



Decoding LoRa Collisions via Parallel Alignment

YUTING WANG, FANHAO ZHANG, XIAOLONG ZHENG, LIANG LIU, and HUADONG MA,
School of Computer Science & Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia,
Beijing University of Posts and Telecommunications, China

The massive connection of LoRa brings serious collision interference. Existing collision decoding methods cannot effectively deal with the adjacent collisions which occur when the collided symbols are adjacent in the frequency spectrum. The decoding features relied on by the existing methods will be corrupted by adjacent collisions. To address these issues, we propose *Paralign* which is the first LoRa collision decoder supporting decoding LoRa collisions with confusing symbols via parallel alignment. The key enabling technology behind *Paralign* is there is no spectrum leakage with periodic truncation of a chirp. *Paralign* leverages the precise spectrum obtained by aligning the de-chirped periodic signals from each packet in parallel for spectrum filtering and power filtering. To aggregate correlation peaks in different windows of the same symbol, *Paralign* matches the peaks of multiple interfering windows to the interested window based on the time offset between collided packets. Moreover, a periodic truncation method is proposed to address the multiple candidate peak problem caused by side lobes of confusing symbols. We evaluate *Paralign* using USRP N210 in a 20-nodes network. Experimental results demonstrate that *Paralign* can significantly improve network throughput, which is over 1.46 \times higher than state-of-the-art methods.

CCS Concepts: • Networks → Sensor networks; Network protocols.

Additional Key Words and Phrases: Decoding LoRa collision, Confusing symbol, Parallel alignment, Spectrum filtering, Power filtering.

1 INTRODUCTION

As a promising technology for long-range communication, LoRa (Long Range) [12, 24, 29, 40] is widely applied to extensive Internet of Things (IoT) applications, such as smart agriculture [18, 30], environment monitoring [2, 20], warehouse management [10], etc. Such widespread deployment of LoRa applications is mainly due to the utilization of CSS (Chirp Spreading Spectrum) modulation which makes LoRa devices possible to have a high receiver sensitivity [4, 19]. Moreover, the star-of-stars topology applied in LoRa networks enables a large number of devices to connect to a common gateway.

While LoRa has been widely adopted, it is still challenging to deploy large-scale LoRa networks in practice, because extensive end devices connected to a same gateway will encounter severe signal collisions, leading to low throughput [15, 16, 25, 32]. To address this problem for LoRa, abundant collision resolution methods have been proposed in MAC and PHY layers. However, due to the limited energy budget and low-cost hardware, it is infeasible to design complex MAC protocols for LoRa devices to resolve conflicts [1, 7, 36]. Existing PHY layer methods mainly distinguish and separate collisions based on the features in the unique hardware[5], time-domain [31, 35], and frequency-domain [3, 23, 27, 28, 33, 38]. Besides, due to the consideration of the decoding features of

Authors' address: Yuting Wang; Fanhao Zhang; Xiaolong Zheng; Liang Liu; Huadong Ma, wangyuting@bupt.edu.cn, liangliu.mhd@bupt.edu.cn,
School of Computer Science & Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia,
Beijing University of Posts and Telecommunications, No.10 Xitucheng Road, Beijing, China, 100876.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.
1550-4859/2022/11-ART \$15.00
<https://doi.org/10.1145/3571586>

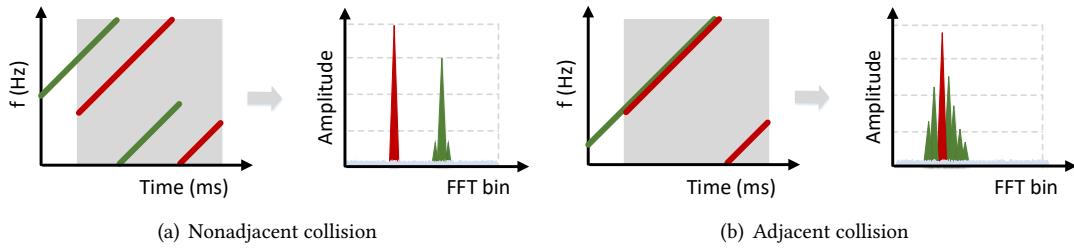


Fig. 1. Frequency relationships between collided symbols.

LoRa, the methods using frequency-domain features outperforms the methods using hardware and time-domain features.

However, a fundamental limitation of existing approaches based on the frequency domain features is that they rely on the precise differentiation of peaks in the frequency spectrum corresponding to the different collided symbols. As shown in Fig. 1(a), the peaks in FFT results can be accurately captured when a non-adjacent collision occurs. Hence, the existing approaches can work well in this case. But in the case of adjacent collision, the peak height and frequency of the spectrum are affected in Fig. 1(b). For instance, the interested symbol (red peak) is affected by the interference symbol (green peak) whose frequency is close to the interested symbol. We name the green ones as a **confusing symbol** to the red ones. Note that when the frequencies of two symbols are adjacent in the frequency spectrum, their collision is called an adjacent collision. Otherwise, it is a nonadjacent collision. As a result, the existing approaches that rely on the peak heights to separate collided symbols, such as CoLoRa [28], NScale [27], cannot match the correct peaks to the corresponding transmitters. Moreover, since the interfering symbols do not span the entire symbol duration, there will be spectrum leakage when performing FFT on the whole symbol. The methods that rely on the peak frequencies, such as CIC [23], cannot correctly capture the interested symbols. Moreover, the occurrence probability of adjacent collision caused by confusing symbols increases with the increase of concurrent packets. It is calculated that when the number of concurrent LoRa devices with SF8 increases to 20, the probability that collided symbols in the same demodulation window have confusing symbols increases sharply to 74.3%. Therefore, an efficient parallel decoding method should not ignore the adjacent collisions with confusing symbols. However, such a method is missing.

To tackle the adjacent collision, we compare the spectrums of the pieces of the original de-chirped signals, called sub-symbols, in Section 2.3. In particular, we observe that the corresponding heights of sub-symbols with different lengths in the spectrum are variant. That is, the peak characteristics of the sub-symbols are closely related to the length. Moreover, as long as the de-chirped signal is not periodically intercepted, the distortion caused by confusing symbols is inevitable. This is why existing methods that ignore signal periodicity cannot work well when an adjacent collision occurs. This observation motivates us to separate the collision packages by periodically intercepting LoRa chirp signals.

In this work, we propose *Paralign* to decode LoRa collisions with confusing symbols via parallel alignment. Since the de-chirp signal of complete chirp is a periodic signal in each reception window, *Paralign* obtains the accurate spectrum of correlation chirps by aligning the demodulation window with the symbols of each packet according to its boundaries at the same time for separating the collision packets. As shown in Fig. 2, although a single spectrum can be affected by the appearance of confusing symbols, after aligning three correlation windows at the same time, the peak of the interested symbol can be easily obtained through the spectrum intersection of parallel aligned windows.

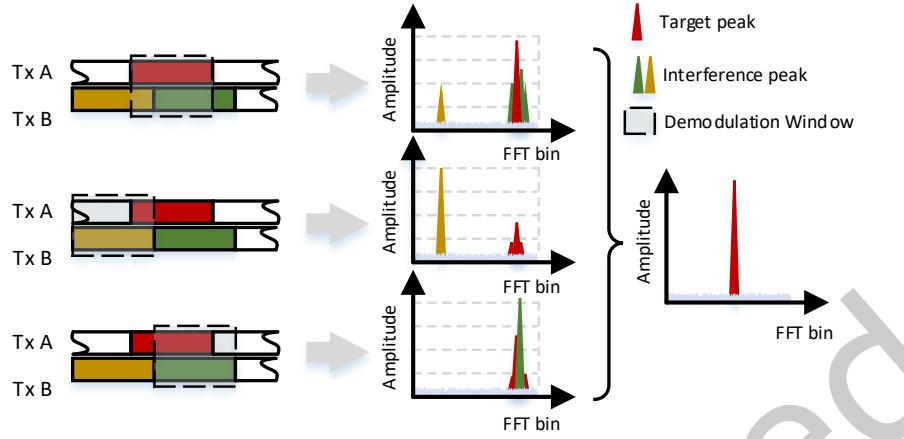


Fig. 2. Illustration of strawman *Paralign* that decodes collision of two packets via parallel alignment, where the red symbol is the interested.

Turning the basic idea of *Paralign* into reality needs to address three challenges. First, since the corresponding frequencies of the same symbol in different demodulation windows are different, it is difficult to correspond the correlated peaks obtained in the parallel demodulation windows to their associated symbols. To address this problem, we leverage the fact that the frequency shift of the same symbol is determined by the time offset between collided symbols. As such, *Paralign* makes the peak-to-symbol matches by using this time offset to aggregate the peaks of the interference windows into the interested window. Second, even if we can match the associated peaks to symbols, it is non-trivial to pick out the desired ones from multiple frequency peaks. Although spectrum intersection of multiple parallel windows can filter out most of the interference peaks, it is difficult to extract the unique interested peak by simple spectrum intersection due to the impact of the side lobes of misaligned confusing symbols. Nevertheless, by analyzing the spectrum of the window associated with the interested window, we observe that the interested symbol appears in the previous interfering window overlapped with the interested window but not in the previous window of the interested window. This means that the interested symbols are the spectrum difference set of the two windows. *Paralign* leverages such spectrum difference to further eliminate redundant peaks. Third, after spectrum filtering, we found that there may still be more than one candidate peak when the time offset between the confusing symbol and the interested symbol is small. To extract the interested symbols, *Paralign* designs a power filtering method. This is based on our key insight that more accurate spectrum peaks can be obtained by periodically intercepting chirp signals.

The main contributions of *Paralign* can be summarized as follows.

- We propose *Paralign* which is the first LoRa collision decoder supporting decoding LoRa collisions with confusing symbols via parallel alignment to the best of our knowledge. The critical insight of *Paralign* is spectrum leakage can dissipate with the integer periodic truncation of a chirp. *Paralign* can extract the precise spectrum of correlation chirps by aligning the de-chirped periodic signals from each packet in parallel.
- To extract the interested symbols, *Paralign* performs spectrum filtering, including spectral intersection and difference, and power filtering in the frequency domain. For the multiple candidate peaks problem caused by side lobes of confusing symbols, *Paralign* selects the interested peaks by the periodic truncation method, i.e., selecting a more stable peak in the spectrum of sub-symbols with integer periods.

- We build a prototype system with 20 commodity off-the-shelf LoRa devices and extensively evaluate the performance indoors and outdoors. Our evaluation results show that *Paralign* can separate collided packets with a small Symbol Error Rate (SER) and improve network throughput by 1.46 \times compared with CIC and 2.33 \times compared with CoLoRa.

The rest of this paper is organized as follows. In Section 2, we introduce the limitations of the existing works and the opportunities of decoding collisions with adjacent symbols, which motivates our work and presents the designs of *Paralign* in Section 3. Evaluation results are shown in Section 4. We discuss the related works and discussion in Section 5 and Section 6. Finally, we conclude this paper in Section 7.

2 PRELIMINARY STUDY

In this section, we first briefly introduce the modulation and demodulation of LoRa and then present the limitations of existing approaches for solving LoRa collision.

2.1 LoRa Modulation and Demodulation

LoRa adopts Chirp Spread Spectrum (CSS) modulation scheme in the physical layer. Symbols are modulated as up-chirp (down-chirp) signals whose frequencies increase (decrease) linearly with time. LoRa varies the initial frequency of an up-chirp to modulate different symbols [37]. The signal of symbol s can be represented as follows.

$$S(t, f_s) = e^{j2\pi(f_0 + \frac{kt}{2})t} \cdot e^{j2\pi f_s t} = C(t) \cdot e^{j2\pi f_s t}, \quad (1)$$

where $C(t) = e^{j2\pi(f_0 + \frac{kt}{2})t}$ is the base chirp and $f_0 = -BW/2$ is the starting frequency of the base chirp. $k = BW/T$ denotes the frequency increasing rate of the chirp and f_s denotes the initial frequency of the encoded symbol s . $T = 2^{SF}/BW$ represents the length of the chirp.

To obtain the initial frequency f_s of the symbol s , the LoRa receiver first multiplies the received signal with the conjugate of the base chirp denoted as $C^*(t)$. The procedure can be written as follows.

$$S'(t, f_s) = C^*(t) \cdot S(t, f_s) = C^*(t) \cdot C(t) \cdot e^{j2\pi f_s t} = e^{j2\pi f_s t}. \quad (2)$$

After that, LoRa receiver performs Fast Fourier Transform (FFT) on the result, i.e., $F(f_s) = FFT(S'(t, f_s))$. The frequency response of symbol s can be obtained by searching for the highest peak.

2.2 Limitations of Prior Art

Due to the dense deployment, there will inevitably be signal collisions when multiple LoRa nodes transmit packets simultaneously. Existing techniques decode multi-packet collision by matching the detected peaks to their corresponding transmitters based on peak heights, such as CoLoRa [28], NScale[27]. In practice, with the increase of packet collisions, using the peak height as a reliable metric to match transmitters is in question. To explore the problem, we simulate randomly generated symbols and observe the height of the peaks of collided symbols. The setting is SF8 and the chirp-level time offset between two collided packets is $0.25T$, where T represents the length of a chirp.

As shown in Fig. 3, we surprisingly find that the peak heights of the same symbol colliding with different symbols are different. Specifically, in nonadjacent collisions, the ratio of the peak height of red symbol in the left demodulation window to that of the right is 1.00, which conforms to the theory that the height is proportional to segment length. Similarly, the ratio of the peak heights of green symbol is $1.50/0.50 = 3$. Nevertheless, in the adjacent collision, the peak height ratio of red symbol is $0.7200/1.136 = 0.634$, which is 0.634 \times that in nonadjacent collision. Similarly, the peak height ratio of green symbol of adjacent collision is 0.2549 \times that of nonadjacent collision. The above analysis reveals that the peak height of a symbol will be affected by the symbols whose frequencies are close to its frequency. For this reason, the existing methods that rely on the peak heights to

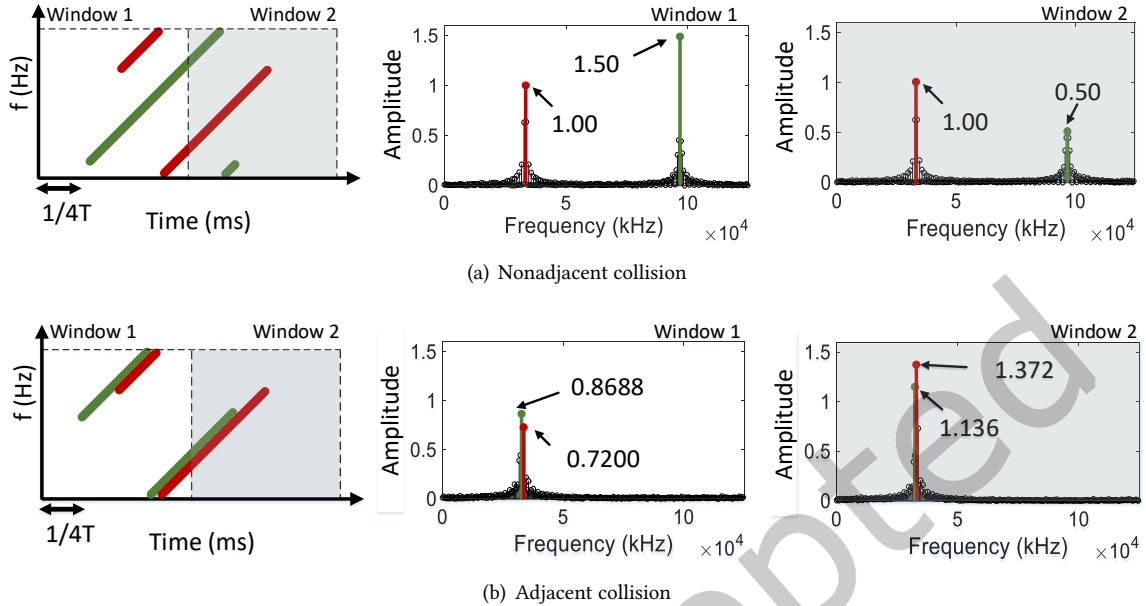


Fig. 3. The FFT results in different demodulation windows.

separate collided symbols, cannot match the correct peaks to the corresponding transmitters when an adjacent collision occurs.

However, how close are two symbols apart will affect the peak height of the interested symbol? To explore this problem, we perform FFT on the de-chirp signals of aligned and misaligned symbols to observe the spectrums, respectively. As shown in Fig. 4, there are several black peaks (peaks of misaligned symbols) of different sizes on both sides of the red peak (peaks of aligned symbols). In other words, after FFT, there are peaks not only at the applied frequency but also at either side of the applied frequency, called side lobes. We name these two symbols as **confusing symbols** of each other when the FFT result of one symbol is within the side lobe of the FFT result of another symbol.

In practice, due to the massive connection of LoRa devices, the probability of confusing symbols occurring increases sharply. In the event of an N packet collision, $2N - 1$ frequencies are included in the de-chirped signal, when the interested symbol is completely aligned with the demodulation window. The probability that the interested symbol will not be confused by the interfering symbol is

$$P(N, D) = \frac{2^{SF} \cdot [2^{SF} - (2D + 1)]^{2N-2}}{(2^{SF})^{2N-1}}, \quad (3)$$

where D is the maximum symbol distance where the interfering symbol is confused with the interested symbol, i.e., maximum distance of symbol confusion. Thus, the probability that the interested symbol is confused by the interfering symbol is $P'(N, D) = 1 - P(N, D)$.

Fig. 5 shows the probability P' when the maximum distance of symbol confusion is 0, 1, 2, 3, and 4 for SF=8, where $D = 0$ means that except for being the same as the interested symbols, other symbols are not confused with it. When the number of concurrent nodes is 20, for $D = 3$ and $D = 4$, the probabilities of the existence of

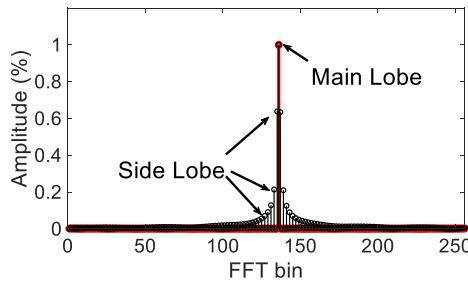


Fig. 4. Sidelobe illustration. The two symbols are confusing symbols with each other when the peak of one symbol is within the side lobe of the FFT result of another symbol.

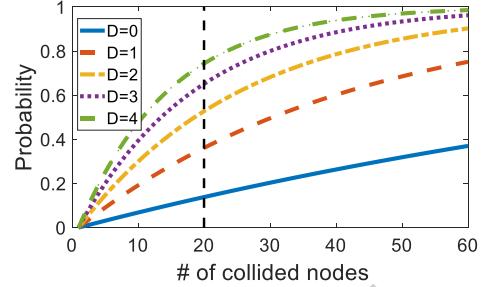


Fig. 5. Probability of existence of confusing symbols, P' , in a demodulation window for SF8, where D is the maximum symbol distance where interfering symbols and interested symbols may be confused.

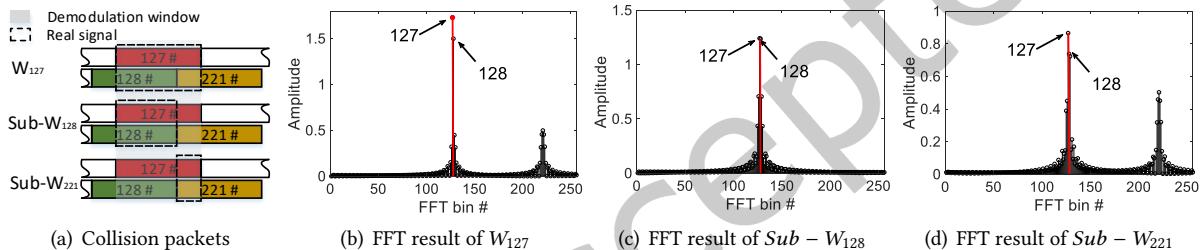


Fig. 6. Spectral unreliability of adjacent collision where 127#, 128#, and 221# represent the FFT bin of the corresponding symbol in the spectral domain.

confusing symbols are 65.1% and 74.3%, respectively. Thus, the occurrence of confusing symbols is not a single case, and the probability of this phenomenon increases with the increase of concurrent packets. Such a high probability can seriously affect the decoding of collided packets.

Then, we may have a question about whether using peak frequencies instead of heights can avoid the effect of confusing symbols. CIC [23] decomposes superimposed symbols by selecting a specific set of sub-symbols to cancel out all other interfering symbols. In these sub-symbols, none of the interfering symbols is common. CIC extracts the common frequency as the interested symbols by estimating spectrum intersection for all sub-symbols. However, as stated in Heisenberg's time-frequency uncertainty principle [26], frequency estimation on signals with a short time-span will lead to poor frequency resolution. This inevitable loss of frequency resolution caused by the cropped signals will cause more than one peak of CIC at the spectrum intersection, especially in the presence of confusing symbols. That is, as long as there are adjacent frequencies in a demodulation window, symbol confusion will occur. Fig. 6 shows a case of decoding failure. We can find that after the spectral intersection of all these spectrums, more than one potential candidate peak (i.e. FFT bin 127 and 128) remains.

To pick the interested peak among them, CIC puts forward a Spectral Edge Difference (SED), i.e, calculating the absolute difference of candidate spectrum energy between the left and right half parts. Then the peak with the least value of SED is selected. However, such blind interception of sub-symbols ignores the periodicity of the signal and yields unreliable peak heights in the spectrum. Coupled with the previous use of an unreliable spectrum to obtain candidate peaks, this unreliable amplitude further exacerbates the probability of incorrect

decoding. Therefore, to support the decoding of adjacent collisions, the accurate acquisition of the height and frequency of the interested symbol is a foremost prerequisite.

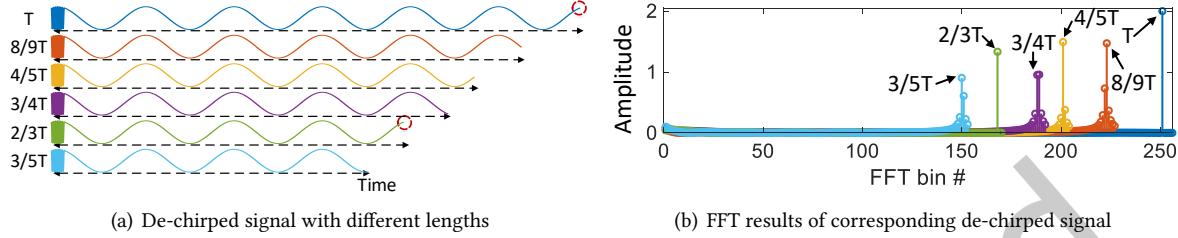


Fig. 7. Spectrum analysis of the de-chirped signal.

2.3 Opportunities for Decoding Collisions

From Eq. 2, we can find that the received signal multiplied by the base down-chirp becomes a periodic signal with one tone frequency in each reception window, i.e., the initial frequency of the interested symbol. Specifically, the specific frequency of the de-chirped signal of symbol s is $f_s = s \cdot BW / 2^{SF}$, and the signal period (N_s) calculated by sampling points is

$$N_s = T_s \cdot F_{samp} = \frac{1}{f_s} \cdot F_{samp} = \frac{2^{SF} \cdot F_{samp}}{s \cdot BW}, \quad (4)$$

where F_{samp} is the sampling rate and T_s represents the period of de-chirped signal of symbol s .

To explore the spectral characteristics of sub-symbols, we perform FFT on sub-symbols with different lengths. As shown in Fig. 7, the frequency and height of the peaks will change accordingly when the length of the sub-symbol changes. Specifically, when the sampled signal is not periodically truncated (i.e., $8/9T$, $4/5T$, $3/4T$, $3/5T$), the FFT results show larger distortion with unpredictable spurious components and spectral leakage, while when the signal is periodically truncated (i.e., T , $2/3T$), there is no leakage in the spectrum. Besides, as stated in [11], periodic truncation is a sufficient and necessary condition for no spectrum leakage and non-periodic truncation is a sufficient and necessary condition for spectrum leakage. However, the existing methods ignore the periodicity of the de-chirped signal and encounter serious sidelobe effects caused by spectrum leakage.

Fortunately, the de-chirp signal of complete chirp has integer periods. Fig. 8 shows the FFT results of the chirps modulated by symbol 1 and symbol 8 at a signal-to-noise ratio of 10 captured in a real experiment when BW is 125kHz and SF is 7. We can find that there is no spectrum leakage in the FFT results on the completed chirp. This observation motivates us to reduce the impact of spectrum leakage by parallel alignment and periodic truncation.

3 DESIGN

Based on the opportunities in Section 2, we design a novel LoRa collision decoder supporting decoding LoRa collisions with confusing symbols via parallel alignment. We first present an overview of *Paralign* and then introduce the major modules.

3.1 Overview

Paralign's design goal is decoding the concurrent collisions with confusing symbols via parallel alignment. The key observation is that periodic truncation is a sufficient and necessary condition for no spectrum leakage.

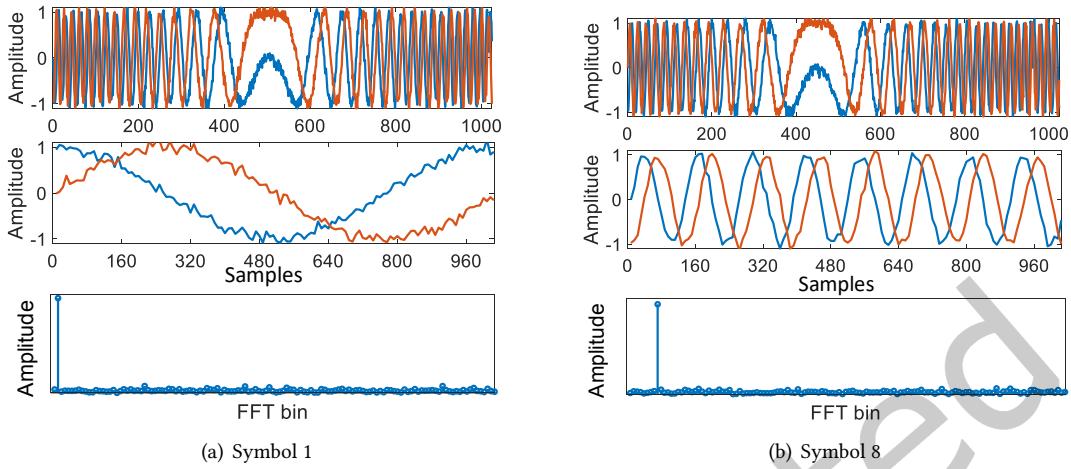


Fig. 8. Periodicity of chirps modulated with symbols 1 and 8 captured in a real experiment when BW is 125kHz and SF is 7. The modulated signal, de-chirped signal, and corresponding FFT results are represented from top to bottom.

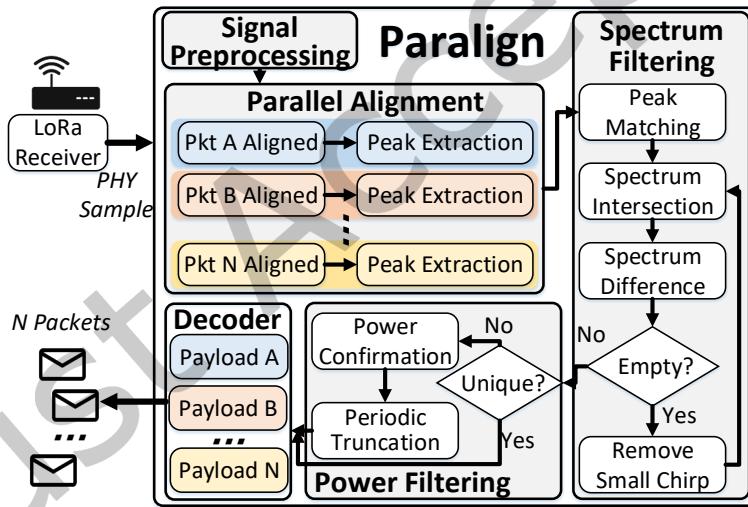


Fig. 9. Workflow of Paralign.

Fig. 9 shows the workflow of *Paralign*. *Paralign* is implemented at the base station and mainly consists of four components: signal preprocessing, parallel alignment, spectrum filtering, and power filtering.

Signal Preprocessing: *Paralign* first detects the preamble existence of LoRa packets by Start Frame Delimiter (SFD) of 2.25 down chirps. Then *Paralign* applies a preamble up-chirp and SFD down-chirp to estimate and compensate Carrier Frequency Offset (CFO) and Sampling Timing Offset (STO). Based on the compensated preamble of detected packets, the time offset of symbol boundaries between collided packets can be captured.

Parallel Alignment: Next, *Paralign* divides the received signal into demodulation windows with a length equal to a chirp in parallel based on the boundaries of each packet. The signal for each window is multiplied by base down-chirp and applied FFT to extract peaks regarding the height of the preamble of interested packets.

Spectrum Filtering: Since the corresponding frequencies of the same symbol in different demodulation windows are different, *Paralign* matches the peak set of multiple interfering windows to the interested window based on the time offset between collided symbols. Next, *Paralign* discovers the potential candidate peaks across all correlation parallel alignment windows by spectral intersection and spectral difference.

Power Filtering: Due to the influence of side lobes of misaligned confusing symbols, more than one candidate peak may remain. For this, *Paralign* utilizes power confirmation to roughly filter the redundant peak. Then, the interested peak is extracted by periodic truncation.

After decoding the target packets, *Paralign* repeats the spectrum filtering and power filtering operations to iteratively decode other concurrent packets. Besides, to better understand *Paralign*, we summarize the notations that are used in the following theoretical introduction in Table 1.

Table 1. Summary of the notations used in the paper.

Notation	Definition
W_i	Demodulation window of the i th packet
s_i	Start sampling points of window W_i
e_i	End sampling points of window W_i
T	Length of the demodulation window
τ_{ab}	Time offset between symbol a and symbol b
Φ_a	Peak sets of demodulation window W_a
P_a	Spectral intersection of correlation windows
H	Peak height when the chirp is aligned with the demodulation window
τ_h	Height variation caused by time offset τ_t
n_s	The number of periods of the de-chirped signal

3.2 Signal Preprocessing

3.2.1 Packet Detection. Preamble detection is usually used to detect ongoing LoRa transmissions by identifying a sequence of eight consecutive peaks with the same frequency and height in the spectrum. This is effective when all data symbols and synchronization words of interfering packets have a frequency offset from the preamble of the interested packets. Once the confusing symbols of the preamble appear, the height of preamble peaks will be changed, resulting in failed preamble detection. Fig. 10 shows an example of the peak height error of the preamble. If eight consecutive peaks with the same frequency and height are still detected to identify the preamble, it will lead to false-negative, i.e. missing the preamble. Specifically, due to the influence of confusing symbols, the peak height of the 5th, 6th, 7th, and 9th demodulation windows are inconsistent with the theoretical peak height (i.e. 2th, 3th, and 4th demodulation windows). If only continuous peaks of the same frequency are considered, the payload symbols of the same frequency in the 10th and 11th demodulation windows will also interfere with the preamble detection. To correctly detect the preamble, we exploit the specific frame structure of SFD composed of 2.25 down-chirps before the start of payload.

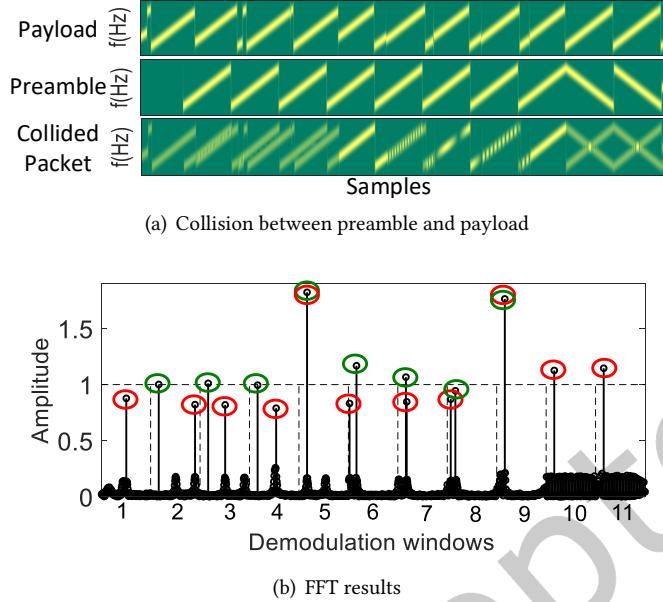


Fig. 10. Failed preamble detection because of wrong peak height and additional same peak frequency. Black dashed lines indicate the demodulation windows edges. The green circles represent the preamble peaks, and the red circles represent the data peaks of interference packets.

3.2.2 CFO and STO Elimination. Due to manufacturing imperfections, there will be a small frequency deviation between the transmitter and the receiver, i.e., CFO and STO, which can result in the frequency shift. Fortunately, the symbols of a same packet share the same CFO and STO. Similar to the existing approaches [27, 28], we eliminate CFO and STO based on the fact that the resulted correlation peaks of up-chirps and down-chirps caused by these manufacturing imperfections shift oppositely. As CFO and STO are determined, *Paralign* compensates for each packet.

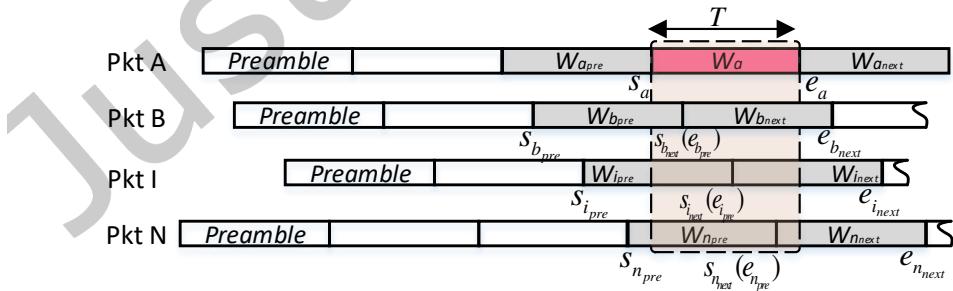


Fig. 11. Symbol boundaries in a collided packet.

3.2.3 Symbol Boundary Detection. After identifying preambles of collided LoRa packets and compensating received signals for CFO and STO, the LoRa receiver can accurately locate the boundaries of packets to demodulate

the concurrent transmissions. Based on the unique structure of LoRa packets, then, the symbol boundaries of the packet's payloads can be obtained. *Paralign* labels the interfering packets based on the time offset between the interfering symbols and the interested symbol. As shown in Fig. 11, in an N packet collision, since symbol boundaries of interfering packets are not aligned with the demodulation windows, the interested window W_a overlaps partially with two consecutive windows from each interfering packet - $W_{i,pre}$ and $W_{i,next}$ from packets $\{B, C, \dots, N\}$. Here, s_a and e_a are the start and end sampling points of window W_a , respectively.

3.3 Parallel Alignment

Based on the symbol boundaries of packet's payloads, the received signal is divided into consecutive demodulation windows with a length equal to a chirp. The signal for each window is multiplied by base down-chirp and applied FFT to observe the spectral features. Based on this spectral information, *Paralign* then seeks out the peaks of the interested symbols. Intuitively, the peaks of interested symbols can be extracted based on the height of the preamble of interested packets. While in practice, due to the time-limited input sequence, heights and even frequencies for peaks are susceptible to interfering symbols. For this, different from the previous demodulation method, *Paralign* divides the window according to the symbol boundaries of each collided packet. As shown in Fig. 11, we can obtain N misaligned parallel demodulation window sets, which is so-called **parallel alignment**.

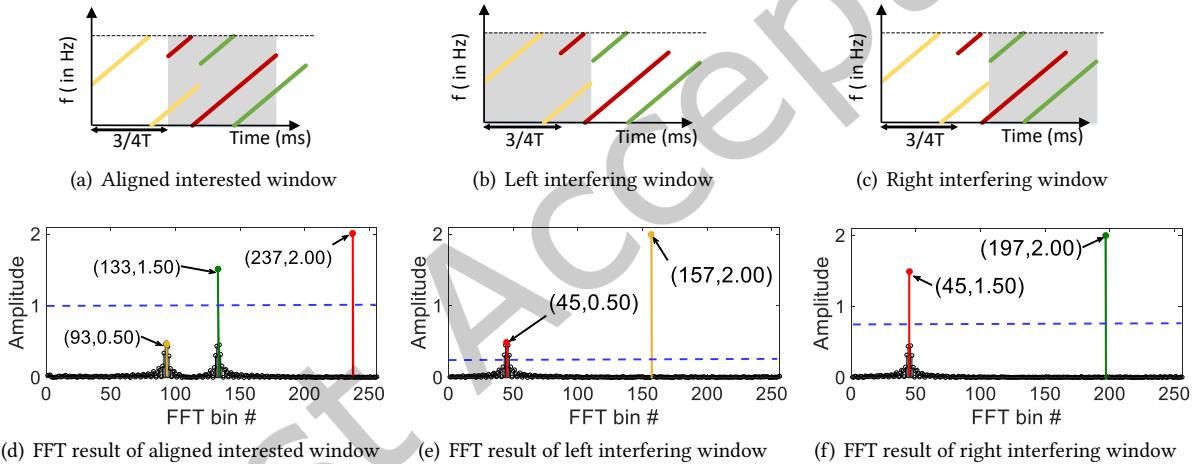


Fig. 12. An example of peak extraction in different demodulation windows. The settings are $SF = 8$ and $BW = 125\text{kHz}$ and the time offset between the two packets is $0.25T$. (a)(b)(c) Spectrogram of collided packets. (d)(e)(f) FFT results corresponding to the demodulation windows.

To extract possible interested peaks, we filter peaks whose heights are lower than the theoretical value by more than $3dB$, which is called power extraction. It is worth noting that $3dB$ represents the point at which the amplitude drops to half of its theoretical level. And the theoretical value is the product of the proportion of the interested signal in the demodulation window and the amplitude of the preamble. Fig. 12 shows an example of peak extraction under different demodulation windows, where the settings are $SF = 8$ and $BW = 125\text{kHz}$ and the time offset between two collided packets is $0.25T$. Specifically, in Fig. 12(d), the theoretical height of the interested red peak is 2. *Paralign* extracts all peaks with a height greater than 1 as candidates. Note that we don't filter symbols whose received power is higher than that of the preamble. Because as the increase of the number of concurrent nodes, there are likely multiple confusing symbols in the demodulation window, which leads to

an abnormally high peak of interested symbol. In our evaluation, we examine the improvements introduced by power extraction.

3.4 Peak Matching

After parallel alignment, the peaks of each corresponding window can be obtained for further spectrum operation. Nevertheless, the corresponding frequencies of the same chirp are distinct in different demodulation windows. In practice, how to accurately aggregate different peaks of the same symbol is a challenge. *Paralign* leverages the fact that the frequency shift of the same symbol is determined by the time offset between collided packets in interfering and interested windows. This process of gathering the peaks of multiple interfering windows to the interested window is called **peak matching**.

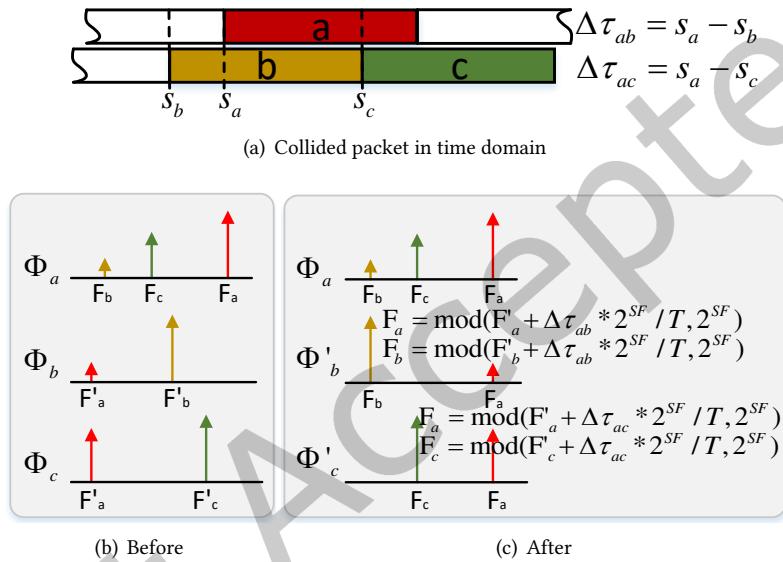


Fig. 13. Peak matching where F_a , F_b , F_c , F'_a , F'_b , and F'_c are the specific bins of corresponding symbols.

Target of peak matching is to translate time offset to frequency shift to map the peaks of the interfering demodulation windows to the interested window. Specifically, after parallel alignment, the segments of each chirp are divided into two demodulation windows. Ideally, the frequency shift of the peak belonging to a same symbol in another demodulation window can be calculated as

$$\Delta f = \frac{\Delta t}{T} \cdot BW, \quad (5)$$

where T represents the length of the chirp. BW is bandwidth and Δt is the time offset between two demodulation windows.

As shown in Fig. 13, the time offset between interference symbol b and the interested symbol a is $\Delta\tau_{ab} = s_a - s_b$. For ease of understanding, we convert the frequency into a specific bin value, i.e., $N_{\text{bin}} = f/(BW/2^{\text{SF}})$. After peak matching, the peak bin corresponding to interfering symbol b in Φ'_b is $F_b = \text{mod}(F'_b + \Delta\tau_{ab} \cdot 2^{\text{SF}}/T, 2^{\text{SF}})$, where F'_b is FFT bin of symbol b in Φ_b . Similarly, the peak bin of the interested symbol a is $F_a = \text{mod}(F'_a + \Delta\tau_{ab} \cdot 2^{\text{SF}}/T, 2^{\text{SF}})$. Take Fig. 12 with SF8 as an example, the bin of yellow chirp in Fig. 12(e) is 157, while the corresponding peak bin

in Fig. 12(d) is $93 = \text{mod}(157 + 0.75T \cdot 2^{SF}/T, 2^{SF})$. Similarly, the peak bin corresponding to the green chirp in Fig. 12(d) is $133 = \text{mod}(197 + (-0.25T) \cdot 2^{SF}/T, 2^{SF})$.

After the peaks of the interfering windows are matched to the interested window, spectrum intersection and spectrum difference will be carried out to extract the interested symbols, as described in Section 3.5.

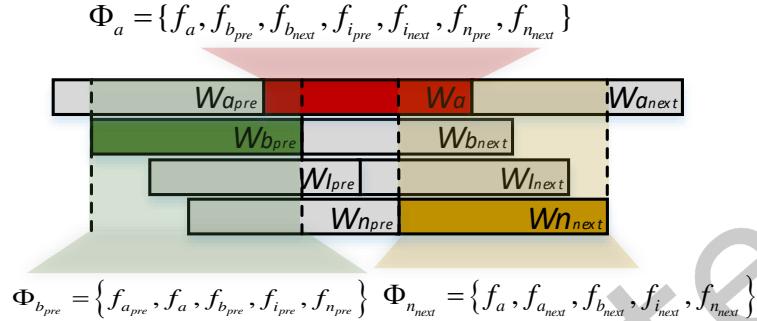


Fig. 14. Frequency sets of associated alignment windows.

3.5 Spectrum Filtering

Target of spectrum filtering is to discover the common frequency across all overlapping parallel alignment windows by spectrum intersection and spectrum difference. Therefore, it is important to analyze the frequencies contained in each aligned window. As an example, Fig. 14 shows the frequency sets corresponding to overlapping windows Φ_a , $\Phi_{b_{pre}}$, and $\Phi_{n_{next}}$. The key property common to these alignment windows is that each of the windows comprises exactly $2(N - 1)$ frequencies. Note that for ease of viewing, in $\Phi_{b_{pre}}$ and $\Phi_{n_{next}}$, we only show the peaks associated with the interested peak. With the frequency sets, spectrum intersection and spectrum difference can be carried out.

3.5.1 Spectrum Intersection. Fig. 15 shows a strawman intuition of how *Paralign* performs spectrum intersection where we use frequency sets of associated alignment windows $\Phi_{b_{pre}}$ and $\Phi_{n_{next}}$. After matching with Φ_a , the spectrum intersection of these three peak sets is the common frequency f_a , i.e., $\Phi_a \cap \Phi_{b_{pre}} \cap \Phi_{n_{next}} = \{f_a\}$.

3.5.2 Spectrum Difference. Due to the influence of the confusing symbol sidelobe, the spectrum intersection P_a may not be unique. For this, *Paralign* uses spectrum difference to eliminate redundant frequencies. This is based on the observation that the interested symbol appears in the previous interfering window overlapped with the interested window but does not appear in the previous window of the interested window. As shown in Fig. 16, f_a is in $\Phi_{b_{pre}}$ but not in $\Phi_{a_{pre}}$, i.e., $\Phi_{b_{pre}} - \Phi_{a_{pre}} = \{f_a\}$.

Algorithm 1 shows the detailed procedure of spectrum filtering, where $\Phi_{a_{pre}}$, Φ_a , $\Phi_{b_{pre}}$, $\Phi_{b_{next}}$, $\Phi_{c_{pre}}$, and $\Phi_{c_{next}}$ are the peak sets of aligned windows $W_{a_{pre}}$, W_a , $W_{b_{pre}}$, $W_{b_{next}}$, $W_{c_{pre}}$, and $W_{c_{next}}$, respectively and P_a is the interested peak of W_a . After matching with Φ_a , *Paralign* first finds the intersection of all associated windows to obtain the initial spectral intersection P_a (Line 2 to 5). If the intersection is empty because of the small overlap between an interfering window and the interested window, *Paralign* removes the peak set of this window and re-intersects the remaining sets until the intersection is not empty (Line 6 to 10). If the intersection is unique, the interested symbol (P_a) is output. Otherwise, the spectrum difference is carried out (Line 14 to 23). After spectrum filtering, *Paralign* adopts the power filtering in Section 3.6 to further acquire interested symbols.

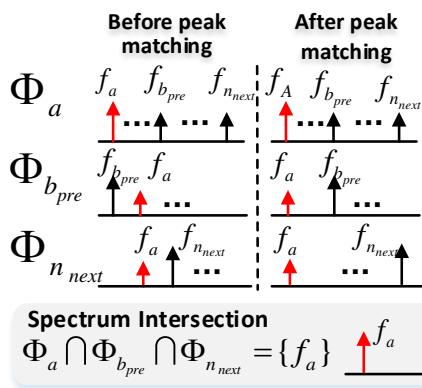


Fig. 15. Peak matching and spectrum intersection.

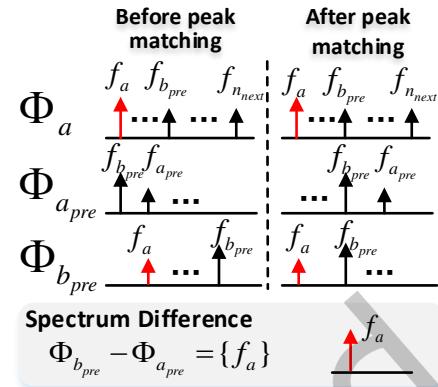


Fig. 16. Peak matching and spectrum difference.

3.6 Power Filtering

Although most of the LoRa collisions can be decoded by spectrum filtering, we found that there may still be more than one candidate peak when the time offset between the confusing symbol and the interested symbol is small. As such, to further extract interested symbols, *Paralign* adopts power filtering including power confirmation and periodic truncation.

3.6.1 Power Confirmation. For power confirmation, we leverage the fact that the peak height in the FFT result is similar across all consecutive windows for an aligned specific LoRa packet. Moreover, the peak height of two segments of a same chirp in adjacent windows is directly proportional to the time-span. For an up-chirp, the height variation caused by time offset Δt can be calculated as

$$\Delta h = \frac{\Delta t}{T} \cdot H, \quad (6)$$

where H represents the peak height when the chirp is aligned with the demodulation window. For example, in Fig. 12(d) with the interested window, the height of the green symbol with time-spans of $0.75T$ is 1.50, while the red symbol with time-spans of T is 2.00. When sliding the window $0.25T$ to the right in Fig. 12(c), the theoretical height of the green symbol is $2 = 1.5 + |-0.25T|/T \cdot 2$ and the red symbol is $1.5 = 2 - |-0.25T|/T \cdot 2$ in Fig. 12(f).

Ideally, the height variation should be directly proportional to the time offset, but there may be a slight change due to the influence of confusing symbols. *Paralign* utilizes a threshold of $3dB$ more than the theoretical value to roughly filter the redundant peaks to realize power confirmation.

3.6.2 Periodic Truncation. After power confirmation, due to the influence of confusing symbols, more than one candidate peak may remain. In such a case, *Paralign* picks the interested peak using the periodic truncation in the time domain. That is, *Paralign* selects chirp segments with integral periods and performs FFT to analyze the spectrum. The key intuition behind this is that the de-chirped signal of the interested symbol has integral periods in the demodulation window, while the interfering signal is not periodically truncated.

We illustrate the intuition for the periodic truncation method in Fig. 17, where the black signal, s' , represents the superposition of the red interested signal, s'_1 , and the green interference signal, s'_2 . It is impossible to distinguish which is an interested symbol when performing FFT on the whole de-chirped signal. Nevertheless, the peak height difference between the left and right chirp segments is different, because, unlike interfering symbols, the frequency of the interested symbol exists uniformly across the entire window. Specifically, as shown in Fig. 17(b)

Algorithm 1 Spectrum Filtering

Input: $\Phi_{a_{pre}}$, Φ_a , $\Phi_{b_{pre}}$, $\Phi_{b_{next}}$, $\Phi_{c_{pre}}$, $\Phi_{c_{next}}$, $W_{a_{pre}}$, W_a , $W_{b_{pre}}$, $W_{b_{next}}$, $W_{c_{pre}}$ and $W_{c_{next}}$

Output: P_a

- 1: **for** W_a in packet A **do**
- 2: Peak matching to get $\Phi_{b_{pre} \rightarrow a}$, $\Phi_{b_{next} \rightarrow a}$, $\Phi_{c_{pre} \rightarrow a}$ and $\Phi_{c_{next} \rightarrow a}$;
- 3: $\Phi = \{\Phi_{b_{pre} \rightarrow a}, \Phi_{b_{next} \rightarrow a}, \Phi_{c_{pre} \rightarrow a}, \Phi_{c_{next} \rightarrow a}\}$;
- 4: $W = \{W_{b_{pre}}, W_{b_{next}}, W_{c_{pre}}, W_{c_{next}}\}$;
- 5: $P_a = \text{intersect}\{\text{each element in } \Phi, \Phi_a\}$;
- 6: **while** $P_a = \emptyset$ **do**
- 7: $x = \min\{W_x \cap W_a\}$, where $W_x \in W$;
- 8: $\Phi = \Phi - \{\Phi_{x \rightarrow a}\}$;
- 9: $P_a = \text{intersect}\{\text{each element in } \Phi, \Phi_a\}$;
- 10: **end while**
- 11: **if** P_a is unique **then**
- 12: **return** P_a ;
- 13: **else**
- 14: $\Phi_{pre} = \Phi - \{\Phi_{b_{next} \rightarrow a}, \Phi_{c_{next} \rightarrow a}\}$;
- 15: **for** Φ_x in Φ_{pre} **do**
- 16: $P = \Phi_x - \Phi_{a_{pre}}$;
- 17: $P_a = P \cap P_a$;
- 18: **if** P_a is unique **then**
- 19: **return** P_a ;
- 20: **else**
- 21: Continue;
- 22: **end if**
- 23: **end for**
- 24: **end if**
- 25: **return** P_a ;
- 26: **end for**

and (c), the height difference of the red peak is $1.00 - 1.00 = 0$, while the difference between the green peaks is $1.00 - 0.75 = 0.25$. Thus, we can infer that the red peak with the small difference is the interested.

Theoretically, the number of periods of the de-chirped signal is related to the modulated symbols s , i.e., $n_s = N/N_s = F_{\text{samp}} \cdot 2^{\text{SF}} / \text{BW} / (F_{\text{samp}} \cdot 2^{\text{SF}} / (\text{BW} \cdot s)) = s$, where N is the number of sampling points for the complete chirp. Thus, there are odd periods in the signal of odd symbols. To avoid the influence of spectrum leakage, we select the de-chirp signal of sub-symbol with integral periods and then calculate the peak height difference between the left and right chirp segments. In our experiments, we intercept chirp segments whose sampling points exceed half of N for the first time. Specifically, for symbol s , the number of sampling points is

$$\begin{aligned} N_{\text{samp}} &= \frac{1}{2} \cdot N + \frac{1}{2} \cdot N_s \cdot \text{mod}(s, 2) \\ &= \frac{F_{\text{samp}} \cdot 2^{\text{SF}}}{2\text{BW}} + \frac{2^{\text{SF}} \cdot F_{\text{samp}}}{2s \cdot \text{BW}} \cdot \text{mod}(s, 2). \end{aligned} \quad (7)$$

For example, in Fig. 18, for symbols 3 and 4, 2/3 and 1/2 of the chirp segments are intercepted to meet the periodicity. Note that for symbols 0 and 1, half the de-chirped signal is used to analyze the spectrum.

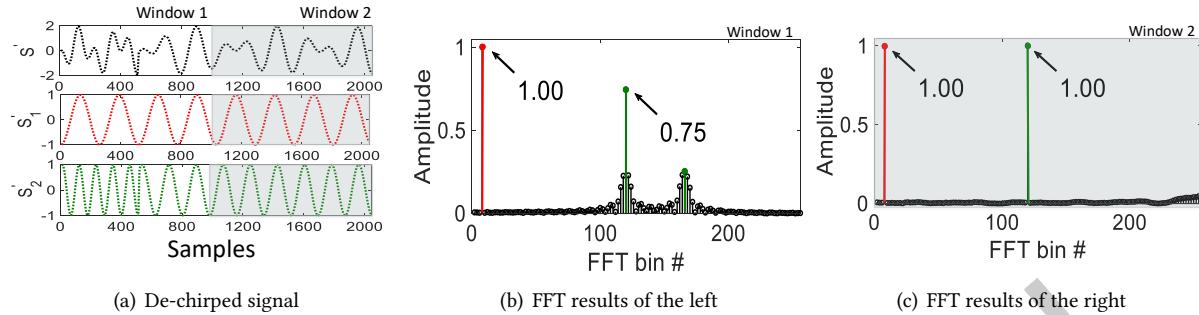


Fig. 17. Illustration of periodic truncation where the red symbol with the small difference is the interested. (a)(b) De-chirped signal in the time domain, where s' represents the superposition of s'_1 and s'_2 . (c)(d) FFT results of left and right chirp segment.

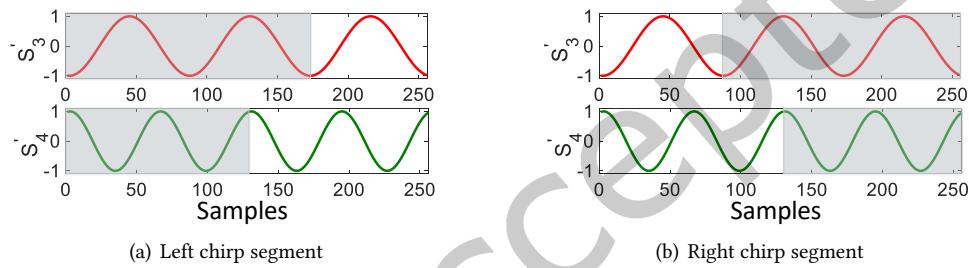


Fig. 18. The number of samples intercepted corresponding to the specific symbol to meet the periodicity, where s'_3 and s'_4 represent the de-chirped signals of symbols 3 and 4, respectively.

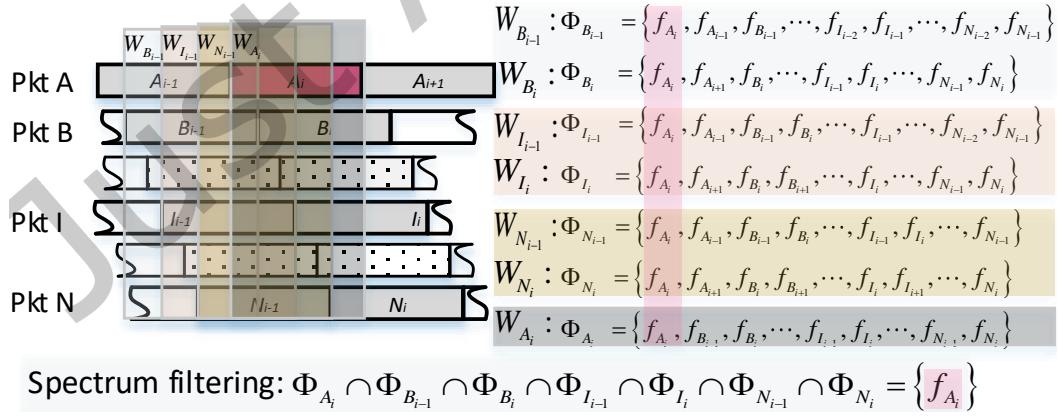


Fig. 19. Strawman Paralign.

3.7 A Strawman Paralign

To provide an intuition into how *Paralign* works, we consider a strawman *Paralign* that uses spectrum filtering to extract the interested symbols. Taking the collision of N packets as an example, after the signal preprocessing (Section 3.2), symbol boundaries of N packets are captured, as shown in Fig. ???. *Paralign* then aligns the collided symbols with the demodulation window in parallel to obtain the peak of the corresponding window, as described in Section 3.3. And Φ_{A_i} , $\Phi_{B_{i-1}}$, Φ_{B_i} , $\Phi_{I_{i-1}}$, Φ_{I_i} , $\Phi_{N_{i-1}}$, and Φ_{N_i} are the peak sets of aligned windows W_{A_i} , $W_{B_{i-1}}$, W_{B_i} , $W_{I_{i-1}}$, W_{I_i} , $W_{N_{i-1}}$, and W_{N_i} , respectively. Next, as described in Section 3.4, *Paralign* matches the peaks of all interference windows with the peaks of interested window to obtain the peak sequence in Fig. ???. The only common component frequency of these peak sets is f_{A_i} . Thus, f_{A_i} can be extracted by spectrum filtering (Section 3.5), i.e., $\Phi_{A_i} \cap \Phi_{B_{i-1}} \cap \Phi_{B_i} \cap \Phi_{I_{i-1}} \cap \Phi_{I_i} \cap \Phi_{N_{i-1}} \cap \Phi_{N_i} = \{f_{A_i}\}$.

3.8 Computation Overhead

Paralign relies on accurate spectrum acquisition by aligning the demodulation window with the symbols of each packet. And the peak of the interested symbol can be easily obtained through the spectrum operation of parallel aligned windows. However, even if the spectrum information of the relevant window is required for each spectrum operation, the spectrum of all collision symbols can be determined simultaneously. That is, after obtaining the symbol boundaries of the collided packet’s symbols, *Paralign* applies FFT by sequentially aligning the symbol boundary with the demodulation window and stores the spectral features. Hence, the major overhead of *Paralign* falls into FFT for each chirp. Besides, the computation cost of FFT is $O(n \log(n))$, where n denotes the number of samples in a chirp. As a result, the computational overhead of collided packet demodulation is $O(mNn \log(n))$, where m and N are the number of collided packets and the payload length. And the computation overhead increases with the total number of nodes.

Compared with past previous approaches, the computational overhead of *Paralign* is acceptable. CoLoRa groups symbols with similar peak ratios across windows to a transmission. FFT only needs to be performed once on the received signal, however, there is a high SER. Although CIC can obtain comparable SER performance with *Paralign*, it needs to analyze the spectrum of all sub-symbols, which increases the computational overhead.

4 EVALUATION

In this section, we extensively evaluate *Paralign*’s performance in various real-world scenarios and present the experimental results compared with the existing methods in different settings.

4.1 Experiment Setting

We implement and perform experimental evaluations of *Paralign* on USRP N210 software radios and LoRa commodity nodes with Semtech SX1276 [22] as the radio frequency chip, which is shown in Fig. 20(a). The USRP N210 connected to a Dell G3 is controlled by Gnu-Radio. Both base station and commodity nodes operate at 433 MHz bands. By default, in our experiments, we set the SF, coding rate (CR), and BW of LoRa communication as 8, 4/8, and 125kHz, respectively. The sampling rate of RTL-SDR is 1 MS/s.

Our experimental environments include the indoor conference room, academic building, and outdoors, as shown in Fig. 20(b)-(c) and each deployed network consists of 20 LoRa devices and a gateway. Each end device is configured to transmit packets with randomly generated 32 bytes. In the experiments, we mainly compare *Paralign* against CoLoRa [28] and CIC [23] which represent the state-of-the-art based on power distribution and spectrum characteristics, respectively. Considering the LoRaWAN receiver without any collision resolution scheme, *Paralign* doesn’t compare with it.

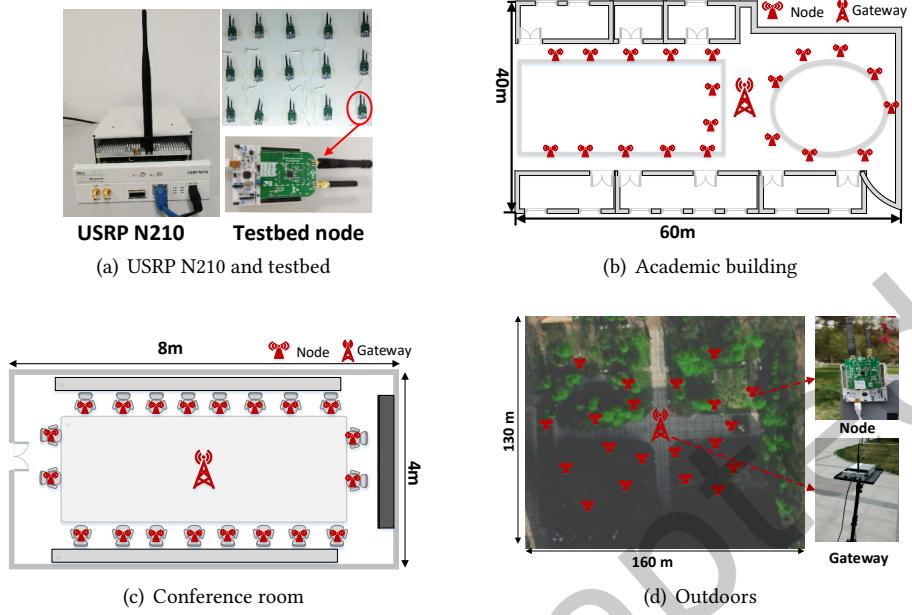


Fig. 20. The experiment setting and the deployed LoRa networks in different scenarios.

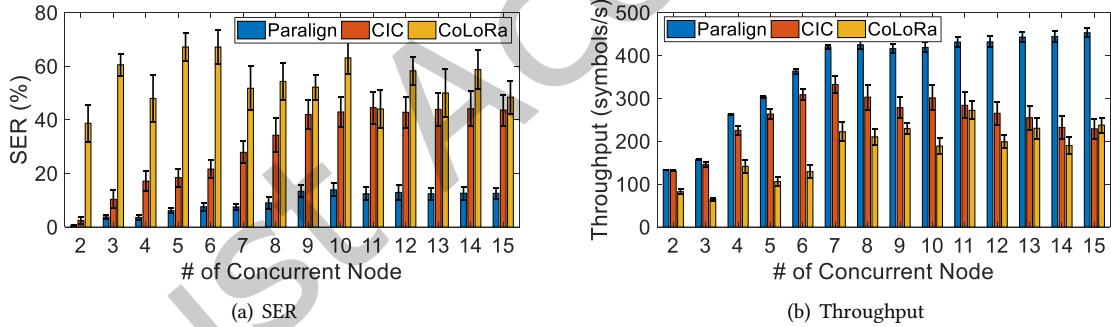


Fig. 21. Decoding performance with increasing concurrency nodes.

4.2 Performance of Multi-Packet Collision

We first examine the performance of multi-packet collision as the number of concurrent nodes increases from 2 to 15. To create collisions, we use a node to control the sending nodes. After receiving the control packet, all sending nodes are awakened and send a known sequence of LoRa packets after a random delay. At the base station, we use *Paralign*, *CoLoRa*, and *CIC* to decompose the collided packets to verify the correctness of the decoded packets and calculate the SER and the throughput.

Fig. 21 shows the averaged SER and the overall network throughput. As the number of concurrent nodes increases, the SERs and the network throughput of both *Paralign* and *CIC* are increasing. But *CIC*'s SER is

improving faster than *Paralign*'s. This is because *Paralign* decodes the concurrent transmissions by parallel alignment to reduce the effects of sidelobes of confusing symbols. But CIC using spectral content of sub-symbol with low-frequency resolution ignores the existence of confusing symbols. Moreover, when the number of concurrent LoRa devices increases, the height difference among chirps in different LoRa packets decreases, and the SER of CoLoRa is up to 67% when six nodes transmit concurrently. Meanwhile, the network throughput of CoLoRa and CIC is lower than that of *Paralign* because of the neglect of the influence of confusing symbols. Specifically, when nine nodes are concurrent transmission, the throughput of *Paralign* is 1.46× compared with CIC and 2.33× compared with CoLoRa.

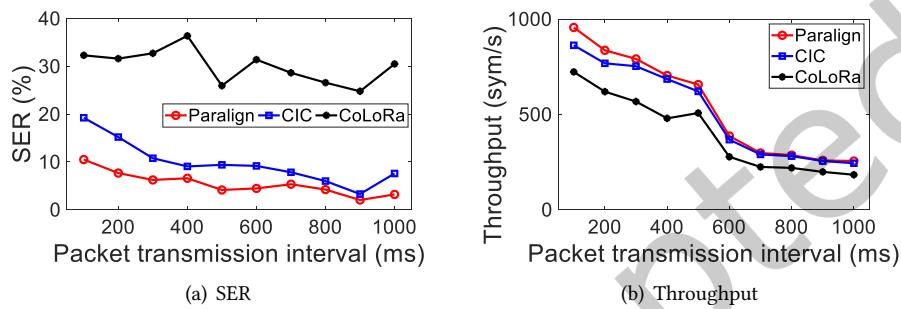


Fig. 22. Comparison under different transmission intervals.

4.3 Comparison under Different Transmission Intervals

In this subsection, we examine *Paralign*'s performance for decoding multiple packets in collisions as the transmission interval from 100ms to 1000ms in terms of SER and throughput in the academic building. As seen in Fig. 22, *Paralign* significantly outperforms CIC and CoLoRa when the transmission interval is 100ms. With the increased packet transmission interval, the SER of *Paralign* generally shows a downward trend. This is because, with the increased transmission interval, the probability of packet collision decreases. Besides, the SER of *Paralign* is getting closer to that of CIC, but it is still lower than that of CIC. This is because CIC ignores the existence of confusing symbols, and the probability of this phenomenon increases with the increase of concurrent packets. For CoLoRa which only uses peak ratio matching, the SER performance is very poor in any case. Similarly, the throughput of *Paralign* is about 1.1× that of CIC and 1.32× that of CoLoRa.

4.4 Impact of Transmission Parameters

The symbol duration of LoRa packets which is related to the Spread Factor (SF) and Bandwidth (BW) can affect the performance of concurrent decoding methods. In this subsection, we compare the performance of *Paralign*, CoLoRa, and CIC for decoding LoRa collisions regarding different SF and BW. Unless otherwise specified, we adopt the default experimental settings.

4.4.1 Impact of SF. We explore the impact of SF on demodulation performance with the bandwidth of 125kHz under the packet transmission interval of 100ms. Fig. 23(a) shows the overall SER for *Paralign*, CoLoRa, and CIC with varied SFs (i.e., SF=8, 9, and 10). The average SER of *Paralign* decreases from 12.6% to 9.68% as SF increases from 8 to 10. This is because the chirp of big SF with a long duration facilitates collision recovery and with the increase of SF, the symbol takes a longer air time, which helps *Paralign* separate collisions in time. That is, higher SF has better tolerance for noise and hence provides comparatively reliable PRR. Compared with CIC and CoLoRa,

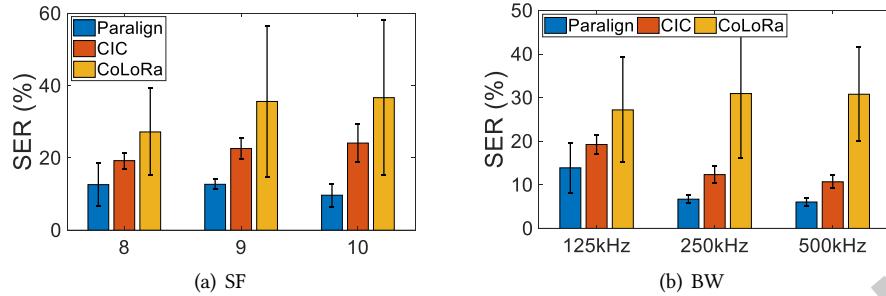


Fig. 23. Impact of transmission parameters.

Paralign achieves lower SERs with any SF. The SERs of CIC and CoLoRa are $1.92\times$ and $3.02\times$ that of *Paralign* when SF is 9, respectively.

4.4.2 Impact of BW. Fig. 23(b) shows the performance of *Paralign* with varied BWs (i.e., BW=125kHz, 250kHz, and 500kHz). The SF is fixed to 8 and the packet transmission interval is 100ms. The results show that *Paralign* performs better with a larger BW. The average SER decreases from 13.9% to 6.06% as the BW increases from 125kHz to 500kHz. That is because with the increase of BW, the duration of packets becomes shorter and the probability of packet collision becomes smaller. Besides, the frequency gap between the symbols coexisting within a demodulation window also increases with bandwidth increases, which helps *Paralign* separate collisions. Similarly, the performance of *Paralign* is better than CIC and CoLoRa. The SERs of CIC and CoLoRa are $1.84\times$ and $4.61\times$ that of *Paralign* when BW is 250kHz, respectively.

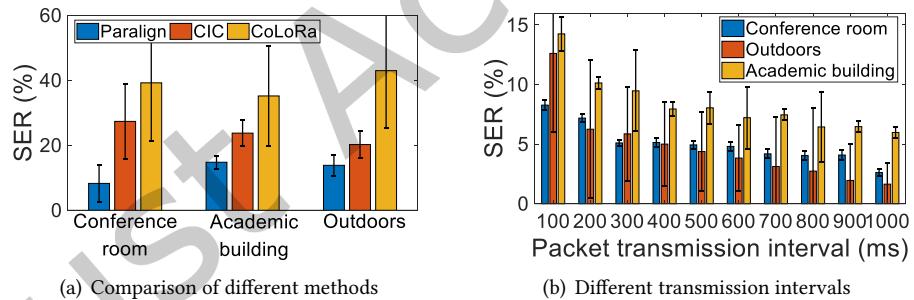


Fig. 24. Impact of different experimental environments.

4.5 Impact of Different Experimental Environments

The locations of deployed LoRa devices can also impact the performance of *Paralign* because the SNR of the received signal is location dependent. To evaluate the performance of *Paralign* under different experimental environments, we deploy 20 LoRa nodes in three scenarios as described in Fig. 20(b)-(d). In order not to lose generality, we use random deployment outdoors, as shown in Fig. 19(d).

4.5.1 Comparison of different methods. We first compare the performance of three decoding methods in different environments. The results are shown in Fig. 24(a). *Paralign* significantly outperforms CIC and CoLoRa. The

average SERs of *Paralign* are 8.27%, 13.84%, and 14.79% in the conference room, academic building, and outdoors, respectively. The SERs of CIC and CoLoRa are 3.3× and 4.74× that of *Paralign* in the conference room. Besides, we find that *Paralign* achieves lower SERs in the conference room. This is because the area of conference room is small, and the signal received by the gateway has a higher signal-to-noise ratio than that of the outdoor, which helps *Paralign* separate collisions.

4.5.2 Different transmission intervals. In the following, we explore the performance of *Paralign* with the variation of packet transmission interval in different environments. Fig. 24(b) shows the SER of *Paralign* in conference room, academic building, and outdoors. We see that the overall SER shows a downward trend with the increasing packet transmission interval. Moreover, the average SERs of *Paralign* are 5.03%, 8.74%, and 4.74% in the conference room, academic building, and outdoors, respectively. Besides, even though the performance of the academic building is lower than the other two locations due to the serious multi-path, the maximum SER is only 14.25%.

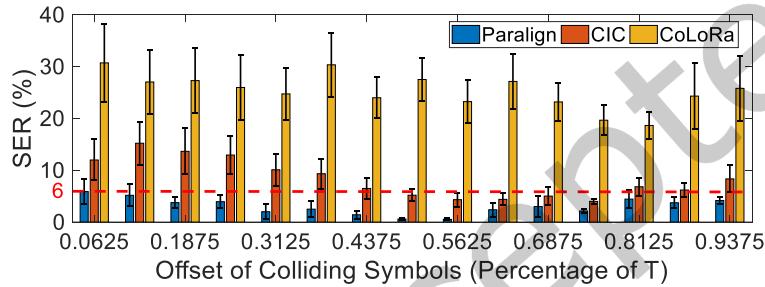


Fig. 25. Simulation results of two packet collisions with increasing time offset.

4.6 Impact of Temporally Close Collisions

As Heisenberg's time-frequency uncertainty principle states, the shorter the time length of the estimated signal, the worse the resolution of frequency estimation. Therefore, we explore the performance of temporally close collisions. In this experiment, similar to CIC, we simulate generating two raw LoRa signals and superimpose them to construct a collision signal within the accuracy range of sampling level, because it is hard for two COTS LoRa devices to transmit synchronously. The time offset of the superimposed signals is in increments of 6.25% of the symbol. We adopt *Paralign*, CIC, and CoLoRa to recover the superimposed symbols respectively to measure the SERs. The results are shown in Fig. 25 where the horizontal axis represents the proportion of the time offset over the symbol duration T . Compared with CIC and CoLoRa, *Paralign* keeps a lower SER (<6%) with the increase of time offset.

4.7 Impact of Confusing Symbols

Paralign decodes the collided packets with confusing symbols via parallel alignment. In this subsection, we try to explore how *Paralign* is affected by the proximity of confusing symbols. Similar to Section 4.6, we simulate generating collision signals with an increasing symbol distance in increments of 1. As shown in Fig. 26, except that the distance is ± 1 , the mean SER of *Paralign* is less than 8.6%. Moreover, the closer the collided symbols are, the more serious the influence of the confusing symbols on the interested symbol. This is because the side lobes of misaligned confusing symbols will affect the peak extraction of interested symbols. Nevertheless, it is unreasonable to directly delete the peak adjacent to the interested symbol in the implementation of CIC [9], resulting in a high SER at the distance of ± 1 and ± 2 . Moreover, the power filtering method adopted by CIC makes

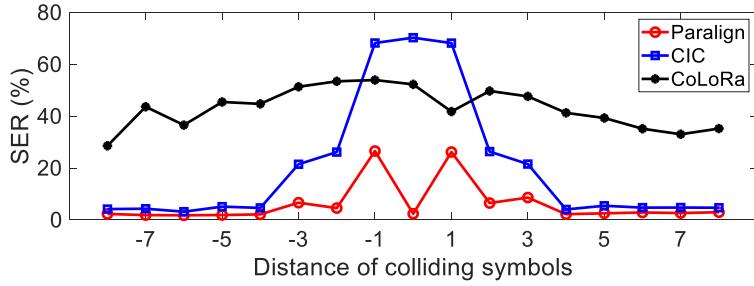


Fig. 26. Simulation results of two packets collision with varying symbol distance.

the interested symbols filtered when the symbols with the same frequency appear in the demodulation window, resulting in a high SER at the distance of 0. For CoLoRa which only uses height ratio to match the transmitter, the SER is high in any case.

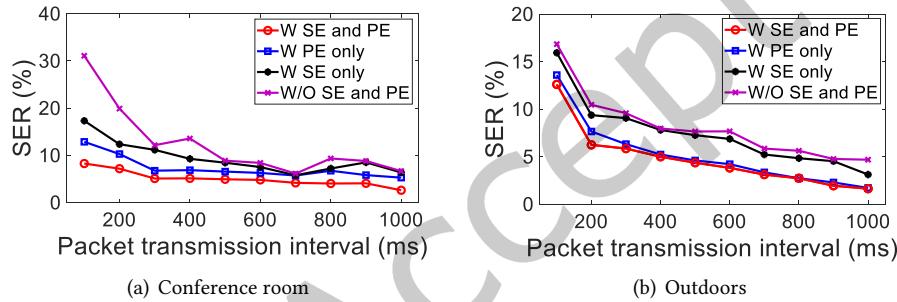


Fig. 27. Performance of Paralign with/without Spectrum difference (SE) and Power Extraction (PE).

4.8 Performance of Paralign's Modules

We then investigate the performance of the key designs of *Paralign*, including spectrum difference and power extraction. Fig. 27 depicts the SERs obtained for each of the modules in conference room and outdoors respectively. The SNR distribution of conference room with a small space is relatively concentrated, while that of outdoors with a large space is broad.

We can find that for all packet transmission intervals, using spectrum difference and power extraction can significantly improve the performance. Compared to the performance of *Paralign* without spectrum difference and without power filter, *Paralign* with all can reduce the SER to 8.27% at a packet transmission interval of 100ms in conference room. Moreover, the performance improvement of the power extraction module is better than the spectrum difference. The SERs of *Paralign* without spectrum difference and without power extraction are 12.87% and 17.33%, respectively. Similarly, in the outdoors large SNR range, the performance of *Paralign* with spectrum difference and power extraction is better than that without spectrum difference and without power extraction.

5 RELATED WORK

Collision resolution in MAC layer: The collision resolution of the MAC layer is mainly implemented by collision avoidance [6–8, 21, 36, 39]. RS-LoRa [21] proposes a new MAC design that determines its transmission

power, SF, and channel based on gateway schedules. Chime [6] enables LPWAN base stations to identify an optimal frequency of operation after the client sends one packet at one frequency. EF-LoRa [8] achieves fair energy consumption among end devices by carefully allocating different network resources. LMAC [7] implements CSMA for LoRa networks to balance loads of the channels defined by frequencies and spreading factors. S-MAC [36] designs a MAC-layer scheduler based on the periodic characteristics of LPWAN applications and transmission features of the LoRa PHY-layer. However, due to limited energy budget and low-cost hardware, it is difficult to design complex MAC protocol for LoRa devices to resolve collisions.

Collision resolution in PHY layer: The collision resolution of PHY layer can decompose multiple collision packets at the same time without mutual and detection information [3, 5, 13, 14, 27, 28, 31, 33–35, 38]. Choir [5] decomposes collided packets based on frequency offset related to the hardware imperfections of LoRa end nodes. But it is difficult to accurately extract the tiny hardware frequency shift for LoRa signals with low SNR. Besides, in a dense LoRa network, a similar frequency shift between end nodes will inevitably lead to decoding failure. To effectively separate collided LoRa packets, time-domain features are extracted [31, 33, 35]. mLoRa [31] iteratively reconstructs and extracts each decoded chirp symbol by successive interference cancellation. FTrack [35] calculates the instantaneous frequency continuity to decode multiple LoRa packets from a collision. However, the methods based on time-domain features have basic limitations in decoding collided signals with low SNR. CoLoRa [28] groups symbols based on a similar spectral peak ratio to transmission because of the consistency of the received power of a packet. NScale [27] adopts a non-stationary signal scaling function to render the controlled frequency peak with different peak scaling factors by amplifying the time domain signal. Nevertheless, the above method of packet matching only using amplitude characteristics will be affected by the side lobe of confusing symbols.

The most related work to ours is CIC [23]. CIC decodes multiple collided packets by concurrent interference cancellation. The key of CIC is adopting the optimal set of sub-symbols to cancel out all other interfering symbols. As the author puts it, using sub-symbols to estimate the spectrum will reduce the spectral resolution, resulting in multiple intersections of these sub-symbols. The characteristics of peak height are then used to filter out the interfering symbols. However, the side lobes generated by confusing symbols will lead to the instability of the peak frequency and height of the interested symbol, which will lead to wrong frequency cancellation. To support the decoding of collision with confusing symbols, we introduce *Paralign* which decodes the concurrent transmissions via parallel alignment.

6 DISCUSSION

Demodulating capacity: Since the performance of concurrent decoding methods depends on the characteristics of relevant collided signals, the demodulation capacity of *Paralign* is not unbounded as the number of colliding nodes increases. If the boundaries of collided symbols are aligned in case of adjacent collision, *Paralign* may not be able to separate them. In practice, some LoRa MAC protocols use slotted-ALOHA to reduce the collision. As a result, some nodes may still collide if they choose the same time slot, especially in densely deployed networks and the boundaries of collided symbols are most likely aligned. To avoid this problem, in *Paralign*, the nodes can slightly back off before transmission.

LoRa network concurrent capacity: In practice, LoRa supports concurrent transmission with different channels, bandwidths, and spreading factors. Specifically, the LoRa clients have hundreds of frequencies to choose from. For example, in Europe, the band from 868.180 MHz to 868.220 MHz is divided into 400 100 Hz orthogonal channels (40 of which are reserved and unused) [17]. For the system with a constant channel, LoRa network supports seven different spreading factors ranging from SF6 to SF12 and multiple bandwidth orthogonality at the same time. Hence, to maximize *Paralign*'s network concurrent capacity, we can select the parameters of LoRa communication appropriately.

7 CONCLUSION

In this paper, we propose *Paralign* which is the first LoRa collision decoder supporting decoding LoRa collisions with confusing symbols via parallel alignment. *Paralign* obtains the accurate spectrum of correlation chirps by aligning the demodulation windows with the symbols of each packet in parallel according to the packet boundaries for further analyzing the specific spectrums. To effectively extract the interested symbols, after peak matching, spectrum intersection and difference for parallel aligned windows are carried out. For the problem caused by confusing symbols that more than one potential candidate peak may remain, a periodic truncation method is proposed to select a more stable peak in the spectrum of sub-symbols with integer periods. Experiment results demonstrate that *Paralign* can separate collided packets with a small SER and improve network throughput by 1.46 \times compared with CIC and 2.33 \times compared with CoLoRa.

ACKNOWLEDGMENTS

This work is supported in part by the National Key Research and Development Program of China (No. 2021YFB2900100), the A3 Foresight Program of NSFC (No. 62061146002), the National Natural Science Foundation of China (No. 62072050, No. 62225204), and the Funds for Creative Research Groups of China (No. 61921003).

REFERENCES

- [1] José Álamos, Peter Kietzmann, Thomas C Schmidt, and Matthias Waehlisch. 2022. DSME-LoRa: Seamless Long Range Communication Between Arbitrary Nodes in the Constrained IoT. *ACM Transactions on Sensor Networks* (2022).
- [2] Nur Aziemah Azmi Ali and Nurul Adilah Abdul Latiff. 2019. Environmental monitoring system based on LoRa technology in island. In *Proceedings of IEEE ICSigSys*.
- [3] Qian Chen and Jiliang Wang. 2021. AlignTrack: Push the Limit of LoRa Collision Decoding. In *Proceedings of IEEE ICNP*.
- [4] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. 2019. Automated estimation of link quality for LoRa: A remote sensing approach. In *Proceedings of IEEE IPSN*.
- [5] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering low-power wide area networks in urban settings. In *Proceedings of ACM SIGCOMM*.
- [6] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. 2020. Frequency configuration for low-power wide-area networks in a heartbeat. In *Proceedings of USENIX NSDI*.
- [7] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. 2020. LMAC: Efficient carrier-sense multiple access for lora. In *Proceedings of ACM MobiCom*.
- [8] Weifeng Gao, Wan Du, Zhiwei Zhao, Geyong Min, and Mukesh Singhal. 2019. Towards energy-fairness in lora networks. In *Proceedings of IEEE ICDCS*.
- [9] CIC implement. 2021. <https://github.com/osama4933/CIC>.
- [10] R.Jedermann, M.Borysov, N.Hartgenbusch, S.Jaeger, M.Sellwig, and W.Lang. 2018. Testing Lora for food applications-Example application for airflow measurements inside cooled warehouses with apples. *Procedia Manufacturing* 24 (2018), 284–289.
- [11] Dah-Jing Jwo, Wei-Yeh Chang, and I-Hua Wu. 2021. Windowing Techniques, the Welch Method for Improvement of Power Spectrum Estimation. *CMC-COMPUTERS MATERIALS & CONTINUA* (2021).
- [12] Chenning Li and Zhichao Cao. 2022. Lora networking techniques for large-scale and long-term iot: A down-to-top survey. *Comput. Surveys* 55, 3 (2022), 1–36.
- [13] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. 2021. NELoRa: Towards Ultra-low SNR LoRa Communication with Neural-enhanced Demodulation. In *Proceedings of ACM SenSys*.
- [14] Chenning Li, Xiuzhen Guo, Longfei Shangguan, Zhichao Cao, and Kyle Jamieson. 2022. {CurvingLoRa} to Boost {LoRa} Network Throughput via Concurrent Transmission. In *Proceedings of USENIX NSDI*.
- [15] Ruinan Li, Xiaolong Zheng, Yuting Wang, Liang Liu, and Huadong Ma. 2022. PolarScheduler: Dynamic Transmission Control for Floating LoRa Networks. In *Proceedings of IEEE INFOCOM*.
- [16] Jansen C Liando, Amalinda Gamage, Agustinus W Tengoutius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks* 15, 2 (2019), 1–35.
- [17] Aamir Mahmood, Emiliano Sisinni, Lakshmikanth Guntupalli, Raúl Rondón, Syed Ali Hassan, and Mikael Gidlund. 2018. Scalability analysis of a LoRa network under imperfect orthogonality. *IEEE Transactions on Industrial Informatics* 15, 3 (2018), 1425–1436.
- [18] Mohammad Shareef Mekala and P.Viswanathan. 2017. A Survey: Smart agriculture IoT with cloud computing. In *Proceedings of IEEE ICMDCS*.

- [19] Tung T Nguyen, Ha H Nguyen, Robert Barton, and Patrick Grossetete. 2019. Efficient design of chirp spread spectrum modulation for low-power wide-area networks. *IEEE Internet of Things Journal* 6, 6 (2019), 9503–9515.
- [20] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. 2018. PLoRa: A passive long-range data network from ambient LoRa transmissions. In *Proceedings of ACM SIGCOMM*.
- [21] Brecht Reynders, Qing Wang, Pere Tuset-Peiro, Xavier Vilajosana, and Sofie Pollin. 2018. Improving reliability and scalability of lorawans through lightweight scheduling. *IEEE Internet of Things Journal* 5, 3 (2018), 1830–1842.
- [22] Semtech. 2017. Semtech SX1276.
- [23] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. 2021. Concurrent interference cancellation: decoding multi-packet collisions in LoRa. In *Proceedings of ACM SIGCOMM*.
- [24] Nicolas Sornin, Miguel Luis, Thomas Eirich, Thorsten Kramp, and Olivier Hersent. 2015. Lorawan specification. *LoRa alliance* (2015).
- [25] Jothi Prasanna Shammuga Sundaram, Wan Du, and Zhiwei Zhao. 2019. A survey on lora networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 371–388.
- [26] Heisenberg's time-frequency uncertainty principle. 2022. https://subsurfwiki.org/wiki/Uncertainty_in_spectral_decomposition.
- [27] Shuai Tong, Jiliang Wang, and Yunhao Liu. 2020. Combating packet collisions using non-stationary signal scaling in LPWANs. In *Proceedings of ACM MobiSys*.
- [28] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. 2020. CoLoRa: Enabling Multi-Packet Reception in LoRa. In *Proceedings of IEEE INFOCOM*.
- [29] Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. 2015. Long-range IoT technologies: The dawn of LoRa™. In *Future access enablers of ubiquitous and intelligent infrastructures*. Springer, 51–58.
- [30] Achim Walter, Robert Finger, Robert Huber, and Nina Buchmann. 2017. Opinion: Smart farming is key to developing sustainable agriculture. *Proceedings of the National Academy of Sciences* (2017).
- [31] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. 2019. MLoRa: A multi-packet reception protocol in LoRa networks. In *Proceedings of IEEE ICNP*.
- [32] Yuting Wang, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2022. PolarTracker: Attitude-aware channel access for floating low power wide area networks. *IEEE/ACM Transactions on Networking* (2022).
- [33] Zhe Wang, Linghe Kong, Kangjie Xu, Liang He, Kaishun Wu, and Guihai Chen. 2020. Online concurrent transmissions at LoRa gateway. In *Proceedings of IEEE INFOCOM*.
- [34] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. 2022. PCube: scaling LoRa concurrent transmissions with reception diversities. *ACM Transactions on Sensor Networks* (2022).
- [35] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2020. FTrack: Parallel decoding for LoRa transmissions. *IEEE/ACM Transactions on Networking* 28, 6 (2020), 2573–2586.
- [36] Zhuqing Xu, Junzhou Luo, Zhimeng Yin, Tian He, and Fang Dong. 2020. S-MAC: Achieving high scalability via adaptive scheduling in LPWAN. In *Proceedings of IEEE INFOCOM*.
- [37] Zhenqiang Xu, Pengjin Xie, Shuai Tong, and Jiliang Wang. 2022. From Demodulation to Decoding: Towards Complete LoRa PHY Understanding and Implementation. *ACM Transactions on Sensor Networks* (2022).
- [38] Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. 2021. Pyramid: Real-Time LoRa Collision Decoding with Peak Tracking. In *Proceedings of IEEE INFOCOM*.
- [39] Fu Yu, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2022. LoRadar: An Efficient LoRa Channel Occupancy Acquirer based on Cross-channel Scanning. In *Proceedings of IEEE INFOCOM*.
- [40] SUN Zehua, YANG Huanqi, LIU Kai, YIN Zhimeng, LI Zhenjiang, and XU Weitao. 2022. Recent Advances in LoRa: A Comprehensive Survey. *ACM Transactions on Sensor Networks* (2022).

AUTHORS' BIOGRAPHIES



Yuting Wang is currently a Ph.D with the School of Computer Science and Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, China. Her received her B.E. degree from the Henan University of science and technology, China, in 2015, and her M.A. degree from the Henan University of science and technology, China, in 2018. Her research interests include Internet of Things, wireless networks.



Fanhao Zhang is currently a postgraduate student with the School of Computer Science and Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, China. He is received his B.E. degree from the Beijing University of Posts and Telecommunications, China, in 2021. His research interests include Internet of Things, wireless networks.



Xiaolong Zheng is currently an Associate Professor with the School of Computer Science and Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, China. He received his B.E. degree from the Dalian University of Technology, China, in 2011, and his Ph.D. degree from the Hong Kong University of Science and Technology, China, in 2015. His research interests include Internet of Things, wireless networks, and ubiquitous computing.



Liang Liu received the B.S. degree from the School of Computer Science and Technology, South China University of Technology, Guangzhou, China, in 2004, and the Ph.D. degree from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2009.

He is currently a Professor with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia and the Dean of the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. He was a Visiting Ph.D. Student with the Networking and Information Systems Laboratory, Texas A&M University, College Station, TX, USA, from 2007 to 2008. He has published more than 150 papers. His current research interests include Internet of Things and intelligent sensing technologies.



Huadong Ma received the B.S. degree in mathematics from Henan Normal University, Xinxiang, China, in 1984, the M.S. degree in computer science from Shenyang Institute of Computing Technology, Chinese Academy of Science, Beijing, China, in 1990, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Science in 1995.

He is currently a Professor of School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. From 1999 to 2000, he held a Visiting Position with the University of Michigan, Ann Arbor, MI, USA. His current research interests include Internet of Things and sensor networks, multimedia computing, and he has published more than 400 papers in journals (such as ACM/IEEE Transactions) or conferences (such as ACM MobiCom/MM, IEEE INFOCOM, ACM SIGCOMM) and five books.

Dr. Ma received the Natural Science Award of the Ministry of Education, China, in 2017. He received the 2019 Prize Paper Award of IEEE TRANSACTIONS ON MULTIMEDIA, 2018 Best Paper Award from IEEE MULTIMEDIA, Best Paper Award in IEEE ICPADS2010, and Best Student Paper Award in IEEE ICME2016 for his coauthored papers. He received the National Funds for Distinguished Young Scientists in 2009. He was/is an Editorial Board Member of the IEEE Transactions on Multimedia, IEEE Internet of Things Journal, ACM Transactions on Internet of Things, and Multimedia Tools and Applications. He is an IEEE Fellow and serves for Chair of ACM SIGMOBILE China.