Ad Hoc Networks 147 (2023) 103203

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

ReLoRaWAN: Reliable data delivery in LoRaWAN networks with multiple gateways

Wenjia Wu [a,*], Hao Wang [b], Zisheng Cheng [a]

[a] School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China
[b] Southeast University-Monash University Joint Graduate School, Southeast University, Suzhou, 215123, China

ARTICLE INFO

Keywords:
LPWAN
LoRa
LoRaWAN
Multi-packet reception
Data recovery

ABSTRACT

LoRa has emerged as a promising technology to provide low-power and long-range communication for IoT devices. LoRaWAN networks build on top of the LoRa physical layer and adopt a centralized network architecture that supports one or more gateways connected with a large number of end devices. Although LoRa communication is resistant to channel noises, it still frequently suffers data delivery failures in LoRaWAN networks due to packet collisions of the concurrent transmissions. The delivery failures trigger the retransmission procedure in LoRaWAN protocol, which causes additional power consumption on end devices and further exacerbates packet collisions. To solve this issue, we investigate the reliable data delivery mechanism that utilizes with the packet reception of multiple gateways to recover the distorted payload. Firstly, we present a ReLoRaWAN framework, where the central server aggregates distorted packet payload from multi-gateway reception of the same packet and executes the corresponding data recovery operations. Then, we design a Tri-operation Integrated Data Recovery (TIDR) algorithm for recovering the distorted packet payload, which involves exclusive-OR based bitwise inversion operation, majority voting based bitwise inversion operation, and weighted bitwise decision operation. Finally, we implement a ReLoRaWAN testbed and conduct real-world experiments to evaluate the performance of our solution. Compared with existing works, the ReLoRaWAN greatly optimizes the quality of service and power consumption in the network. It improves the packet delivery ratio by 35% and reduces the average power consumption of the end device by 30%.

1. Introduction

Low-power wide-area networks (LPWANs) are emerging networking technologies that connect tens of billions of devices for large-scale Internet of Things (IoT) paradigms [1]. The promising LPWANs provide wide area connectivity with low-cost energy consumption and low data rate [2], e.g., LoRa, Sigfox, NB-IoT, Weightless, and LTE-M. It can even reach up to tens of kilometers at a few tenths of Kbps. According to market research, there will be hundreds of billions of LPWAN devices in the following decades around the world.

LoRa is a leading LPWAN technique, where a single battery-powered end device can provide communication for several years [3]. Different from other LPWAN technologies, LoRa works at sub-GHz license-free shared industrial scientific medical (ISM) bands, and it has been widely applied in smart cities, smart parks, buildings monitoring, smart security, and other fields [4].

Based on chirp spreading spectrum (CSS) modulation technology, LoRa is inherently robust to ambient noises. The characteristics of high reliability seem to work very well. Nevertheless, LoRa transmission is inevitably affected by several kinds of interference from concurrency transmission at ISM bands. Besides, in dynamic multi-floor or multi-room indoor scenarios, the presence of moving individuals, interferences from jammers, and other ambient noises in non-line-of-sight (NLOS) propagation can induce shadow fading of signals, resulting in performance degradation and worse communication quality even in a short area range [5–7]. Such an indoor environment also causes various kinds of collisions, and all the packets may fail the cyclic redundancy check (CRC) verification. The packets with erroneous physical payload (PHYPayload) are abandoned. From recent works [8,9], the redundant packet reception from multiple gateways benefits network transmission with macro diversity gains. In this context, industry and academia have proposed many collaborative approaches for reliable LoRa transmission in the past few years [10]. For instance, Charm [11] leverages multiple gateways to jointly decode weak signals in the cloud to combat carrier frequency offset. FDR [12] relies on self-designed error detection codes

* Corresponding author.
E-mail address: wjwu@seu.edu.cn (W. Wu).

https://doi.org/10.1016/j.adhoc.2023.103203
Received 16 January 2023; Received in revised form 24 April 2023; Accepted 8 May 2023
Available online 15 May 2023

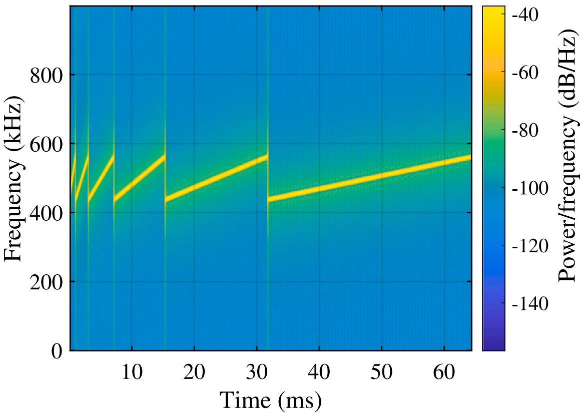


Fig. 1. Spreading factors in LoRa modulation (7 to 12).

for data recovery through multiple gateway cloud–edge collaborative mitigation. WIP [13] gets rid of the frame erasure by appending the bitwise Exclusive-OR (XOR) value of the last conveyed frame to the data frame. eLoRa [14] cooperatively uses LT code to recover lost data frames in multiple gateway environments. MIMO-LoRa [15] introduces precoding into a multiple-input multiple-output (MIMO) system. BiDiCom [16] realizes a weighted combining of the same copy from multiple gateways for interference cancellation. However, they are not comprehensive enough to solve this problem and have many fundamental limitations. For example, Charm works at the physical layer and needs software-defined radio (SDR) processing for not fine-grained LoRa physical features. MIMO-LoRa, WIP, FDR, and eLoRa append extra data bits to the payload of the packet.

Recently a novel link layer method OPR [17] performs error control by utilizing the traditional wireless network error control methods, such as the SPaC [18] and majority logic decoding in a pure software implementation. Unlike traditional application layer methods, this method does not adopt encoding strategies and therefore does not require additional transmission redundancy. However, there are still some defects in OPR. Firstly, the error correction mechanism is relatively simple, thus the performance is not good in the case of severe interference environments like multi-floor or multi-room sceneries. Secondly, the energy saving effect is not mentioned or discussed, which has great significance for the IoTs. Besides, it is not fine enough for OPR to take a benchmark to determine the hardcoded limit of error correction capability because it depends on hardware processing capability.

LoRaWAN defines the MAC layer that operates on top of the LoRa PHY layer. The concurrent transmissions in LoRaWAN with pure Aloha access protocol intrinsically interfere with each other [19]. According to LoRaWAN specification, LoRaWAN relies on the acknowledgment (ACK) mechanism to realize reliable transmission. However, when there are too many lost packets, the retransmission procedure leads to more energy consumption for end devices.

To address these challenges, we propose a reliable data delivery mechanism based on a multi-operation error correction algorithm in LoRaWAN networks, called ReLoRaWAN, which aims to recover the distorted data through the received packets of multiple gateways. It takes real end-to-end delay and the cloud processing latency into consideration to get a flexible processing time upper constraint. The real-world experiment results also verify the energy-saving improvements. We mainly focus on the distorted packets where the preamble is well received but the PHYPayload is corrupt, that is, the PHYPayload is unable to pass CRC verification. Most of these transmissions are unable to decode online via any reception of individual gateway. Motivated by existing coherent decoding paradigms, the key insight of our work is coordinating distributed gateways to make error corrections. It is a novel optimization that ReLoRaWAN introduces much less redundant data in transmission. Due to the solely software-based implementation that works above the physical layer, the raw modulation and demodulation details are isolated from ReLoRaWAN. As a result, the ReLoRaWAN mechanism is compliant with standard LoRaWAN network protocol and can be implemented with commercial off-the-shelf (COTS) gateways and end devices. Meanwhile, we reduce the average power consumption of LoRa devices without prohibitively high SDR hardware costs.

The contributions of this paper can be summarized as follows.

- We present the ReLoRaWAN framework to reduce packet retransmissions through data recovery, where the central server aggregates the distorted packet PHYPayload copies from multi-gateway reception of the same packet and executes the corresponding data recovery operations.
- We design a Tri-operation Integrated Data Recovery (TIDR) algorithm for recovering the distorted packet PHYPayload, which involves exclusive-OR based bitwise inversion operation, majority voting based bitwise inversion operation, and weighted bitwise decision operation. The three bitwise operations are executed in sequence to improve the probability of successful recovery.
- We implement a ReLoRaWAN testbed with COTS gateways and end devices, and conduct real-world experiments to evaluate the performance of our solution. The results demonstrate that the packet delivery ratio (PDR) of ReLoRaWAN has increased to 1.35 times of the existing method OPR and the power consumption of the end device is reduced by 30%.

2. Preliminaries

In this section, we first present the modulation and demodulation of LoRa and then introduce the LoRaWAN protocol.

2.1. LoRa modulation and demodulation

LoRa is a spread spectrum modulation scheme, which employs the CSS technique and modulates data bits to up-chirps and down-chirps.

Modulation. There are several parameters characterizing LoRa modulation, i.e., spreading factor (SF), transmission power (TP), coding rate (CR), and bandwidth (BW). The frequency of an up-chirp sweep linearly from $-\frac{BW}{2}$ to $\frac{BW}{2}$ with time, which is also a base chirp. When modulation, a symbol is modulated into 2^{SF} chips, where a chip is data sent per second per Hz of bandwidth BW. The higher the SF is, the higher the sensitivity, which contributes to robustness. Nevertheless, the increased data drains the battery violently. The data rate is inversely proportional to SF, so the time on air (TOA) also increases with the SF. When taking a higher SF to compensate for interference, there is an increased collision possibility in severe interference environments like multi-floor or multi-room sceneries. Fig. 1 shows a spectrogram of LoRa up-chirps in Phase/in Quadrature (I/Q) to modulate several physical symbols with different SF. According to prior works, transmissions with the same SF may interfere with each other.

Demodulation. When the receiver carries out the demodulation, it multiplies the base chirp with a down-chirp and performs a fast fourier transform (FFT) operation on the result. The energy focuses on the unique maximal peak of FFT bins and this one is identified as the demodulated symbol (chirp). However, the multiple peaks resulting from collisions may cause the failure of demodulation.

Physical Packet Structure. Fig. 2 shows the structure of an uplink physical packet from the transceiver datasheet.[1] The CRC field detects the errors of the PHYPayload with the CCITT-16 CRC algorithm.

[1] SX1276/77/78/79–137 MHz to 1020 MHz Low Power Long Range Transceiver, https://www.semtech.com/products/wireless-rf/lora-connect/sx1276

Preamble	Sync	PHDR	PHYPayload	CRC

Fig. 2. LoRa physical packet structure.

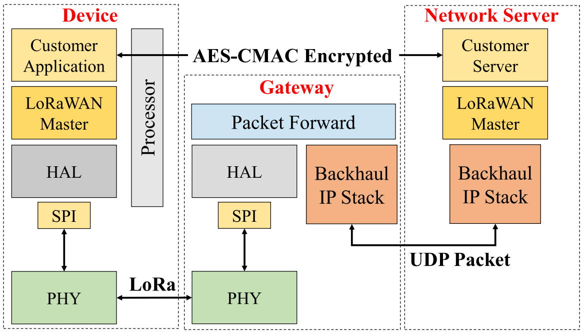


Fig. 3. LoRaWAN stack.

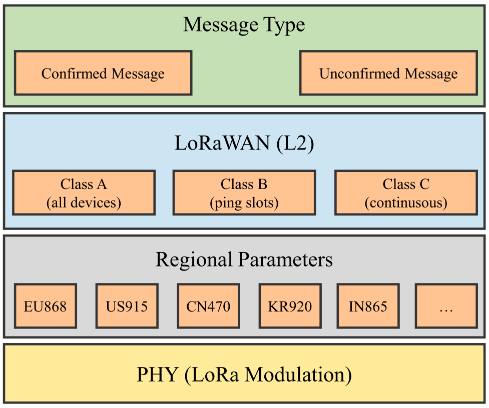


Fig. 4. Device configuration.

2.2. LoRaWAN

LoRaWAN is an Aloha-like MAC and network layer protocol upon LoRa, which is quite simple and designed for energy-constrained devices. It is proposed by LoRa Alliance, and the International Telecommunication Union (ITU) recently recognizes it as the official LPWAN communication standard, i.e., Recommendation ITU-T Y.4480. Due to its open-source character, LoRaWAN-certified devices are spread all over the world.

LoRaWAN Architecture. A typical LoRaWAN network is a star-of-stars topology, and the single-hop structure applies to resource-limited devices. The devices are not associated with any specific gateway but are assigned to the central network server. The gateways just forward UDP (User Datagram Protocol) packets to the upper server through standard Internet Protocol (IP) backhaul, so they are transparent to devices. When the network server receives multiple copies from the same device with multiple gateways, it performs deduplication based on the signal quality. Fig. 3 indicates the standard LoRaWAN design.

LoRaWAN Device. The configuration of a LoRaWAN device is shown in Fig. 4. The traffic in LoRaWAN is always asymmetric, so the focus of this work is dominant uplink communication. The packets from the end device are either confirmed or unconfirmed. When the device adopts a confirmed message, it requests a downlink ACK from the LoRaWAN network server for reliable communication. More specifically, the network server chooses the gateway with the highest signal-to-noise ratio (SNR) to send ACK. The device would retransmit several times after a random back-off time if no ACK is received, which is referred to as the retransmission procedure. This process resembles an improved type-I hybrid automatic repeat request (Type-I HARQ) mode. It increases power consumption dramatically and renders network congestion. The devices can be categorized into the following 3 types according to downlink schedule styles:

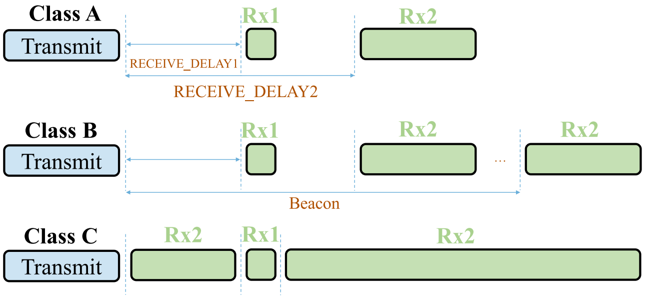
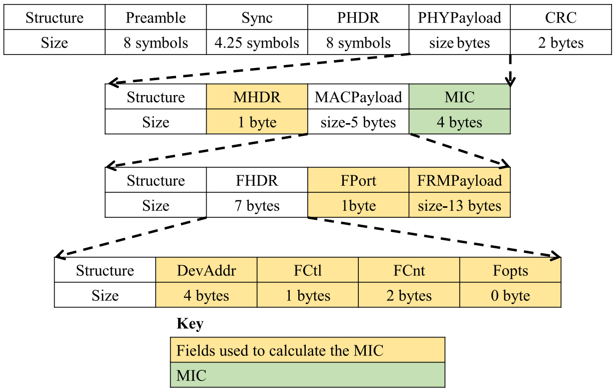


Fig. 5. Taxonomy of LoRaWAN device.



Structure	Preamble	Sync	PHDR	PHYPayload	CRC
Size	8 symbols	4.25 symbols	8 symbols	size bytes	2 bytes

Structure	MHDR	MACPayload	MIC
Size	1 byte	size-5 bytes	4 bytes

Structure	FHDR	FPort	FRMPayload
Size	7 bytes	1byte	size-13 bytes

Structure	DevAddr	FCtl	FCnt	Fopts
Size	4 bytes	1 bytes	2 bytes	0 byte

Fig. 6. LoRaWAN MAC message format.

- *Class A device*: It sleeps most of the time. After sending an uplink packet, it opens a first reception window **RX1** for a short duration of RECEIVE_DELAY1. If no downlink is received, it will open another slot **RX2**. In this work, we only consider *class A* device for the lowest power consumption.
- *Class B device*: It opens additional reception windows based on *the class A device*, which is synchronized by coordination beacons of the network server forwarded by the gateway.
- *Class C device*: It always opens the receive window unless sending uplink data packets.

Fig. 5 indicates different work models of the devices.

MAC Message Format. Fig. 6 shows the MAC message format that makes up of PHYPayload according to the LoRaWAN specification.[2]

The Message Integrity Check (MIC) verification utilizes a Cipher-based Message Authentication Code with Advanced Encryption Standard (AES-CMAC) encryption to verify the integrity of the entire PHYPayload. The FRMPayload (frame payload field) is the application layer payload for integration with external services, which is a part of PHYPayload.

3. ReLoRaWAN design

In this section, we present the network structure of ReLoRaWAN and design its overall framework.

[2] LoRaWAN® Specification v1.0.3, https://lora-alliance.org/resource_hub/lorawan-specification-v1-0-3/

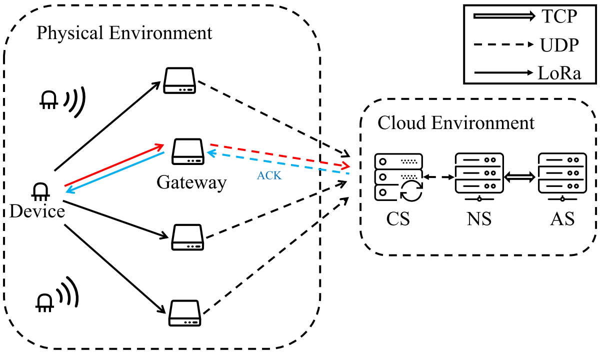


Fig. 7. Network structure.

3.1. Network structure

ReLoRaWAN is optimized for the typical LoRaWAN network architecture that involves two parts, i.e, the physical environment and the cloud environment, as shown in Fig. 7.

In the physical environment, we design a single busy LoRaWAN cell, consisting of N homogeneous gateways and M static end devices in the edge deployment. Given the predominant LoRaWAN indoor deployment of environmental sensors that are inherently static, there is no pressing need to account for mobility. Similar to OPR, our work is only designed for general LoRaWAN communication, without special optimization for mobile nodes during data recovery. In the uplink, when a device transmits a packet, the copies of this packet are obtained at all gateways located within the range of coverage. The coverage area of the end device is large enough that it can broadcast signals to nearby gateways through different wireless branches. As for the propagation model, these branches between end devices and gateways are time-varying and non-stationary, so packets are easily corrupted. Besides, the packets also interfere with each other in this dense network. When transmitting data packets in parallel across channels that are independent and identically distributed in this multi-packet reception system, these packets have nearly equal probabilities of being corrupted. To make matters worse, in the harsh multi-path propagation from the end device to the gateway, the burst errors are not evenly spread throughout the copies but disjoint across receivers even when subject to the same interference source respectively.

In the cloud environment, there are three server entities: central server (CS), LoRaWAN network server (NS), and LoRaWAN application server (AS). Firstly, the CS integrated with the TIDR algorithm performs error correction on the PHYPayload of the corrupt packets. Then, the NS checks the MIC of the recovered PHYPayload with the network session key (NwkSKey) for validity. In the end, the AS utilizes the application session key (AppSKey) to decrypt FRMPayload from PHYPayload.

Table 1 lists all notations used in this paper.

3.2. Overall framework

Fig. 8 shows the ReLoRaWAN framework. The LoRa packet physical structure is transparent to the ReLoRaWAN because the TIDR algorithm only works on the bit value of UDP packets at the application layer. In the implementation of the packet forwarder, upon receiving a packet from the end device, the gateway serializes the upstream JSON data structure defined in the protocol[3] to format the embedded data field and associated metadata through the driver. Every serialized UDP packet has fixed metadata as identifiers whose characteristics are multi-dimensional, e.g., UTC time, GPS time, gateway internal timestamp, CRC value, size, etc. If the gateway drops the packet that cannot pass the CRC verification like the traditional LoRaWAN communication process, there is no PUSH_ACK packet in downlink traffic because NS does not receive packets correctly, which results in retransmissions of end devices. To avoid this, our solution like cooperative communication is that all the gateways have to forward the packet copies to CS from different branches, whether they can pass the CRC verification or not. The gateways do not classify which data packets belong to an identical device. They forward the packets received to CS by the packet forwarder without further judgment. Then, the packets are aggregated and proceeded by the TIDR algorithm. Notwithstanding the increased uplink end-to-end delay resulting from the additional forward transmission process, a judicious trade-off may be achieved between the aforementioned delay and an improved transmission success rate engendered by effective data recovery, which is discussed later.

Table 1
Summary of notations.

Notations	Description
N	Total number of gateways
M	Total number of end devices
δ	Duty cycle
T_g	Packet transmission interval
λ	Packet arrival rate
$L_{payload}$	PHYPayload size in bits
size	PHYPayload size in bytes
X	Hamming weight
ω	Weight of copy
σ	Set of SNR values
P_r	PHYPayload from different copies
P_c	Candidate PHYPayload
C	Confidence information of copies
S_v	Soft value
P_p	Proportion of occurrences of specific bit
P_e	Estimated correctly decoded PHYPayload
b_p	Bitmap
b_k	Bitmask
$\mathrm{FCS}_{\mathrm{CRC}}$	CRC value
$\mathrm{FCS}_{\mathrm{MIC}}$	MIC value
FCnt	Frame counter
DevAddr	Device address

We divide the cloud environment in the system into three virtual layers: the data layer, the execution layer, and the output layer. Firstly, the data layer is responsible for data aggregation and prepossessing. Secondly, the execution layer runs the data recovery algorithm for error control. At last, the output layer has validity verification and acknowledgment modules.

In the data layer, CS first stores all packet copies in memory and parses the serialized JSON data structures to extract data fields and metadata. Then it performs data aggregation. The idea is that only after CS receives and aggregates the same number of distorted packet copies as the gateway at the same time, can it determine that all the packet copies in the same transmission received by different gateways from the identical device cannot pass the CRC verification and need to be recovered. Then CS calculates frame counter (FCnt), device address (DevAddr), and MIC value based on the received PHYPayload and metadata. Secondly, CS carries out the data preprocessing. It classifies and filters out coexisting packets with different metadata values. CS only performs error control on the packet copies from the same transmission of an identical device, So it checks the DevAddr of packets to make sure they are at least from an identical device and checks the UTC time, GPS time, size, FCnt, and CRC value of the packets to make sure they are from the same transmission. It relays all packet copies to the NS directly if there exists at least one error-free. Otherwise, all the error packet copies are redirected to the execution layer. As for the

[3] Basic communication protocol between LoRa gateway and Network Server, https://github.com/Lora-net/sx1302_hal/blob/master/packet_forwarder/PROTOCOL.md#4-upstream-json-data-structure

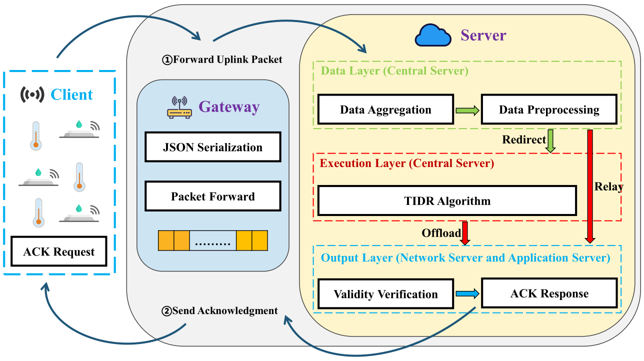


Fig. 8. ReLoRaWAN framework.

buffer management strategy, it is important to avoid buffer overflow by clearing the stored packet copies in the memory buffer after every call to the algorithm.

In the execution layer, CS collaboratively merges all the aggregated packet copies and executes the TIDR algorithm to finish joint decoding efficiently. The TIDR algorithm has to process the data format during the runtime for further bitwise operations. Firstly, it performs a base64-decoding of the data field to get the hex format PHYPayload byte codewords before processing. Then the PHYPayload bytes should be converted to bits in binary representation when the algorithm performs error correction. Finally, the Frame Check Sequence (FCS) verification needs the binary PHYPayload bits to reconvert to hex format bytes. If and only if the PHYPayload is restored, the CS will replace the original PHYPayload with the correctly decoded one, offload it to NS eventually, and record packet statistics.

In the output layer, the NS discards the received packets that fail the MIC verification. Then we calculate the real statistics at the AS to verify the validity of the algorithm. The acknowledge module works when it receives a correct packet.

4. TIDR algorithm

To correct bit errors in distorted packets, we design the TIDR algorithm.

4.1. Design overview

The algorithm requires at most three bitwise operations to finish error correction opportunistically. It practically leverages the multiple reception gain of receivers to reduce the number of device retransmissions with the sacrifice of cloud computing overhead. The goal is to provide bit-level error control. Innovatively, the TIDR algorithm reuses traditional error-detection checksum to generate the estimated correctly decoded PHYPayload based on two generator polynomials for CRC and MIC. If this PHYPayload is generated, the TIDR algorithm is announced to be valid finally. To see how it works, Fig. 9 depicts the detailed workflow of the algorithm.

Firstly, TIDR obtains copies of all distorted packets that have been redirected from the data layer and aggregates them. Then the SNR based packet selection pretreatment, exclusive-OR based bitwise inversion operation (EO), majority voting based bitwise inversion operation (MO), and weighted bitwise decision operation (WO) for different tasks together constitute the TIDR algorithm, which provides great flexibility. As for the initial pretreatment, it yields a candidate packet for the later error detection and correction processing of EO and MO. When it comes to the EO and MO, they continue to finish the major calculation to recover the candidate packet till an arbitrarily estimated repaired PHYPayload is available. The goal of WO is to make a final effort: it takes advantage of the ordered weighted averaging (OWA) operator based on different levels of physical layer information to calculate the estimated result.

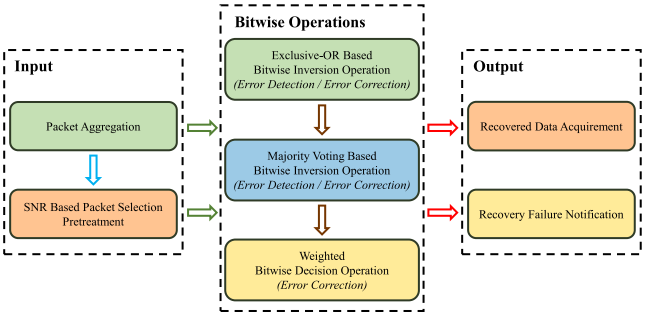

Input
Packet Aggregation
SNR Based Packet Selection Pretreatment
Bitwise Operations
Exclusive-OR Based Bitwise Inversion Operation
(Error Detection / Error Correction)
Majority Voting Based Bitwise Inversion Operation
(Error Detection / Error Correction)
Weighted Bitwise Decision Operation
(Error Correction)
Output
Recovered Data Acquirement
Recovery Failure Notification

Fig. 9. TIDR workflow.

In the first operation, the EO performs some bitwise modulo 2 sum calculation [20] for error detection and correction. As for the MO, it is based on the majority voting rule [21] for error detection and correction. In the last operation, WO is a cross-layer two-dimensional design that exploits soft decision decoding [22] for error correction.

4.2. Three bitwise operations

This algorithm consists of one pretreatment with three bitwise operations. Firstly, it performs pretreatment for further process. Then it carries out EO, MO, and WO in sequence.

4.2.1. SNR based packet selection pretreatment

The pretreatment is one of the simplest ways to benefit from multiple receptions.

Before decoding, the pretreatment determines the most robust signal because this one is likely to suffer less error. It selects the copy with maximal SNR as output to minimize the bit error rate. This output is regarded as P_c to be correctly decoded in further operations EO and MO.

Weighted by SNR values, the jth bit of candidate PHYPayload P_c can be formulated by the OWA operator as:

$$P_{c_j} = \sum_{i=1}^{N} \omega_i * P_{r_{i,j}} \tag{1}$$

where $P_{r_{i,j}}$ represents the jth bit of ith aggregated PHYPayload copy and the ω_i is the weight coefficient of this copy. The weight coefficient is uneven: the copy with the highest SNR value is 1 while the others are 0.

Similar to the capture effect, the procedure can also be simplified to find the packet with the highest SNR value. Firstly, we can locate the packet as follows:

$$i = \arg\max_{i} \sigma_{(i)} \tag{2}$$

where $\sigma_{(i)}$ is the SNR value of ith buffered PHYPayload copy and it is considered as a function of i.

Then copy the located packet to the candidate packet as follows:

$$P_{c_j} = P_{r_{i,j}} \tag{3}$$

4.2.2. Exclusive-OR based bitwise inversion operation

To recover P_c, EO performs a bitwise logical operation on the aggregated PHYPayload copies to generate a bitmap, which identifies the assumed error bit location of P_c for error detection.

The jth bit of bitmap can be formed as:

$$b_{p_j} = \underbrace{P_{r_{1,j}} \oplus P_{r_{2,j}} \oplus \dots \oplus P_{r_{N-1,j}} \oplus P_{r_{N,j}}}_{N} \tag{4}$$

Position with a bit value 1 on the bitmap b_p reveals to some extent that the bit of P_c at this location may be erroneous because some PHYPayload copies differ from each other at this common bit location. However, if there is a bit with a value of 0, the error at this bit location can be overlapping and undetected. This phenomenon is termed a hidden error.

There are X nonzero bits in the bitmap, which is referred to as hamming weight and can be defined as follows:

$$X = \sum_{j=1}^{L_{payload}} b_{p_j} \tag{5}$$

where $L_{payload}$ is the bit length of the PHYPayload.

Then it utilizes the improved search mechanism to yield multiple bitmasks exponentially by bit-by-bit inversion according to the bitmap for error correction, that is, the bitmask is gradually generated as the hamming weight increases from 0 to X. Each bitmask corresponds to a 'trial' error pattern, and b_k is the set of all $2^X - 1$ bitmasks, which is represented as follows:

$$b_k = \left.\begin{matrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ & & \vdots & & & & \\ 1 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 1 & 0 & 1 \\ & & \vdots & & & & \\ 1 & 1 & 1 & \cdots & 1 & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 & 1 \end{matrix}\right\} 2^X - 1 \tag{6}$$

Then EO combines the bitmask with bitwise logical operations to indirectly modify and replace the potential error bits of P_c. The modification process can be modeled as follows:

$$P_{e_{i,j}} = P_{c_j} \oplus b_{k_{i,j}} \tag{7}$$

Where $b_{k_{i,j}}$ denotes the jth bit of the ith bitmask.

The traditional wireless network error control algorithms need to repeat this procedure till all $2^X - 1$ permutations have been probed. If the worst comes to the worst, the false positives phenomenon occurs, which introduces great computation overhead. The term refers to a scenario where the algorithm finds more than one correctly decoded P_e that passes the FCS verification like the CRC verification but is not the original PHYPayload because of the hash collision effect.

As a result, unlike the above brute-force way, it introduces the greedy conditional stop mechanism to reduce the computation overhead: as soon as there is a successful retrieval at any attempt to find a P_e, EO aborts. The criteria is represented as follows:

$$(CRC(P_e) \oplus \mathrm{FCS}_{\mathrm{CRC}} = 0) \wedge (MIC(P_e) \oplus \mathrm{FCS}_{\mathrm{MIC}} = 0) \tag{8}$$

Compared to the brute-force method with $2^X - 1$ iterations, the total number of iterations required in EO with the conditional stop mechanism is considerably diminished, with only a marginal increase in the likelihood of false positives. This trade-off is deemed acceptable and justifiable. If no P_e generates, it is declared as a failure, and the MO is triggered.

4.2.3. Majority voting based bitwise inversion operation

If EO fails, the MO will carry out the more powerful decoding based on a bit-by-bit majority voting operation. The idea is that the bit value of P_e should be consistent with the most bit values of all the aggregated PHYPayload copies at the same location. To avoid a tie, the situation of equal shares is considered.

Firstly, it performs error correction directly by modeling the jth bit a newly built P_e as:

$$P_{e_j} = \begin{cases} 0, & \sum_{i=1}^{N} P_{r_{i,j}} < \lfloor N/2 \rfloor \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

Then the new P_e is verified, if it cannot satisfy the FCS verification, subsequent error detection and correction steps are required.

More specifically, MO creates another b_p by bitwise threshold judgment for error detection. When not all PHYPayload copies have the same value in a bit location, the bit value of the corresponding location in this new b_p is recorded as 1, and the bit value of P_c at this location is considered unreliable. In comparison with EO, MO maximizes the hamming weight of new b_p to reduce hidden error possibilities at the expense of higher computational complexity.

The jth bit of bitmap can be calculated as follows:

$$b_{p_j} = \begin{cases} 0, & (\sum_{i=1}^{N} P_{r_{i,j}} = 0) \vee (\sum_{i=1}^{N} P_{r_{i,j}} = N) \\ 1, & \text{otherwise} \end{cases} \tag{10}$$

The remaining error correction procedure of MO is the same as that of EO: the improved search mechanism is provided in conjunction with the conditional stop mechanism to get a new P_e that passes FCS verification by recovering P_c. If no P_e is generated, MO fails and the WO is called.

4.2.4. Weighted bitwise decision operation

In the WO, it maps bit values by comparing confidence information with a threshold for error correction.

Firstly, the soft value at jth bit of the ith buffered PHYPayload copy is calculated based on physical layer hint SNR as follows:

$$S_{v_{i,j}} = \begin{cases} \sigma_i, & P_{r_{i,j}} = 1 \\ -\sigma_i, & P_{r_{i,j}} = 0 \end{cases} \tag{11}$$

In conventional wireless network error control methods, the next steps typically involve mapping bit values by comparing soft values with a predefined threshold or selecting suboptimal majority voting schemes. However, they are not fine-grained enough. On the contrary, WO carries out the OWA operator on soft values, which are jointly weighted according to the occurrence probability of bits to get the confidence value. Then it compares the confidence value with a threshold to get the bit value.

WO exhibits unequal treatment of original bits based on their relative occurrence in copies, under the assumption that the probability of encountering either 0 or 1 in a given bit is equal to the probability of encountering the corresponding value in the same bit across all PHYPayload copies. Therefore, the proportion coefficient of the jth bit of the ith aggregated PHYPayload copy can be mathematically calculated by the following equation:

$$P_{p_{i,j}} = \begin{cases} (\sum_{i=1}^{N} P_{r_{i,j}})/N, & P_{r_{i,j}} = 1 \\ 1 - (\sum_{i=1}^{N} P_{r_{i,j}})/N, & P_{r_{i,j}} = 0 \end{cases} \tag{12}$$

Then the confidence value of the jth bit of the aggregated PHYPayload copies can be represented based on the soft value with proportion coefficient as follows:

$$C_j = \sum_{i=1}^{N} P_{p_{i,j}} * S_{v_{i,j}} \tag{13}$$

Finally, the jth bit of newly built P_e can be mapped by comparing the confidence value with the threshold 0 as follows:

$$P_{e_j} = \begin{cases} 1, & (P_{p_{i,j}} = 1) \vee (C_j > 0) \\ 0, & (P_{p_{i,j}} = 0) \vee (C_j \leq 0) \end{cases} \tag{14}$$

If the new P_e satisfies the conditional stop mechanism, it works, otherwise, the TIDR algorithm declares a failure finally.

4.3. Algorithm description

Algorithm 1 displays the detailed workflow of the algorithm. Line 1 first filters out packets with different metadata. Line 2 regards the 'best' packet from all copies as the candidate. Line 3–6 takes the candidate as part of the input to generate the corrected packet. Line 7–10 continues to correct the candidate. In lines 11–16, it attempts the final error correction.

Algorithm 1 TIDR Algorithm

```
Require: $P_r$, $P_c$, $\sigma$, FCS, size, FCnt, DevAddr
 1: if metadata match then
 2:     $P_c \leftarrow \mathrm{Pretreatment}(P_r, \sigma)$
 3:     $q \leftarrow (P_r, P_c, \mathrm{FCS}, \mathrm{size}, \mathrm{FCnt}, \mathrm{DevAddr})$
 4:     $P_e \leftarrow \mathrm{EO}(q)$
 5:     if $Len(P_e) \neq 0$ then
 6:         send $P_e$
 7:     else
 8:         $P_e \leftarrow \mathrm{MO}(q)$
 9:         if $Len(P_e) \neq 0$ then
10:             send $P_e$
11:         else
12:             $q \leftarrow (P_r, \sigma, \mathrm{FCS}, \mathrm{size}, \mathrm{FCnt}, \mathrm{DevAddr})$
13:             $P_e \leftarrow \mathrm{WO}(q)$
14:             if $Len(P_e) \neq 0$ then
15:                 send $P_e$
16:             else
17:                 return null
18:             end if
19:         end if
20:     end if
21: end if
```

4.4. Discussion

4.4.1. Time complexity

When it comes to the procedure of improved search and conditional stop mechanism, processing latency is an important problem to be discussed. In the worst-case of hidden error, there is a possibility that none of the bitmasks helps to meet the conditional stop mechanism after an energy-wasteful exhaustive search. In such a case, the exponential complexity of $O(2^X)$ may lead to an almost infinite runtime. Therefore, it is necessary to set a maximum processing delay that can force the error correction to stop in time. In the LoRaWAN, two types of messages are specified, i.e., unconfirmed message and confirmed message, and the corresponding settings of maximum processing latency are different.

Unconfirmed Message. When a device sends an unconfirmed message, there is almost no need for restriction because no ACK is required, and the maximum processing latency is virtually the packet transmission interval T_g, which may even reach tens of minutes in delay-tolerant LPWAN applications with sparse traffic. In short, there is ample processing time for error correction.

Confirmed Message. As for the stricter confirmed message, the uplink transmission of the end device requires ACK. Therefore, the whole process should be within the reception window length regardless of the packet transmission interval length, and a large processing latency causes the timeout. One method to make the algorithm implementable is to define an upper bound on hamming weight through a microbenchmark. If X exceeds this guaranteed hard-coded bit constraint, no further calculation will be attempted because the runtime is likely to exceed the window length. However, this strategy is too conservative and not universal because the benchmark depends on hardware processing capability. To make a flexible processing latency upper constraint, we take the Round-trip Time (RTT) of bidirectional communication in LoRaWAN into consideration, which can be approximately modeled as follows:

$$\mathrm{RECEIVE\_DELAY1} \geq \mathrm{Latency} + \mathrm{ToA}_{\mathrm{up}} + \mathrm{ToA}_{\mathrm{down}} \tag{15}$$

where $\mathrm{ToA}_{\mathrm{up}}$ and $\mathrm{ToA}_{\mathrm{down}}$ are uplink and downlink TOA end-to-end delay, respectively.

When the end-device set an *SF* value, it is easy to calculate the total TOA according to the transceiver datasheet. Then we can find out the maximum processing latency.

For generality, ReLoRaWAN enables the latency constraint in both cases, even though it is not needed in the case of unconfirmed messages.

4.4.2. Recovery validity

In the case of the conditional stop mechanism, if ReLoRaWAN only relies on CRC verification like OPR, the recovered PHYPayload may not pass the MIC verification at NS and be discarded. Instead, we take advantage of the second generator polynomial for MIC for supplementary hashing, which is initially designed for security in LoRaWAN specification. By combining the MIC security feature from AES-CMAC encryption in LoRaWAN with the original CRC verification, ReLoRaWAN greatly increases the validity of data recovery and eliminates the false positives problem without any extra coding strategy.

CRC Verification: Most of the traditional wireless network error control methods only take CRC verification as correction criteria, so it is likely to cause false positives that can be expressed mathematically as:

$$CRC(P_{e_i}) = CRC(P_{e_m}) \tag{16}$$

where P_{e_i} and P_{e_m} are two different P_e.

The high probability of false positives is $\frac{(2^X-1)}{2^{16}}$ under the 16-bit CRC verification, which may result in the collision attack and amplified traffic. Therefore, CRC verification is only the minimal requirement for error correction and is inadequate.

MIC Verification: When the P_e passes the CRC verification, CS queries whether the end device it comes from is registered, that is, whether the DevAddr is pre-stored in memory or not. If found, it will use the corresponding stored NwkSKey to check the integrity of the P_e by 32-bit AES-CMAC algorithm. The integrity check works as a generator polynomial.

Then the false positives probability is reduced to $\frac{(2^X-1)}{2^{16+32}}$ because of collision resistance. The follow-up experiments proved that incorporating CRC with MIC is enough to eliminate the false positives even if it is affected by quite huge noise.

4.4.3. Security features

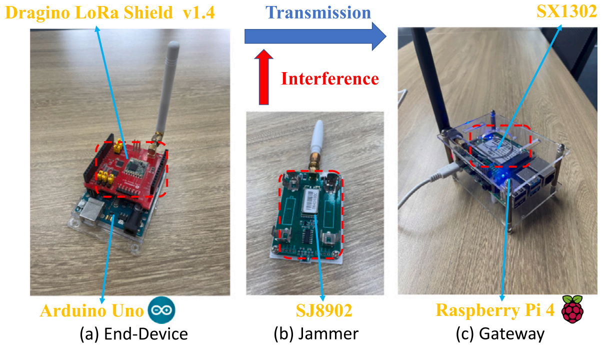
Because ReLoRaWAN utilizes some LoRaWAN security-related features, the security concerns need to be discussed from the two following aspects:

On the one hand, there is no possibility of passive network attacks. The ReLoRaWAN does not lack compliance with the LoRaWAN specification because CS does not have access to the AppSKey. Only the AS in the original LoRaWAN stack can store AppSKey for security and it only performs decryption on the packets that already pass the MIC verification. In other words, the CS does not need to decrypt FRMPayload with the AppSKey by the scheme described in LoRaWAN specification because if the restored PHYPayload passes the MIC verification, it can be inferred that the FRMPayload contained in it is intact. As a result, the bitwise operations of TIDR algorithm all perform on the PHYPayload, and it is not leaked. As for the AES-CMAC algorithm for the MIC verification of the PHYPayload only needs the NwkSKey for calculation, which can be referred to LoRaWAN specification.

On the other hand, ReloRaWAN has the capability to resist active network attacks and ensure the seucirty and reliability of the communication system. We do not snoop on the traffic between the gateway and NS like the OPR server which may break the security of LoRaWAN

Table 2
Experiment parameters.

Parameters	Values
N	6
M	8
T_g	10 s
λ	0.1 packet/s
size	28 bytes
SF	7 to 12
TP	0 to 7
Regional Band	CN470-510 MHz



(a) End-Device
(b) Jammer
(c) Gateway

Fig. 10. Transmitter and receiver hardware.

when access to these keys is not kept separate. On the contrary, we intend to build an authorized CS which receives the packets directly from the gateways and stores the NwkSKey with negotiation. The CS and NS are deployed in the same cloud server environment, So they have the same protective ability to resist active network attacks. Assuming that the original packets at NS are reliable, it can be inferred that the packets at CS can be securely transmitted without any tampering or alterations. Otherwise, even without CS, NS has enormous security risks. As a result, the security of storing the NwkSKey in CS and performing MIC verification is the same as that of NS and does not attract any additional risk to the system.

In summary, ReLoRaWAN does not break the security of LoRaWAN or pose security issues.

5. Implementation and evaluation

5.1. Experiental setup

In this section, we present the hardware, testbed, metric settings, and baselines. All the default experiment parameters are summarized in Table 2.

5.1.1. Hardware implementation

We implement gateways and end devices with low-cost COTS hardware and accessible open-source libraries, as shown in Fig. 10.

As for the hardware part of a gateway, the raspberry pi powered by tethered power supplies is equipped with an SX1302 chip. At the software level, it runs the packet forwarder.[4]

Connecting the Arduino motherboard with the LoRa shield, the end device runs the LMIC library.[5] The device is equipped with a DHT22 temperature/humidity sensor and a CCS811 air-quality sensor to build a sensing application. It generates packets according to low-power payload format, i.e., CayenneLPP 2.0 payload. There are existing temperature sensor and humidity sensor data types in CayenneLPP, which occupy two bytes and one byte of data size respectively. As for the eCO2 (equivalent CO2) and TVOC (Total Volatile Organic Compounds) from the air-quality sensor, we take two bytes data size structure for each one after comprehensively considering both data size and resolution. Overall, the FRMPayload size of each packet is 15 bytes in size with the data channel and data type flags, which is compressed as small as possible. According to Figs. 2 and 6, the total PHYPayload size is 28 bytes. After this data compression, the PHYPayload is short. When the AS receives the data, it decrypts the FRMPayload and leverages the payload codec to get the original uncompressed data. Fig. 11 indicates the reference design software of the gateway and device.

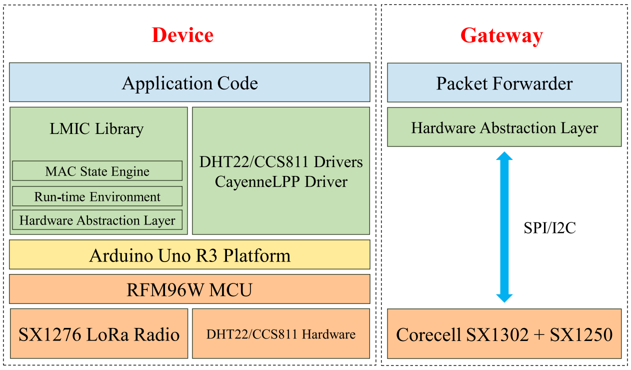


Fig. 11. The reference design software.

The device runs in CN470–510 MHz band and the packet transmission interval T_g is as short as 10 s in LoRaWAN networks to get abundant samples because there is no duty cycle regulation, which also means the packet arrival rate λ is 0.1 packet per second. When taking different SF parameters, the $\mathrm{ToA_{up}}$ is different and can be used to get the duty cycle δ as follows:

$$\delta = \frac{\mathrm{ToA_{up}}}{\mathrm{ToA_{up}} + T_g} \tag{17}$$

The default settings for BW and CR are 125 kHz, 4/5 respectively, which conform to the fixed values in the regional parameters.[6] The SF and TP available in this band range from 7 to 12 and 0 to 7 respectively. For simplicity, we divide different TP into three degrees, i.e., respectively as low power (TP=6), mid power (TP=3), and high power (TP=1). To test the resilience of ReLoRaWAN, Rejeee SJ8902 sensors work as jammers and make concurrent emissions to generate noises. They are designed with SX1278 chip and SHT30 sensor and all have fixed configurations when no special statement.

5.1.2. Testbed setup

We explore the performance of ReLoRaWAN over months in a real-world platform. Inside an indoor scene (e.g., lab institute), we conduct experiments on two different floors and each floor spans an urban region of 40.5 m × 39.3 m. There are static obstacles such as walls and chairs, and the RF packets are very likely to suffer severe attenuation. The presence of moving people also causes shadow fading of signals in a dynamic environment. We place multiple static transceivers and gateways on different floors at fixed positions, which are as depicted in Fig. 12. The marked red hexagon, green triangle, and blue circle represent the gateway, jammer, and device respectively.

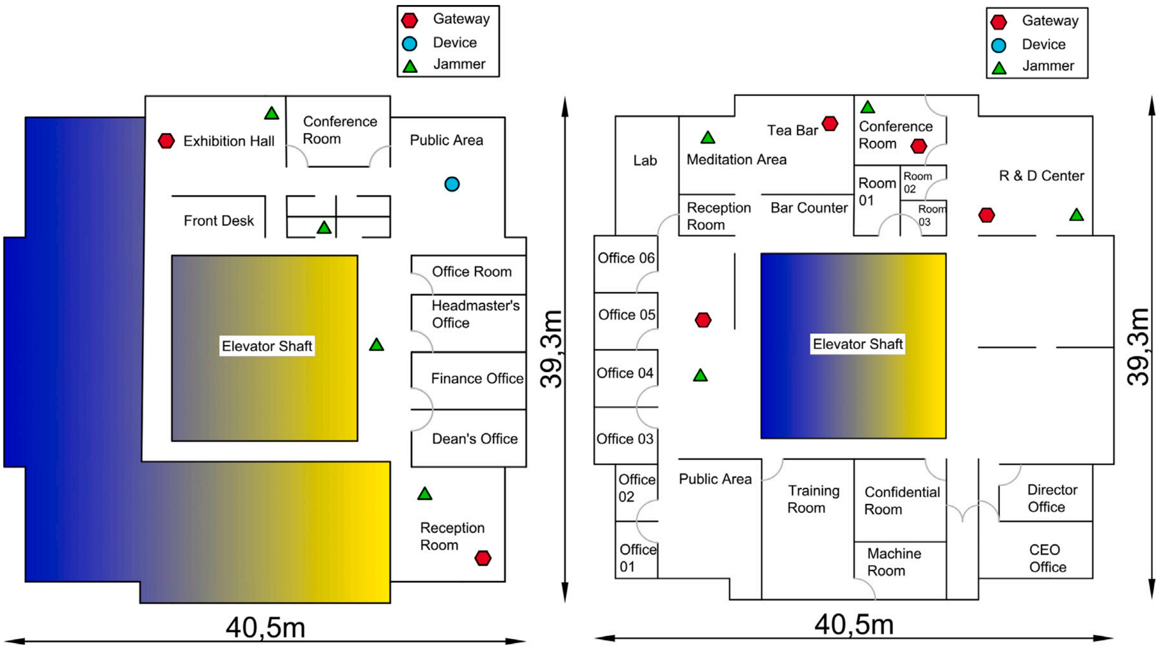
We deploy all servers at a cloud virtual private server. An open-source LoRaWAN network server stack called ChirpStack[7] is established with Docker Compose. ReLoRaWAN is implemented in C++.

[4] SX1302/SX1303 Hardware Abstraction Layer and Tools, https://github.com/Lora-net/sx1302_hal

[5] MCCI LoRaWAN LMIC Library, https://github.com/mcci-catena/arduino-lmic

[6] RP002-1.0.3 LoRaWAN® Regional Parameters, https://lora-alliance.org/resource_hub/rp2-1-0-3-lorawan-regional-parameters/

[7] ChirpStack, https://www.chirpstack.io/



(a) The 7th floor layout.

(b) The 8th floor layout.

Fig. 12. Experimental layout deployment.

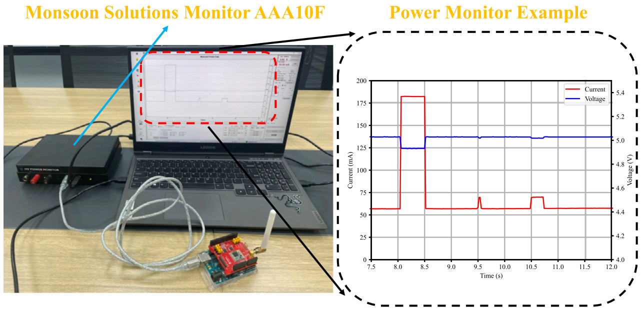


Fig. 13. Power monitor setup.

5.1.3. Metrics

There are two metrics to investigate the overall performance of real-time decoding:

- *Quality of Service (QoS)*: QoS is an important indicator for evaluating computer network. We mainly consider reliability and scalability, which are determined by PDR and throughput at the receiving end respectively.

  As for the PDR, we get inspiration from the recommendation document.[8] We record the FCnt of the delivered packets of the device in sequence to form an array and compute PDR, that is, dividing the number of delivered FCnt by the original length of the entire array. The formula is as follows:

  $$\mathrm{PDR}_{system} = \frac{\mathrm{FCnt}_{delivered}}{\mathrm{FCnt}_{max}} \tag{18}$$

  where the $\mathrm{FCnt}_{delivered}$ is the total number of delivered FCnt and the FCnt_{max} is the FCnt of the latest delivered packet plus one.

  The average throughput of the system is defined as follows:

  $$\mathrm{throughput}_{system} = \frac{L_{payload} * \mathrm{FCnt}_{delivered}}{\mathrm{Runtime}_{system}} \tag{19}$$

  where the $\mathrm{Runtime}_{system}$ is the total runtime of the receiving end.

  For example, to calculate the PDR, we record the FCnt of the delivered packets in sequence, which forms an array such as [0, 1, 2, 5, 7]. At this point, the FCnt_{max} can be obtained by deriving it from the FCnt of the most recently delivered packet when considering the value 0, which equals 8. The FCnt values of the undelivered packets are 3, 4, and 6. Consequently, by dividing the number of delivered packets 5 by the total number of transmitted packets 8, the PDR is calculated to be 5/8. To calculate the average throughput, ReLoRaWAN has to first multiply the bit length of the PHYPayload $L_{payload}$ by the length of the delivered FCnt array 5 to determine the total amount of data delivered. Next, this value is divided by the total running time of the receiving end to obtain the average throughput.
- *Power consumption*: In the LPWAN field, it is crucial to find out the battery life of an end device, which is calculated indirectly by energy consumption. When the power consumption is reduced, the energy consumption of the same time duration decreases proportionally. We use a power monitor (Monsoon AAA10F) to measure the overall power consumption for evaluation, as shown in Fig. 13. The average power consumption is computed as follows:

  $$P_{avg} = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} V_{(t)} I_{(t)} dt \tag{20}$$

  where the $V_{(t)}$ and $I_{(t)}$ are the instantaneous voltage and current at time t separately including transmission and sleep state.

On the one hand, the reduction of PDR inevitably causes retransmissions that waste energy and increases the average power. In this context, the calculation of PDR in the unconfirmed mode that does not require retransmissions is an intuitive indicator to show the indirect impact on power consumption. The higher PDR is, the lower power consumption is guaranteed in some way.

[8] LoRaWAN — simple rate adaptation recommended algorithm, https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f515d14a1.pdf

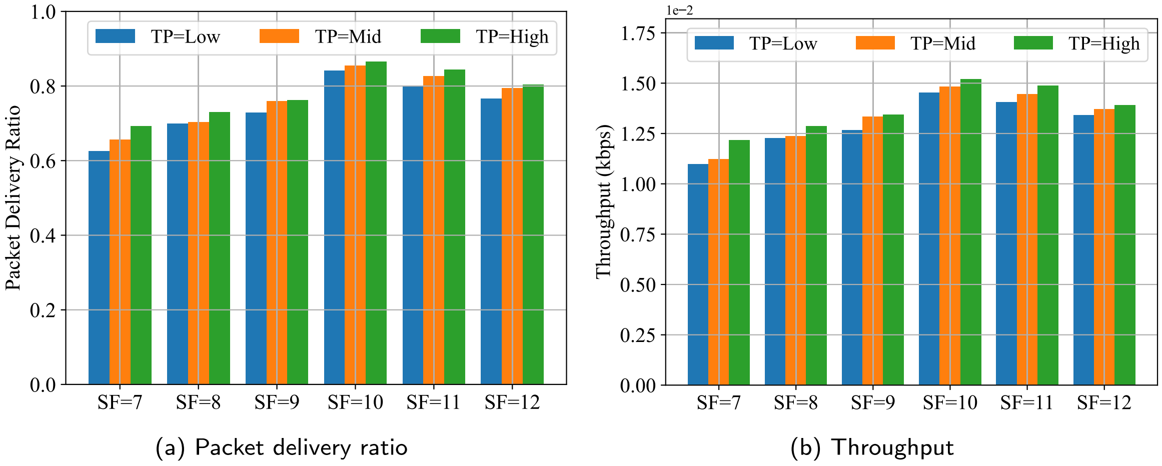


Fig. 14. Impacts of *SF* and *TP* on (a) PDR (b) throughput.

On the other hand, computing the average power consumption in the confirmed mode can yield the direct impact of retransmission on energy and average power. As a result, there is no necessity to compute PDR in this context. In the LoRaWAN specification, the retransmissions will happen several times each time there is a packet loss according to the real-time packet loss rate arbitrarily.

5.1.4. Baselines

We compare ReLoRaWAN with two existing benchmarks:

- *Legacy LoRaWAN*: The legacy LoRaWAN protocol, which is established based on LoRa modulation, does not incorporate any specific measures for data recovery control. In this case, distorted packets are more likely to be dropped. The number of retransmissions increases when devices adopt the confirmed messages, which leads to higher average power consumption.
- *OPR [17]*: OPR performs opportunistically cloud-optimized link-layer bit error control. The bitwise operation processing capacity of the OPR is contingent upon the number of gateways utilized.

5.2. Experimental results

5.2.1. Impact of physical parameters

According to the modulation process, the *SF* and *TP* are the two most important physical parameters. Therefore, we analyze the performance of ReLoRaWAN under various combinations of fixed parameter settings.

The impact of configuration on collision resolution and weak signal decoding is analyzed. On the one hand, as shown in Fig. 14, due to the improved SNR condition, higher *TP* does improve the demodulation process. On the other hand, when *SF* becomes large, the demodulation sensitivity is increased and the symbol is less affected by collisions. However, when *SF* is high enough, they have a long TOA because of the large symbol duration and become vulnerable to jamming when transmitting. Therefore, the higher *SF* does not always guarantee transmission performance.

5.2.2. Comparison of existing methods

We compare ReLoRaWAN with baselines with the following settings under fixed parameter settings (*SF*=10, *TP*=High):

- *Jammer number:* The level of concurrency is based on the jammer amount.
- *Gateway number:* The processing capacity of the approach varies with the number of gateways.

We note from the results in Fig. 15 that the increased gateway number coherently improves the packet delivery capacity of both algorithms. This is because of the impact of redundant multiple receptions for mitigating interference. Compared with OPR, ReLoRaWAN can recover more data even under huge interference because it exploits more physical information. After ReLoRaWAN theoretically corrects all errors except for unrecoverable hidden errors, the PDR is improved. Additionally, ReLoRaWAN enhances throughput by facilitating the successful delivery of data and thereby improves transmission efficiency. In the end, ReLoRaWAN increases PDR to about 1.35× compared with OPR at most.

5.2.3. Verification of recovery validity

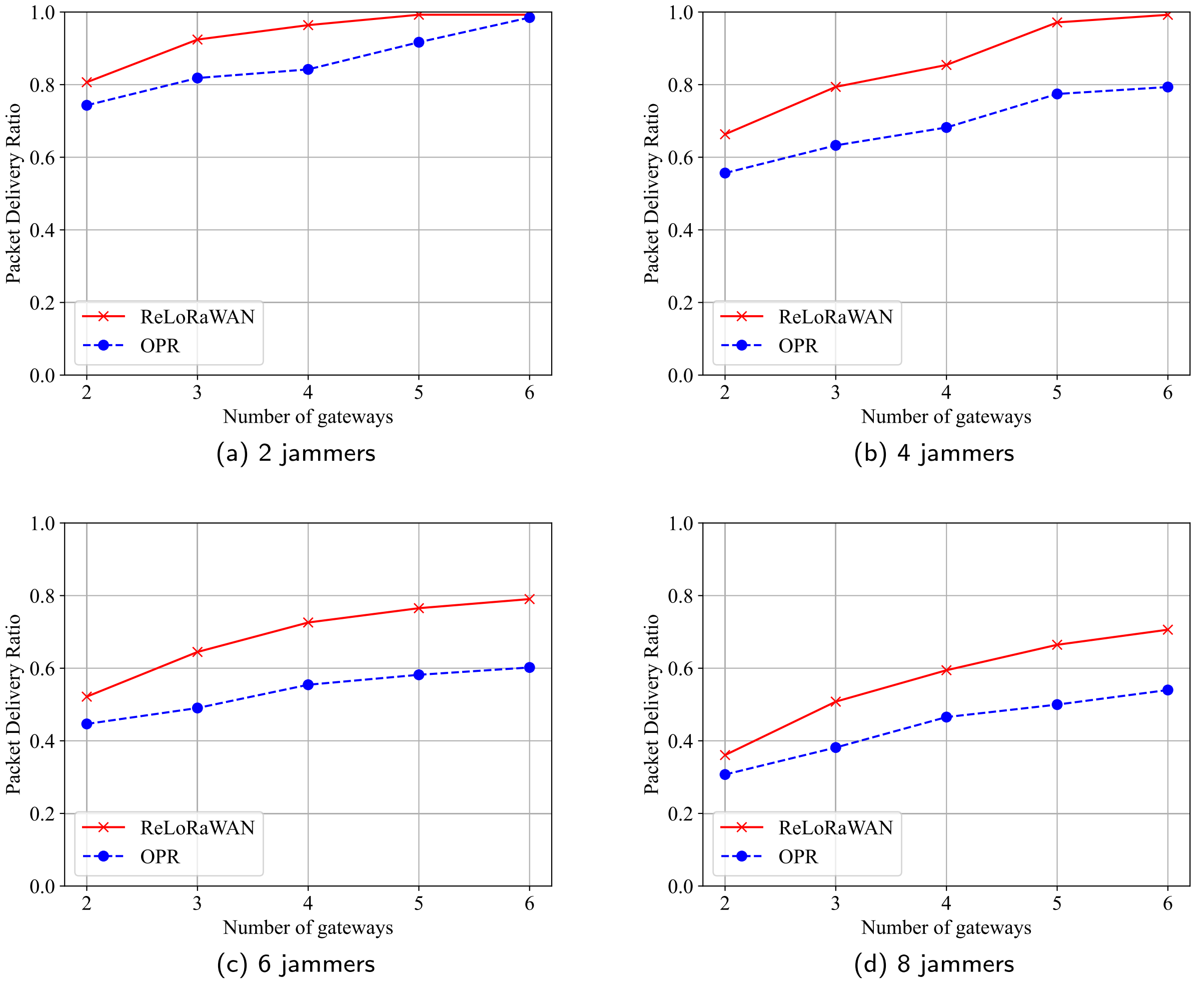
Because of the conditional stop mechanism, there is an increased chance of false positives. As a result, the validity of error correction needs to be verified by real experiments. We measure the throughput at CS and AS at different parts of the framework when ReLoRaWAN is enabled.

It is worth noting that regardless of whether the data recovery is successful or not, the information at CS is updated every time all packet copies are aggregated. While the information at AS does not update unless it receives the packet that passes FCS verification.

The observed results in Fig. 16 show that almost no false positives happen because there is no difference between the statistics at CS and AS most of the time. It means that the NS rarely drops packets because ReLoRaWAN really corrects most of the error packets. The result verifies the excellent capacity for error correction.

5.2.4. Power consumption evaluation

When confirmation messages are employed by the device, an increase in the number of uncorrectable packets results in a corresponding increase in the number of retransmissions. This, in turn, leads to higher power consumption. The ReLoRaWAN is designed as an application layer amendment for LoRaWAN to achieve data recovery for lower power consumption. Therefore, to evaluate the energy-saving performance of ReLoRaWAN, we compare it with legacy LoRaWAN. Fig. 17 indicates almost no retransmissions with ReLoRaWAN because it corrects most of the erroneous PHYPayload copies. The legacy LoRaWAN without data recovery capacity introduces unnecessary retransmissions more frequently. The average power of legacy LoRaWAN and ReLoRaWAN is 410 mW and 289 mW respectively when the sleep state is considered, which means that our method reduces the overall power by nearly 30%.



(a) 2 jammers

(b) 4 jammers

(c) 6 jammers

(d) 8 jammers

Fig. 15. Impact of gateway and jammer number on packet delivery ratio.

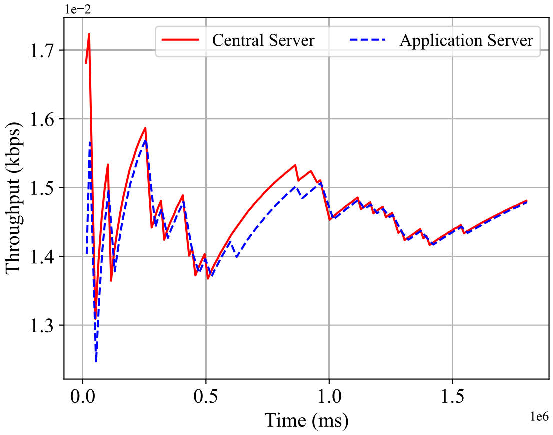


Fig. 16. Throughput comparison.

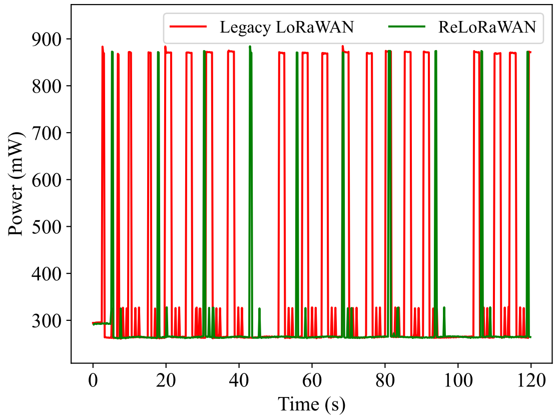


Fig. 17. Instant power consumption comparison.

6. Related work

LoRa and LoRaWAN Performance Analysis. Up to now, researchers have conducted a wide range of studies on the overall performance of LoRa and LoRaWAN [23,24].

Tu et al. [25] introduce the energy model. Plenty of literature [26,27] conclude the error rate under interference. Stusek et al. [28] show that multiple gateways are beneficial to the whole system. In [5,6], the authors conduct extensive research efforts on the propagation capabilities of LoRa in indoor scenarios such as factories and offices. Borkotoky et al. [29] perform detailed experiments on the retransmission influence.

LoRa Collision Recovery and Weak Signal Decoding. There is lots of pioneer solution that aims at mitigating the collisions in LoRa and decoding the weak physical signals. They can be mainly categorized into physical layer methods, MAC layer methods, and application layer methods.

The physical layer approaches works on physical modulation to disentangle overlapping symbols to alleviate interference based on frequency or time-domain features [30]. CurvingLoRa [31] replaces the standard chirp with a non-linear chirp to decode collided signals. MALoRa [32] utilizes the gain of the multi-antenna gateway to design a phase-based parallel decoder and resist carrier frequency offset. Pyramid [33] uses multiple windows to decode conflict signals in real-time. CIC [34] cancels out all the interfering symbols to decode colliding packets. NELoRa [35] decodes collision signals using DNN. The physical layer approaches have the advantages of low computational complexity and processing latency but mostly rely on specific expensive SDR equipment. They cannot be deployed on a large scale because of the additional hardware cost.

The MAC layer solutions leverage synchronization to schedule transmissions for collision avoidance instead of pure Aloha protocol [36]. CurveALOHA [37] introduces a new access mechanism to support non-linear chirps. FCA-LoRA [38] broadcasts beacons with the gateway to synchronize devices and increases transmission fairness. PolarTracker [39] acquires physical attitude information of the device to model propagation link model and proposes a slotted-based ALOHA protocol. SBTS-LoRa [40] divides the timeslots based on the distance from a node to the gateway for large-scale networks. TS-LoRa [41] establishes a fine-grained synchronous time slot access control slotted Aloha protocol on top of the physical layer. Despite these complex MAC layer methods compliant with LoRaWAN devices, they need to redesign the standard frame structure. In some cases, they deplete the battery and degrade the overall throughput performance of the network. Moreover, LoRaWAN has become a defacto LPWAN standard, So the cost of modifying, popularizing, and deploying private protocols is incalculable. What is worse, the MAC layer methods are mainly based on simulation, and verification, without actual large-scale experiments.

The traditional application layer methods are software-only algorithms without specialized SDR platform costs. They depend on channel encoding primitives from information theory to ensure the transmission between sender and receiver. NCC-LoRa [42] utilizes Network-Coded Cooperation based on D2D communication to improve outage probability. DaRe [43] utilizes fountain codes and convolutional codes to recover data from previously received data frames when frame erasure occurs. ReDCoS [44] introduces a lightweight encoder for payload for reliable transmission. However, like the traditional FEC, the extra overhead from application layer methods may increase energy waste, even under successful transmission.

By merging multiple non-coded packets, ReLoRaWAN does not need to provide wasteful channel coding strategies like prior works. It relies on collaborative decoding to achieve error control with the least transmission redundancy.

LoRaWAN Resource Allocation. The fixed physical modulation parameters affect the maximal scalability. As a result, Semtech proposes the first standard NS-side Adaptive Data Rate (ADR) algorithm to adjust the parameters dynamically according to the link quality. However, the legacy ADR design is simple and does not consider the collision probability in the Aloha-like LoRaWAN system [45]. As a result, there are some collision-aware works recently. STEPS [46] adjusts the parameters intelligently with reinforcement learning. EFLoRa [8] formulates a max–min fairness problem, taking into account the energy efficiency of the entire network. AAPC [47] proposes a semi-decentralized algorithm for application requirements like PDR and energy consumption per packet. Aimi et al. [48] isolate clusters of devices and meet differentiated PDR targets to address network congestion. ADR-Lite [49] even designs a link-based ADR algorithm in a mobile scenario.

Wireless Network Error Control. Wireless error control is a traditional problem that flourish for decades. There are mainly two types of methods to solve it: hard decision decoding and soft decision decoding.

In hard decision decoding, PSC [50] selects the best signal independent of physical layer implementation in cooperative communications. iPM [20] works in cooperative communication and broadcast communication. It performs the bitwise modulo-2 operation on multiple data packets from branches to determine the error bit position and to repair the error with brute force search. MRD [51] utilizes block-based packet merging and determines the location of errors with blocks of different packets. APC [21] incorporates packet combining with the majority voting to deal with multi-bit errors.

As for soft decision decoding, SOFT [22] and PPR [52] both leverage a physical layer-independent interface that informs higher layers of the confidence of each bit in WLAN to recover errors.

7. Conclusion

In this paper, we propose a reliable data delivery mechanism ReLoRaWAN that is a novel application-layer amendment to LoRaWAN. It aims to aggregate distorted packet PHYPayload copies from multi-gateway reception of the same packet and execute the corresponding data recovery operations. To correct bit errors in malformed packets as much as possible, we design the tri-operation integrated data recovery algorithm, which involves exclusive-OR based bitwise inversion operation, majority voting based bitwise inversion operation, and weighted bitwise decision operation. On this basis, we implement the ReLoRaWAN testbed with COTS hardware and evaluate the performance through real-world experiments. The experimental results demonstrate that the packet delivery ratio of ReLoRaWAN has increased to 1.35 times of the existing method OPR and the average power consumption of the end device is reduced by 30%.

In the future, we plan to integrate more data recovery operations to further improve the performance of ReLoRaWAN. We will also investigate fine-grained physical-layer decoding techniques and involve them in our solution. Moreover, we will combine our solution with dynamic resource allocation algorithms like ADR.

CRediT authorship contribution statement

Wenjia Wu: Conceptualization, Methodology. **Hao Wang:** Investigation, Writing, Software. **Zisheng Cheng:** Validation.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. 62072102, 62132009); Key R&D Program of Jiangsu Province (No. BE2022680); Jiangsu Provincial Key Laboratory of Network and Information Security (No. BM2003201); the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China (No. 93K-9).

References

[1] Augustine Ikpehai, Bamidele Adebisi, Khaled M. Rabie, Kelvin Anoh, Ruth E. Ande, Mohammad Hammoudeh, Haris Gacanin, Uche M. Mbanaso, Low-power wide area network technologies for Internet-of-Things: A comparative review, IEEE Internet Things J. 6 (2) (2019) 2225–2240.

[2] Riccardo Marini, Konstantin Mikhaylov, Gianni Pasolini, Chiara Buratti, Low-power wide-area networks: Comparison of LoRaWAN and NB-IoT performance, IEEE Internet Things J. (2022) 1.

[3] Zehua Sun, Huanqi Yang, Kai Liu, Zhimeng Yin, Zhenjiang Li, Weitao Xu, Recent advances in LoRa: A comprehensive survey, ACM Trans. Sen. Netw. (2022).

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

[5] Davide Magrin, Martina Capuzzo, Andrea Zanella, Lorenzo Vangelista, Michele Zorzi, Performance analysis of LoRaWAN in industrial scenarios, IEEE Trans. Ind. Inform. 17 (9) (2021) 6241–6250.

[6] Weitao Xu, Jun Young Kim, Walter Huang, Salil S. Kanhere, Sanjay K. Jha, Wen Hu, Measurement, characterization, and modeling of LoRa technology in multifloor buildings, IEEE Internet Things J. 7 (1) (2020) 298–310.

[7] Jetmir Haxhibeqiri, Abdulkadir Karaagac, Floris Van den Abeele, Wout Joseph, Ingrid Moerman, Jeroen Hoebeke, Lora indoor coverage and performance in an industrial environment: Case study, in: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2017, pp. 1–8.

[8] Zhiwei Zhao, Weifeng Gao, Wan Du, Geyong Min, Wenliang Mao, Mukesh Singhal, Towards energy-fairness in LoRa networks, IEEE Trans. Mob. Comput. (2022) 1.

[9] Verónica Toro-Betancur, Gopika Premsankar, Mariusz Slabicki, Mario Di Francesco, Modeling communication reliability in LoRa networks with device-level accuracy, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–10.

[10] Alexandre Guitton, Megumi Kaneko, Multi-gateway demodulation in LoRa, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2008–2013.

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

[12] Peiyuan Qin, Luoyu Mei, Qi Jing, Shuai Wang, Zhimeng Yin, Xiaolei Zhou, Edge-cloud collaborative interference mitigation with fuzzy detection recovery for LPWANs, in: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design, CSCWD, 2022, pp. 792–797.

[13] Henrik Rosenberg, Andreas Reinhardt, WIP: Collaborative approaches to mitigate links of variable quality in LoRa networks, in: 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, 2021, pp. 244–247.

[14] Jiamei Lv, Gonglong Chen, Wei Dong, Exploiting rateless codes and cross-layer optimization for low-power wide-area networks, ACM Transactions on Sensor Networks 18 (4) (2023) 1–24.

[15] Jae-Mo Kang, MIMO-LoRa for high-data-rate IoT: Concept and precoding design, IEEE Internet Things J. 9 (12) (2022) 10368–10369.

[16] Andrea Petroni, Mauro Biagi, Interference mitigation and decoding through gateway diversity in LoRaWAN, IEEE Trans. Wireless Commun. (2022).

[17] Artur Balanuta, Nuno Pereira, Swarun Kumar, Anthony Rowe, A cloud-optimized link layer for low-power wide-area networks, in: Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services, MobiSys '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 247–259.

[18] Henri Dubois-Ferrière, Deborah Estrin, Martin Vetterli, Packet combining in sensor networks, in: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, SenSys '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 102–115.

[19] Chenglong Shao, Osamu Muta, Qinghe Du, Kapil R. Dandekar, Xianpeng Wang, Multiple access in large-scale LoRaWAN: Challenges, solutions, and future perspectives, IEEE Consum. Electron. Mag. (2022) 1–9.

[20] Damien O'Rourke, Conor Brennan, Practical packet combining for use with cooperative and non-cooperative ARQ schemes in resource-constrained wireless sensor networks, Ad Hoc Netw. 10 (3) (2012) 339–355.

[21] Yiu-Wing Leung, Aggressive packet combining for error control in wireless networks, IEICE Trans. Commun. 83 (2) (2000) 380–385.

[22] Grace R. Woo, Pouya Kheradpour, Dawei Shen, Dina Katabi, Beyond the bits: Cooperative packet recovery using physical layer information, in: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 147–158.

[23] Davide Magrin, Martina Capuzzo, Andrea Zanella, A thorough study of LoRaWAN performance under different parameter settings, IEEE Internet Things J. 7 (1) (2020) 116–127.

[24] Jothi Prasanna Shanmuga Sundaram, Wan Du, Zhiwei Zhao, A survey on LoRa networking: Research problems, current solutions, and open issues, IEEE Commun. Surv. Tutor. 22 (1) (2020) 371–388.

[25] Lam-Thanh Tu, Abbas Bradai, Yannis Pousset, Alexis I. Aravanis, Energy efficiency analysis of LoRa networks, IEEE Wirel. Commun. Lett. 10 (9) (2021) 1881–1885.

[26] Panagiotis Gkotsiopoulos, Dimitrios Zorbas, Christos Douligeris, Performance determinants in LoRa networks: A literature review, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1721–1758.

[27] Qahhar Muhammad Qadir, Analysis of the reliability of LoRa, IEEE Commun. Lett. 25 (3) (2021) 1037–1040.

[28] Martin Stusek, Dmitri Moltchanov, Pavel Masek, Konstantin Mikhaylov, Jiri Hosek, Sergey Andreev, Yevgeni Koucheryavy, Pavel Kustarev, Otto Zeman, Martin Roubicek, LPWAN coverage assessment planning without explicit knowledge of base station locations, IEEE Internet Things J. 9 (6) (2022) 4031–4050.

[29] Siddhartha S. Borkotoky, Jorge F. Schmidt, Udo Schilcher, Prameela Battula, Sonu Rathi, Reliability and energy consumption of LoRa with bidirectional traffic, IEEE Commun. Lett. 25 (11) (2021) 3743–3747.

[30] Ningning Hou, Xianjin Xia, Yuanqing Zheng, Jamming of LoRa PHY and countermeasure, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–10.

[31] Chenning Li, Xiuzhen Guo, Longfei Shangguan, Zhichao Cao, Kyle Jamieson, CurvingLoRa to boost LoRa network throughput via concurrent transmission, in: 19th USENIX Symposium on Networked Systems Design and Implementation, NSDI 22, USENIX Association, Renton, WA, 2022, pp. 879–895.

[32] Ningning Hou, Xianjin Xia, Yuanqing Zheng, Don't miss weak packets: Boosting LoRa reception with antenna diversities, ACM Trans. Sen. Netw. 19 (2) (2023).

[33] Zhenqiang Xu, Pengjin Xie, Jiliang Wang, Pyramid: Real-time LoRa collision decoding with peak tracking, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–9.

[34] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, Bhuvana Krishnaswamy, Concurrent interference cancellation: Decoding multi-packet collisions in LoRa, in: Proceedings of the 2021 ACM SIGCOMM 2021 Conference, SIGCOMM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 503–515.

[35] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, Yunhao Liu, NELoRa: Towards ultra-low SNR LoRa communication with neural-enhanced demodulation, in: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, SenSys '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 56–68.

[36] Luca Beltramelli, Aamir Mahmood, Patrik Österberg, Mikael Gidlund, LoRa beyond ALOHA: An investigation of alternative random access protocols, IEEE Trans. Ind. Inform. 17 (5) (2021) 3544–3554.

[37] Chenning Li, Zhichao Cao, Li Xiao, CurveALOHA: Non-linear chirps enabled high throughput random channel access for LoRa, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, IEEE, 2022, pp. 520–529.

[38] Anna Triantafyllou, Panagiotis Sarigiannidis, Thomas Lagkas, Ioannis D. Moscholios, Antonios Sarigiannidis, Leveraging fairness in LoRaWAN: A novel scheduling scheme for collision avoidance, Comput. Netw. 186 (2021) 107735.

[39] Yuting Wang, Xiaolong Zheng, Liang Liu, Huadong Ma, PolarTracker: Attitude-aware channel access for floating low power wide area networks, IEEE/ACM Trans. Netw. (2022) 1–15.

[40] Hanan Alahmadi, Fatma Bouabdallah, Ahmed Al-Dubai, A novel time-slotted LoRa MAC protocol for scalable IoT networks, Future Gener. Comput. Syst. 134 (2022) 287–302.

[41] Dimitrios Zorbas, Khaled Abdelfadeel, Panayiotis Kotzanikolaou, Dirk Pesch, TS-LoRa: Time-slotted LoRaWAN for the industrial Internet of Things, Comput. Commun. 153 (2020) 1–10.

[42] Luis Henrique de Oliveira Alves, João Luiz Rebelatto, Richard Demo Souza, Glauber Brante, Network-coded cooperative LoRa network with D2D communication, IEEE Internet Things J. 9 (7) (2022) 4997–5008.

[43] Paul J. Marcelis, Nikolaos Kouvelas, Vijay S. Rao, R. Venkatesha Prasad, DaRe: Data recovery through application layer coding for LoRaWAN, IEEE Trans. Mob. Comput. 21 (3) (2022) 895–910.

[44] Niloofar Yazdani, Nikolaos Kouvelas, R Venkatesha Prasad, Daniel E. Lucani, Energy efficient data recovery from corrupted LoRa frames, in: 2021 IEEE Global Communications Conference, GLOBECOM, 2021, pp. 1–6.

[45] F. Helder C. Santos F., Plínio S. Dester, Pedro H.J. Nardelli, Elvis M.G. Stancanelli, P. Cardieri, Dick Carrillo, Hirley Alves, Multi-class random access wireless network: General results and performance analysis of LoRaWAN, Ad Hoc Netw. 135 (2022) 102946.

[46] Mi Chen, Lynda Mokdad, Jalel Ben-Othman, Jean-Michel Fourneau, Dynamic parameter allocation with reinforcement learning for LoRaWAN, IEEE Internet Things J. (2023) 1.

[47] Ameer Ivoghlian, Kevin I-Kai Wang, Zoran Salcic, Application-aware adaptive parameter control for LoRaWAN, J. Parallel Distrib. Comput. 166 (2022) 166–177.

[48] Alessandro Aimi, Fabrice Guillemin, Stéphane Rovedakis, Stefano Secci, Packet delivery ratio guarantees for differentiated LoRaWanServices, in: GLOBECOM 2022 - 2022 IEEE Global Communications Conference, 2022, pp. 2014–2019.

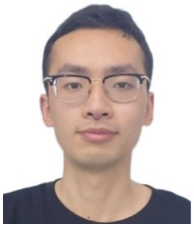
[49] Reza Serati, Benyamin Teymuri, Nikolaos Athanasios Anagnostopoulos, Mehdi Rasti, ADR-Lite: A low-complexity adaptive data rate scheme for the LoRa network, in: 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, 2022, pp. 296–301.

[50] S. Valentin, D.H. Woldegebreal, T. Volkhausen, H. Karl, Combining for cooperative WLANs - A reality check based on prototype measurements, in: 2009 IEEE International Conference on Communications Workshops, 2009, pp. 1–5.

[51] Allen Miu, Hari Balakrishnan, Can Emre Koksal, Improving loss resilience with multi-radio diversity in wireless networks, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, MobiCom '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 16–30.

[52] Kyle Jamieson, Hari Balakrishnan, PPR: Partial packet recovery for wireless networks, in: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 409–420.

Wenjia Wu received the B.S. and Ph.D. degrees in computer science from Southeast University, Nanjing, China, in 2006 and 2013, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University. His research interests include wireless and mobile networks.



Hao Wang received the B.S. degree from Hohai University, Nanjing, China, in 2020. He is currently a M.S. student in the Southeast University-Monash University Joint Graduate School in Southeast University, Nanjing, China. His research interests include wireless and mobile networks.

Zisheng Cheng received the B.S. degree from Southeast University, Nanjing, China, in 2022. He is currently a M.S. student in the School of Computer Science and Engineering in Southeast University, Nanjing, China. Her research interests include wireless and mobile networks.