CoLoRa: Enabling Multi-Packet Reception in LoRa Networks

Shuai Tong, *Student Member, IEEE*,
Zhenqiang Xu, *Student Member, IEEE*, and Jiliang Wang, *Senior Member, IEEE*

Abstract—LoRa, as a representative Low-Power Wide Area Network (LPWAN) technology, has emerged as a promising platform for connecting the Internet of Things (IoTs). It enables low-rate communications over upto tens of kilometers with a 10-year battery lifetime. However, practical LoRa deployments suffer from collisions, given the dense deployment of devices and the wide coverage area. We propose CoLoRa, an approach to decompose large numbers of concurrent transmissions from one collision and enable multi-packet reception in LoRa networks. At the heart of CoLoRa, we utilize the packet time offset to disentangle collided packets. CoLoRa incorporates several novel techniques to address practical challenges. (1) We translate time offset, which is difficult to measure, to frequency features that can be reliably measured. (2) We propose a method to extract peak features from low-SNR LoRa signals iteratively. (3) We address frequency shift incurred by carrier frequency offset and time offset for LoRa decoding. We implement CoLoRa on USRP N210 and evaluate its performance in both indoor and outdoor networks. CoLoRa is implemented in software at the base station, and it can work for COTS LoRa nodes. The evaluations show that CoLoRa improves the network throughput by 3.4× compared with Choir and 14× compared with LoRaWAN.

Index Terms—Internet of Things, LPWAN, LoRa, multi-packet reception

1 Introduction

The success of the Internet of Things (IoTs) highly depends on connecting large-scale IoT devices. As a promising communication platform, Low-Power Wide Area Networks (LPWANs) can provide low-cost, long-range communication with very low energy consumption for large-scale IoT devices [1], [2]. LoRa is a widely used and industry applied LPWAN technique, working on the sub-GHz or 2.4 GHz unlicensed ISM bands [3]. A typical LoRa client can communicate to the remote LoRa gateways at the range of kilometers or even tens of kilometers with a data rate of a few kbps [4]. The energy consumption of the LoRa client is extremely low, and thus the client can work for nearly 10 years powered by a button cell battery [5].

Deploying LoRa networks in practice, however, is very challenging. LoRa adopts a star-of-the-star network topology, where thousands of LoRa clients connect to the same gateway. There are severe signal collisions when multiple LoRa clients transmit packets to the gateway concurrently, decreasing network throughput and ratcheting up energy consumption and network delay. Moreover, the limited energy budget and low-cost hardware of LoRa clients make it difficult to apply sophisticated MAC to resolve collisions. The relatively long packet duration further accelerates the collision problem. All these challenges lead to a gap between LoRa's vision to provide low-power large-scale connections and its practical capability [6], [7].

Existing Approaches. There exist a large collection of parallel decoding approaches in traditional wireless [8], [9], [10], [11]. As a representative one, ZigZag [8] can decode collided Wi-Fi packets, but it requires m retransmissions to resolve an m-packet collision. mZig [9] decodes multiple ZigBee packets from one collision directly by leveraging ZigBee's unique encoding characteristics. However, these approaches cannot be applied to solve LoRa collisions due to the extremely low SNR of LoRa transmissions.

Recently, Choir [12] shows that the hardware imperfections of low-cost LoRa end nodes can be used for separating LoRa collisions. Its main idea is to disentangle overlapped LoRa symbols based on the difference of their frequency offsets. However, extracting tiny frequency offsets from low SNR LoRa signals is very challenging, which diminishes Choir's benefits in practice. mLoRa [13] and FTrack [14] apply a time-domain Successive Interference Cancellation (SIC) approach to LoRa collisions. mLoRa [13] starts with a collision-free chunk and then iteratively reconstructs and extracts each decoded symbol. FTrack [14] decodes multiple LoRa packets from a collision by calculating the instantaneous frequency continuity. Both mLoRa and FTrack have limitations in decoding low SNR LoRa collisions, as they only use the time domain features of LoRa signals, without leveraging the property of LoRa modulation. Temim *et al.* propose a frequency-domain SIC approach [15] which decodes LoRa collisions by iteratively decoding and canceling the highest peak in the frequency domain. It decodes LoRa collisions at a low SNR by energy concentration of FFT, but is severely affected by variations in signal power

• *The authors are with the School of Software, Tsinghua University, Beijing 100084, China. E-mail: {tl19, xu-zq17}@mails.tsinghua.edu.cn, jiliangwang@tsinghua.edu.cn.*

Manuscript received 10 May 2021; revised 26 Oct. 2021; accepted 15 Dec. 2021. Date of publication 28 Dec. 2021; date of current version 5 May 2023.
This work was supported in part by NSFC under Grants 62172250 and 61932013, and in part by the Tsinghua University Initiative Scientific Research Program.
(Corresponding author: Jiliang Wang.)
Digital Object Identifier no. 10.1109/TMC.2021.3138495

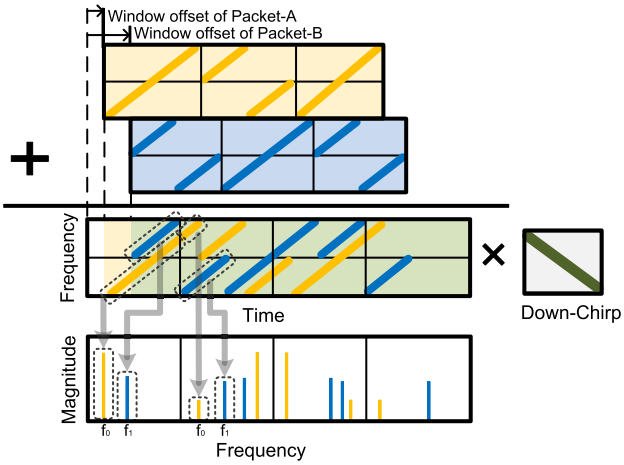

Fig. 1. An example of decomposing a two-packet collision based on the packet time offset. CoLoRa decomposes collided packets by transforming the packet time offset to frequency domain features.

levels. Besides, the SIC approaches also suffer from the error accumulation due to their imperfect signal cancellation.

Our Approach. To resolve the LoRa packet collisions, we proposed CoLoRa, an approach which enables Multi-Packet Reception (MPR) in LoRa networks. CoLoRa utilizes the packet time offsets and signal strength to decompose multiple concurrent transmissions from one collision directly. We implement CoLoRa at a software defined radio (SDR) based LoRa gateway without requiring any hardware or software modifications on the clients. The source code and evaluation data for CoLoRa are available at [16].

To see how CoLoRa works, consider the scenario in Fig. 1, where two packets collide. Both packet-A and packet-B consist of multiple chirp symbols. LoRa modulates signals with chirp spread spectrum (CSS), where chirps with different frequency shifts represent different data bits. At the receiver, each received LoRa chirp is multiplied with a standard down-chirp whose frequency linearly decreases. When there is no collision, the multiplication results in a single tone which translates to a single peak in the frequency domain. Otherwise, there will be multiple energy peaks in the frequency domain, as shown in Fig. 1. The data bits of the two collided packets are mixed up and hence the decoding algorithm fails to decode any of the collided packets.

At the heart of CoLoRa is a physical layer algorithm that utilizes the packet time offset to disentangle the collisions. Upon receiving a collision, CoLoRa first splits the received signal into a series of reception windows, each window with the length equal to a chirp. As shown in Fig. 1, after the signal splitting, each chirp is divided into two segments by two consecutive windows. Then for signal in each window, we transform the chirp segments to frequency peaks by multiplying a standard down-chirp and applying the Fourier transformation. We formally prove that the height of the frequency peak is proportional to the length of the corresponding chirp segment in Section 5. We also show that the two peaks of the same chirp locate the same frequency, e.g., f_0 for packet-A's first chirp. For two peaks belonging to the same chirp, we define the peak ratio as the height of the latter one divided by that of the former one. We can see that the peak ratio is identical for chirps of the same packet, while it is distinct for chirps of different packets. Thus, by grouping chirps with the same peak ratio, CoLoRa can finally disentangle the collided packets.

The benefits of using the peak ratio as features to separate collided packets are three-fold. (1) It concentrates chirp energy by transforming time domain signals to frequency peaks, and thus it works well for low SNR LoRa signals with inter-chirp interferences. (2) It is resilient to signal dynamics and environment complication, as the peak ratio is only determined by the packet time offset which is stable during the whole transmission. (3) There is no error accumulation as the peak ratio of each chirp is calculated independently.

Challenges. Using the peak ratios to disentangle collided packets in CoLoRa also faces practical challenges: First, CoLoRa relies on identifying the existence of LoRa collisions and splitting collision signals into continuous reception windows. Collision identification is difficult due to the extremely low SNR of LoRa transmissions. Besides, we find that the reception window selection affects the peak estimation. For example, an improper reception window selection may result in two peaks of the same chirp having unbalanced heights, where the lower peak is easily distorted or even be masked in noise. We propose a correlation-based collision identification approach and an interleaved window selection strategy to achieve a bounded division ratio for arbitrary incoming LoRa packet. Second, it is non-trivial to obtain accurate peak estimations as the inter-peak interference leads to peak distortion. We propose an iterative peak recovery algorithm where the highest peak component is iteratively estimated, recovered and extracted. Thus, we can eliminate the inter-peak interference as well as solving the near-far problem. Third, after collision separation, we find the packet decoding is severely impeded by the mixed impact of the carrier frequency offset (CFO) and reception window time offset. And the peak overlaps in the frequency domain also impacts the packet decoding as it hinders the estimation of FFT peaks. Based on the structure of LoRa preamble, we design a technique to estimate and compensate the CFO and window time offset for LoRa decoding. We also present a overlapping peak detection approach to identify peak overlaps from the FFT results based on the phase rotation of the Fourier transformation.

Main Results and Contributions. The main results and contributions of this paper are as follows:

- We proposed CoLoRa, a protocol to decompose multiple concurrent transmissions from one collision in LoRa, with the goal to bridge the gap between LoRa's vision to provide low-power long-distance connection to large scale IoT devices, and its practical capability.
- We address practical challenges in CoLoRa design. The performance of CoLoRa highly relies on accurate estimation of FFT peaks. We propose an efficient reception window selection strategy to generate balanced peaks. We design an iterative peak recovery algorithm to address inter-peak interference and recover peak information accurately. Finally, we remove the impact of CFO and time offset to accurately decode packets.

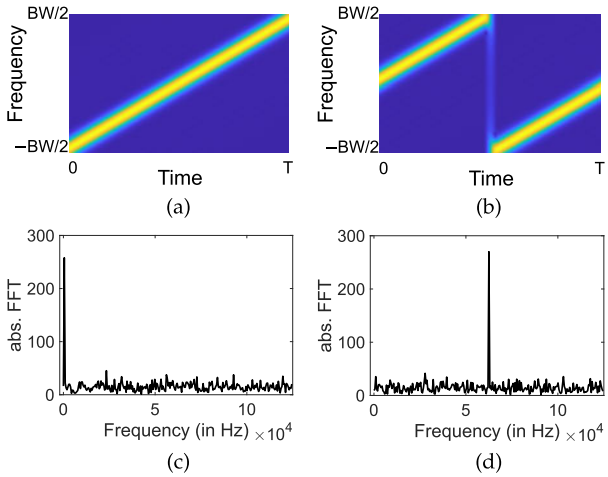

Fig. 2. LoRa Modulation/Demodulation Primer: (a)(b) Spectrograms of the standard chirp and the shifted chirp. (c)(d) Demodulation results of the standard chirp and the shifted chirp.

- We implement CoLoRa on USRP N210 and thoroughly evaluate its performance in three different scenarios. CoLoRa is completely implemented in software at the gateway without requiring any modifications at the end nodes. The experiments results show that CoLoRa can improve the network throughput by **3.4**× compared with Choir and **14**× compared with LoRaWAN.

The rest of the paper is organized as follows. Section 2 introduces the background and motivation of this work. Section 3 shows the basic architecture of CoLoRa. Sections 4, 5, and 6 describes detailed designs of CoLoRa. Section 7 presents discussions of CoLoRa regarding to the computational overhead and deployment cost of CoLoRa. Sections 8 and 9 presents our implementation and evaluation results. Section 10 introduces some related work. And finally, Section 11 concludes this paper.

2 Background and Motivation

2.1 LoRa Background

LoRa employs the chirp spread spectrum (CSS) as the physical layer modulation technique. CSS modulates signals into chirps of linearly increasing/decreasing frequency, i.e., up-chirps and down-chirps, making the signal occupying the entire spectral band. Chirp symbols are inherently robust against in-band interference and other channel degradations, and hence they can be detected and decoded even under extremely low SNR, which makes low power and long range communication be possible for LoRa end nodes [17], [18].

LoRa modulates data bits by cyclically shifting the frequency of the base up-chirp. As shown in Fig. 2a, the frequency of the base up-chirp increases linearly from $-\frac{BW}{2}$ to $\frac{BW}{2}$, and the length of the chirp is T. Thus, the frequency of the base up-chirp can be represented as $kt - \frac{BW}{2}$, where $k = \frac{BW}{T}$ is the gradient of frequency sweeping. And the base up-chirp $C(t)$ can be represented as

$$C(t) = e^{j2\pi \int_0^t (k\tau - \frac{BW}{2})d\tau} = e^{j2\pi(\frac{k}{2}t^2 - \frac{BW}{2}t)}.$$

Given the frequency shift f of the base up-chirp, an encoded up-chirp is then represented as $C(t)e^{j2\pi ft}$. Given the bandwidth BW, after the frequency shift, all frequencies higher than $\frac{BW}{2}$ will align down to $-\frac{BW}{2}$ as shown in Fig. 2b. LoRa defines N different frequency shifts, resulting in N uniformly shaped up-chirps to encode $SF = log_2 N$ bits.

LoRa receivers extract the encoded data bits by multiplying each received symbol with a base down-chirp, i.e., the conjugate of the base up-chirp $C^*(t)$, whose frequency decreases linearly over time. After the multiplication, each encoded up-chirp is despread as

$$C^*(t) \times C(t)e^{j2\pi ft} = e^{j2\pi ft},$$

which is a single tone with the frequency of f. Then an FFT is performed on the despread signal, leading to an energy peak in an associated FFT bin corresponding to the frequency shift f, as shown in Figs. 2c and 2d. The location of the FFT energy peak is finally used for determining the encoded data bits.

At the physical layer, a typical LoRa packet is composed of multiple preamble symbols, 2 mandatory sync word symbols, 2.25 Start Frame Delimiter (SFD) symbols and a variable number of payload symbols. The preambles are all identical base up-chirps and the SFDs are identical base down-chirps. The payload symbols are shifted base up-chirps.

LoRaWAN [19] is a widely adopted MAC protocol for LoRa networks. In LoRaWAN, end-devices are directly connected to a gateway. LoRaWAN mainly adopts an Aloha based MAC for different nodes to transmit packets. It is also known that LoRaWAN scales poorly when the network density becomes high due to packet collisions [20], [21].

2.2 Motivation

LoRa is robust against channel noise as it can concentrate time domain signal energy into a single tone energy peak by despreading and the FFT. Therefore, LoRa signals can be detected and decoded even under extremely low SNR, enabling low power and long range communications. When signals from multiple LoRa clients collide at the receiver, multiple chirp segments overlap in the same reception window, which are transformed to multiple energy peak in the frequency domain. The LoRa demodulator cannot map each of these collided energy peaks to its corresponding transmitter, and thus it fails to decode the collided signals.

We leverage the fact that collided LoRa packets are likely to be misaligned in time. As shown in Fig. 1, when selecting reception windows misaligned with each received packet, chirps of each packet are divided by consecutive reception windows. The length of each reception window, denoted as L, is the same as the chirp length. Assuming the lengths of the two chirp segments from the same chirp are $\gamma_1 L$ and $\gamma_2 L$, respectively, we have $\gamma_1 L + \gamma_2 L = L$. We have two observations: (1) Given a specific LoRa packet, the ratio of the length for the two segments, i.e., $\gamma_1 : \gamma_2$, are identical for all chirps belonging to this packet. (2) For two misaligned LoRa packets, the segment length ratios of their chirps should be different. The ratios, therefore, can be applied to disentangle different packets in a collision.

3 Overview

Fig. 3 illustrates the overall architecture of CoLoRa. CoLoRa consists of three key components: a signal preprocessing module for identifying packet collisions and selecting the

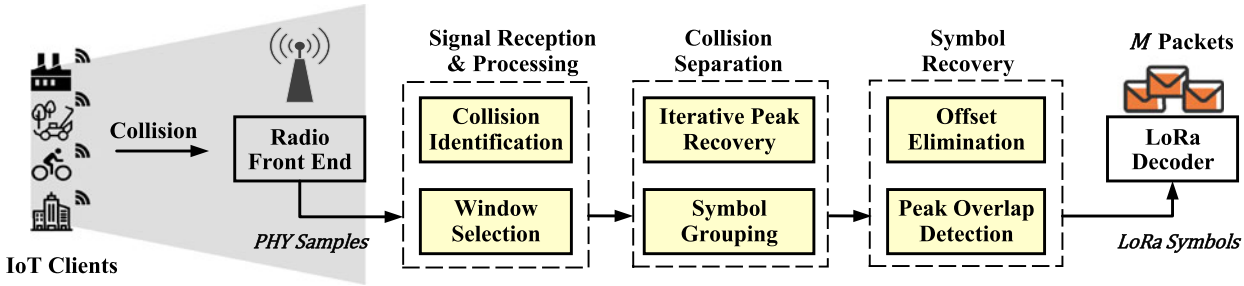

Fig. 3. Illustration of signal processing in CoLoRa.

optimal reception window; a collision separation module for estimating signal characteristics from collisions and clustering overlapped chirps to groups; a symbol recovery module for eliminating time/frequency offsets from separated symbols and resolving chirps of overlapped peaks. CoLoRa at the gateway mainly operates as following: (1) For a received signal, CoLoRa first detects whether the received packet suffers from a collision, and selects the optimal reception window to divide the received chirps into segments with the window length equal to a chirp. (2) For each reception window, CoLoRa transforms the low-SNR signals into robust FFT peaks and then accurately recovers the features of peaks in the presence of noise and inter-peak interference. Based on the peak estimation, CoLoRa calculates the peak ratios and clusters the peaks into multiple groups where each group contains the peaks of the same packet. (3) Finally, CoLoRa decodes each group of peaks while addressing the challenge of time/frequency offsets as well as resolving collided chirp symbols of overlapped FFT energy peaks. We will show the details of each component and how to address practical challenges in the following.

4 Signal Reception & Preprocessing

In this section, we present the design of CoLoRa on collision identification and optimal reception window selection for decoding LoRa collisions.

4.1 Collision Identification

Like in conventional LoRa, CoLoRa assumes no collision after receiving a signal and tries to decode the signal with a standard LoRa decoder. If the decoding fails (e.g., the decoded packet does not satisfy the CRC), CoLoRa then enables the collision identification process.

A straightforward method for collision identification is to analyze the strength of the received signal, i.e., from the Received Signal Strength (RSS). A sudden rise of the RSS indicates a packet collision. However, such RSS-based collision identification approaches are not appropriate in LoRa, as the signal strength of a LoRa packet can be extremely weak and usually overwhelmed by the channel noise. Besides, CoLoRa aims to identify and decode each collided packet from the collision, therefore, it has to determine the start of every collided packet preciously, which, however, cannot be satisfied by the RSS-based approaches.

We propose a correlation-based approach for LoRa collision detection, which can detect the existence and accurate start for each collided LoRa packet. Mathematically, the correlation is computed as follows. Let the samples $s[k]$, $1 \leq k \leq L$ refer to the standard LoRa preamble. Denote $y[k]$ as the received signal, which is the sum of the received preamble $\alpha \cdot s[k]$ and the noise term $w[k]$. We move the start of the standard preamble through the received signal and calculate their correlation Γ. The correlation at position Δ is

$$\begin{aligned}\Gamma(\Delta) &= \sum_{k=1}^{L} s^*[k]y[k+\Delta] \\ &= \sum_{k=1}^{L} s^*[k](\alpha \cdot s[k+\Delta] + w[k+\Delta]),\end{aligned} \tag{1}$$

where $s^*[k]$ is the complex conjugate of $s[k]$. When the standard LoRa preamble is strictly aligned with the preamble in the received signal, i.e., Δ is zero, the correlation, Γ can be expressed as

$$\Gamma(0) = \alpha \sum_{k=1}^{L} s^*[k]s[k] + \sum_{k=1}^{L} s^*[k]w[k]. \tag{2}$$

Through the correlation, the energy of the received preamble is efficiently concentrated by $\alpha \sum_{k=1}^{L} s^*[k]s[k]$, while the noise signal adds up incoherently. Therefore, the correlation produces a significant energy peak for the received preamble signal as shown in Fig. 4, which can be used for detecting the existence and accurate start for the received LoRa packet, even when the received LoRa signal is below the noise.

Putting the correlation approach into practice, however, is still very challenging. When there exists a CFO between the transmitter and the receiver, up-chirps in the received preamble have different initial phases. The initial phase of the ith preamble chirp is drifted by $f_{CFO} \times i \cdot T_{chirp}$. Thus,

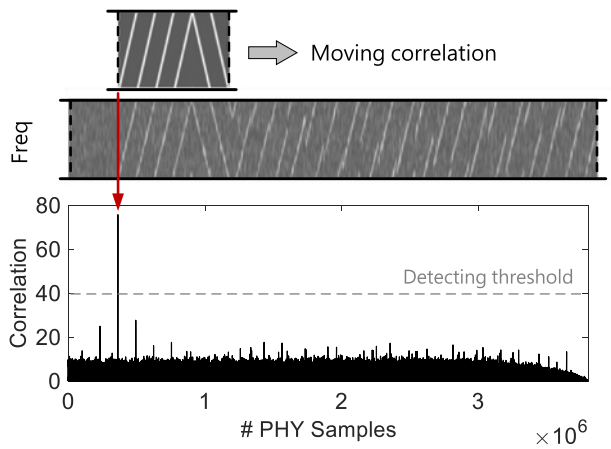

Fig. 4. Detecting LoRa packets by correlating the received signal with a standard LoRa preamble.

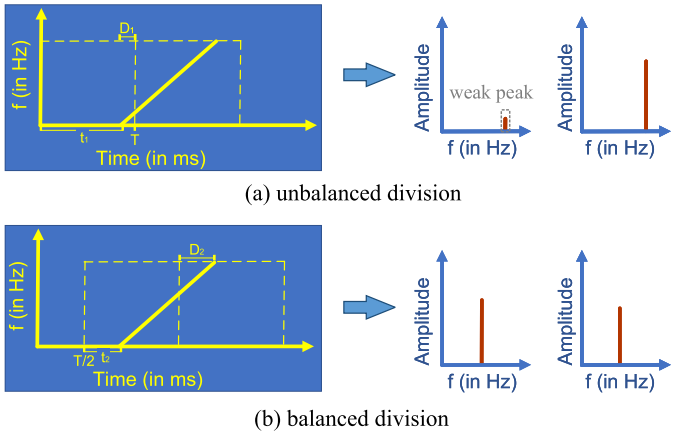

Fig. 5. Examples of interleaved windows for solving unbalanced window division.

the up-chirps in the generated standard LoRa preamble may have different initial phases compared to the received up-chirps. Such phase differences make samples of received preambles being accumulated destructively during the correlation, making the correlation peak of the entire preamble become low or disappear, even when the correlated sequence accurately aligns with the received packet.

We propose a enhanced two-stage preamble detection approach. We leverage the fact that CFO influences the initial phases of up-chirps in the received LoRa preamble, but the phase changes are in a predictable way, where phases of consecutive up-chirps drift linearly. Thus, we first manually adjust the initial phases of each generated ideal up-chirp, making their phases drift linearly with a specific step of $\Delta\varphi$. We use four different phase steps for generating four ideal LoRa preambles, i.e., making $\Delta\varphi$ equal to $\pi/2$, π, $3\pi/2$, and 2π. Then, we move these phase drifted ideal preambles through the received signal, and calculate their correlation coefficient, respectively. Our experiments show that given a received LoRa packet with arbitrary CFO, at least one of the four generated ideal LoRa preambles can produce a significant correlation peak, indicating a LoRa packet is detected from the received collisions. Meanwhile, we can also determine the start of each collided packet by identifying the position of each correlation peak.

4.2 Reception Window Selection

Upon receiving a signal, CoLoRa needs to cut the signal into continuous reception windows, each with the length equal to a chirp. A practical challenge is that the selection of reception windows impacts the decoding performance. For example, an improper selection of reception windows will result in an unbalanced division, where chirps are divided into very short segments corresponding to very low peaks, which are easily distorted or even masked in noises as shown in the Fig. 5a. Considering there are multiple packets in a collision, this is even difficult to achieve balanced division for all packets.

CoLoRa adopts an interleaved window selection strategy to achieve balanced division for all collided LoRa chirps. In the absence of collisions, this strategy acts like conventional LoRa receivers, where reception windows are selected to be aligned with each LoRa chirp. Upon receiving a signal, CoLoRa first detects the beginning of the first arrived LoRa packet. We use an Akaike Information Criterion (AIC) [22] based algorithm for detecting the precious packet beginning. The AIC algorithm is widely used in estimating the arrival time of seismic waves. At the heart of this algorithm is an auto-regressive model, which calculates the dissimilarities for signals before and after each sample point. The sample point of the highest dissimilarity estimation is determined as the beginning of the first arrived packet. Evaluations in [23] show that the AIC based signal detection algorithm can find an accurate packet beginning even under an extremely low SNR (i.e., averaged time-stamping error of $20\mu s$ even under a -20dB SNR).

Based on the result of the AIC detection, we select a set of reception windows W_1 whose beginning exactly aligned with the detected packet beginning. Then we use the W_1 to divide and demodulate the received signal, and we estimate the FFT peaks for the signal in each reception window. If only one peak is detected in each window, it indicates no collision, and then the packet is decoded like in conventional LoRa. Otherwise, as shown in Fig. 5b, we select a new interleaved reception window W_2 by moving the start of W_1 with a time offset of $\frac{T}{2}$, where T is the chirp length.

Here, we show that given any packet involved in the collision, at least one of those two interleaved reception windows W_1 and W_2 can give a balanced division with a bounded division ratio in $[\frac{1}{3}, 3]$. We first prove that given any received chirp, at least one of these two windows can divide the chirp with the minimum segment length (i.e., D) longer than $T/4$. As shown in Fig. 5, the time offset between the starts of the two reception windows W_1 and W_2 is $T/2$. Without loss of generality, assume the chirp-level time offset between pkt and W_1 longer than $T/2$, i.e., $t_1 > T/2$. Let the shorter chirp segments divided by these two windows are D_1 and D_2. Both D_1 and D_2 are smaller than $T/2$ and $D_2 = T/2 - D_1$. For accurate peak estimation, we choose the more balanced division between the results of these two reception windows, i.e.,

$$D = \max(D_1, D_2).$$

If D_1 is smaller than $T/4$, we have $D = D_2 > T/4$; otherwise, $D = D_1 > T/4$. Therefore, the shorter segment, resulted from either W_1 or W_2, should be limited in the range of $[T/4, T/2]$. Based on this, we calculate the division ratio R from the more balanced window, i.e.,

$$R \in \left[\frac{D}{T-D}, \frac{T-D}{D}\right] = \left[\frac{1}{3}, 3\right].$$

Therefore, by leveraging the interleaved window selection, we achieve the balanced division for any received LoRa chirp with a bounded division ratio.

5 Collision Separation

This section presents the iterative peak recovery algorithm for recovering accurate peak features in the presence of channel noise and inter-peak interference. We also show how to separate LoRa collisions by clustering collided LoRa chirps to multiple groups based on the estimated peak features.

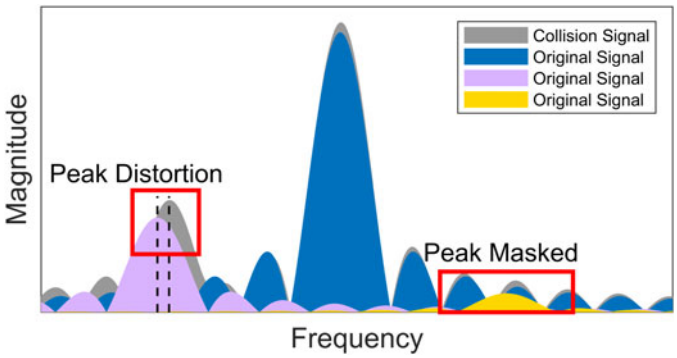

Fig. 6. Peak distortion and peak masked caused by inter-peak interference.

5.1 Iterative Peak Recovery

In each reception window, we transform the time domain signal to energy peaks in the frequency domain by despreading the received signal with base down-chirps and applying the FFT. We perform the FFT on the dechirped signals over a wider window by zero-padding the signal, which introduces more computation complexity but provides better resolution for FFT peaks in the presence of energy leakage. Then we need to estimate the frequency and height for each FFT peak to separate collided LoRa packets. While in practice, height estimation for peaks is prone to inter-peak interference. Revisiting the progress of LoRa decoding, we multiply down-chirp and apply FFT for signals in each reception window. Fig. 6 plots the result of the Fourier transform. Observe that there are periodical side lobes around each main peak, a property that stems from the time limited input sequence. The side lobes affect the peak estimation from two aspects. On the one hand, side lobes distort other main peaks and impede accurate measurement of the height and frequency of the impacted peaks. On the other hand, low-height peaks tend to be masked by side lobes of other strong peaks.

We propose an iterative peak recovery algorithm. The pseudo-code for the iterative peak recovery algorithm is shown in Algorithm 1. We first find the highest peak from the reception window. The benefit of using the highest peak is two-fold. First, the relative distortion of the highest peak is smaller than other peaks. Second, using the highest peak avoids incorrectly using side lobes as peaks since the highest peak normally cannot be any side lobe. By measuring the frequency and height of the highest peak, we can obtain a coarse estimation of the real peak. Later, we show how to refine the highest peak by searching and reconstructing an optimal chirp segment corresponding to the peak in the time domain. The reconstructed chirp segment can be used to recover the accurate peak information and remove the side lobe impacts to other peaks.

The rest of this section shows how to obtain an accurate recovery of the highest peak. Assume the highest peak has a height of h_0, a center frequency of f_0 and an initial phase of ϕ_0, which can be obtained from the FFT result. To obtain a coarse estimation of corresponding chirp segment, we need further to obtain the position, amplitude and length of the chirp segment.

Position Estimation. First, we need to determine the position of the chirp segment, i.e., whether the segment is adjacent to the former window or the latter window. We use the signal from both the former and the latter windows to help determine the chirp position. Denote the signal of the current window is x_0, and the signal of the former and the latter windows are x_1 and x_2. If we combine the signals of each two adjacent windows together, we can get $y_1 = [x_1, x_0]$ and $y_2 = [x_0, x_2]$, each with the length of two chirps. Then, we multiply two continuous base down-chirp with y_1 and y_2, respectively. After the FFT on the multiplication of y_1 and y_2, we can obtain two peaks both corresponding to our target chirp segment. These two peaks have the same frequency, i.e., f_0, each with the height of h_1 and h_2 . If the chirp segment is adjacent to the latter window, the height of h_2 should be higher than h_1, and vice versa. Note that if the chirp segment is balanced divided, the height of the unselected peak should be lower than $\frac{3}{4}$ of the selected one. Therefore, we can easily determine the position of the chirp segment and determine whether the chirp division for the current reception window is balanced.

Algorithm 1. Iterative Peak Recovery

Input: Signal in a reception window: Sig
Output: Height and frequency of peaks: $[H, F]$
$DemodSig = Sig \otimes DownChirp$;
while SUM(FFT($DemodSig$)) $> threshold$ **do**
  $[f_0, \phi_0, h_0]$ = HIGHESTPEAK(FFT($DemodSig$));
  $[Loc, h]$ = SEGMENTLOCATION();
  $L = \frac{Th_0}{h}$;
  $H = \frac{h_0}{f_s L}$;
  $\tilde{s}$ = INITIALCHIRPSEG(f_0, ϕ_0, H, L, Loc);
  S = ITERATIVEREFINE($\tilde{s}$);
  $[H_i, F_i]$ = PEAKMEASURE(FFT($S \otimes DownChirp$));
  CANCEL S FROM Sig;
  $DemodSig = Sig \otimes DownChirp$;
end
return $[H, F]$;

Amplitude and Length Estimation. We estimate the amplitude of the chirp segment based on y_1 and y_2. Assume h_2 is the height of the higher peak corresponding to y_2. Then, as a chirp symbol can only span two windows, y_2 should contain the complete chirp symbol. As the peak height is proportional to the length of segment (i.e., L), we have $\frac{h_2}{h_0} = \frac{T}{L}$. Thus, we can estimate the length of chirp segment in x_0 as $L = \frac{h_0}{h_2}T$. According to the principles of the discrete Fourier transform, we can further calculate the amplitude to chirp segment as $H = \frac{h_0}{f_s L}$, where f_s is the sampling frequency.

Accurate Peak Recovery. Based on estimated peak frequency f_0, peak phase ϕ_0, segment length L and chirp amplitude H, we can reconstruct the chirp segment in the time domain as

$$\tilde{s}(t) = He^{j2\pi ft + \phi_0}C(t), \tag{3}$$

$t \in [0, L]$ if the segment is adjacent to the previous window, or $t \in [T - L, T]$ if the segment is adjacent to the following window. The chirp segment $\tilde{s}(t)$ is a coarse estimation of real chirp segment as its corresponding peak may be distorted by side lobes of other chirps. We determine whether a chirp segment is estimated accurately based on how much energy remains after the signal is eliminated. For a more accurate estimation of a chirp segment, we will obtain less

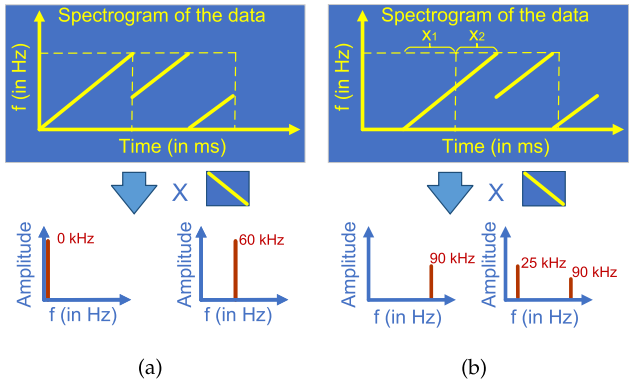

Fig. 7. Signal demodulation with aligned and unaligned reception windows: (a) Conventional receiver uses aligned windows and demodulates chirp by chirp. (b) CoLoRa receiver uses unaligned windows and transforms each chirp symbol to two energy peaks in the frequency domain.

residual energy in the remaining signal after canceling it from the collision signal. Therefore, we find the optimal chirp segment estimation by searching from the near space of the initial chirp segment, i.e., $\tilde{s}(t)$. We iteratively vary the phase, amplitude, start frequency and length of the initial chirp segment, and reconstruct and cancel it from the original signal to calculate the energy of the residual signal. The goal for the searching is to minimize the residual energy, which indicates the optimal estimation of our target chirp segment, i.e., S in Algorithm 1. Then we obtain the accurate peak by multiplying S with a base down-chirp and applying FFT. Moreover, by canceling S, we also cancel its interference to other peaks. We iteratively recover accurate peaks in the remaining signal until the residual energy is lower than a threshold. Using our iterative peak recovery algorithm, we can also address the near-far problem where a strong signal from a near transmitter interferes with a weaker signal from a far transmitter.

5.2 Peak-Ratio Based Symbol Grouping

Till now, accurate FFT energy peaks have been recovered. We show how to use these estimated peaks to separate LoRa packet collisions. In LoRa, an encode chirp symbol is an up-chirp with a frequency shift, i.e., $x(t) = He^{j2\pi ft}C(t)$, where H is the signal amplitude and f is the shifted frequency to encoded data bits. When the reception window is strictly aligned with the received packet as shown in Fig. 7a, each chirp in the packet is transformed to an FFT peak whose frequency corresponds to the coding words of the chirp symbol. CoLoRa leverages the unaligned windows for extracting peak ratio features for each received chirp. Denote τ is the window-level time offset between the packet start and the reception window in Fig. 7b. For a chirp, the first chirp segment $x_1(t)$ in the first reception window can be written as

$$x_1(t) = x(t-\tau) = He^{j2\pi f(t-\tau)}C(t-\tau) \qquad \tau \le t < T.$$

As time shift can be translated to frequency shift, we have $C(t-\tau) = e^{-j2\pi k\tau}C(t)$. Thus, $x_1(t)$ can be rewritten as

$$x_1(t) = He^{j2\pi(-f\tau+(f-k\tau)t)}C(t) \qquad \tau \le t < T. \tag{4}$$

Similarly, the second chirp segment $x_2(t)$ in the second reception window is written as

$$\begin{aligned} x_2(t) &= x(t-(T-\tau)) \\ &= He^{j2\pi(-f(T-\tau)+(f+k(T-\tau))t)}C(t) \quad 0 < t < \tau. \end{aligned} \tag{5}$$

We multiply the signal in each reception window with a standard down-chirp, despreading each chirp segment to a single tone. Mathematically, for $x_1(t)$ in the first reception window, after the multiplication with the standard down-chirp, we have

$$\begin{aligned} \hat{x}_1(t) &= He^{j2\pi(-f\tau+(f-k\tau)t)}C(t)\cdot C^*(t) \\ &= He^{j2\pi(-f\tau+(f-k\tau)t)}, \end{aligned} \tag{6}$$

which is a single tone at the frequency of $f_1 = f - k\tau$. $x_2(t)$ in the second window can also be despreaded to a single tone, whose frequency is $f_2 = f + k(T-\tau) = f - k\tau + BW$. When using a sample rate equaling to the bandwidth (i.e., BW), these two despreaded signals ($\hat{x}_1(t)$ and $\hat{x}_2(t)$) share the same frequency in the spectrum, i.e., $f_1 = f_2 = f - k\tau$. We apply the FFT to the despreaded signal at each reception window, transforming the chirp segments to energy peaks in the frequency domain. As shown in Fig. 7b, two segments of the same chirp are transformed to two peaks located at the same FFT bin of $f - k\tau$ (90 kHz in this example). For the $x_1(t)$ in the first reception window, the height of its energy peak is

$$h_1 = \sum_{n=0}^{N-1} \hat{x}_1[n]e^{-j2\pi(f-k\tau)\frac{nT}{N}}, \tag{7}$$

where $\hat{x}_1[n]$ is nth sampling point of $\hat{x}_1(t)$ and N is the total number of sample points of a chirp. Substituting $\hat{x}_1[n]$ with the Eq. (6), we have

$$h_1 = Hf_s(T-\tau). \tag{8}$$

This shows that the height of each energy peak is proportional to the length of the corresponding chirp segment. Similarly, for the second segment $x_2(t)$, the peak height is

$$h_2 = Hf_s\tau. \tag{9}$$

For each chirp, we define peak ratio P as the height of the second energy peak divided by that of the first peak, i.e.,

$$P = \frac{h_2}{h_1} = \frac{\tau}{T-\tau}. \tag{10}$$

It can be seen that the peak ratio is determined by the window-level time offset τ. Thus, the peak ratio is identical for all chirps of the same packet. To summary, through the analysis, we have the following results:

- The peaks of two segments of the same chirp are located at the same frequency, with height proportional to segment length.
- The peak ratio is determined by the window offset, i.e., $P = \frac{\tau}{T-\tau}$, and thus is identical for all chirps of the same packet.
- The peak ratio is different for packets with different arrival time.

CoLoRa leverages the peak ratio to distinguish packets and then uses peak information to facilitate decoding. Specifically, as the peaks of two segments from the same chirp

are located at the same frequency, we can pair these two peaks by matching the peaks of the same frequency in two consecutive windows. We calculate the peak ratio for each pair of peaks, i.e., the last peak's height divided by the former peak's height. The peak ratio is identical for chirp symbols of the same packet but different from packet to packet. Thus, we group the collided peaks into k different clusters based on their peak ratios, where each cluster of peaks corresponds to symbols of the same LoRa packet.

The performance of peak ratio based collision separation is highly related to the chirp level time offset among packets. For rare cases that the time offsets between collided LoRa packets are very small or even strictly synchronized, the collided packets may have similar peak ratios. We propose an advanced peak ratio design for separating collided LoRa chirp by utilizing the signal strength as a characteristic, named *Chirp Power IDentification (CPID)* . The key idea is that the two collided packets may be similar in arriving time but have a difference in signal strength. Thus, for each pair of peaks, the advanced peak ratio is represented as a vector (h_1, h_2), where h_1 is the peak height for the former segment, and h_2 is the height of peak for the latter segment. With the advanced peak ratio (i.e., *CPID*), we group peaks to the same packet based on the original peak ratio h_1/h_2 as well as the euclidean distance of vectors. As *CPID* considers both the division ratio as well as the signal strength, it can distinguish collided LoRa packets with short or zero time offset when their received signal strength exhibits difference. For example, consider two packets, packet-A and packet-B, arriving at the receiver with a very small chirp-level time offset. After the segmentation, both packets have similar peak ratio ($3:1$ in our case). Hence the receiver fails to decode by naive peak ratio. However, each chirp from packet-A is transformed into two peaks with the height of 9 and 3, and the advanced peak ratio can be represented as $(9, 3)$. Similarly, the advanced peak ratio for packet-B can be represented as $(6, 2)$. The euclidean distance between the advanced peak ratio of two packets is very large, and hence we can distinguish those two packets. In the evaluation, we will show the performance improvement for the advanced peak ratio design.

Finally, we use an improved k-means grouping approach for grouping collided peaks to corresponding LoRa packets. Traditional k-means randomly chooses each cluster's initial center and uses the euclidean distance for grouping points.. Differently, our improved k-means is a time-series-based approach, in which we process peaks in time order. The improvement of the grouping method is from the following aspects. First, we can set the number of clusters, i.e., k, to the number of collided packets. Second, the number of clusters (i.e., packets) at any time is constrained by the packets observed so far. Therefore, a new cluster can only be added after every start of a new packet. Third, for two adjacent windows, there should be one and only one chirp for each packet. Thus, during the process of k-means, we exactly group one pair of peaks to each cluster in every two adjacent windows. Based on the improved k-means approach, we can efficiently and accurately obtain k groups of peaks, each corresponding to peaks of the same packet. Therefore, the collided packets can be finally separated through the grouping process.

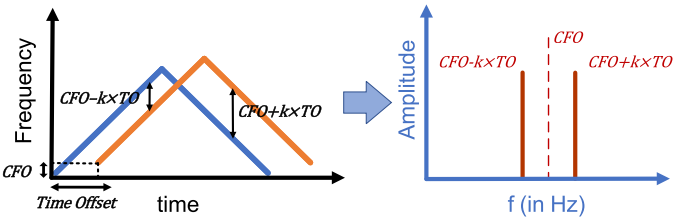

Fig. 8. Examples of carrier frequency offset and window time offset estimation.

6 Symbol Recovery

This section presents how to recover separated chirp symbols for LoRa decoding by eliminating their time/frequency offsets, and we show how to resolve collided chirp symbols of overlapped FFT energy peaks.

6.1 Frequency Offset Elimination

During the peak-ratio-based symbol grouping, collided packets are translated to multiple groups of peaks, each group corresponding to symbols of the same packet. To decode the separated signal and recover data contents from grouped peaks, we first need to address the peak frequency shifts introduced by carrier frequency offsets (CFO) and window time offsets. Both the CFO and window time offsets cause chirp decoding error as they introduce peak frequency biases. Normally, we can measure the frequency shift of the preamble to obtain the CFO. The challenge is that window time offset between the chirp and reception window also results in frequency shifts. We utilize the unique structure of the LoRa packet to resolve this challenge.

In a LoRa packet, the preamble contains base up-chirps, and the SFD are base down-chirps. When there is no window time offset and CFO, both base up-chirps and base down-chirps are transformed to peaks with zero frequency shift. Otherwise, each chirp will be transformed to a peak with a non-zero frequency shift related to the CFO and window time offset. Though both CFO and window time offset incur frequency shifts, their impacts differ. As shown in Fig. 8, CFO causes identical frequency shift for both up-chirp and down-chirp, while window time offset causes opposite frequency shift for up-chirp and down-chirp. Denote $TO = \tau$ is the window time offset, for the base up-chirp, the total frequency shift can be calculated as

$$\delta f_{up} = -\tau k + CFO, \tag{11}$$

where $k = \frac{BW}{T}$ and $-\tau k$ is frequency shift caused by window time offset. Similarly, for the base down-chirp, the frequency shift is

$$\delta f_{down} = \tau k + CFO.$$

We calculate δf_{up} and δf_{down} by multiplying the preamble with standard down-chirps and SFD with standard up-chirps. Then we can estimate the CFO as

$$CFO = \frac{\delta f_{up} + \delta f_{down}}{2}.$$

Meanwhile, by substituting CFO into Eq. (11), we can also obtain the estimation of the window time offset τ. We compensate for each peak with the calculated CFO and window

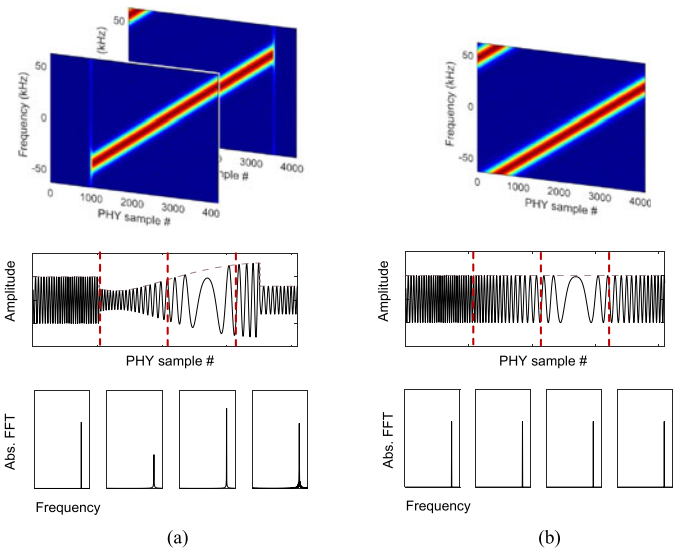

Fig. 9. Illustration of overlapped peak detection.

time offset. Finally, we can decode a packet with a group of peaks using a standard LoRa decoder.

6.2 Accounting for Peak Overlapping

Previous collision decoding process exploits the fact that energy peaks of two collided chirp segments, typically, do not locate at the same FFT bin. Therefore, peaks of different symbols can be estimated separately in the frequency domain. When two chirps with close initial frequencies collid, their FFT peaks will overlap in the frequency domain, hindering the estimation of peak features and causing symbol clustering errors.We address this problem by identifying overlapped peak from the FFT results.

We use a two-chirp collision to illustrate how CoLoRa identifies overlapped peaks. Denote the initial frequencies of two collided chirps are f and f', where $|f - f'| < BW/2^{SF}$ as the chirps are transformed to peaks at the same FFT bin. CoLoRa leverage the *phase rotation property* of the Fourier transform [10] to amplify the frequency differences and detect overlapped peaks for separating collided packets. A shift in the time domain can be translated to a phase rotation in the frequency domain for the Fourier transform. Specifically, denoting an FFT peak $R(f)$ corresponds to a single chirp segment $r(t)$, FFT on the same signal with a time shift τ causes the peak phase rotating by $2\pi f\tau$, i.e.,

$$\mathcal{F}\{r(t+\tau)\} = R(f) \cdot e^{j2\pi f\tau}.$$

Based on this, the same time shift on the two collided chirps will introduce different phase rotations for peaks in the frequency domain, i.e., $R(f) \cdot e^{j2\pi f\tau}$ and $R(f') \cdot e^{j2\pi f'\tau}$ due to their frequency difference. Therefore, the peaks of two collided chirps will overlap either constructively or destructively depending on the phase rotation value of the two overlapped peaks.

For detecting the peak overlapping, CoLoRa first divides the time-domain chirp into multiple sub-segments (e.g., 4 in our example as Fig. 9). Then, for signals in each segment, CoLoRa transforms them into FFT peaks, all at the same FFT bin. Suppose the signal consists of multiple overlapped chirps. In that case, the peak heights corresponding to different sub-segments will vary, as shown in Fig. 9a, due to the phase rotations for overlapped chirps. While for normal peaks without signal overlapping, FFT peaks corresponding to different sub-segments have similar heights, as shown in Fig. 9b. Therefore, CoLoRa can distinguish a normal peak from the overlapped one by comparing the peak heights of different sub-segments. If peak heights of all sub-segments vary more than a noise threshold, CoLoRa identifies the peak as an overlapped peak. The detected overlapping peaks are finally grouped into clusters with no matching item at the corresponding windows.

7 Discussion

The Near-Far Problem. The near-far problem or hear-ability problem is a situation that is common in wireless communication systems. Consider a group of colliding transmitters in which some are physically closer to the receiver than others. The signal strength of the nearby transmitters will be much stronger than those of far-away transmitters due to the difference in signal propagation attenuations. Hence, after demodulation at the receiver, the nearby transmitters will have clear peaks that are readily discernible. In contrast, the far-away transmitters may have weak peaks that strong peaks from nearby transmitters can easily distort. CoLoRa leverages the iterative peak recovery algorithm for settling the near-far problem. When packets from different transmitters are received with significantly different signal strength, CoLoRa uses the peak recovery algorithm to recover LoRa peaks with weak strength by iteratively canceling the interference of prominent peaks from the collisions. Moreover, the iterative peak recovery algorithm also refines the frequency and height estimation for weak signal peaks by adaptively searching the frequency, amplitude, and segment length for each chirp segment.

Computation Overhead. Compared with the standard LoRa demodulation, the computation complexity of CoLoRa mainly stems from the peak estimation in each reception window. CoLoRa uses an iterative peak recovery algorithm for removing the inter-peak interference in FFT peak estimation. Therefore, it must reconstruct and cancel chirp segments from collision signals in each reception window multiple times. Besides, to better detect the energy leakage and estimate accurate peak height, CoLoRa performs the Fourier transform of the collision chirps over a wider window ($10\times$ larger) by zero-padding the signal which increases the computation complexity of the FFT. The computation overhead of CoLoRa is similar to that of Choir, as both require accurate estimations for all FFT peaks in the frequency domain. CoLoRa achieves a performance improvement compared with Choir as it leverages more signal features for collision separation (i.e., frequency and amplitude features for CoLoRa and only frequency for Choir). Note that all the computation overhead introduced by CoLoRa happens at the high-end gateway, which requires no modification on the low-power end nodes. As a result, most state-of-the-art commercial gateways with high-performance processors and wall-plugin power supplies can afford this computation overhead of CoLoRa. We can also adopt a Cloud Radio Access Network (C-RAN) architecture where the gateway offloads the computation overhead to a cloud

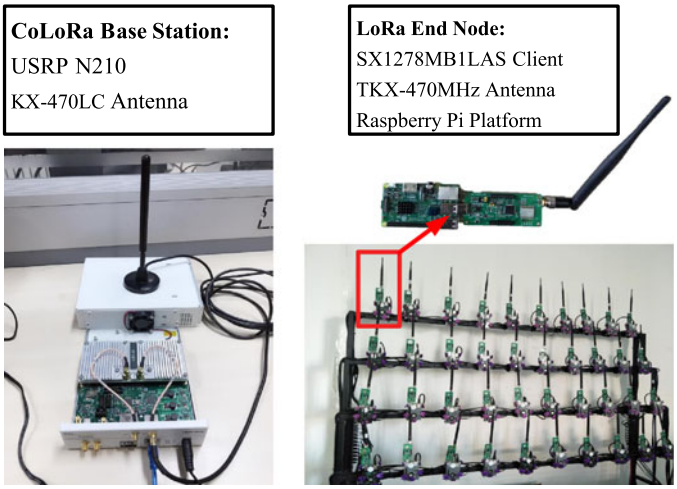

Fig. 10. CoLoRa gateway on USRP N210 and the indoor *LoRaNet* testbed.

server by forwarding the received signals through a backhaul connection for further signal processing.

Storage Requirements. The current implementation of CoLoRa requires the gateway the store IQ samples for signal reprocessing in case of packet collisions. The storage requirements depend on the transmission duration of end nodes and the sample rate at the gateway. For example, for transmissions in our experiments where each packet consists of 35 payload symbols with a sample rate of 1M/s, the storage buffer is 16 MB, where IQ samples can be recorded over the past two seconds. As collision decoding requires more computation complexity than the standard LoRa decoding, a holistic demodulation method that integrates standard LoRa decoding and collision resolving is still an open issue in our future work. Besides, with the development of low-cost storage and high-performance computation platforms, most gateways will be equipped with signal storage and processing abilities to support the deployment of CoLoRa.

Compatibility With Commercial Gateways. CoLoRa is expected to be compatible with commercial LoRa gateways to reduce deployment costs. State-of-the-art LoRa gateways consist of integrated RF front-end chips (e.g., SX1255) for capturing digital I and Q samples and a digital baseband processing chip (e.g., SX1301) for simultaneously receiving LoRa packets. An FPGA/CPLD is connected between the RF front-end chips and the digital baseband processing chip to adapt the digitized I/Q stream from the RF front-end to the specific format required by the digital baseband processing. Therefore, CoLoRa can be implemented on the FPGA component without disturbing the normal functions of the LoRa gateway. A holistic collision separation and decoding design perfectly compatible with the commercial LoRa gateways is our future work.

8 Implementation & Evaluation

8.1 Implementation

Hardware. We implement CoLoRa gateway on USRP N210 software defined radios (SDRs) [24] with a UBX daughterboard [25] as shown in Fig. 10. The gateway uses a single antenna with 2 dBi gain and can receive signals at 470 MHz bands for LoRa. The sample rate of the SDR receiver is 1M/s. CoLoRa is independent of hardware platforms and can be implemented on multiple platforms as long as the physical samplings can be obtained. CoLoRa requires no modification on end nodes, and thus it works well for LoRa deployments with commercial end devices. In our experiment, we use commercial LoRa end nodes, each with an SX1278 chip [26].

Software. We use the UHD+GNU-Radio library [27] for developing our own LoRa demodulator and implement CoLoRa in MATLAB to process PHY samples. The functionality of CoLoRa is to decompose an m-packet collision into m sequences of collision-free symbols and then translate them into m packets. To facilitate the experiment, we also implement a LoRa decoder in MATLAB.

8.2 Evaluation

Scenario. We evaluate the performance of CoLoRa in the following scenarios.

- LoRa testbed (*LoRaNet*) which consists of 40 LoRa end nodes as shown in Fig. 10. Each end node consists of an SX1278 chip, working at the frequency of 470 MHz and placed at a fixed position of a shelf. All the LoRa nodes are connected to a backbone network through the Raspberry Pis, and thus information from them can be efficiently collected. We also design the software to support online parallel programming and data collection from all nodes to facilitate the experiment. We can also control each LoRa node through Raspberry Pi, facilitating precise collision generation. By default, the testbed nodes in our experiments transmit LoRa packets of 35 payload symbols using the spreading factor SF = 12, coding rate CR = $4/5$ and bandwidth BW = 125 kHz.
- Outdoor real LoRa network, where 20 LoRa temperature and humidity sensors are placed at different locations of the campus such as buildings, roads, and parking lots as shown in Fig. 13. Each sensor node can collect and transmit temperature and humidity data to the gateway by LoRa packets. The distance between the sensors and the gateway is in the range of 10 meters to 500 meters.
- For scalability evaluation, we emulate a large-scale LoRa network scenario where 1,000 LoRa nodes work simultaneously and generate collisions with additional white Gaussian noise.

Baseline. Both the FTrack [14] and mLoRa [13] receivers decode LoRa collisions based on the time domain signal analysis. They cannot deal with low SNR LoRa signals from the outdoor deployed LoRa network. Thus, in this experiment, we only compare the performance of CoLoRa with Choir and two other LoRaWAN based approaches:

- Pure LoRaWAN [19]: The widely used standard LoRaWAN baseline using ALOHA.
- LoRaWAN+Oracle: LoRaWAN with an oracle scheduler that schedules transmissions optimally to avoid collisions.
- Choir [12]: A recent LoRa collision resolution approach that decouples collisions using hardware imperfection of end nodes. In practice, the hardware offsets of low-cost LoRa end nodes fluctuate according

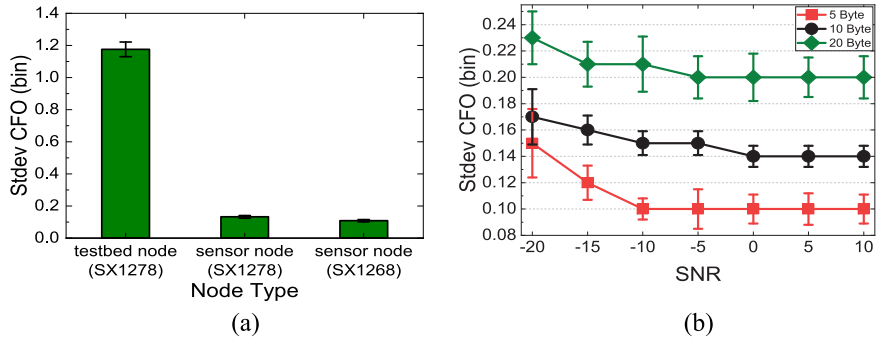

Fig. 11. Characterizing Hardware Offsets: the root mean-squared error of the frequency offset within a packet for (a) different type of end nodes with the same configuration (SF12, packet length of 10 Byte). (b) different packet length and SNRs (sensor node with SX1268).

to the node type, packet length, and SNR, as shown in Fig. 11. Thus, we use both the fractional FFT bin and the peak magnitude to match correct transmitters in our implementation.

Metrics. We mainly evaluate CoLoRa with the following metrics.

- Symbol Error Rate (SER): The # of error symbols with respect to the total. Chirp symbols are the basic unit for a LoRa packet, and we use SER to evaluate the physical layer demodulating accuracy.
- Packet Loss Rate (PLR): the # of miss-detected and undecodable packets with respect to the total. LoRa transmitter encodes data into symbols with certain encoding techniques, e.g., Gray indexing, data whitening, interleaving and Forward Error Correction (FEC). This tolerates some symbol errors for packet decoding. Hence the PLR should be closely related to but not the same as the SER.
- Network Throughput (NT): the # of received bits or symbols divided by time.

9 Results

9.1 Decoding Multi-Packet Collision

This experiment examines CoLoRa's performance for separating multiple packets in collisions. As LoRaWAN cannot separate packets in collisions, their performance under collision is extremely low. Thus, we only show the performance of CoLoRa and Choir, which can separate multi-packet reception in LoRa.

Method. We use the *LoRaNet* testbed to efficiently generate multi-packet collision where we can control each collided node accurately. The SNR survey for signals from the testbed nodes is shown in Fig. 12. The SNR measurements at the gateway are estimated based on the received signals. For calculating the SNR, the gateway first dechirps the received LoRa preambles by multiplying them with a standard down-chirp. Then, the gateway obtains the power spectral density (PSD) by transforming the signal from the time domain to the frequency domain. As the frequency of the transmitted signal is already known (e.g., the initial frequency of a preamble chirp is $-BW/2$), the gateway estimates the SNR by comparing the energy of the target frequency and energy of all remaining frequencies. To produce a collision with M overlapped packets, we use a beacon to synchronize the transmission for M different end nodes. All M end nodes wake up and transmit a LoRa packet upon receiving a beacon. We allow a random processing delay for each end node. All packets are generated with a specific known sequence of bytes. At the gateway, packets from the M end nodes overlap, leading to an M-packet collision. We produce collisions with different overlapped packets by changing the number of involved end nodes M. We use CoLoRa and Choir to decompose the collided packets at the gateway. Then for each decomposed packet, we use a standard LoRa decoder to translate the chirp symbols into data bits. The packets sent by each end node are known prior. Thus, we can verify the correctness of decoded packet and calculate the SER, PLR, and network throughput of all nodes in this experiment.

Results. Fig. 14a shows the SER for CoLoRa and Choir. As concurrent nodes increase from 1 to 20, the SERs of both approaches grow. The SER of CoLoRa increases much more slowly than that of Choir. This is because CoLoRa extracts more efficient features to separate packets. Choir uses the hardware imperfection which is less stable and difficult to detect especially under inter-chirp interference and channel

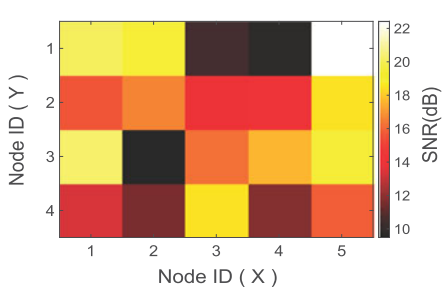

Fig. 12. SNR survey for LoRa nodes on the *LoRaNet* testbed.

Fig. 13. The outdoor deployed LoRa network, each sensor node transmits temperature and humidity data by LoRa packets periodically.

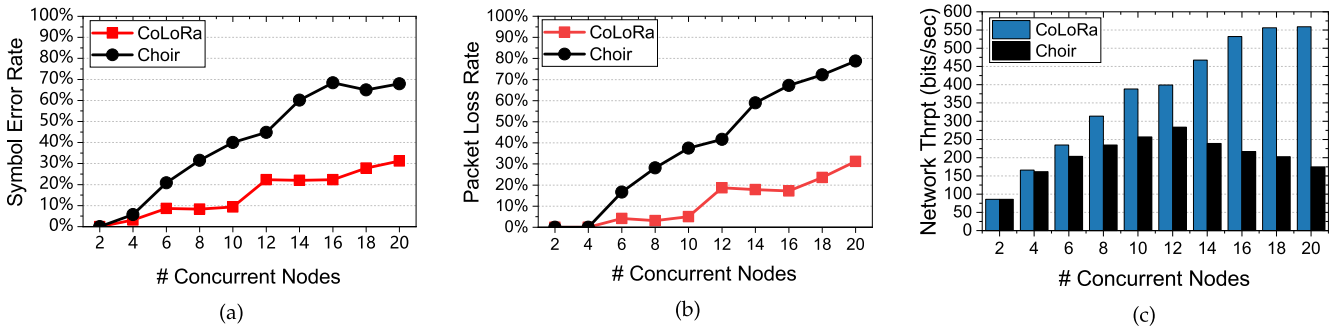

Fig. 14. Decoding collisions with different concurrent transmitters at a single-antenna USRP gateway. (a) Symbol Error Rate. (b) Packet Loss Rate. (c) Network throughput.

noise. This also coincides with the result in Choir[12] that it supports less than 6 concurrent transmissions, and with more concurrent transmissions, its performance degrades significantly. We further investigate the performance of CoLoRa and find that packet loss usually happens when two packets have similar peak ratios causing peaks to be clustered into a wrong group. We can also see that when the number of current nodes is less than 4, CoLoRa and Choir have similar performance. However, when the number of nodes increases, CoLoRa quickly outperforms Choir.

It should be noted that we can also leverage the coding in LoRa implementation. LoRa adopts the forward error correction (FEC) mechanism, where every four data bits are encoded with 1 to 4 additional bits of redundancy corresponding to CR4/5 to CR4/8. Some symbol errors can be corrected through the decoding process. Therefore, the packet loss rate can be similar or lower than the symbol error rate. LoRa can use different coding rates to compensate for the errors for different symbol error rates. We send decomposed chirp symbols to the implemented LoRa decoder to extract the packet's content. In our experiment, we initialize the testbed nodes with a coding rate of 4/8, where each four useful data bits are encoded eight bits along with FEC code. Under this coding rate, the PLR of CoLoRa and Choir are shown in Fig. 14b. We can see that the performance of CoLoRa is better than Choir, especially under a high concurrent collision.

After decoding, we can get the overall network throughput as shown in Fig. 14c. The network throughput of CoLoRa increases as the number of concurrent nodes increases from 2 to 20. This is due to the benefit of multi-packet reception in CoLoRa. Note that when the number of concurrent transmitters is more than 16, the network throughput increase starts to get slow and even be flat. This is because when the number of concurrent nodes is large, more packets become undecodable for CoLoRa. Meanwhile, we can see that the network throughput of Choir is much lower than that of CoLoRa. Moreover, the network throughput even decreases when concurrent nodes are larger than 12 due to too many packets that Choir cannot decode. When the number of concurrent nodes is 20, the network throughput of CoLoRa (552 bps) is about 3.4× of Choir (162 bps).

9.2 Micro-Benchmarks

In this experiment, we verify CoLoRa's functionality with micro-benchmarks to explain the peak recovery and overlap detection designs.

Method. We first experiment with the iterative peak recovery algorithm by evaluating its peak estimation accuracy over different SF configurations. CoLoRa decomposes LoRa collisions based on accuracy FFT peak estimation, including both the frequency and height of the FFT peak. A peak is considered as accurately estimated when its frequency bias is within ± 1 FFT bin, and the height error is within $\pm 10\%$. For generating peak collisions with known ground truth, we collect IQ samples of LoRa packets from ten *LoRaNet* testbed nodes at different SFs, each packet with known contents and amplitudes. We overlay the collected samples of different packets for generating LoRa collisions and use the iterative peak recovery algorithm for estimating peaks from the collision signals. We implement a threshold-based peak detection algorithm as the baseline approach for the performance comparison. Then, we experiment with the functionality of the overlapped peak detection mechanism by evaluating the SER performance of the CoLoRa receiver with and without the overlap detection module. In this experiment, we also verify the impact of different packet configurations, i.e., different SFs and bandwidths.

Results. Fig. 15 shows the peak estimation accuracy for the iterative peak recovery and the baseline approach over different SFs. The baseline approach detects FFT peaks when the peak is higher than a threshold. Moreover, it estimates peak heights and frequencies without considering peak distortion and power leakage. Thus, it introduces more peak estimation errors than the iterative peak recovery. As the SF decreases, the peak estimation accuracy for both approaches decreases. This is because the chirp symbols in low SFs are demodulated into fewer FFT bins. Thus, the energy of different peaks is more likely to interfere with each other, causing peak estimation errors.

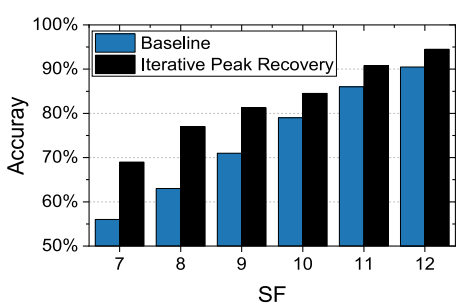

Fig. 15. Peak estimation accuracy for the iterative peak recovery and baseline peak estimation approaches over different SFs.

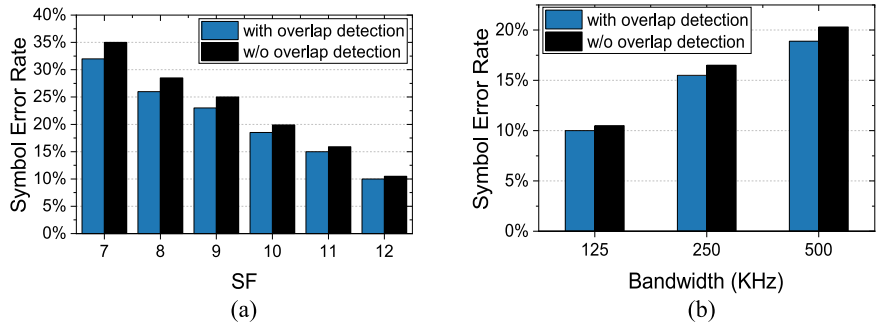

Fig. 16. Symbol Error Rates for CoLoRa with and without overlapped peak detection module with (a) different SF configurations and (b) different bandwidth configurations.

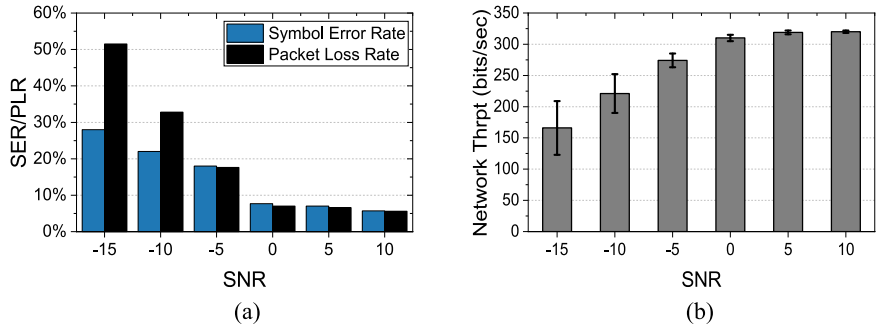

Fig. 17. CoLoRa performance for decoding collisions under different level of SNR. (a) Symbol Error Rates and Packet Loss Rates. (b) Network Throughput.

Fig. 16 shows the SER performance of the CoLoRa receiver with and without overlapped peak detection under different SF and bandwidth configurations. Peak overlap in the frequency domain can lead to symbol grouping errors and further disturbs packet decoding. We detect overlapped peaks by exploiting the time features of the collided signal and group the identified overlapped peaks to clusters that have no matching item at the corresponding windows. Results in Fig. 16 show that the overlapped peak detection can significantly decrease the symbol errors for LoRa packet decoding, as it avoids grouping errors for those overlapped peaks. Fig. 16 also reveals that LoRa chirps with the high SF and low bandwidth can be more resistant to the inter-packet interfere and channel noise, as chirps of those configurations have a long symbol duration and thus can concentrating more energy for signal decoding.

9.3 Impact of SNR

In this experiment, we show the impact of SNR on the performance of CoLoRa.

Method. High channel noise will cause peaks of chirp segments to suffer from distortion, which further disturbs the calculation of peak ratios. To characterize the impact of channel noise, we use the testbed to produce packet collisions where each end node transmits a randomly chosen sequence of bits concurrently. We use ten testbed nodes for the experiment, where one node transmits as the beacon transmitter and other nodes replay for generating LoRa collisions.

Results. We define the SNR of a collision signal as the SNR of its strongest signal component. As shown in Fig. 12, the SNRs of different nodes are diverse, but all above the noise floor. Thus, for precise SNR control and emulating low SNR scenarios, we artificially add noise traces to the received collision signals. By controlling the magnitude of the added noise traces, we can achieve a certain SNR in dB defined as $10log_{10}(A^2 \ / \ \mathbb{E}[Z_Q^2(t) + Z_I^2(t)])$, where $z_Q(t)$ and $z_I(t)$ are the Q and I traces of the added Gaussian noise, and A is the signal's amplitude.

Fig. 17a shows the SER and PLR for CoLoRa under different levels of SNR. CoLoRa works well for low SNR signals as it concentrates the energy of a chirp to a signal peak in the frequency domain. As the SNR decreases, both the SER and PLR increase as the channel noise disturbs the peak detection and frequency estimation. Note that the PLR increases faster than the SER when the SNR is very low. The high channel noise introduces severe interference on most received packets that cannot be corrected through the LoRa FEC mechanism. While for a high SNR, some symbol errors caused by the channel noise can be corrected through the LoRa coding mechanism, and thus the PLR is similar even slightly lower than SER in high-SNR scenarios. Fig. 17b shows the overall network throughput for CoLoRa under different SNRs. The total throughput is stable when SNR is higher than -5dB, and the performance slightly degrades for lower SNR scenarios.

9.4 Impact of Packet Offset

In this experiment, we evaluate the impact of the packet time offset between two collided packets. It has been shown CoLoRa leverages the time offset information to separate packets. Thus the packet time offset affects CoLoRa performance. We use two LoRa nodes to produce packet collisions with different packet time offsets. Then we decompose the two collided packets using both the basic design of CoLoRa and the advanced design of CoLoRa.

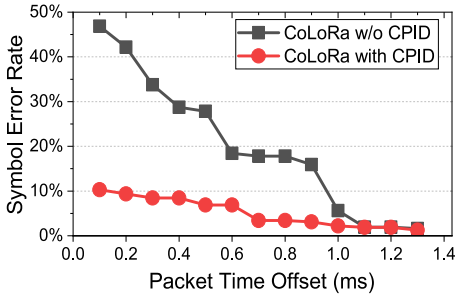

Fig. 18. Symbol Error Rates for decoding two-packet collisions with different packet time offsets.

Results. Fig. 18 shows the SER of basic CoLoRa and advanced CoLoRa with CPID in terms of the time offset between two collided packets. The SER of the basic CoLoRa is relatively high when two collided packets only have a tiny time offset. This is because a small offset leads to a small difference between the division ratio, which further makes it difficult to distinguish two packets. The performance of the advanced design is much better than the basic design. This is because we use an advanced peak ratio to disentangle collided symbols with a small time offset.

9.5 Addressing Collision in Real LoRa Networks

In this experiment, we verify the performance of CoLoRa in a real deployed LoRa network consisting of 20 LoRa end nodes in the campus.

Method. Fig. 13 shows the deployment environment, which has several multi-story buildings, trees, and hills. Each end node is equipped with a temperature sensor and a humidity sensor. The sensed data is transmitted to the gateway periodically in a regular interval (duty cycle of 0.1 in our experiment). The reference time of the duty cycle is 32.768s (i.e., the length of 1,000 LoRa chirps when SF = 12). The transmitted data is encoded with a spreading factor of 12 and a coding rate of 4/8, and the length of the packet is no more than 20 Bytes. We adopt LoRaWAN MAC based on pure ALOHA for all the end nodes on the MAC layer. Each node transmits to the gateway without consulting with others. There is a high probability that packets from different end nodes will collide. Hence our performance evaluation is carried out in terms of the network size, i.e., the number of activated nodes in the LoRa network.

Results. Fig. 19 shows the performance of three different LoRa receivers under the real deployed LoRa networks of size less than 20. Considering the LoRaWAN receiver without a collision resolution scheme, its SER increases rapidly when the network scales. The Network Throughput of the LoRaWAN receiver first grows up and then rapidly drops down as the size of the network increases. When the network size is small, the increase of concurrent nodes improves channel utilization. However, when the network scales, collisions frequently happen, which significantly degrades the performance of the LoRaWAN receiver. Overall, we consider the case for 10 nodes with 10% duty cycle ratio. There are enough data in the channel for transmission in such a case, and LoRaWAN+Oracle should achieve its highest throughput, i.e., 100%. We can see that LoRaWAN only achieves 7.2% of Oracle. The throughput of CoLoRa is about 14× that of LoRaWAN. Choir can separate overlapped packets from a collision based on the hardware offset. For the low duty cycle network with a small number of nodes, Choir also performs well as the number of concurrent transmissions is expected to be low (e.g., ≤ 4). This coincides with the results in Fig. 14a where CoLoRa and Choir have similar performance for low concurrency. However, the performance of Choir also degrades when the size of the network increases.

As shown in Fig. 19a, SER of Choir is around 30% when there are 20 concurrent nodes in the network. CoLoRa outperforms the other two approaches when the network scales, and it keeps a relatively low SER and PLR. The resulted network throughput grows nearly linearly as the size of the network increases, indicating CoLoRa can decompose most of the collided packets.

9.6 Performance in an Emulated Large Scale Network

In this experiment, we emulate a large-scale LoRa network for scalability evaluation.

Method. We use a software program to generate the signal of LoRa packets with white Gaussian noise. We emulate a LoRa network work with 1,000 nodes, each working at a certain duty cycle.

Results. We evaluate the performance of CoLoRa and the other three baselines in terms of a different duty cycle for end nodes. We change the duty cycle of each end node from 0.2% to 2%. Fig. 20a shows the SER performance. Fig. 20b shows the packet loss rate. We can see that LoRaWAN performs poorly in this scenario since most of the packets collide. LoRaWAN+oracle uses the optimal scheduler to avoid

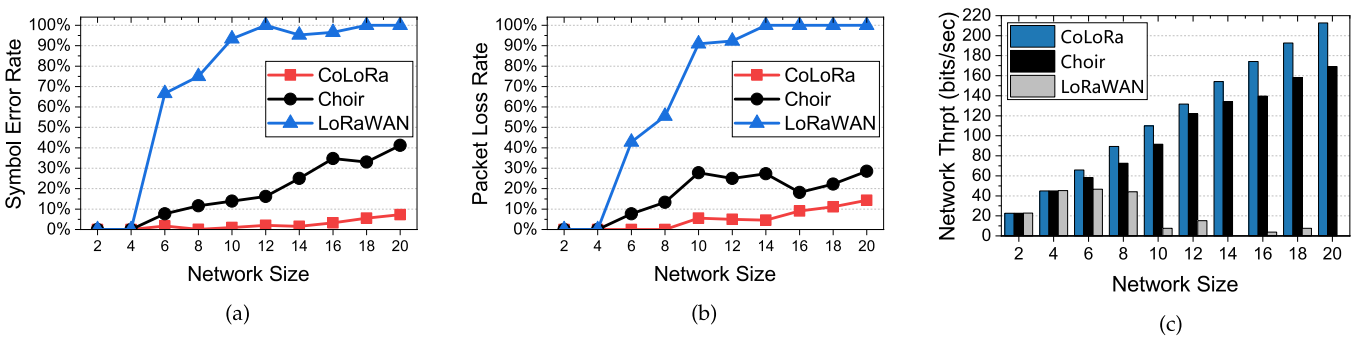

Fig. 19. Performance in a real deployed sensor network with different network sizes: (a) Symbol Error Rate. (b) Packet Loss Rate. (c) Network Throughput.

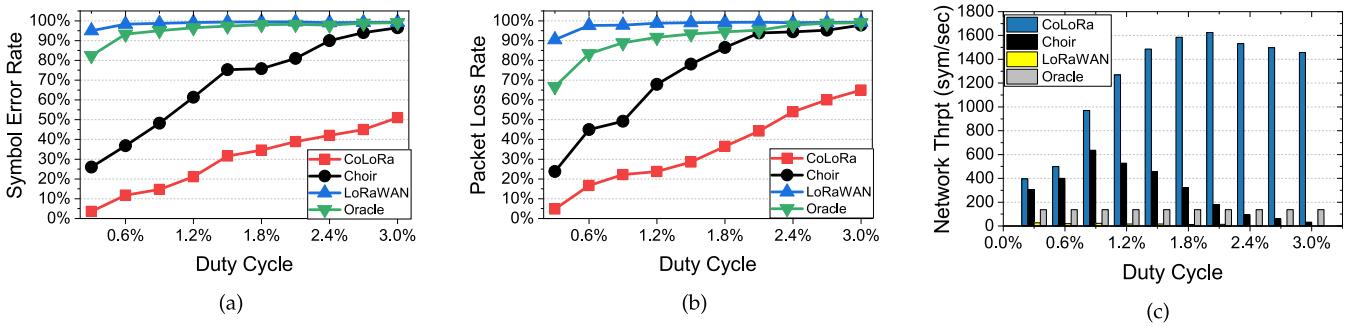

Fig. 20. Performance in emulated large scale LoRa network with different duty cycles: (a) Symbol Error Rate. (b) Packet Loss Rate. (c) Network Throughput.

collisions. However, it still loses many packets due to limited channel time. The network throughput of LoRaWAN with oracle converges to a certain bound when the channel is fully utilized. CoLoRa and Choir can deal with collisions, and the former performs best even in a large-scale network. As a result, the network throughput of CoLoRa, as shown in Fig. 20c, outperforms the remaining three approaches, especially when the network size increases. When the transmit duty cycle increases from 0.3% to 1.8%, the network throughput shows near-linear growth due to the benefit of multi-packet reception of CoLoRa. When the duty cycle is larger than 2.1%, the network throughput starts to get flat and slightly decreases. When the number of concurrent nodes is large, more packets become undecodable for CoLoRa. As such high concurrent transmissions are not frequent in real-world deployments, the simulation results still show the advantage of CoLoRa for large-scale networks.

10 Related Work

Collision Decoding in Traditional Wireless. Extensive works focus on collision resolution and parallel decoding in various wireless systems [11], [28], [29], [30], [31], [32], [33] (e.g., Wi-Fi, RFIDs and cellular networks). Some advocate Multiple-Input Multiple-Output (MIMO) to exploit spatial diversity across multiple paths [34], [35], [36]. MIMO-based approaches, which significantly improve the throughput, cannot be used in LoRa with a single antenna. Successive interference cancellation (SIC) based approaches resolves collisions by iteratively canceling interference from collided signals [37], [38], [39], [40]. These schemes work only when the colliding senders transmit under strict power control, which is usually used in cellular networks. ZigZag [8] combats inter-packet collisions in 802.11. It utilizes different collision-free parts of different collisions to decode the overlapped packets iteratively. In this way, ZigZag decodes an m-packet collision based on m repeated collisions. mZig [9] decompose m concurrent ZigBee packets from one collision directly. It starts with a collision-free chunk and then iteratively reconstructs and extracts each decoded symbol.

Concurrent Transmissions in LoRa. Recently, NetScatter [41] proposes a multi-packet reception strategy that enables hundreds of concurrent transmissions in LoRa backscatter networks [42]. The key innovation of NetScatter is a distributed coding mechanism where each node is assigned a chirp symbol with a different frequency shift and uses On-Off Keying (OOK) to modulate data. NetScatter requires that all the transmitters are synchronized, and hence it cannot work for unsynchronized transmission in existing LoRa communications. DeepSense [43] enables random access and coexistence for different LoRa configurations by exploring machine learning algorithms on-board. It identifies the presence of LoRa collisions using neural networks. However, in the emergence of collisions with the same LoRa configuration, DeepSense cannot recover any collided data bits. Choir [12] proposes a collision resolution method for LoRa. It depends on the fact that the hardware imperfection of a low-cost LoRa end node causes a frequency offset of the corresponding generated chirp signal. Choir utilizes this frequency offset to decompose collided packets of different end nodes. However, accurately extracting the tiny frequency offset is very difficult, especially for low SNR LoRa signals. Meanwhile, the frequency offsets of low-cost end nodes are changeable over time, impacting its practice performance. mLoRa [13] applies time-domain SIC to LoRa collisions. It starts with a collision-free chunk and then iteratively reconstructs and extracts each decoded symbol. FTrack [14] decodes multiple LoRa packets from a collision by calculating the instantaneous frequency continuity. Both mLoRa and FTrack have fundamental limitations in decoding low SNR LoRa signals. They focus more on the time domain signal analysis and interference cancellation and do not consider the encoding features of LoRa. NScale [44] decompose concurrent transmissions by leveraging subtle inter-packet time offsets for low SNR LoRa collisions. The key idea of NScale is to translate the subtle time offsets, which are vulnerable to noise, to robust frequency features, and further amplify the time offsets by non-stationary signal scaling. Recent literature proposes frequency-domain LoRa SIC for resolving LoRa collisions by iteratively decoding and canceling packets of different power levels [15], [45], [46], [47]. The key novelty is detecting and decoding the strongest signal information from the frequency domain with delicate signal detection and synchronization algorithms. However, the performance of LoRa SIC approaches is affected by variations in signal power levels. Given the long transmission duration of a LoRa packet, the power level varies for different parts of the same packet. LoRa SIC based on signal energy will incorrectly match symbols of different packets to the same transmitter, leading to decoding errors. CoLoRa leverages the time offsets between the LoRa packets and the reception windows for decoding collisions. Thus, it can be more robust when the power level of the received packet varies.

11 Conclusion

We present CoLoRa, a multi-packet reception approach in LoRa to address the practical collision problem of LPWAN deployments. CoLoRa utilizes, perhaps counter-intuitively, packet time offset to decompose multiple packets from a single collision. We propose several novel techniques to address practical challenges in CoLoRa design. We translate time offset, which is difficult to measure for symbols in collisions, to robust frequency features, i.e., peak ratios, for low SNR LoRa signal. We design a method to extract accurate peak ratios by iteratively canceling inter-packet interference. Finally, we address frequency shifts incurred by CFO and time offset to decode LoRa packets. CoLoRa is completely implemented in software at the gateway without any modifications to end nodes. The evaluation results show that CoLoRa improves the network throughput by **3.4**× compared with Choir and **14**× compared with LoRaWAN. We believe CoLoRa can be easily applied to today's LoRa networks with a very small overhead.

References

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, Second Quarter 2017.

[2] C. Li and Z. Cao, "LoRa networking techniques for large-scale and long-term IoT: A down-to-top survey," *ACM Comput. Surv.*, vol. 1, 2021, Art. no. 1.

[3] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 371–388, First Quarter 2020.

[4] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology," in *Proc. Int. Conf. ITS Telecommun.*, 2015, pp. 55–59.

[5] Y. Lin, W. Dong, Y. Gao, and T. Gu, "SateLoc: A virtual fingerprinting approach to outdoor LoRa localization using satellite images," *ACM Trans. Sensor Netw.*, vol. 17, no. 4, 2021, Art. no. 43.

[6] B. Ghena, J. Adkins, L. Shangguan, K. Jamieson, P. Levis, and P. Dutta, "Challenge: Unlicensed LPWANs are not yet the path to ubiquitous connectivity," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, Art. no. 43.

[7] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," *ACM Trans. Sensor Netw.*, vol. 15, no. 2, 2019, Art. no. 16.

[8] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 159–170.

[9] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in ZigBee linghe," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 552–565.

[10] O. Salehi-Abari , D. Vasisht, D. Katabi, and A. Chandrakasan, "Caraoke: An e-toll transponder network for smart cities," in *Proc. ACM Conf. Special Interest Group Data Commun.*, 2015, pp. 297–310.

[11] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy, "CrossZig: Combating cross-technology interference in low-power wireless networks," in *Proc. 15th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2016, pp. 1–12.

[12] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2017, pp. 309–321.

[13] X. Wang, L. Kong, L. He, and G. Chen, "mLoRa: A multi-packet reception protocol for LoRa communications," in *Proc. IEEE 27th Int. Conf. Netw. Protocols*, 2019, pp. 1–11.

[14] X. Xianjin, Z. Yuanqing, and G. Tao, "FTrack: Parallel decoding for LoRa transmissions," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, 2019, pp. 192–204.

[15] M. A. Temim, F. Guillaume, B. Laporte-Fauret , D. Dallet, B. Minger, and L. Fuche, "An enhanced receiver to decode superposed LoRa-like signals," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7419–7431, Aug. 2020.

[16] Accessed: Dec. 30, 2021. [Online]. Available: https://github.com/tongsh/colora-tmc

[17] Semtech, "LoRa modulation basics," 2015. [Online]. Available: https://www.semtech.com/uploads/documents/an1200.22.pdf

[18] LoRa alliance, "A technical overview of LoRa and lorawan," *White paper*, 2015.

[19] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN specification version: V1.1," 2017. Accessed: Dec. 30, 2021. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_-v1.1.pdf

[20] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?," in *Proc. 19th ACM Int. Conf. Model. Anal. Simul. Wireless Mobile Syst.*, 2016, pp. 59–67.

[21] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for LoRa," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–13.

[22] R. Sleeman and T. van Eck, "Robust automatic P-phase picking: An on-line implementation in the analysis of broadband seismogram recordings," *Phys. Earth Planetary Interiors*, vol. 113, pp. 265–275, 1999.

[23] C. Gu, L. Jiang, R. Tan, M. Li, and J. Huang, "Attack-aware synchronization-free data timestamping in LoRaWAN," *ACM Trans. Sen. Netw.*, Assoc. Comput. Machinery, New York, NY, USA, vol. 18, no. 1, p. 31, Oct. 2021, Art. no. 10. [Online]. Available: https://doi.org/10.1145/3474368

[24] USRP Ettus, "N210 datasheet." Accessed: Dec. 30, 2021. [Online]. Available: https://www.ettus.com/wp-content/uploads/2019/01/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf

[25] UBX Ettus, "Ubx 10–6000 MHz rx/tx." Accessed: Dec. 30, 2021. [Online]. Available: https://www.ettus.com/all-products/ubx160/

[26] Semtch, "Sx1278 datasheet." Accessed: Dec. 30, 2021. [Online]. Available: https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf

[27] G. FSF, "GNU radio - GNU FSF project." Accessed: Dec. 30, 2021. [Online]. Available: http://gnu.ist.utl.pt/software/gnuradio/gnuradio.html

[28] J. Ou, M. Li, and Y. Zheng, "Come and be served: Parallel decoding for COTS RFID tags," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 500–511.

[29] Y. Zheng and M. Li, "Read bulk data from computational RFIDs," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 495–503.

[30] L. Yang, J. Han, Y. Qi, C. Wang, and Y. Liu, "Revisiting tag collision problem in RFID systems," in *Proc. IEEE 39th Int. Conf. Parallel Process.*, 2010, pp. 178–187.

[31] Y. Du, E. Aryafar, J. Camp, and M. Chiang, "iBeam: Intelligent client-side multi-user beamforming in wireless networks," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 817–825.

[32] E. Zihan, K. W. Choi, and D. I. Kim, "Distributed random access scheme for collision avoidance in cellular device-to-device communication," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 3571–3585, Jul. 2015.

[33] J. Li, Y. Zhang, M. Shi, Q. Liu, and Y. Chen, "Collision avoidance strategy supported by LTE-V-based vehicle automation and communication systems for car following," *Tsinghua Sci. Technol.*, vol. 25, no. 1, pp. 127–139, Feb. 2020.

[34] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous MIMO networks," in *Proc. ACM SIGCOMM Conf.*, 2011, pp. 146–157.

[35] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's wireless LANs," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, 2013, pp. 387–398.

[36] Y. Xie, Y. Zhang, J. C. Liando, and M. Li, "SWAN: Stitched Wi-Fi antennas," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 51–66.

[37] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 339–350.

[38] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 5, pp. 796–807, Jan. 1994.

[39] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 159–170.

[40] B. Dezfouli, M. Radi, K. Whitehouse, S. Abd Razak , and H.-P. Tan, "CAMA: Efficient modeling of the capture effect for low-power wireless networks," *ACM Trans. Sensor Netw.*, vol. 11, no. 1, 2014, Art. no. 20.
[41] M. Hessar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *Proc. 16th USENIX Conf. Netw. Syst. Des. Implementation*, 2019, pp. 271–283.
[42] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa backscatter: Enabling the vision of ubiquitous connectivity," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, 2017, Art. no. 105.
[43] J. Chan, A. Wang, A. Krishnamurthy, and S. Gollakota, "DeepSense: Enabling carrier sense in low-power wide area networks using deep learning," 2019, *arXiv:1904.10607*.
[44] T. Shuai, W. Jiliang, and L. Yunhao, "Combating packet collisions using non-stationary signal scaling in LPWANs," in *Proc. 18th Int. Conf. Mobile Syst. Appl. Serv.*, 2020, pp. 234–246.
[45] R. N. El, G. Alexandre, and K. Megumi, "Decoding superposed LoRa signals," in *Proc. IEEE 43rd Conf. Local Comput. Netw.*, 2018, pp. 184–190.
[46] N. Umber, C. Laurent, and B. Ahcene, "LoRa-like CSS-based PHY layer, capture effect and serial interference cancellation," in *Proc. 24th Eur. Wireless Conf.*, 2018, pp. 1–6.
[47] L.-F. Baptiste, T. M. A. Ben, F. Guillaume, D. Dominique, M. Bryce, and F. Loic, "An enhanced LoRa-like receiver for the simultaneous reception of two interfering signals," in *Proc. IEEE 30th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2018, pp. 1–6.

Shuai Tong (Student member, IEEE) received the BE degree from the College of Computer Science, Nankai University, China, in 2019. He is currently working toward the PhD degree in the School of Software, Tsinghua University, China. His research interests include low-power wide-area networks, Internet of Things, and mobile computing.

Zhenqiang Xu (Student member, IEEE) received the BE degree from the College of Computer Science and Technology, University of Science and Technology of Chnia, China, in 2017. He is currently working toward the PhD degree in the School of Software, Tsinghua University, China. His research interests include low-power wide-area networks, wireless security, and bioinformatics.

Jiliang Wang (Senior member, IEEE) received the BE degree in computer science and technology from the University of Science and Technology of China, China, and the PhD degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong. He is currently an associate professor with the School of Software, Tsinghua University, China. His research interests include wireless and sensor networks, Internet of Things, and mobile computing.