# Edge-Cloud Collaborative Interference Mitigation with Fuzzy Detection Recovery for LPWANs

Peiyuan Qin[1], Luoyu Mei[1], Qi Jing[1], Shuai Wang[1*], Zhimeng Yin[2] and Xiaolei Zhou[3*]
[1] Southeast University, Nanjing, China
[2] City University of Hong Kong, Hong Kong, China
[2] The Sixty-Third Research Institute, National University of Defense Technology, Changsha, China
{peiyuanqin, lymei-, qijing, shuaiwang}@seu.edu.cn, {zhimeyin}@cityu.edu.hk, {zhouxiaolei}@nudt.edu.cn



***Abstract*—Recent researches have mitigated interference by utilizing cloud assistance or cloud-edge collaboration for Low-Power Wide-Area Networks. However, the issue of long interference recovery time prevents these methods from being well utilized in practical scenarios. In this paper, we propose a novel method, called FDR, for Edge-Cloud collaborative interference mitigation with Fuzzy Detection Recovery, which recovers errors in real-time. Our design (i) utilizes gateways and cloud servers and (ii) reduces data transmissions with fuzzy detection codes for real-time error recovery. In our design, each gateway detects and reports the fuzzy positions of errors to the cloud. Then the cloud restores packets with fuzzy detection results. FDR takes the advantage of both the computational ability of the cloud and the error detection benefit of each gateway. We design and implement FDR with commodity devices including LoRa SX1280 and the USRP-B210 platform. Experimental results show that FDR reduces recovery time by 78.53% compared with the state-of-art, and recovers interfered data packets accurately when the packet damage rate reaches 45.72%.**

***Keywords*—Low-Power Wide-Area Networks (LP-WANs), Fuzzy Detection, Signal Real-Time Recovery**

## I. Introduction

LoRa is a promising Low-Power Wide Area Network (LP-WAN) technology [1]. LoRa recently has become the mainstream technology for the Internet of Things (IoT) for both industrial and civilian networks worldwide. However many of today's wireless technologies are designed to share the same frequency band including WiFi, Bluetooth and 2.4 GHz LoRa, which inevitably brings to signal overlapping, causing either isomorphic or heterogeneous interference [18].

Traditionally, the methods to mitigate interference tend to utilize physical and MAC layers solutions [3]–[6]. Some researchers propose transparent solutions to re-design and synchronize LoRa senders [5], [6] while others reduce corruptions at base stations [1], [6]. Those efforts bring deployment complexity or hardware overhead. Recently, some studies utilize cloud resources to mitigate interference, which does not need extra hardware modification. For example, OPR [2] recovers interfered data packets by transmitting data packets and RSSI samples to the cloud which achieves a great recovery effect. However, the cloud-based methods need excessive data transmission (e.g. 200 bytes RSSI for a 25-byte payload packet [2]), which limits their feasibility in practice [7].

To recover the packets in real-time, the challenge is to detect packet corruptions rapidly. In this paper, we propose a novel real-time method for LoRa interference mitigation with fuzzy error detection, namely Fuzzy Detection Recovery (FDR). We design and insert the error fuzzy detection codes after the encoding of the LoRa physical payload. Instead of directly transmitting a large amount of data such as RSSI samples to detect errors, we detect errors with fuzzy detection codes. The gateways upload concise detection results to the cloud. Then we utilize comparing replacement algorithm to recover packets. The recovered packets are stored in the cloud if it is necessary. In summary, the contributions of FDR are as follows:

- We propose a novel real-time interference mitigation design called FDR, which is the first fuzzy detection recovery method based on Edge-Cloud collaborative interference mitigation for LPWANs. FDR is a software-based algorithm, which can be directly applied to the deployed LoRa infrastructure.
- In order to mitigate interference in real-time, we address a few challenges including (i) detecting the fuzzy position of errors before decoding packets and (ii) collaborating data packets from multiple gateways for packet recovery. Our techniques provide guidance for the range extension of real-time interference mitigation.
- We conduct extensive experiments on commercial devices, i.e., the sx1280 LoRa chip and USRP-N210 platform. Our experiments show that FDR achieves correctly decoding packets when the packet damage rate reaches 45.72% with the cost of a 9.35% increase of recovery time compared to the standard LoRa. Compared with the algorithms based on cloud such as OPR [2], FDR better meets the real-time requirement.

## II. Background and Motivation

In this section, we first introduce the basic concepts of LoRa and LoRaWAN and then explain the design motivation of FDR. Finally, we conduct a preliminary experiment to motivate our work.

### A. What LoRa is

LoRa is a low-power wide-area network wireless standard created by Semtech [7]. The LoRa signal has properties that resemble those of noise, rendering detecting or jamming more

TABLE I
FDR COMPARED WITH RELATED STUDIES

| | **Computational complexity** | **Data recovery capability** | **Data Transmission** |
|---|---|---|---|
| LoRa | Low | High | Low |
| OPR | High | Low | High |
| **FDR** | **Low** | **High** | **Low** |

difficult. The CSS processing increases resistance to both noise and interference [21].

### B. What LoRaWAN is

LoRaWAN is a Media Access Control (MAC) protocol for controlling low-power devices in wide-area networks. Its first specification was released by the LoRa Alliance in 2015. Fig.1 shows the architecture of LoRaWAN. It contains several functional modules: Receivers, Gateways, Network Servers, and Applications. After the gateway receives the signal, it demodulates the signal into a data packet. These data packets are finally transmitted to the cloud application for use.

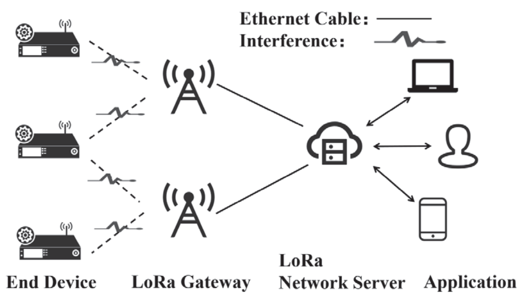



Fig. 1. Structure of the LoRaWAN

### C. Motivation

Modifying the hardware on the physical layer improves the performance of LoRa interference mitigation. However, the cost of extra hardware costs makes it difficult to popularize. To solve this problem, recent research proposes a cloud-based approach to recover the most likely corrupt bits. Without hardware modification, experiments show that the OPR can recover 72% of the failed data packets [2] by uploading the corrupt packets and RSSI samples. However, it requires the gateways to upload RSSI to the cloud which greatly increases the overhead of data recovery (e.g. 200 bytes RSSI for a 25-byte payload packet [2]). We compare the performance of Standard LoRa, Cloud-based approach, and fuzzy error detection approach in Table I. In this paper, we try to design an approach that achieves all the ideal performances at the same time. To achieve this, we design a real-time recovery algorithm based on edge-cloud collaborative interference mitigation. The corruption is detected at the gateway side while the packets are restored at the cloud side, which reduces the data transmission used for detecting errors. Besides, the algorithm utilizes packets from multiple gateways, to achieve strong error correction capability. With a carefully designed packet recovery algorithm, it is able to restore packets with the time approaching standard LoRa.

**Disjoint Interference in Multi-Packets.** Consider an example of three reception at three different gateways of the same packet. These three packets have been interfered with at different positions, where one packet has errors in the header, and the other two have errors in other parts. Packets have the interference in disjoint position. The interesting finding is called disjoint interference [22], by which the wrong part can be recovered by using the same and correct part in another packet.

**Feasibility of Fuzzy Detection.** Fuzzy detection refers to detecting whether an error exists in a fixed-length data segment, rather than being accurate to a certain bit. In order to verify the feasibility of fuzzy detection, we design a preliminary experiment and set the length of data segment to 7 bits.

In the preliminary experiment, a one-bit parity check code is used to detect data segment errors. For comparison, we utilize the known error detection results of data segments (These results are obtained by comparing with the correct data segments). Under the condition that SF=9, our experiments(Fig.2) show that the average recovery rate reached 70.4% by fuzzy detection. Some packets cannot be recovered because an even number of errors occur. This is the limitation of the parity check code itself. The experimental results show that it is feasible to utilize fuzzy detection to recover data packets in the real world. This motivates us to design a fuzzy detection algorithm with better performance.

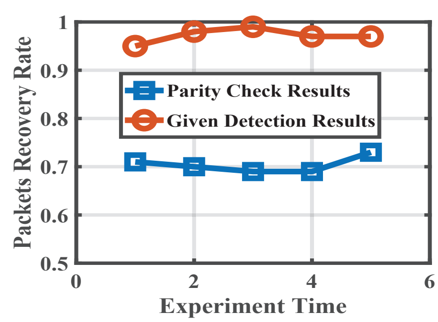



Fig. 2. Feasibility verification of fuzzy detection

**Benefit of Low Recovery Delay** Received Signal Strength Indication (RSSI) is an indication of the strength of the received signal. The algorithm based on cloud requires gateways to upload RSSI sample to the cloud for detecting errors. However, in LoRa protocols, RSSI is eight times the length of the payload (e.g. 200 bytes for a 25-byte payload packet [2]). Compared to the payload, the RSSI sample is very long, even after compression.

In terms of computational time consumption, the time consumption of error fuzzy detection is linearly related to the amount of the data transmissions and data packet damage degree. Compared with sending RSSI to recover data packets, using fuzzy detection codes takes less recovery time. In terms of communication time consumption, FDR reduces the time consumption for sending the RSSI to the cloud.

TABLE II
THE RELATIONSHIP BETWEEN N, R AND REDUNDANCY RATE

| $r$ | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|
| $n$ | 2-4 | 5-11 | 12-26 | 27-57 | 58-120 |
| $x$ | 42.86% | 26.67% | 16.13% | 9.52% | 5.51% |

## III. MAIN DESIGN

In this section, we first describe the overview of FDR, then explain the key components of FDR, i.e., fuzzy detection and data recovery, respectively.

### A. Overview of FDR

FDR is a signal recovery algorithm based on software. It can recover the corrupted packets due to interference, especially in the case of strong interference or weak signal strength.

Fig.1 illustrates the structure of LoRaWAN. The FDR recovers packets based on the LoRaWAN structure. Before the LoRa signal is sent, FDR adds fuzzy detection codes to the physical payload for detecting errors. Multiple gateways within the coverage of the LoRa receive the interfered data packets. Then the gateways detect the packets and upload the fuzzy detection results to the cloud. The cloud server restores the corrupted packets with a recovery algorithm called comparing replacement.

### B. Fuzzy Detection for Errors

First, we need to know what fuzzy detection is. When we detect errors in packets, if we get the specific position of error, it is called accurate detection. For example, the error occurs in the 21st bit of the payload. On the contrary, if we get the approximate position of the error, it is called fuzzy detection. For example, an error occurs between the 10th and 21st bits of the payload.

In order to realize the fuzzy detection for errors, we first design the Fuzzy Detection Codes, which draw on the design principle of Hamming codes. FDR adds error fuzzy detection codes in the LoRa physical payloads so that the gateway can identify whether a received packet is corrupted before decoding.

We calculate the number of fuzzy detection codes using the formula: $2^r \geq n + r + 1$ where r is the num of detection bits and n is the num of data bits. For example, when the data bits are 7, r should be 4. This means that 7 bits of data require at least 4 bits of detection codes to detect errors. If we increase the number of detection codes, redundancy will be increased. Table II shows the relationship between $n$, $r$ and redundancy rate: $x$.

Fig.3 shows the structure of the payload after we insert the detection codes. Here we divide the payload of the data packet into several 7-bit data segments, and each data segment is divided into 4 fuzzy detection groups after inserting 4-bit error fuzzy detection codes. The fuzzy detection codes are inserted at $2^n$-th, where $n$=1,2,3,...$r$. In this example, they are 1,2,4,8,respectively and each bit represents for one fuzzy detection group(e.g.$G_1$-$G_4$). Group $G_k$ indexes for the bits

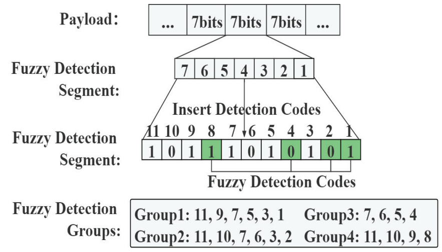



Fig. 3. The packet structure partition of fuzzy detection algorithm

which are located in which the k-th bit of binary representation is 1. As for the optimal length of the fuzzy detection section, we will explain it in the IV part.

After demodulating the signal, the gateway utilizes fuzzy detection codes to detect corruption before decoding packets. We design that '1' is used to represent for error, while '0' for correct. FDR takes advantage of this to detect fuzzy error positions. For example, if the $bit_j$, where,$j$=1,2,...11 is wrong, the correct and wrong situation of the $Fuzzy\ Detection\ Group_i$ and $Fuzzy\ Detection\ Segment$ are obtained by the following equation:

$$\begin{cases} \mathbf{P_k} = \{c | c \in [t \times 2^k, t \times (2^{k+1} - 1)], c < n\} \\ G_i = \bigoplus_{j=1}^{\mathbf{P_k}} C_j \\ S = \sum_{i=1}^{r} G_i \end{cases} \tag{1}$$

Here $P_k$, $G_i$, $C_j$, $S$, $r$ represent the index collection of $G_k$, the correct and wrong condition of $i$-th Fuzzy Detection Group, the $j$-th bit in Fuzzy Detection Code, the Fuzzy Detection Segment and the num of detection code respectively. $t$ is a positive odd number.

When FDR works, once it finds errors in a group, it immediately ends the detection of the segment to which the wrong group belongs. If any one group in segment $k$ is wrong, the segment with the index $k$ will fail the error detecting. After all the segments are detected, the fuzzy detecting results will be utilized to recover packets.

To sum up, when a fuzzy detection group finds an error, the whole data segment is marked with '1', which indicates that there is an error in this segment. After all the detection of segments in the data packet is completed, the packet and detection result is transmitted to the cloud for recovery. The cloud utilizes comparing replacement algorithms to recover the data packets. Next, we explain how comparing replacement algorithm works.

### C. Package Recovery in the Cloud

Gateways upload the fuzzy detection results to the cloud through reliable Ethernet connections during which LoRaWAN utilizes 128 bits AES for integrity protection and data encryption. Each detection result is stored in an array. Then the cloud recovers the data packets by comparing replacement algorithm, which is signed according to the equation:

$$C_n[j] = \sum_{k=0}^{m-1} (P_n[j] == P_n[k])$$

TABLE III
COMPARING REPLACEMENT

| | | | | | |
|---|---|---|---|---|---|
| **Packet: from Base1** | **Segment1** | **Segment2** | **Error** | **Error** | **Segment5** |
| **Packet: from Base2** | **Error** | **Segment2** | **Error** | **Segment4** | **Error** |
| **Packet: from Base3** | **Error** | **Segment2** | **Segment3** | **Error** | **Segment5** |
| **Packet: from Base4** | **Segment1** | **Segment2** | **Error** | **Segment4** | **Segment5** |

Here $C_n[j]$ represents for the num of the segments which are same as $j$-th segment in $n$-th packet, $P_n$ represents for the $n$-th packet. The $V_n$ can be calculated by the equation:

$$V_n = (1 == R_n[j]) \times C_n[j]$$

Here $V_n$ represents for the value of the one segment in $n$-th packet, $R_n$ represents the detection result of the $n$-th packet.

The packets are uploaded by multiple gateways. Then the cloud utilizes these packets and replaces the wrong data segment with the right data segment based on the detection results. Taking four data packets with different interference positions and degrees as an example, the detection results are shown in Table III.

After observation, we find that when packets have been seriously interfered, the four packets may report the error in the same segment, which requires more packets to recover errors. However, if there are not enough gateways, we utilize the error correction capability of fuzzy detection codes to correct one-bit error. In each segment, the wrong bit position is calculated using the following equation:

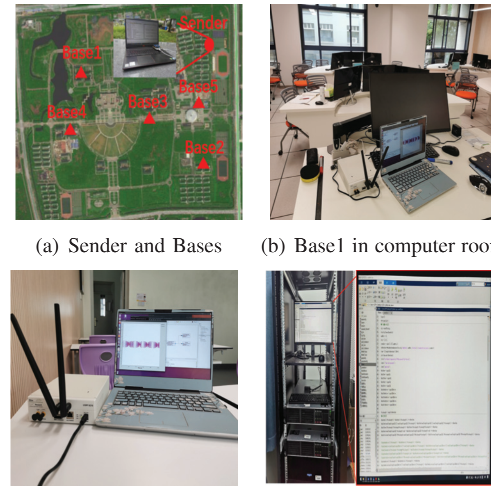
$$P = \sum_{i=0}^{r}(H(2)_i \times 2^k)$$

Where the $P$ represents for the position of the error bit, $H(2)$ for the binary representation of fuzzy detection codes.

## IV. Performance Evaluation

In this part, we first design an experiment to explore the optimal fuzzy detection segment length with the known packet length. And then it is extended to the length selection of fuzzy detection segment under different levels of interference. After we select the optimal fuzzy detection segment length, we evaluate the performance of FDR, such as packet recovery rate and recovery time overhead.

In our experiment, we deploy the sender and gateways at The Jiulonghu campus of Southeast University. As shown in Fig.4, we place the signal sender in the open space, and the gateways are respectively placed in the computer room (base1), dormitory (Base2), teaching building (Base3), work and training center (Base4) and meeting room (Base5). We also conduct extensive experiments with emulation. Our evaluations include different situations of Wi-Fi interference. To ensure limit-testing the performance of FDR, we emulate In-Phase/Quadrature (I/Q) of LoRa packets using LoRaMatlab [14] and utilize WLAN Waveform Generator [15] to generate Wi-Fi packets as interference. We control the levels of interference by extending the time of WiFi packets.




(a) Sender and Bases (b) Base1 in computer room

(c) Base3 in buildings (d) The cloud server

Fig. 4. Commodity devices and the cloud server

According to Table II, we choose segments with lengths of 4, 11, 18, 26, 57, and 120. We transmit 40 bytes of payload and apply about 15 error bits interference. The experiment results are shown in Fig.5. With the increase of segment length, the symbol error rate shows a rapid growth trend. When the segment length reaches 26 bits, the symbol error rate exceeds 90%. We analyze this phenomenon in theory. In the urban environment, about 89% of the 40-byte payloads have errors less than 15 bits, accounting for 3.6% of the total packet length. At the same time, these interferences are very evenly distributed in the whole packet [2]. When the segment length is 26 bits, the payload can be divided into 13 segments on average and almost all segments are detected as errors. However, although shortening the segment length decreases the symbol error rate, it leads to a higher redundancy rate. For example, when the segment length is 4 bits, the detection code length should be 3 bits, which increases the redundancy of the payload by 75%. This directly increases the signal demodulation time by 20.72%. Therefore, we need to find a balance between time cost and data recovery.

When the detection code is 4 bits long and 11 bits long, the symbol error rate and demodulation time overhead are acceptable. When the detection code length is 5 bits, the segment length is at least 18 bits, so that the redundancy rate can be as low as when the detection code length is 4 bits. But now the symbol error rate has reached 49.53%. Theoretically, we subtract the number of error segments from the number of payload segments (each segment has an average error of 1 bit) and then multiply the result by the number of gateways that send packets. As long as the result is larger than the number of segments in the payload, the low symbol error rate can be satisfied. Then we take the maximum length of this segment to obtain the minimum redundancy that meets the recovery requirements. The equation is: $(\lceil l/n \rceil - l \times x) \times m \geq \lceil l/n \rceil$. Here $l$, $n$, $x$, and $m$ represent the length of the payload, length of the segment, error bits rate, and the num of gateways. The equation is simplified as follows: $n \leq (m-1)/(m \times x)$.

In the above example, according to the equation, the segment length should be less than 14.6 bits, but in order to ensure the minimum redundancy rate, the segment length should be 11 bits. This is consistent with our experiment results.

We utilize the conclusion derived from the formula as the condition of the following experiment. In order to evaluate FDR performance in different aspects, we choose four evaluations indicators: (i) the effect of LoRa's spread-spectrum factor (SF) on LoRa and FDR recovery time; (ii) The influence of payload length on OPR and FDR data recovery time; (iii) The impact of the number of gateways on the data packet recovery rate; (iv) The influence of signal interference degree on the data recovery time of LoRa and FDR.

### *A. Effect of LoRa Spread Factor on the Recovery Time*

By testing the effects of different spread spectrum factors on packet recovery time, we explore whether the FDR algorithm can maintain great time performance in different scenarios.

Fig.6 shows the packets decoding time of standard LoRa and FDR when the SF increases. The time consumption of LoRa and FDR increased exponentially (observed in the bar chart) when the SF increases. We notice that with the increase of payload, the proportion of time consumption increased by FDR nearly remains unchanged. The results show that in the face of different SF, the FDR time performance increases by 9.35% compared with LoRa.

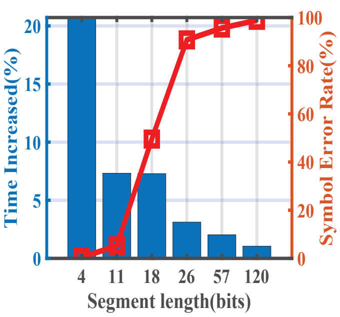



Fig. 5. Influence of segment lengths on error rate and recovery time

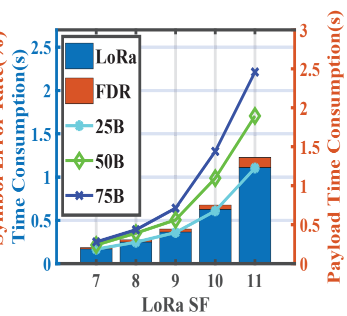



Fig. 6. Influence of LoRa spread spectrum factor on recovery time

### *B. Effects of Payload Length on Data Recovery Time*

Fig.9 compares the recovery time of LoRa and FDR for four payloads with different lengths. In the case of SF=9 and 15 fuzzy detection group errors, the data recovery time of LoRa and FDR is tested. The results show that with the increase of Payload, LoRa time and FDR time increase linearly. The average time of the FDR algorithm is 8.62% more than that of LoRa.This is consistent with the conclusion of our first experiment.

### *C. The Influence of the Number of Gateways*

In each position the gateways are tested indoors and outdoors. Fig 7 shows the recovery results in indoor and outdoor conditions. Notice that when the number of outdoor gateways is 4, FDR can recover 90% of the previously unacceptable packets. When the number is greater than or equal to 5, the data is completely recovered. Because of the stronger interference of electronic devices and wifi, it is more difficult to recover the signals received indoors than outdoors.

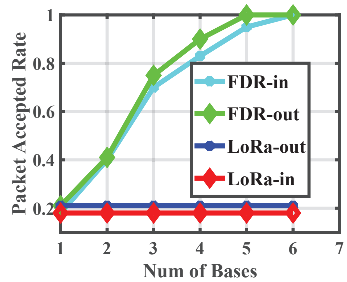



Fig. 7. Influence of the number of gateways on packet accepted rate

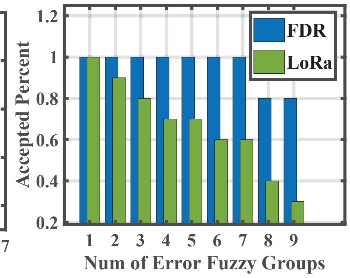



Fig. 8. Influence of the interference strength on packets accepted rate

### *D. The Influence of Signal Interference Strength*

We test the accepted rate of packets after recovery when the number of fuzzy detection group errors increases. We utilize the number of fuzzy detection groups errors to evaluate the degree of signal interference.

**Packet Accepted Rate**: As shown in Fig 8, the FDR maintains a 100% probability of packets correct recovery before the number of the fuzzy detection group errors reaches 22. Standard LoRa can recover a certain amount of data packets under a small degree of interference by itself. With the enhancement of interference degree, the accepted rate of data packets recovered by standard LoRa decreases rapidly.

From the point of view of time consumption, more specifically, we compare the time cost of fuzzy detection and precise detection, comparing replacement algorithm and voting weight algorithm. We select: SF=8, payload=75B.

**Error Detection Time**: Fig.10 shows that LoRa time increases linearly slowly with the increase of interference degree. Compared with LoRa, the average time spent by precise error detection increased by 17.35%, while the average time spent by FDR for error fuzzy detection increased by only 6.23%. And with the deepening of the interference degree, the increased cost fluctuates in a small range around these two values.

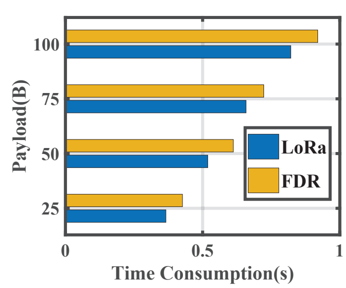



Fig. 9. Influence of payload length on recovery time

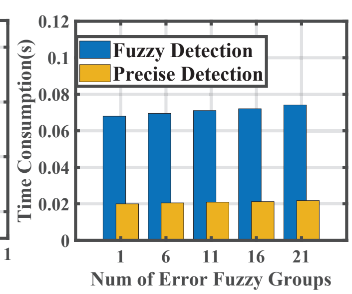



Fig. 10. Influence of the interference strength on error detection time

**Packet Error Recovery Time**: We selected two recovery methods, comparing replacement and weight voting. Fig.14 shows that weight voting increases 10.4% of the average time cost for LoRa, FDR in the cloud of comparison replacement algorithm to increase costs for 3.12% of the average time when the num of errors in the fuzzy detection groups increases.

**Average Recovery Time**: Fig.12 compares the LoRa, FDR, algorithm based on the cloud when the number of errors in fuzzy detection groups reaches 20, the average increase of FDR computing cost is 8.44% of decoding time of standard LoRa. FDR reduces the time cost by 78.53% compared with the algorithm based on the cloud.

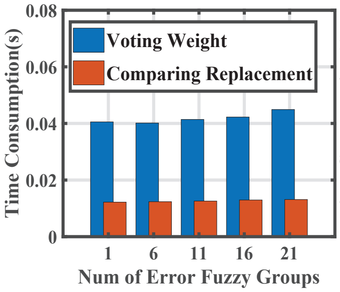



Fig. 11. The influence of the interference strength on error recovery time

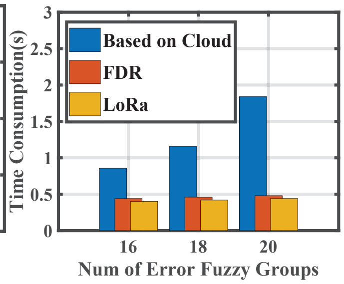



Fig. 12. The average recovery time of different methods

## V. Related Works

This section summarizes the work related to this paper. At present, the anti-interference work in LP-WAN can be divided into two categories: protocol modification methods and software modification methods.

**Protocol Modification Methods**:The methods based on physical layer modification include SCLoRa [8], Choir [16], FTrack [17], mLoRa [18] and so on. The methods based on MAC layer modification include S-MAC [19], LMAC [20], and so on.

**Software Modification Methods**: A cloud-based algorithm OPR sends RSSI samples and interfered data packets to the cloud. It does not need additional hardware deployment costs. However, the consumption caused by uploading RSSI samples is difficult to meet the demand of real-time recovery.

## VI. Conclution

This work presents the FDR, a real-time signal recovery algorithm based on fuzzy error detection. FDR takes the advantage of the powerful computational ability of the cloud and multiple gateways cooperation. Our experiments show that FDR achieves to recover packets accurately in the case of packets damage rate up to 45.72% and reduces the time consumption by 78.53% compared with the state-of-art. In the case of the ultimate damage rate of FDR recovery, FDR only increases the time consumption by 9.35% compared to the decoding process of standard LoRa. FDR is real-time, low-cost and easy to deploy, and it has reliable anti-interference ability.

## Acknowledgement

This work was supported in part by Science and Technology Innovation 2030 - Major Project 2021ZD0114202, National Natural Science Foundation of China under Grant No. 61902066, Natural Science Foundation of Jiangsu Province under Grant No. BK20190336 and Fundamental Research Funds for the Central Universities under Grant No. 2242021R41068.

## References


[1] Inc Gartner, "Gartner Report," URL:http://cloudtimes.org/2013/12/20/gartner-theinternet-of-things-will-grow-30 times-to-26-billion-by-2020/, 2016.

[2] Balanuta,A., Pereria,N., Kumar,S., and Rowe. A, "A cloud-optimized link layer for low power wide-area networks," Proceedings of the 18th International, Conference on Mobile Systems, Applications, aand Services, 247-259, 2020.

[3] Dongare, A., Narayanan, R., Gadre, A., Luong, A., Balanuta, A., Kumar, S., and Rowe, A, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 60-71, 2018.

[4] letreby, R., Zhang, D., Kumar, S., and Yağan, O, "Empowering low-power wide area networks in urban settings," Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 309-321, 2017.

[5] Hessar, M.,Najafi, A., and Gollakota,S. Netscatter, "Enabling large-scale backscatter networks," 16th USENIX Symposium on Networked Systems Design andImplementation (NSDI'19), 271-284, 2019.

[6] Xia, X., Zheng, Y., and Gu, T. "FTrack: Parallel decoding for LoRa transmissions," Proceedings of the 17th Conference on Embedded Networked Sensor Systems, 192-204, 2019.

[7] Gu, C., Tan, R.2021.LoRaWAN technology for enabling the new generation of internet of things [J]. Journal of Internet of Things, 2021,5(02),18-25.

[8] Hu, B., Yin, Z., Wang, S., Xu, Z., and He, T, "SCLoRa: Leveraging MultiDimensionality in Decoding Collided LoRa Transmissions," 2020 IEEE 28th International Conference on Network Protocols (ICNP) , 1-11, 2020.

[9] Wang, S., Yin, Z., Li, Z., and He, T, "Networking support for physical-layer cross technology communication," 2018 IEEE 26th International Conference on Network Protocols (ICNP), 259-269, 2018.

[10] Wang, S., Kim, S. M., Yin, Z., and He, T, " Correlated coding: Efficient network coding under correlated unreliable wireless links," 2014 IEEE 22nd International Conference on Network Protocols, 433-444, 2014.

[11] Jiang, W., Yin, Z., Liu, R., Li, Z., Kim, S. M., and He, T, "Bluebee: a 10,000 xfaster cross-technology communication via phy emulation," 15th ACM Conference on Embedded Network Sensor Systems, 1-13, 2017.

[12] Jiang, W., Yin, Z., Kim, S. M., and He, T, Transparent cross-technology communication over data traffic. IEEE INFOCOM 2017-IEEE Conference on Computer Communications, 1-9, 2017.

[13] Xu, Z., Luo, J., Yin, Z., He, T., and Dong, F, "S-MAC: Achieving high scalability via adaptive scheduling in LPWAN," IEEE INFOCOM 2020-IEEE Conference on Computer Communications, 506-515, 2020.

[14] MATHWORKS Help Center. https://ww2.mathworks.cn/help/. last accessed 2021/04/05

[15] WLAN Waveform Generator. https://ww2.mathworks.cn/help/wlan/. last accessed 2021/04/05

[16] R. Eletreby, D. Zhang, S. Kumar, and O. Ya.gan. "Empowering low power wide area networks in urban settings," Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 309-321, 2017.

[17] X. Xia, Y. Zheng, and T. Gu, "Ftrack: parallel decoding for lora transmissions," Proceedings of the 17th Conference on Embedded Networked Sensor Systems, 192–204, 2019.

[18] X.Wang, L. Kong, L. He, and G. Chen, "mlora: A multi-packet reception protocol in lora networks," IEEE 27th International Conference on Network Protocols, 1–11, 2019.

[19] Xu, Z., Luo, J., Yin, Z., He, T., and Dong, F. "S-MAC: Achieving high scalability via adaptive scheduling in LPWAN," IEEE INFOCOM 2020-IEEE Conference on Computer Communications, 506-515, 2020.

[20] Gamage, A., Liando, J. C., Gu, C., Tan, R., and Li, M. "Lmac: Efficient carriersense multiple access for lora," Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, 1-13, 2020.

[21] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in 2016 23rd International Conference on Telecommunications (ICT), May 2016, 1–6, 2016.

[22] Mei L., Yin Z., Zhou X., Wang S., Sun K, "ECCR: Edge-Cloud Collaborative Recovery for Low-Power Wide-Area Networks Interference Mitigation," 2021.