sender, we perform experiments with payload lengths ranging from 10 to 100. Figure 21(a) presents the experimental results of encoding time consumption. ECRLoRa exhibits a longer encoding time than DaRe and OPR due to its interleaving error detection code. However, the error detection code effectively leverages the disjointness of LoRa edges to achieve rapid error detection.

Figure 21(b) illustrates the consumption of data transmission utilizing the transmit data amount as a metric, comparing the proposed method with OPR and DaRe. The horizontal axis represents the Payload Length, ranging from 10 to 100 bytes, while the vertical axis indicates the Transmit Data Amount (in bytes). ECRLoRa achieves exponential savings in data transmission by detecting errors at the LoRa edge and transmitting correct frames instead of the RSSI sequence. Although ECRLoRa has a minor drawback in encoding speed due to utilizing redundancy, it is faster and more reliable in error detection and packet recovery. Employing a small amount of redundancy for error detection is a wise tradeoff. Specifically, at SF = 7, ECRLoRa disables redundancy and maintains a consistent data transmission amount between 10 and 100 bytes, while DaRe contains higher redundancy amount and has data transmission amount varies considerably from 32 to 455 bytes. For higher SF values, such as SF = 10 and SF = 12, ECRLoRa sustains stable data transmission amounts within the ranges of 30–300 bytes and 60–600 bytes, respectively. Conversely, DaRe demonstrates significantly higher data transmission amounts at these SF levels, ranging from 76 to 754 bytes for SF = 10 and 89 to 801 bytes for SF = 12.

Additionally, when comparing ECRLoRa with OPR, we observe a distinct advantage in data transmission amount for ECRLoRa. OPR relies on data repetition to combat interference, which leads to a larger data transmission amount compared to our method. For example, at SF = 7, OPR's data transmission amount ranges from 105 to 834 bytes, which is considerably larger than ECRLoRa's range. This trend continues at higher SF values such as SF = 10 and SF = 12. At SF = 10, OPR's data transmission amount varies between 406 and 2194 bytes, while at SF = 12, it ranges from 1,414 to 6,040 bytes.

Figure 21(c) illustrates the packet recovery time consumption for various payload lengths. ECRLoRa and DaRe exhibit linear packet recovery time, while OPR shows exponential recovery time. To clearly demonstrate ECRLoRa's packet recovery capability, we recover packets that disregard the ACK threshold, wait until the packet is fully recovered, or exceed the time limitation. In our settings, the maximum recovery time is one hundred seconds. Benefiting from the group-weighted voting algorithm and store-then-feedback mechanism, ECRLoRa is, on average, 7.5× faster than OPR when the payload length is less than 30. When the payload is longer than 30 but shorter than 60, OPR's packet recovery time exceeds the LoRa ACK threshold, and ECRLoRa performs 192.7× faster on average under such conditions. Furthermore, OPR fails to recover packets when the payload length surpasses 60 in our experimental environment. In contrast, ECRLoRa successfully recovers packets with time consumption below the LoRa ACK threshold. In conclusion, ECRLoRa is an brilliant design that employs minimal redundancy at the LoRa sender, detects errors, and transmits accurate frames at the LoRa edge while rapidly recovering packets in the cloud.

7 RELATED WORKS

This section summarizes the related works of this article. Some studies focus on essential error detection algorithms, e.g., EEC [2] and TVA[19]. The majority of recent LPWAN interference mitigation efforts fall into the following two categories. The first category includes protocol-based interference mitigation approaches that involve redesigning LoRa edges and end devices [3, 5, 10, 32, 39]. The second category has recently emerged, utilizing cloud computing resources to recover corrupted packets as a means of information collaboration for interference mitigation. These cloud-optimized application layer approaches, such as in References [1, 20], demonstrate excellent compatibility with existing LPWAN systems.

Protocol-based Approaches. Early efforts in LPWAN interference mitigation have concentrated on (i) the physical layer, encompassing SCLoRa [11], PSR [28], Choir [6], FTrack [32], and mLoRa [31], and (ii) the MAC layer, which includes S-MAC [35], LoRaFFEC [3], and LMAC [8]. These protocol-based solutions necessitate the redesign of the PHY or MAC layers in LPWAN senders or edges. The need for specialized devices significantly constrains the large-scale application of these approaches.

Cloud-optimized Application Layer Approaches. Taking advantage of the LPWAN system architecture, it is feasible to employ cloud resources for interference mitigation [1, 20]. For example, DaRe [20] integrates convolution and fountain codes for spatial and temporal information exploration in LoRa communication channels. However, these redundancy coding approaches are limited to specific LoRa edges and cannot fully collaborate with multiple edges for distributed error detection. A collaborative method is necessary to fully utilize this information, enabling LoRa to enhance data accuracy while reducing coding redundancy for increased throughput and reliability. OPR [1] offloads RSSI samples, sends corrupted packets to the cloud, and recovers them. Cloud-optimized application layer approaches have made significant strides in recent research by leveraging the cloud's ample computational resources and global management capabilities. Moreover, these approaches are compatible with existing LPWAN systems as they do not require hardware modifications. Nonetheless, offloading RSSI samples to the cloud results in substantial

Table 3. Related Cloud-optimized Application Layer Approaches

	Cost	Delay	PRR	Low SNR
LoRaWAN Standard				
Standard [24]	Low	Low	Low	Support
Cloud-optimized				
DaRe [20]	High	Low	High	Support
OPR [1]	Low	High	High	Not Support
Edge–Cloud Collaboration				
ECRLoRa	**Low**	**Low**	**High**	**Support**

communication overhead in uplink bandwidth. Our preliminary work [21] demonstrated the feasibility of edge–cloud collaboration and achieved packet recovery with 51.76% corruption. This article introduces ECRLoRa to further enhance packet recovery capacity at the edge side. We conduct a series of empirical studies to support our insights as follows.

ECRLoRa is designed for edge–cloud collaborative packet recovery under low SNR conditions. Our approach involves utilizing LoRa edges to detect errors in corrupted packets accurately. This way, only the error bits need to be transmitted to the cloud instead of a large volume of RSSI samples. Generally, the length of RSSI is several times the length of the LoRa payload. As in-field interference worsens, transmitting RSSI to the cloud results in a heavy network load. Minimizing the required data transmission amount and enhancing error detection capabilities at the edge contribute to reducing the overall time cost for packet recovery. As shown in Table 3, error detection at the LoRa edge significantly decreases the data transmission amount. Moreover, ECRLoRa employs a weighted voting algorithm to quickly recover packets by collaborating with correct frames from multiple LoRa edges, thus recovering severely damaged packets beneath the ACK threshold.

8 CONCLUSION

This article introduces ECRLoRa, an edge–cloud collaborative approach for packet recovery under low SNR conditions. It leverages the signal perception advantage of LoRa edge and the global management capabilities of the cloud. To the best of our knowledge, ECRLoRa is the first work implementing packet recovery for interference mitigation through edge–cloud collaboration. It achieves low cost, reduced packet recovery delay, and a high packet recovery ratio while supporting low SNR. ECRLoRa attains a 96% Packet Recovery Ratio at low SNR, below the ACK threshold of LoRaWAN. It demonstrates 1.8× higher throughput, 7.5× faster recovery time, and 4.92× greater accuracy compared to the state-of-the-art under such conditions. ECRLoRa explores a novel methodology for interference mitigation utilizing edge–cloud collaboration and strikes an optimal balance between reliability, flexibility, deployability, and complexity.

APPENDIX

Proof of Lemma 4.1.

Lemma 1. *Given LoRa spreading factor SF, and the noise power spectral density σ^2. For transmitting symbol 'k', the SER of LoRa is as follows:*

$$P(\widehat{k} \neq k|k) = \int_0^{\infty} \left[1 - \left[1 - exp\left[-\frac{\mu^2}{2\sigma^2}\right]\right]^{2^{SF}-1}\right] f_{\mu_k}(\mu)d\mu. \tag{5}$$

Here "k" is the transmitted symbol, μ follows Rician distribution, SF is the spreading factor at LoRa edge, and $\sigma^2 = \frac{N_0}{2}$ is the noise power spectral density.

Proof. The proof involves two steps: (1) First, we show the signal received by LoRa edge under interference in Appendix 8, then for transmitting every symbol "k," we have the bit error ratio in Lemma 10.

(2) Next, we show that noise satisfies the Rayleigh distribution of the Lemma 4.1. The proof is completed by calculating the symbol error ratio using the Rayleigh distribution function.

To prove Equation (1), we first show that the LoRa communication is under interference. LoRa devices are highly sensitive to environmental noise. Due to the interference, the signal received by the LoRa edge is as follows:

$$y[n] = hx[n] + z[n] + I[n], \tag{6}$$

Here $n \in \{0, \ldots, 2^{SF} - 1\}$, $x[n]$ is the original signal from the sender, $z[n]$ represents plural white Gaussian noise, $I[n]$ is the interference received by LoRa edge, $h \in \mathbb{C}$ is the channel signal gain.

The interference contained in the received signal influences the accuracy of demodulation results. For the received signal $y[n]$:

$$\sum_{n=0}^{2^{SF}-1} y[n] * x_i^*[n] = \begin{cases} 2^{SF}(\sqrt{E_S} + \phi_i + I_i), & i = k \\ 2^{SF}(\phi_i + I_i), & i \neq k \end{cases}. \tag{7}$$

Here ϕ_i represents the process of plural white Gaussian noise, and I_i is the interference received by the LoRa edge. We define $\delta_i = |\phi_i + I_i|$ as the amplitude of the envelope composed of noise and interference.

Thus, the symbol error ratio is as follows:

$$P_{e|k} = P_r[max_{i,i\neq k}(\delta_i) > |\sqrt{E_S} + \phi_k + I_K|]. \tag{8}$$

Here ϕ_k represents the plural Gaussian white noise, and the plural noise envelope is $\delta_i = |\phi_i|$. δ_i satisfies Rayleigh distribution, and the cumulative distribution function is as follows:

$$F_{\delta_i}(\delta) = 1 - exp\left[-\frac{\delta^2}{2\sigma^2}\right]. \tag{9}$$

Here $\sigma^2 = \frac{N_0}{2}$ is noise power spectral density. Thus, the bit error ratio of the transmitting symbol "k" is as follows:

$$P(\widehat{k} \neq k|k) = Pr[max_{i,i\neq k}(\delta_i) > \mu_k]. \tag{10}$$

$\mu_k = |\sqrt{E_S} + \phi_k|$ follows Rician distribution $f(\mu_k|\sqrt{E_S}, \sigma)$:

$$f_{\mu_k}(y) = \frac{y}{\sigma^2} exp\left[-\frac{(y^2 + E_S)}{2\sigma^2}\right] I_0\left(\frac{y\sqrt{E_S}}{\sigma^2}\right). \tag{11}$$

The cumulative distribution function of $max_{i,i\neq k}$ is as follows:

$$F_{max_{i,i\neq k}}(x) = \left[1 - exp\left[-\frac{x^2}{2\sigma^2}\right]\right]^{2^{SF}-1}. \tag{12}$$

Thus, we obtain the symbol error ratio in Lemma 4.1. □

REFERENCES

[1] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. 2020. A cloud-optimized link layer for low-power wide-area networks. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 247–259.

[2] Binbin Chen, Ziling Zhou, Yuda Zhao, and Haifeng Yu. 2010. Efficient error estimating coding: Feasibility and applications. *ACM SIGCOMM Comput. Commun. Rev.* 40, 4 (2010), 3–14.

[3] Ulysse Coutaud, Martin Heusse, and Bernard Tourancheau. 2020. Fragmentation and forward error correction for LoRaWAN small MTU networks. In *Proceedings of the International Conference on Embedded Wireless Systems and Networks*. 289–294.

[4] Xianjun Deng, Yuan Tian, Lingzhi Yi, Laurence T. Yang, Yunzhi Xia, Xiao Tang, and Chenlu Zhu. 2022. Resilient deployment of smart nodes for improving confident information coverage in 5G IoT. *ACM Trans. Sens. Netw.* 18, 3 (2022).

[5] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'18)*. IEEE, 60–71.

[6] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 309–321.

[7] Xiaoxuan Fan, Xianjun Deng, Yunzhi Xia, Lingzhi Yi, Laurence T. Yang, and Chenlu Zhu. 2023. Tensor-based confident information coverage reliability of mobile Internet of Things. *IEEE Trans. Mobile Comput.* (2023), 1–15.

[8] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. 2023. Lmac: Efficient carrier-sense multiple access for lora. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* 19, 2 (2023), 1–13.

[9] Wei Gong, Longzhi Yuan, Qiwei Wang, and Jia Zhao. 2020. Multiprotocol backscatter for personal IoT sensors. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*. 261–273.

[10] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. 2019. NetScatter: Enabling large-scale backscatter networks. In *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI'19)*. 271–284.

[11] Bin Hu, Zhimeng Yin, Shuai Wang, Zhuqing Xu, and Tian He. 2020. SCLoRa: Leveraging multi-dimensionality in decoding collided LoRa transmissions. In *Proceedings of the IEEE 28th International Conference on Network Protocols (ICNP'20)*. IEEE, 1–11.

[12] Wenchao Jiang, Feng Li, Luoyu Mei, Ruofeng Liu, and Shuai Wang. 2022. VisBLE: Vision-enhanced BLE device tracking. In *Proceedings of the 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON'22)*. 217–225. https://doi.org/10.1109/SECON55815.2022.9918581

[13] Wenchao Jiang, Zhimeng Yin, Ruofeng Liu, Zhijun Li, Song Min Kim, and Tian He. 2019. Boosting the bitrate of cross-technology communication on commodity IoT devices. *IEEE/ACM Trans. Network.* 27, 3 (2019), 1069–1083. https://doi.org/10.1109/TNET.2019.2913980

[14] Chenning Li and Zhichao Cao. 2022. Lora networking techniques for large-scale and long-term iot: A down-to-top survey. *ACM Comput. Surv.* 55, 3 (2022), 1–36.

[15] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. 2021. NELoRa: Towards ultra-low SNR LoRa communication with neural-enhanced demodulation. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 56–68.

[16] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Trans. Sens. Netw.* 15, 2 (2019), 1–35.

[17] Ruofeng Liu, Zhimeng Yin, Wenchao Jiang, and Tian He. 2019. LTE2B: Time-domain cross-technology emulation under LTE constraints. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. https://doi.org/10.1145/3356250.3360022

[18] Ruofeng Liu, Zhimeng Yin, Wenchao Jiang, and Tian He. 2020. XFi: Cross-technology IoT data collection via commodity WiFi. In *Proceedings of the IEEE 28th International Conference on Network Protocols (ICNP'20)*, 1–11. https://doi.org/10.1109/ICNP49622.2020.9259363

[19] Travis Mandel and Jens Mache. 2013. Practical error correction for resource-constrained wireless networks: Unlocking the full power of the CRC. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. 1–14.

[20] Paul Marcelis, Nikolaos Kouvelas, Vijay S. Rao, and Venkatesha Prasad. 2020. DaRe: Data recovery through application layer coding for LoRaWAN. *IEEE Trans. Mobile Comput.* (2020).

[21] Luoyu Mei, Zhimeng Yin, Xiaolei Zhou, Shuai Wang, and Kai Sun. 2021. ECCR: Edge-cloud collaborative recovery for low-power wide-area networks interference mitigation. In *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 494–507.

[22] Congduc Pham. 2016. QoS for long-range wireless sensors under duty-cycle regulations with shared activity time usage. *ACM Trans. Sens. Netw.* 12, 4 (2016), 1–31.
[23] Peiyuan Qin, Luoyu Mei, Qi Jing, Shuai Wang, Zhimeng Yin, and Xiaolei Zhou. 2022. Edge-cloud collaborative interference mitigation with fuzzy detection recovery for LPWANs. In *Proceedings of the IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD'22)*. 792–797. https://doi.org/10.1109/CSCWD54268.2022.9776037
[24] Semtech. Retrieved May 18, 2023 from https://www.semtech.com/products/wireless-rf/lora-core/.
[25] Semtech. *SX1272/3/6/7/8: LoRa Modem Designer's Guide.*
[26] Junyang Shi, Di Mu, and Mo Sha. 2021. Enabling cross-technology communication from LoRa to ZigBee via payload encoding in sub-1 GHz bands. *ACM Trans. Sens. Netw.* 18, 1 (2021), 1–26.
[27] Sigfox. Retrieved May 18, 2023 from https://www.sigfox.com/.
[28] Kai Sun, Zhimeng Yin, Weiwei Chen, Shuai Wang, Zeyu Zhang, and Tian He. 2021. Partial symbol recovery for interference resilience in low-power wide area networks. In *Proceedings of the IEEE 29th International Conference on Network Protocols (ICNP'21)*. IEEE, 1–11.
[29] Shuai Tong, Zilin Shen, Yunhao Liu, and Jiliang Wang. 2021. Combating link dynamics for reliable lora connection in urban settings. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 642–655.
[30] Weizheng Wang, Hao Xu, Mamoun Alazab, Thippa Reddy Gadekallu, Zhaoyang Han, and Chunhua Su. 2022. Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Trans. Industr. Inf.* 18, 10 (2022), 7059–7067. https://doi.org/10.1109/TII.2021.3084753
[31] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. 2019. mlora: A multi-packet reception protocol in lora networks. In *Proceedings of the IEEE 27th International Conference on Network Protocols (ICNP'19)*. IEEE, 1–11.
[32] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: Parallel decoding for LoRa transmissions. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 192–204.
[33] Yunzhi Xia, Xianjun Deng, Lingzhi Yi, Laurence T. Yang, Xiao Tang, Chenlu Zhu, and Zhongping Tian. 2022. AI-driven and MEC-empowered confident information coverage hole recovery in 6G-Enabled IoT. *IEEE Trans. Netw. Sci. Eng.* 10, 3 (2022), 1256–1269.
[34] Binbin Xie and Jie Xiong. 2020. Combating interference for long range LoRa sensing. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 69–81.
[35] Zhuqing Xu, Junzhou Luo, Zhimeng Yin, Tian He, and Fang Dong. 2020. S-MAC: Achieving high scalability via adaptive scheduling in LPWAN. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'20)*. IEEE, 506–515.
[36] Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. 2021. Pyramid: Real-time LoRa collision decoding with peak tracking. In *Proceedings of the IEEE Conference on Computer Communications*. 1–9.
[37] Yifan Yang, Longzhi Yuan, Jia Zhao, and Wei Gong. 2022. Content-agnostic backscatter from thin air. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys'22)*. Association for Computing Machinery, New York, NY, 343–356. https://doi.org/10.1145/3498361.3538930
[38] Longzhi Yuan and Wei Gong. 2022. SubScatter: Sub-symbol WiFi backscatter for high throughput. In *Proceedings of the IEEE 30th International Conference on Network Protocols (ICNP'22)*. 1–11. https://doi.org/10.1109/ICNP55882.2022.9940419
[39] Zeyu Zhang, Weiwei Chen, Junwen Wang, Shuai Wang, and Tian He. 2022. CONST: Exploiting spatial-temporal correlation for multi-gateway based reliable LoRa reception. In *Proceedings of the IEEE 30th International Conference on Network Protocols (ICNP'22)*. 1–11. https://doi.org/10.1109/ICNP55882.2022.9940424
[40] Jia Zhao, Wei Gong, and Jiangchuan Liu. 2018. X-Tandem: Towards multi-hop backscatter communication with commodity WiFi. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 497–511.

Received 6 July 2022; revised 1 June 2023; accepted 8 June 2023