FlipLoRa: Resolving Collisions with Up-Down Quasi-Orthogonality

Zhenqiang Xu, Shuai Tong, Pengjin Xie and Jiliang Wang
School of Software and BNRist, Tsinghua University, P. R. China
{xu-zq17, tl19, xpj15}@mails.tsinghua.edu.cn, jiliangwang@tsinghua.edu.cn

***Abstract*—LoRa is recently a rising star in Low Power Wide Area Network (LPWAN) family to provide low power and long range communication for large number of devices in Internet of Things. LoRa is based on Chirp Spread Spectrum (CSS) and uses chirp frequency shift to encode data. It has been shown that collision significantly degrades LoRa performance in practice. We propose FlipLoRa, a new mechanism to disentangle LoRa collisions, which allows concurrent transmission of multiple packets. The key idea of FlipLoRa is to utilize the quasi-orthogonality between upchirp and downchirp. FlipLoRa encodes packets with interleaved upchirps and downchirps instead of only using upchirps as in LoRa. We then propose a novel method to disentangle chirps and decode multiple collided packets. To evaluate the performance, we formally prove the quasi-orthogonality and analyze its applicable conditions. We validate the performance improvement by theoretical analysis. Further, we implement FlipLoRa on software-defined radio and extensively evaluate its performance for real LoRa networks. The evaluation results show that FlipLoRa can improve the throughput by 3.84x over LoRa physical layer.**

***Index Terms*—LPWAN, LoRa, CSS, collision decoding**

I. Introduction

Recently, Low Power Wide Area Network (LPWAN) offers a promising solution for Internet of Things (IoT) devices that need long communication range (placed 1∼10 km away from gateway) and low power consumption (work for 10 years without replacing battery). It has attracted much attention from both industry and academia. Being high energy efficient for embedded devices, LPWAN brings more opportunities to IoT applications which require low power and long range communication (e.g. smart agriculture, smart city, large area wild animal monitoring, etc.). Many LPWAN technologies are raised to suit such situations, such as LoRa, SigFox, NB-IoT and RPMA [1], [2], [3], [4]. LoRa, as a promising competitor in LPWAN family, is one of the representative LPWAN technologies and has attracted many research efforts. LoRa's physical layer adopts CSS to achieve the goal of long range communication. Chirp is a kind of signal widely used in many areas, like radar, acoustics, satellite communications [5], [6], [7]. LoRa uses linear chirp whose frequency changes linearly with time. The energy of LoRa chirp could be accumulated through an operation called dechirp [8], which overcomes the attenuation of long range communication, with a tradeoff of relatively low bit rate, normally $0.3 \sim 50$ kbps.

LoRa is proposed to connect a large number of low data rate IoT devices. However, practical LoRa deployments suffer from performance degradation due to collisions. Moreover, current standard MAC layer protocol LoRaWAN adopts an Aloha-like mechanism which aggravates collisions. This further limits the capacity and scalability of LoRa network. To address the problem, many efforts have been made. Choir [9] proposes a physical layer collision separation mechanism by leveraging hardware imperfection. The minor frequency offsets are used to distinguish different packets and then improve throughput. FTrack [10] exploits the time misalignment for the overlapped packets to separate collisions. Applying these approaches in practice, however, is still challenging as it is difficult to extract the tiny features from low SNR LoRa signals.

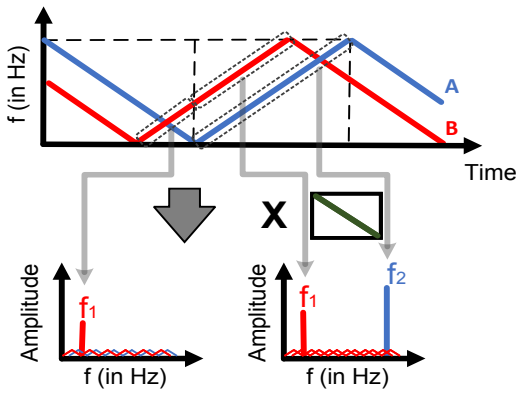


Fig. 1. FlipLoRa encoding and decoding mechanism

We disentangle LoRa collisions from another novel perspective. Our key observation is the quasi-orthogonality between upchirp and downchirp. An upchirp has a linearly increasing frequency while a downchirp has a decreasing frequency. In LoRa decoding for an upchirp symbol, after dechirping by multiplying the upchirp symbol with a basis downchirp (frequency from $\frac{B}{2}$ to $-\frac{B}{2}$, B: Bandwidth), the energy of the upchirp symbol is accumulated in frequency domain. Due to quasi-orthogonality, while multiplying the upchirp symbol with a basis upchirp (frequency from $-\frac{B}{2}$ to $\frac{B}{2}$), the energy of the given upchirp is spread to the entire spectrum.

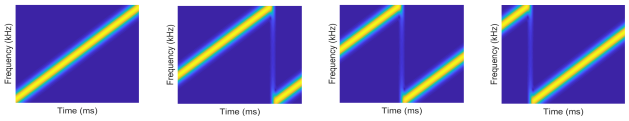
We propose FlipLoRa, a new encoding and decoding mechanism with interleaved up-down chirps to fully utilize quasi-orthogonality in collision separation. Figure 1 depicts a simplified example of FlipLoRa with two collided packets A and B. To decode those two packets, we first align the window with

978-1-7281-6630-8/20/$31.00 2020 © IEEE

one of the packets (e.g., A). Then we perform dechirping by multiplying the collided signal in each window with the basis downchirp. After FFT, in the first window, there should be only one peak transformed from a part of chirp from packet B. The energy of all other chirps is averaged in the entire spectrum due to quasi-orthogonality. Similarly, in the second window, there are two peaks after dechirping. The first peak at f_1 comes from the upchirp part of packet B. The second peak at f_2 comes from the chirp of packet A. Due to quasi-orthogonality, we can see that packet B leads to a peak in each window both at the same frequency f_1, while packet A only leads to one peak at f_2 in the second window. Therefore, the aligned packet has only a single peak in the second window. Combining the peaks in both windows, we can decode the chirp for packet A in the second window based on peak f_2. Similarly, we can also decode the chirps of packet A in other windows by combining two consecutive windows. After that, we can cancel chirps of packet A from the collided signal and then decode the remaining packets.

Challenges: The practical design of FlipLoRa to disentangle packets faces challenges. (1) The above decoding method requires accurate window alignment for low SNR LoRa signal. Meanwhile, carrier frequency offset (CFO), i.e., the frequency offset among different devices, also impacts the decoding accuracy. Moreover, the CFO measurement and window alignment are often coupled, both leading to peaks offset in the frequency domain. (2) Chirp cancellation is required to reduce cross chirp interference and to decode packets iteratively. The traditional cancellation method is based on iterative search, which has high computational complexity and a low cancellation rate. For low SNR LoRa decoding, the residual energy after cancellation may be amplified to considerable packet decoding errors. (3) The quasi-orthogonality is related to transmission parameters such as Spreading Factor (SF). Low transmission SF may lead to a relatively low quasi-orthogonality. Thus simply applying interleaved chirps does not perform well under low SF.

To address the first challenge, we achieve accurate window alignment and CFO measurement using combination of up-down chirp pair preceding each packet. Our method decouples the window alignment and CFO measurement. When the decoding window is not aligned with a packet, the basis upchirp and downchirp would lead to two different peaks in the frequency domain after dechirping. If and only if the decoding window is aligned with the packet, the basis upchirp and downchirp lead to peaks in the same position. Then the CFO can be measured by the peak offset when the window is aligned. To address the second challenge, we propose a *frequency domain construction* method for chirp symbol cancellation. Considering the low SNR signal, our main idea is to accurately reconstruct the chirp signal based on the frequency domain information and then subtract the signal from the time domain. Our method costs about constant computation time and can reduce the energy of signal by up to 30dB. To address the third challenge, we design a multi-dimensionality based decoding method by combining features from different aspects such as magnitude and phase together with quasi-orthogonality.



(a) (b) (c) (d)

Fig. 2. Chirp symbols of LoRa modulation with SF=2. (a) Symbol for '00'. (b) Symbol for '01'. (c) Symbol for '10'. (d) Symbol for '11'.

The performance of FlipLoRa is validated and evaluated by theoretical analysis, simulation and real experiments. We implement FlipLoRa on software-defined radio (SDR) boards, a single antenna USRP N210 gateway and multiple HackRF One nodes. Our evaluation results show that FlipLoRa achieves a throughput gain of 3.84x over original LoRa. Furthermore, the design of FlipLoRa can be easily extended to existing hardware since LoRa chip supports sending both upchirp and downchirp.

Contributions: Our contributions are three-fold: (1) We formally analyze the quasi-orthogonality between upchirp and downchirp. To the best of our knowledge, the quasi-orthogonality between upchirp and downchirp has not yet been thoroughly discussed in LoRa. (2) We propose two novel methods for CFO elimination and symbol cancellation. They are ubiquitous methods and could be used in other CSS based work. (3) We propose a new coding mechanism FlipLoRa to disentangle collisions. We validate the performance of FlipLoRa by theoretical analysis, simulation and SDR implementation. Our experiments show that FlipLoRa improves throughput by 3.84x.

II. Background

LoRa uses Chirp Spread Spectrum (CSS) for modulation and demodulation. A basic baseband LoRa symbol with unit power can be expressed as

$$chirp(t) = \exp\left(j2\pi\frac{B}{2T}t(t-T)\right) \tag{1}$$

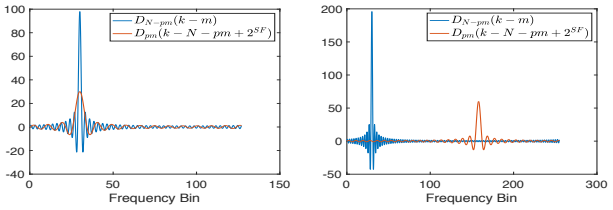
where B is the bandwidth of chirp and T is the duration of one single chirp. Due to the anti-interference feature of CSS, LoRa achieves long range while sacrificing its data rate. An important parameter of LoRa to control data rate is Spreading Factor(SF), whose definition is

$$2^{SF} = B \cdot T. \tag{2}$$

SF determines the frequency changing rate of a chirp. For a larger SF, the time duration for a chirp is longer given the bandwidth B.

Note that $chirp(t)$ is an upchirp whose frequency increases linearly with time while the complex conjugate of $chirp(t)$ is a downchirp whose frequency decreases linearly with time.

The strategy of LoRa to encode data is shifting the basis chirp in time-frequency domain. For instance, Figure 2 depicts the LoRa encoding scheme when SF=2. Figure 2(a)



(a) (b)

Fig. 3. Dirichlet kernel. (a) $p = 1, SF = 7, N = 128, m = 30$. (b) $p = 2, SF = 7, N = 256, m = 30$.

is an unshifted basis upchirp and encodes data bits "00". Figure 2(b)(c)(d) shifts the start frequency of the symbol to $-\frac{B}{4}, 0, \frac{B}{4}$ and encodes "01", "10", "11" respectively.

To understand the decoding operation, we need to discretize Equation (1). Let sampling frequency $fs = B \cdot p$, where p is an integer. Let $t = \frac{n}{N}T, n = 0, 1, 2 \cdots, N-1$ where $N = 2^{SF} \cdot p$. Then a chirp can be represented as

$$\omega_0(n) = \exp\left(j2\pi \frac{n(n-N)}{2Np}\right). \tag{3}$$

Denote $\omega_m(n), m \in \{0, 1, \cdots, 2^{SF}-1\}$ as the m^{th} cyclic-shifted form of basis LoRa chirp. Then we have

$$\omega_m(n) = \exp\left(j2\pi \frac{n'(n'-N)}{2Np}\right) \omega_0^*(pm) \tag{4}$$

where $n' = (n + pm) \mod N$, $*$ denotes complex conjugate. The constant $\omega_0^*(pm)$ makes $\angle\omega_m(0) = 0$ and thus ensures the phase continuation between adjacent chirp symbols.

The task for receiver is to extract the decoded information, i.e. m. The decoder first multiplies the symbol with a basis downchirp with starting frequency at $\frac{B}{2}$.

$$\overline{\omega_m}(n) = \omega_m(n) \cdot \omega_0^*(n). \tag{5}$$

Notice that m, n are integers and $\exp(j2\pi \cdot integers) = 1$. $\overline{\omega_m}(n)$ could be expressed using rectangular window function

$$\overline{\omega_m}(n) = s_1(n)r_1(n) + s_2(n)r_2(n), \tag{6}$$

where $s_1(n) = \exp(\frac{j2\pi mn}{N})$, $s_2(n) = s_1(n)\exp(\frac{-j2\pi n}{p})$, $r_1(n) = rect_{0,N-pm-1}(n)$, $r_2(n) = rect_{N-pm,N-1}(n)$. The definition of rectangular function $rect$ is

$$rect_{a,b}(n) = \begin{cases} 1, & a \le n \le b \\ 0, & otherwise. \end{cases} \tag{7}$$

Then we apply Discrete Fourier Transform (DFT) to $\overline{\omega_m}(n)$, the kth output is

$$\begin{aligned} \mathcal{F}(\overline{\omega_m})_k &= \mathcal{F}(s_1 \cdot r_1)_k + \mathcal{F}(s_2 \cdot r_2)_k \\ &= d_1 D_{N-pm}(k-m) + d_2 D_{pm}(k-N-m+2^{SF}) \end{aligned} \tag{8}$$

where $D_L(k) = \frac{\sin(\pi kL/N)}{\sin(\pi k/N)}$ is Dirichlet kernel, $d_1 = \exp(j\pi(N-pm-1)(m-k)/N)$, $d_2 = d_1 \exp(j\pi(m-k-(N-2^{SF})(pm+1)/N))$. Equation (8) is an important result and we will use it in Section IV-D for symbol cancellation.

We see from Equation (8) that the transformation of a LoRa chirp produces two FFT peaks (except basis chirp). Each represents a chirp segment in a LoRa symbol. The amplitude of each segment in frequency domain is a Dirichlet kernel and the phase is linear. An example image of Dirichlet kernels with parameter setting $SF = 7, m = 30$ is shown in Figure 3. The main peak position of $D_{N-pm}(k-m)$ and $D_{pm}(k-N-m+2^{SF})$ is m and $N+pm-2^{SF}$. If $p = 1$, we have $m = N+pm-2^{SF}$. Then the two Dirichlet kernels would share the same main peak position as shown in Figure 3(a). Otherwise, they are separated like Figure 3(b). The decoder could extract the encoded data m by searching the peak index.

III. Up-Down quasi-orthogonality

Section II describes the decoding process of an upchirp. Relatively, the decoding rule for downchirp is a conjugate replication of upchirp, i.e., the incoming symbol is multiplied with ω_0 instead of ω_0^*. We define two operations to represent the decoding process for the two kinds of chirp:

$$\begin{aligned} \mathcal{U}(\omega) &= \mathcal{F}(\omega \cdot \omega_0^*) \\ \mathcal{D}(\omega) &= \mathcal{F}(\omega \cdot \omega_0). \end{aligned} \tag{9}$$

$\mathcal{U}$ represents upchirp decoding and $\mathcal{D}$ represents downchirp decoding. For an upchirp, applying $\mathcal{U}$ accumulates its energy and leads to a peak in frequency domain. However, for a downchirp, the energy after $\mathcal{U}$ is spread over the whole spectrum. As a result, the decoding result of an upchirp would not change with the existence of downchirps. Next, we theoretically analyze this feature, i.e., quasi-orthogonality between upchirp and downchirp.

The quasi-orthogonality means $\mathcal{U}(\omega_{up} + \omega_{down})$ gives the same decoding result of $\mathcal{U}(\omega_{up})$ [1]. Without loss of generality, we only prove the situation where $\omega_{up} = \omega_0, \omega_{down} = \omega_0^*$ and $p = 1$.

If ω_0^* is sent but we try to decode it with $\mathcal{U}$, the output is [2]

$$\mathcal{U}(\omega_0^*)_k = \begin{cases} 2^{\frac{SF+1}{2}} \exp\left(\frac{j\pi}{4} - \frac{j\pi k^2}{2^{SF+1}}\right), & 2 \mid k \\ 0, & 2 \nmid k. \end{cases} \tag{10}$$

According to Equation (8), $\mathcal{U}(\omega_0)_k = d_1 D_N(k)$. Because of the linearity of Fourier Transform, $\mathcal{U}(\omega_0 + \omega_0^*) = \mathcal{U}(\omega_0) + \mathcal{U}(\omega_0^*)$. Thus we have

$$\arg\max_k |\mathcal{U}(\omega_0 + \omega_0^*)_k| = \arg\max_k |\mathcal{U}(\omega_0)_k| = 0. \tag{11}$$

Therefore, $\mathcal{U}(\omega_{up} + \omega_{down})$ has the same decoding result "0" with $\mathcal{U}(\omega_{up})$ and the quasi-orthogonality is proved.

IV. FlipLoRa Design

A. FlipLoRa In A Nutshell

In this section, we provide an overview of FlipLoRa's main idea and show how to utilize the quasi-orthogonality in LoRa for collision separation. Figure 4 gives an example of the

[1] Here we suppose upchirp and downchirp have the same power. The quasi-orthogonality does not hold when the energy gap is too large.

[2] We leave the calculation details in Appendix A.

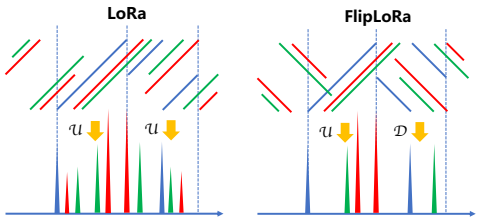


Fig. 4. (a) LoRa collision. (b) FlipLoRa collision.

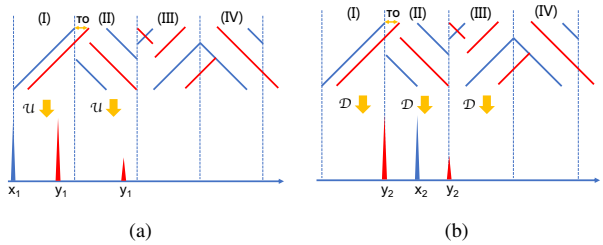


Fig. 5. (a) FlipLoRa decodes blue upchirp in collision. (b) FlipLoRa decodes blue downchirp in collision.

difference between LoRa and FlipLoRa collisions. The bottom part of Figure 4 indicates the output of upchirp decoding or downchirp decoding. The main difference between LoRa and FlipLoRa is that FlipLoRa uses interleaved chirps (upchirp followed by downchirp, and vice versa) to encode data while LoRa only uses upchirps (or downchirps). Intuitively, this design achieves the same coding rate with the original LoRa. We show that such a coding design enables better collision separation ability. To understand the key idea, it is essential to know the problems in current LoRa decoding with collisions.

According to the decoding process discussed in Section II, LoRa searches the highest peak in each FFT window and takes it as the decoding result. In the presence of collisions, there will be more peaks in a single window. For example, in Figure 4(a), three packets collide. The decoding window is aligned with the blue one. When using operation $\mathcal{U}$ to decode the blue packet, we find that there are too many peaks in the FFT result. Therefore, it is difficult to divide peaks into groups and map peaks to different packets.

When we switch to Figure 4(b), we find that with FlipLoRa, nearly one-third of the peaks disappeared due to quasi-orthogonality. To extract the correct symbol, FlipLoRa applies operation $\mathcal{U}$ (or $\mathcal{D}$) on two consecutive windows. As shown in Figure 5(a), the first red upchirp spans over window (I) and (II). The segment in window (I) after operation $\mathcal{U}$ transforms to a red peak y_1 (x_i and y_i indicate the FFT bin index). The segment in window (II) after operation $\mathcal{U}$ changes to a relatively lower red peak y_1. We can see that two segments of an upchirp transform to two peaks at the same position in the frequency domain after operation $\mathcal{U}$. Meanwhile, our target blue chirp is aligned with a decoding window and it does not generate a replica peak in window (II). From window (I), we have a peak set $peaks_1 = \{x_1, y_1\}$. From window (II), we have a peak set $peaks_2 = \{y_1\}$. $peaks_1 - peaks_2 = \{x_1\}$ gives the final decoding result. Similarly, by replacing $\mathcal{D}$ with $\mathcal{U}$, we could decode downchirp encoded data as shown in Figure 5(b).

In the rest of this section, we focus on how to leverage the basic idea of FlipLoRa to develop a practical collision decoding framework:

- Window alignment: FlipLoRa decoding method requires the current decoding window to align with a packet. The challenge for alignment comes from CFO, which cannot be ignored in real environments. Under collision, it becomes much more difficult for accurate window alignment because different packets have different CFO. We propose a CFO elimination algorithm utilizing up-down pair and optimize it for collision situation.
- Packet tracking: Once a window corresponding to a packet is aligned, we need to track all the symbols belonging to the packet. Our main method is based on quasi-orthogonality. We find that tracking error can be reduced with the assistance of multi-dimensional information like signal magnitude and phase. We combine quasi-orthogonality and multi-dimensional information to accomplish the packet tracking algorithm.
- Symbol cancellation: Near-Far problem exists in FlipLoRa communication, i.e., strong signals would mask weak signals. Our solution is extracting and canceling symbols while decoding.

Figure 6 shows the block diagram of FlipLoRa. It is composed of three main blocks: window alignment, packet tracking and symbol cancellation. FlipLoRa decoder first consumes IQ sequence of the signal and tries to find the preamble. The preamble detection is done by calculating the correlation of the pre-constructed ideal preamble and the real received signal. Once we detect a correlation value larger than a threshold, the decoder switches to window alignment mode. After the packet is aligned with the decoding window, the decoder starts to track symbols belonging to the same packet with quasi-orthogonality and multi-dimensional information. During packet tracking, decoded symbols are canceled from the received signal. If a stronger signal is detected, we push the current packet into a LIFO called Unfinished Packet List and start to locate the preamble of the stronger signal.

B. Window Alignment

An important feature of linear chirp is that the time offset is equivalent to frequency shift. Therefore, a misaligned window causes the FFT peak shift and thus leads to wrong data. If the start time t of a packet is not accurately measured, we obtain a measurement of $t - TO$ where TO is the Time Offset. Then all our data symbols are shifted due to TO, i.e., ω_m becomes $\omega_{m-round(\frac{TO}{T})}$. For a LoRa packet, time offset for data symbol could be calibrated by referring the preamble peaks since preamble experiences the same offset and is already known. However, the calibration fails when collisions occur because we could not distinguish peaks

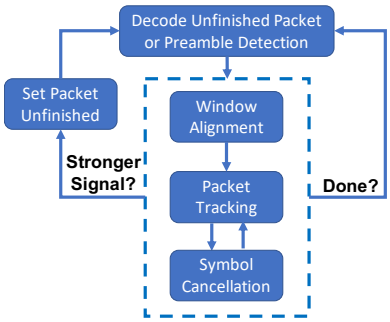


Fig. 6. FlipLoRa block diagram

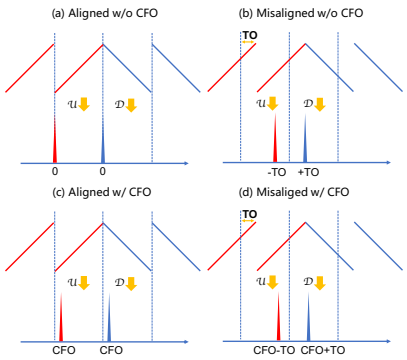


Fig. 7. Up-Down pair alignment and CFO elimination

from different packets. When CFO is considered, shifted data symbols become $\omega_{m-round(\frac{TO}{T})+round(\frac{CFO}{B})}$. It is difficult to distinguish the frequency shift caused by time offset together with CFO.

We leverage the up-down chirp pair for CFO elimination and window alignment. LoRa physical layer packet naturally provides such a pair. Preamble contains basis upchirps and SFD (Start Frame Delimiter) contains basis downchirps. Figure 7 shows the principle of up-down alignment. If TO = 0 and CFO = 0 (Figure 7(a)), both up peak and down peak locate at the first FFT bin when the window is aligned with the up-down chirp pair. If TO $\neq$ 0 and CFO = 0 (Figure 7(b)), the peak of upchirp shifts to left (cyclic shift) and the peak of downchirp has an equal-sized right shift. If TO = 0 and CFO $\neq$ 0 (Figure 7(c)), up peak and down peak are shifted in the same direction. If TO $\neq$ 0 and CFO $\neq$ 0, the combination of TO and CFO results in Figure 7(d). We derive that the packet is accurately aligned if and only if the up peak and the down peak are both located at the same FFT bin.

The above method solves how to align a single packet. However, it is challenging to conduct window alignment under collisions since pairing upchirp and downchirp is a problem. Our strategy is first shifting the preamble peak to position 0. Assume $|\text{CFO}| < \text{CFO}_{limit}$, the bin shift caused by CFO has an upper bound

$$bin_{shift} < \delta = \frac{\text{CFO}_{limit} \cdot 2^{SF}}{B} \tag{12}$$

According to the alignment method, the occurrence of the related downchirp peak will be located at $[0, 2\delta] \cup [2^{SF} - 2\delta + 1, 2^{SF}]$. As δ is usually much less than B, there is a high probability that only one downchirp peak in such an interval satisfies our requirements. Using this up-down pair eliminates the CFO of the current decoding packet.

C. Packet Tracking

When a packet is aligned, we begin to track its data symbols. The way to extract them is using quasi-orthogonality as discussed in Section IV-A. However, the set subtraction would not always succeed due to interference and noise. Peaks may be masked by imperfect symbol cancellation (will be discussed in Section IV-D). Therefore, we propose the concept of peak distance to improve the accuracy of quasi-orthogonality decoding. We denote the peak set after operation $\mathcal{U}$ (or $\mathcal{D}$) in current decoding window as $peaks_C$. The peak set after the same operation in the left (right) window is called $peaks_L$ ($peaks_R$). The target peak bin index x in $peaks_C$ would be

$$x = \arg\max_{x \in peaks_C} \min(dist(x, peaks_L), dist(x, peaks_R)) \tag{13}$$

where

$$dist(x, peaks) = \begin{cases} \min\limits_{x' \in peaks} |x - x'|, & peaks \neq \varnothing \\ +\infty, & peaks = \varnothing \end{cases} \tag{14}$$

Take Figure 5(b) as an example, $peaks_C$ is the peak set $\{x_2, y_2\}$ in window (II) and $peaks_R$ is the peak set $\{y_2\}$ in window (III). Since there is no peak in window (I), $peaks_L = \varnothing$. According to Equation (13) and (14), we have $x = x_2$, which means x_2 is what we require in current window. In real deployments, the peak index would be distorted by interference and noise. Our goal is to find the best-fit peak index. Suppose $peaks_R$ has a small distortion and becomes $\{y_2 + 1\}$ (the peak index is not stable), the above calculation still gives $x = x_2$ which strengthens the reliability of FlipLoRa.

Besides the quasi-orthogonality feature, multi-dimensional information is also exploited for packet tracking. The wireless channel is considered to be static during a short packet transmission time, and thus the received signal power is stable. In FFT results, peak height represents power level. Therefore we can track peaks of the packet based on their power level, i.e., peak height.

At the implementation level, to get an accurate peak height, a technology called zero-padding is needed. Before doing FFT, we append $(r-1)N$ zeros to the signal where r is an integer representing zero-padding ratio. As shown in Figure 8(b), zero-padding increases the frequency resolution. If zero-padding is not adopted, a large peak height fluctuation would occur. Figure 8(a) shows the non-zero-padding condition. The red dots are the output of FFT and their maximum modulus does not reflect the actual Dirichlet magnitude.

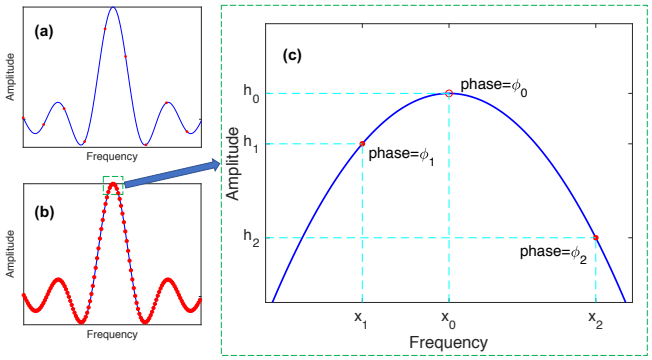


Fig. 8. (a) Frequency bin peak without zero-padding. (b) Frequency bin peak with zero-padding (c) Use peak side points to reconstruct peak

D. Symbol Cancellation

Every time we extract a symbol in an aligned window, we cancel it from the received signal S_r. On one hand, it reduces the number of peaks and enables the decoding for other packets. On the other hand, it solves the near-far problem because high power signals are canceled iteratively. Suppose we want to cancel the symbol ω_m, a traditional way is to construct a similar symbol as $S_c = Ae^{j(2\pi\Delta f t+\varphi)}\omega_m$ where A is amplitude, Δf is the estimation of CFO and φ is the initial phase. We can estimate parameters $A^*, \Delta f^*, \varphi^*$ through solving the optimization problem below

$$(A^*, \Delta f^*, \varphi^*) = \underset{A,\Delta f,\varphi}{\arg\min}\ R(A, \Delta f, \varphi), \tag{15}$$

where $R(A, \Delta f, \varphi) = ||S_r - S_c||$ is residual energy. However, such method has three drawbacks:

- It is a search-based method and relies on multiple iterations. The computation complexity is high.
- The result may be a local minimum and the symbol is not completely canceled.
- The calculation of residual energy does not consider the high sensitivity of CSS. Since LoRa works under low SNR, the variation of time domain energy may be small even if the symbol is correctly canceled.

To overcome these drawbacks, we propose a novel *frequency domain construction* method with very low complexity. The method can cancel the signal by up to 30dB and reduces the energy of the signal to noise level. The goal of symbol cancellation is to minimize the residual chirp influence after operation $\mathcal{U}/\mathcal{D}$, i.e., the residual peak height. We can directly construct a peak in frequency domain and then subtract the peak from real signal FFT results to check the cancellation effect. If the cancellation suits requirements, we can apply inverse fourier transform to obtain a time domain constructed signal. According to Equation (8), a LoRa symbol in frequency domain is one or two Dirichlet kernels with linearly changing phases. The peak heights of the two Dirichlet kernels vary with encoded data m. We choose the higher one, say, $\mathcal{P}$. The peak $\mathcal{P}$ with apex index x_0 has form

$$\mathcal{P} = \phi(k)D(k) = \phi(k) \cdot A \cdot \frac{\sin\left(\pi(k - x_0)L/N\right)}{\sin\left(\pi(k - x_0)/N\right)} \tag{16}$$

where $\phi(k)$ is a linear phase sequence and Dirichlet kernel $D(k)$ represents the peak amplitude. Figure 8(c) shows the local view of a peak. In real experiments, the apex (x_0, h_0) of Dirichlet kernel does not accurately locate at an FFT bin as shown in Figure 8(c). Solid red dot (x_1, h_1) and (x_2, h_2) are real frequency sample points while the actual apex, hollow red dot (x_0, h_0) is not. According to Equation (16), we have

$$A \cdot \frac{\sin\left(\pi(x_1 - x_0)L/N\right)}{\sin\left(\pi(x_1 - x_0)/N\right)} = h_1 \tag{17}$$

$$A \cdot \frac{\sin\left(\pi(x_2 - x_0)L/N\right)}{\sin\left(\pi(x_2 - x_0)/N\right)} = h_2 \tag{18}$$

Since N is large, $\sin\left(\pi(x_i - x_0)/N\right) \approx \pi(x_i - x_0)/N$. Denote $\Delta x = x_0 - x_1 \in [0, \frac{1}{r}]$, after applying approximation and triangular expansion to the ratio of Equation (17) and Equation (18), we have

$$\Delta x\left(\sin\frac{\pi L}{rN}\cot\pi\Delta x + \frac{h_2}{h_1} - \cos\frac{\pi L}{rN}\right) = \frac{h_2}{h_1} \cdot \frac{L}{rN} \tag{19}$$

The above equation can be solved by numerical methods with a small overhead. After we derive Δx, the amplitude is calculated as $A = \frac{h_1\pi\Delta x}{N\sin(\pi\Delta x L/N)}$. Till now, the Dirichlet kernel is successfully constructed as

$$D(k) = \frac{h_1\pi\Delta x}{N\sin\left(\pi\Delta x L/N\right)} \cdot \frac{\sin\left(\pi(k - x_1 - \Delta x)L/N\right)}{\sin\left(\pi(k - x_1 - \Delta x)/N\right)}. \tag{20}$$

Then we need to estimate $\varphi(k)$. According to Equation (8), the FFT bin phase of an ideal LoRa chirp is linear. Thus $\varphi(k)$ could be derived by interpolation with the phase of bin x_1 and x_2. Hence one constructed chirp can be expressed as $S_{c1} = IFFT(\phi \cdot D)/\omega_0^*$. Similarly, we get another constructed peak S_{c2}. The final constructed chirp symbol is

$$S_c = S_{c1} + S_{c2} \tag{21}$$

The above frequency domain construction method costs nearly constant time and the constructed symbol is accurate due to the utilization of Equation (8).

V. Evaluation

A. Simulation

In our settings, the number of concurrent nodes is called Parallel Degree (PD). Every node sends a given packet. We analyze the Symbol Error Rate (SER) and Bit Error Rate (BER) of a decoded packet. SER is the number of error symbols decoded with respect to total numbers. BER is the number of bit errors with respect to total bits. To make PD nodes collide in the worst situation, we apply a high duty cycle, i.e. 0.5, in the simulation.

First, we evaluate the performance of FlipLoRa under different Signal-to-Noise Ratio (SNR) environments. We define four SNR levels for the experiments: high SNR ($> 15dB$),

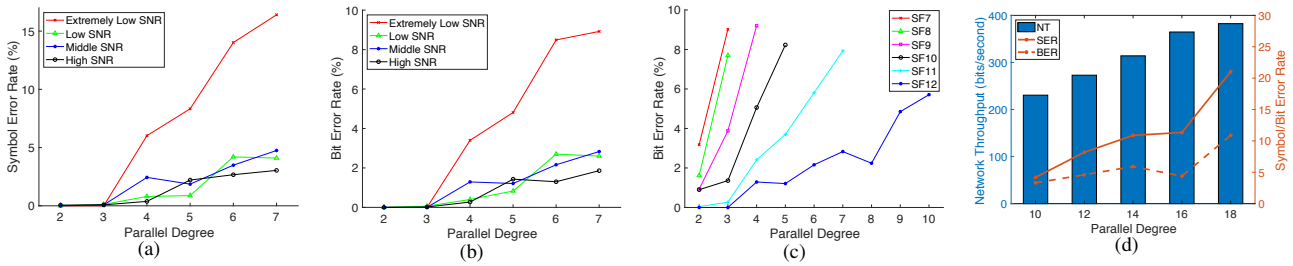


Fig. 9. (a) SER under different SNRs (b) BER under different SNRs (c) Exploring the parallel degree limitation of FlipLoRa on uniformly distributed packets (SF=12, Middle SNR, CR=$\frac{4}{7}$). (d) FlipLoRa's Maximum parallel degree supported by different SFs within 10% BER (SF=12, Middle SNR, CR=$\frac{4}{7}$).

middle SNR ($5 \sim 15dB$), low SNR($-5 \sim 5dB$) and extremely low SNR($< -5dB$). To simulate real packet collisions, we randomly set the transmitting power for each node in the range of SNR setting. In this experiment, we set the packet length to 21 bytes. The packet is sent with $SF = 12, CR = \frac{4}{7}$(coding rate), $B = 125kHz$. Thus each packet has 43 symbols according to [11]. Figure 9(a) and (b) show the SER and BER with different PDs. One nature observation is that both SER and BER increase with the increase of PD. That is because a larger PD brings a higher probability of shorter time offset in FlipLoRa decoding. Within a reasonable error rate, we find that FlipLoRa supports 10 concurrent transmissions. Besides, Figure 9(a)and (b) also reveal that the performance of FlipLoRa is not sensitive to SNR, except extremely low SNR. Since we select $CR = \frac{4}{7}$, $(7, 4)$ Hamming Code is used to correct 1-bit error. Thus, BER is usually lower than SER.

SF impacts the performance of FlipLoRa. Figure 9(c) shows the relation of BER and PD under different SFs with parameters setting: Middle SNR, SF=12, CR=$\frac{4}{7}$. We find that it is consistent with our observation in Section III. Since higher SF provides better quasi-orthogonality, FlipLoRa performs better. To explore the max PD FlipLoRa supports, we create an ideal environment that all packets are uniformly distributed, which means the chirp offset is constant. The results in Figure 9(d) show that FlipLoRa could allow more than 10 concurrent transmissions. Even if in the extremely low SNR, FlipLoRa supports 5+ parallel degree.

B. Performance in Real Deployment

We implement FlipLoRa on the software-defined radio platforms. As shown in Figure 11, the FlipLoRa gateway is implemented on a high-end software-defined radio (i.e. USRP N210) with a single TKX-470LC antenna. The end nodes are implemented on the low-cost HackRF One with a RaspberryPi. Both the FlipLoRa gateway and end nodes operate at 470MHz with a bandwidth of 125kHz. They can also work at any legal LoRa band depending on the configuration of users. We develop our FlipLoRa transceiver based on the GNU Radio library and implement FlipLoRa decoder on MATLAB. By default, we use the spreading factor and coding rate of FlipLoRa communication as 12 and $\frac{4}{7}$ respectively. The sampling rate of the USRP is set to 1 MHz. We place the FlipLoRa gateway on the roof of a building and distribute end nodes at different locations in the building.

We first show the relationship between FlipLoRa performance and SNR in real environments. By changing the transmitting power of the FlipLoRa end nodes, we can receive the signal under different SNR. Figure 10(a) shows the SER and BER under different SNRs. The parallel degree is set to 4. We observe that the symbol error rate of FlipLoRa remains lower than 6% even under extremely low SNR. In addition, by increasing the SNR of received signals, FlipLoRa can produce better demodulation results. Considering that LoRa applies a forward error correction (FEC) for data encoding, the decoding process can correct some of the errors from the symbol demodulation stage. Consequently, the bit error rate in Figure 10(a) is much lower than the symbol error rate, which is below 5% even when the SNR is extremely low.

Next, we explore FlipLoRa's performance under different PDs. Figure 10(b) shows the evaluation result of the demodulation performance of FlipLoRa with different number of concurrent packets. We set SF=12 and test the performance under low SNR environment. As concurrent packets increasing from 1 to 5, both the SER and BER of FlipLoRa grow up. But they increase slowly and remain below 6% even when 5 packets are overlapped together. This is because FlipLoRa extracts the feature of quasi-orthogonality between upchirp and downchirp to separate packets, which is stable and easy to detect even under high concurrency. Figure 10(c) compares the throughput of FlipLoRa and nominal LoRa. Since nominal LoRa physical layer does not support receiving multiple packets with the same configuration simultaneously, we see that the throughput of nominal LoRa drops. The throughput of FlipLoRa first grows linearly as PD increases because of the low SER/BER. When PD reaches 5, the interference between packets becomes stronger which results in high SER. FlipLoRa then fails to correct these bit errors with FEC mechanism and reaches its upper limit. The results show that FlipLoRa improves the max throughput of LoRa by 3.84x.

We evaluate the performance of our symbol cancellation method in real environment. Figure 10(d) shows the algorithm performance on canceling the target signal's power. For most cases (over 60%), our construction based cancellation algorithm can reduce the power of the target signal by $25dB$. Considering the SNR of low-cost LPWAN nodes is usually under $20dB$, our approach is sufficient for canceling interference from collisions. Besides, we also compare search-

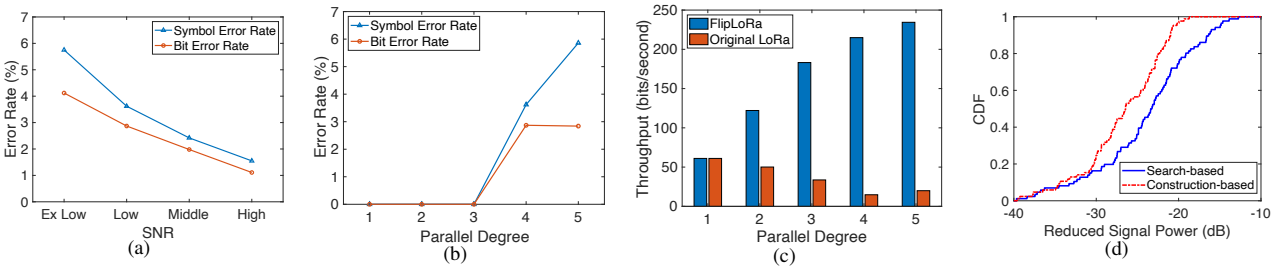


Fig. 10. (a) SER/BER of five real FlipLoRa nodes under different SNRs. (b) SER/BER of real FlipLoRa nodes under low SNR. (c) Throughput of FlipLoRa and nominal LoRa. (d) CDF of reduced signal power in symbol cancellation.

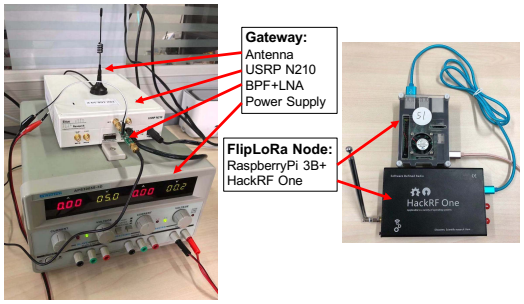


Fig. 11. FlipLoRa Setup: USRP N210 gateway and RaspberryPi+HackRF node

based cancellation algorithm with our method. The experiment results in Figure 10(d) show that our construction based cancellation algorithm is better than traditional methods.

VI. Related Work

Collisions in Wireless Networks: With the development of diverse wireless technologies [12], [13], [14], collisions and interferences have handicapped the performance of wireless networks [15], [16]. Many efforts have been carried out for solving collisions in wireless networks [17]. Some of them use multiple antennas to achieve MIMO on wireless devices [18], [19]. The multi-antenna systems can enable concurrent transmissions for both uplinks and downlinks. Although the MIMO based solutions improve the network throughput significantly, they cannot be used in common LPWAN devices with a single antenna. Successive Interference Cancellation (SIC) recovers packets from collisions by iteratively estimating and extracting the strongest component from the received signal. This technique is widely used in cellular networks, where the signal strength of each end device is known and can be scheduled by the base stations. ZigZag [20] and mZig [21] are two representative approaches for decoding collisions in WiFi and ZigBee networks. However, those works cannot work well for low SNR LoRa signals without leveraging the coding properties of LoRa.

Parallel Decoding for LoRa: Our work is inspired by some recent works of parallel decoding and collision recovery in LoRa. Netscatter [22] migrates LoRa encoding mechanism to backscatter devices and enables parallel decoding for hundreds of concurrent transmissions. The heart of Netscatter is a distributed coding mechanism that symbols from different devices shifted by different frequencies and each device uses On-Off Keying (OOK) to encode data. Devices in Netscatter should be strictly synchronized. Thus, this approach cannot be applied in current LoRa networks. Choir [9] proposes a collision recovery method for LoRa, which exploits the hardware imperfection of low-cost LoRa devices to decompose overlapped signals. However, as demonstrated in [22], this approach does not scale to more than 5 to 10 concurrent devices since the tiny frequency offset is difficult to extract especially for low SNR signal. More recently, mLoRa [23] and FTrack [10] exploits the misaligned edges of LoRa symbols to separate collisions. FlipLoRa is also based on such an idea but better leverages the characteristic of LoRa.

LoRa Technology Enhancement: Apart from collision decoding, many efforts also have been made to strengthen the LoRa technology. Charm [24] utilizes the multi-gateway feature in LoRa for weak signal decoding. Chime [25] empowers LoRa network by estimating the channel quality and selecting the best channel for transmission. Combining LoRa with backscatter technology is also appealing. *LoRa Backscatter* [26] presents the first wide-area backscatter system. PLoRa [27] first demonstrates an ambient backscatter system achieving kilometer communication range using LoRa. Sensing and localization are important areas in wireless research. LoRa Alliance releases a whitepaper [28] for LoRa localization. The main idea is using TDoA based on widely deployed gateways. But currently the accuracy is sub-100 meters level. Widesee [29] explores the possibility of LoRa sensing. The system can detect and localize human targets, which is especially useful in emergency scenarios.

VII. Conclusion

In this paper, we formally demonstrate the quasi-orthogonality between upchirp and downchirp in LoRa and leverage it to improve the performance of LoRa network. We propose FlipLoRa, a new coding mechanism to disentangle LoRa collisions, which allows concurrent transmission of multiple packets. The key idea of FlipLoRa is to utilize the quasi-orthogonality between upchirp and downchirp. FlipLoRa encodes packets with interleaved upchirps and downchirps instead of only using upchirp. We propose a decoding method for multiple collided FlipLoRa packets. We validate FlipLoRa

performance by both theoretical analysis and simulation. Further, we implement the FlipLoRa on software-defined radio and extensively evaluate its performance on real hardware. The evaluation results show that FlipLoRa can improve the throughput by more than 3.84x over LoRa physical layer.

ACKNOWLEDGEMENTS

This work is in part supported by National Natural Science Fund for Excellent Young Scholars (No. 61722210), National Natural Science Foundation of China (No. 61932013, 61532012, 61632008).

REFERENCES

[1] Semtech. LoRa™ Modulation Basics, AN1200. 22, Revision 2. 2015.
[2] Sigfox. Sigfox Connected Objects: Radio Specifications. Revision 1.3. 2019.
[3] Rapeepat Ratasuk, Vejlgaard Benny, Mangalvedhe Nitin, and Ghosh Amitava. NB-IoT system for M2M communication. In *Proceedings of IEEE WCNC*, 2016.
[4] Ingenu. RPMA Technology for the Internet of Things. In *Rpma White Paper*, 2015.
[5] Wenguang Mao, Jian He, and Lili Qiu. CAT: high-precision acoustic motion tracking. In *Proceedings of ACM MobiCom*, 2016.
[6] Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota. 3D Localization for Sub-Centimeter Sized Devices. In *Proceedings of ACM SenSys*, 2018.
[7] Pengjin Xie, Jingchao Feng, Zhichao Cao, and Jiliang Wang. GeneWave: Fast authentication and key agreement on commodity mobile devices. *IEEE/ACM Transactions on Networking (TON)*, 2018.
[8] Matthew Knight. Decoding LoRa : Realizing a Modern LPWAN with SDR. 2016.
[9] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering Low-Power Wide Area Networks in Urban Settings. In *Proceedings of ACM SIGCOMM*, 2017.
[10] Xia Xianjin, Zheng Yuanqing, and Gu Tao. FTrack: Parallel Decoding for LoRa Transmissions. In *Proceedings of IEEE SenSys*, 2019.
[11] Jansen C Liando, Amalinda Gamage, Agustinus W Tengourtius, and Mo Li. Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study. *ACM Transactions on Sensor Networks (TOSN)*, 2019.
[12] Zimu Zhou, Chenshu Wu, Zheng Yang, and Yunhao Liu. Sensorless sensing with WiFi. *Tsinghua Science and Technology*, 2015.
[13] Linsong Cheng and Jiliang Wang. How can I guard my AP? Non-intrusive user identification for mobile devices using WiFi signals. In *Proceedings of ACM MobiHoc*, 2016.
[14] Jiliang Wang, Zhichao Cao, Xufei Mao, Xiang-Yang Li, and Yunhao Liu. Towards energy efficient duty-cycled networks: analysis, implications and improvement. *IEEE Transactions on Computers (TC)*, 2015.
[15] Xiaolong Zheng, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. Interference resilient duty cycling for sensor networks under co-existing environments. *IEEE Transactions on Communications*, 2017.
[16] Xiaolong Zheng, Yuan He, and Xiuzhen Guo. Stripcomm: Interference-resilient cross-technology communication in coexisting environments. In *Proceedings of IEEE INFOCOM*, 2018.
[17] Xiaoyu Ji, Yuan He, Jiliang Wang, Kaishun Wu, Ke Yi, and Yunhao Liu. Voice over the dins: improving wireless channel utilization with collision tolerance. In *Proceedings of IEEE ICNP*. IEEE.
[18] Kate Ching-Ju Lin, Shyamnath Gollakota, and Dina Katabi. Random access heterogeneous MIMO networks. In *Proceedings of ACM SIGCOMM*, 2011.
[19] Swarun Kumar, Diego Cifuentes, Shyamnath Gollakota, and Dina Katabi. Bringing cross-layer MIMO to today's wireless LANs. In *Proceedings of ACM SIGCOMM*, 2013.
[20] Shyamnath Gollakota and Dina Katabi. ZigZag Decoding: Combating Hidden Terminals in Wireless Networks. In *Proceedings of ACM SIGCOMM*, 2008.
[21] Linghe Kong and Xue Liu. mZig: Enabling Multi-Packet Reception in ZigBee. In *Proceedings of ACM MobiCom*, 2015.
[22] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. NetScatter: Enabling Large-Scale Backscatter Networks. In *Proceedings of USENIX NSDI*, 2019.
[23] Wang Xiong, Kong Linghe, He Liang, and Chen Guihai. mLoRa: A Multi-Packet Reception Protocol for LoRa Communications. In *Proceedings of IEEE ICNP*, 2019.
[24] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks. In *Processing of ACM/IEEE IPSN*, 2018.
[25] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. Frequency Configuration for Low-Power Wide-Area Networks in a Heartbeat. In *Proceedings of USENIX NSDI*, 2020.
[26] Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R Smith, and Shyamnath Gollakota. LoRa backscatter: Enabling the vision of ubiquitous connectivity. In *Proceedings of ACM Ubicomp*, 2017.
[27] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. PLoRa: a passive long-range data network from ambient LoRa transmissions. In *Proceedings of ACM SIGCOMM*, 2018.
[28] LoRa Alliance. LoRa Alliance Geolocation Whitepaper. 2018.
[29] Lili Chen, Jie Xiong, Xiaojiang Chen, Sunghoon Ivan Lee, Kai Chen, Dianhe Han, Dingyi Fang, Zhanyong Tang, and Zheng Wang. WideSee: towards wide-area contactless wireless sensing. In *Proceedings of ACM SenSys*, 2019.

APPENDIX

$$\begin{aligned}\mathcal{U}(\omega_0^*)_k &= \sum_{n=0}^{2^{SF}-1} (\omega_0^*)^2 \cdot \exp\left(-j2\pi \cdot \frac{kn}{2^{SF}}\right) \\ &= \sum_{n=0}^{2^{SF}-1} \exp\left(j2\pi \frac{n(n+k-2^{SF})}{2^{SF}}\right) \\ &= \sum_{n=0}^{2^{SF}-1} \exp\left(j2\pi \frac{n(n+k)}{2^{SF}}\right).\end{aligned} \tag{22}$$

We define

$$\begin{aligned}f_N &= \sum_{n=0}^{N-1} a_n = \sum_{n=0}^{N-1} \exp\left(j2\pi \frac{n^2}{N}\right), \\ g_N &= \sum_{n=0}^{N-1} b_n = \sum_{n=0}^{N-1} \exp\left(j2\pi \frac{(n+\frac{1}{2})^2}{N}\right).\end{aligned} \tag{23}$$

where $N = 2^s, s \in \mathbb{N}^+$. When $N \geq 4$, we know $b_{\frac{N}{2}+n} = -b_n$ and then $g_N \equiv 0$. During the following process, we assume $N \geq 16$. If so, we have $a_{N-n} = a_n, a_{\frac{N}{2}} = a_0$. Therefore,

$$\begin{aligned}f_N &= 2\sum_{n=0}^{\frac{N}{2}-1} a_n \\ &= 2\sum_{n=0}^{N/4-1} \left(\exp\left(j2\pi \frac{n^2}{N/4}\right) + \exp\left(j2\pi \frac{(n+\frac{1}{2})^2}{N/4}\right)\right) \\ &= 2f_{\frac{N}{4}} + 2g_{\frac{N}{4}} \\ &= 2f_{\frac{N}{4}}.\end{aligned} \tag{24}$$

By the first few f values, we conclude that

$$f_N = 2^{(\log_2 N+1)/2} \exp\left(\frac{j\pi}{4}\right), \quad N \geq 4. \tag{25}$$

Using the trick of completing the square, we have

$$\mathcal{U}(\omega_0^*)_k = \exp\left(\frac{-j\pi k^2}{2^{SF+2}}\right) \sum_{n=0}^{2^{SF}-1} \exp\left(j2\pi \frac{(n+\frac{k}{2})^2}{2^{SF}}\right). \tag{26}$$

It is easy to get that the summation equals f_N when k is even and g_N when k is odd. Finally,

$$\mathcal{U}(\omega_0^*)_k = \begin{cases} 2^{\frac{SF+1}{2}} \exp\left(\frac{j\pi}{4} - \frac{j\pi k^2}{2^{SF+1}}\right), & 2 \mid k \\ 0, & 2 \nmid k. \end{cases} \tag{27}$$