Combating Packet Collisions Using Non-Stationary Signal Scaling in LPWANs

Shuai Tong[1], Jiliang Wang[1], Yunhao Liu[1,2]
[1]Tsinghua University, Beijing, China
[2]Michigan State University, Michigan, USA
tl19@mails.tsinghua.edu.cn, jiliangwang@tsinghua.edu.cn, yunhaoliu@gmail.com

ABSTRACT

LoRa, a representative Low-Power Wide Area Network (LPWAN) technology, has been shown as a promising platform to connect Internet of Things. Practical LoRa deployments, however, suffer from collisions, especially in dense networks and wide coverage areas expected by LoRa applications. Existing collision resolution approaches do not exploit the coding properties of LoRa and thus cannot work well for low SNR LoRa signals. We propose *NScale* to decompose concurrent transmissions by leveraging subtle inter-packet time offsets for low SNR LoRa collisions. NScale (1) translates subtle time offsets, which are vulnerable to noise, to robust frequency features, and (2) further amplifies the time offsets by non-stationary signal scaling, i.e., scaling the amplitude of a symbol differently at different positions. In practical implementation, we propose a noise resistant iterative symbol recovery method to combat symbol distortion in low SNR, and address frequency shifts incurred by CFO and packet time offsets in decoding. We theoretically show that NScale introduces < 1.7 dB SNR loss compared with the original LoRa. We implement NScale on USRP N210 and evaluate its performance in both indoor and outdoor networks. NScale is implemented in software at the gateway and can work for COTS LoRa nodes without any modification. The evaluation results show that NScale improves the network throughput by 3.3× for low SNR collided signals compared with other state-of-the-art methods.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Link-layer protocols**; **Wireless access points, base stations and infrastructure**.

KEYWORDS

Internet of Things, LPWAN, LoRa, Collision Resolution

ACM Reference Format:
Shuai Tong, Jiliang Wang and Yunhao Liu. 2020. Combating Packet Collisions Using Non-Stationary Signal Scaling in LPWANs. In *International Conference on Mobile Systems, Applications, and Services (MobiSys '20), June 15–19, 2020, Toronto, ON, Canada.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3386901.3388913

1 INTRODUCTION

As a promising technology for Low-Power Wide Area Networks (LPWANs), LoRa is recently drawing extensive interests from both academia and industry. Different from high-power and high-bitrate Wi-Fi or 4G/5G, LoRa focuses on the field of low-power, low-cost, and long-range communications for millions of Internet of Things (IoT) devices [1]. As one of the key communication technologies for IoTs, LoRa is widely used in various IoT applications such as environment monitoring [2], wild animal tracking [3], disaster rescue [4], warehouse management [5], etc.

However, LoRa networks in practice suffer from packet collisions, especially when connecting a large number of devices, which is expected for most LoRa applications [6, 7]. Collisions adversely cause packet loss and throughput degradation, which also drain battery life and waste precious air time and spectrum. Moreover, LoRa usually uses a star-of-stars topology where thousands of LoRa nodes connect to a single LoRa gateway [8]. For design simplicity and energy conservation, LoRa adopts a simple MAC layer design (e.g., ALOHA based MAC protocol in LoRaWAN), which further exacerbates packet collisions in LoRa networks [9, 10].

Existing Approaches. The collision problem should be carefully addressed before applying LoRa as the main technique for connecting millions of IoT devices. Although there exist a large collection of collision decoding approaches, they do not exploit the coding properties of LoRa. Thus, they (e.g., time domain Successive Interference Cancellation, SIC [11, 12]) cannot work well for low SNR LoRa signals. For example, mLoRa [13] applies SIC to LoRa collisions. It starts with a collision-free chunk and then iteratively reconstructs and extracts each decoded symbols. According to their experiment results, mLoRa mainly works for signals with SNR > 5 dB. Choir [14] exploits the hardware imperfections of low-cost LoRa nodes to separate collided packets. FTrack [15] decodes multiple LoRa packets from a collision by calculating the instantaneous frequency continuity by short time spectrum analysis.

Fundamental limitations: Existing collision decoding approaches [13, 15] have limitations in decoding low SNR LoRa signals (e.g., $SNR < 0$). They focus more on the time domain signal analysis and interference cancellation. But they do not consider the decoding features of LoRa which can concentrate energy in the frequency domain. As a result, those methods have a high SNR loss compared with the original LoRa decoding and cannot work for low SNR LoRa signals.

Our Approach. To resolve collisions in low SNR LoRa signals, we present NScale to decode packets from collided LoRa signals. The heart of NScale is to leverage the subtle packet time offsets to disentangle collided packets. To achieve low SNR collision decoding, NScale (1) translates packet time offsets, which are vulnerable

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

MobiSys '20, June 15–19, 2020, Toronto, ON, Canada
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7954-0/20/06…$15.00
https://doi.org/10.1145/3386901.3388913

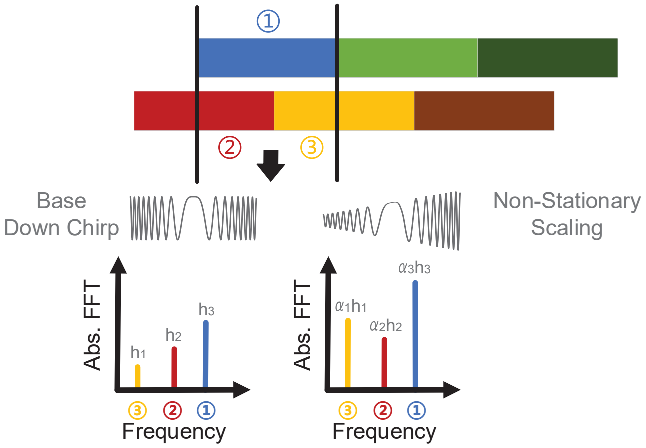


Figure 1: NScale decoding example: for each collided symbol, NScale transforms and amplifies the time domain features into frequency features through non-stationary scaling. Chirp segments at different positions of the window are amplified with different peak scaling factors.

to noise, to more robust frequency features, and (2) amplifies the time offsets by non-stationary signal scaling, i.e., scaling a symbol differently at different positions. NScale then leverages the frequency features after non-stationary scaling to decompose concurrent transmissions.

To see how NScale works, consider a simplified collision scenario in Figure 1, where two packets - each with three chirps (symbols) - collide. The PHY layer of LoRa uses the Chirp Spread Spectrum (CSS) modulation to modulate data into chirp symbols of linearly *increasing* frequency. In demodulation, each window containing a chirp symbol is multiplied with a base down-chirp of linearly *decreasing* frequency. When there is no collision, this dechirp operation results in a single frequency tone (a single peak after FFT) which represents the modulated data. When there are collisions, the dechirp results in multiple frequency tones in a window, making it difficult to distinguish symbols from different packets.

To decode collided packets, as shown in Figure 1, we leverage the down-chirp with non-stationary amplitude to translate the time misalignment of packets into frequency features, which can be leveraged to separate packets from different senders. NScale first applies the standard demodulation with the base down-chirp to each window. The three chirp segments, as shown in Figure 1, result in three FFT peaks (h_1, h_2 and h_3) with height proportional to the segment length and the signal amplitude. Those three peaks also show why traditional LoRa cannot decode the collision. Further, NScale dechirps each window with a *non-stationary* scaled down-chirp, i.e., a down-chirp with varying amplification along time. We multiply the three chirp segments in the window by a non-stationary scaled down-chirp with different amplitudes at different positions. This results in FFT peaks amplified with different factors ($\alpha_1 h_1$, $\alpha_2 h_2$ and $\alpha_3 h_3$), depending on which part on the non-stationary scaled down-chirp is multiplied with the chirp segment. Combining the dechirp results, we can obtain the scaling factors α_i. By carefully designing the non-stationary scaled down-chirp, we can make α_i distinguishable and derive the chirp segment length and position related to the non-stationary scaled down-chirp. Based on this, we can group chirp segments for each packet with the same misalignment and then decode each packet.

Challenges. Turning the idea into reality, however, entails non-trivial challenges. First, NScale relies on accurate measurement of peaks for frequency tones after dechirping, which is difficult due to low SNR signal and the phase rotation property of the Fourier transformation. We propose a noise resistant iterative peak recovery algorithm to combat the peak distortion, and achieve an accurate estimation of both frequency and height for each peak. Second, it is non-trivial to design the non-stationary scaled down-chirp, which affects the performance of NScale. We make an in-depth analysis of the relationship between non-stationary scaled down-chirps and the decoding performance, and propose the designing strategy for non-stationary scaled down-chirps to optimize NScale's decoding performance in practice. Third, after grouping chirp segments to packets for decoding, we find it is difficult to decode each group of chirp segments due to the mixed impact of the Central Frequency Offset (CFO) and symbol-window time offset. We design a technique to calculate CFO and symbol-to-window time offset based on the combination of up-chirps and down-chirps in the preamble and SFD of LoRa packets. Finally, we decode each group of peaks after compensating the CFO and time offset.

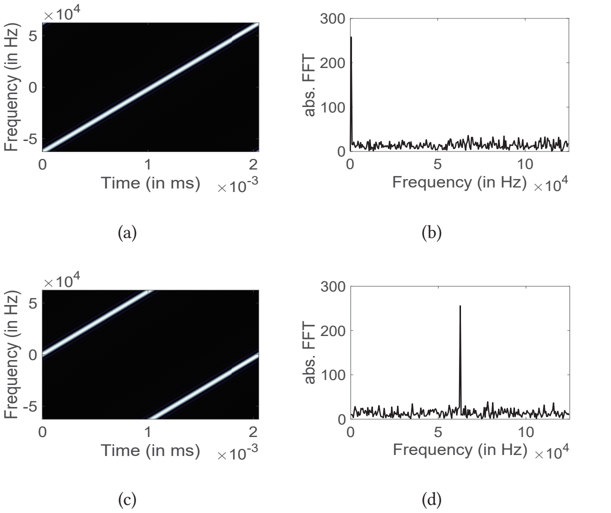
Main Results and Contributions.

- We propose NScale, a protocol leveraging non-stationary scaling to decompose concurrent transmissions for low SNR LoRa collisions, trying to bridge the gap between LoRa vision to provide low-power long-distance connection to large scale IoT devices, and its practical limitations. To address practical challenges in NScale design, we propose a noise-resistant iterative peak recovery algorithm to resolve peak distortion in low SNR LoRa signal, and remove the impact of the CFO and time offset to accurately decode packets.
- We theoretically analyze NScale performance and show that NScale incurs SNR loss <**1.7 dB** to original LoRa.
- We implement NScale on the SDR platform USRP N210. NScale is completely implemented in software at the LoRa gateway without any modification to LoRa end nodes. Thus, it can be easily applied to COTS LoRa nodes and existing LoRa networks.
- We thoroughly evaluate NScale's performance in both indoor and outdoor LoRa networks. The experiment results show that NScale can improve the network throughput in collisions by **3.3**× for low SNR LoRa signals compared with other state-of-the-art methods.

2 BACKGROUND AND MOTIVATION

2.1 LoRa Background

LoRa physical layer adopts the Chirp Spreading Spectrum (CSS) technique for modulation [16]. CSS modulates data into chirp symbols whose frequency change linearly over time. As shown in Figure 2(a), given a predefined bandwidth BW, the frequency of the base up-chirp $C(t)$ linearly increases from $-\frac{BW}{2}$ to $\frac{BW}{2}$. Thus, the frequency of the base up-chirp can be represented as $kt - \frac{BW}{2}$, where



(a)

(b)

(c)

(d)

Figure 2: LoRa Physical Layer: (a)-(b) Spectrogram and demodulation result of a base chirp symbol. (c)-(d) Spectrogram and demodulation result of a shifted chirp symbol.

k denotes the frequency increasing rate of the chirp. The base up-chirp $C(t)$ can be represented as

$$C(t) = e^{j2\pi(-\frac{BW}{2}+\frac{kt}{2})t} \tag{1}$$

LoRa encodes data bits into symbols by shifting the initial frequency of the base up-chirp. A LoRa symbol with initial frequency f_{sym} is denoted as $C(t)e^{j2\pi f_{sym}t}$. Given the bandwidth BW, frequency for a symbol higher than $\frac{BW}{2}$ aligns down to $-\frac{BW}{2}$ as shown in Figure 2(c). LoRa defines N different shifted initial frequencies, which results in N uniformly shaped up-chirps to encode $SF = log_2 N$ bits [17].

A typical LoRa receiver demodulates an incoming LoRa chirp as follows. It first multiplies the received signal with a base down-chirp $C^{-1}(t)$, i.e., the conjugate of a base up-chirp $C(t)$. After multiplying, the received signal is dechirped into a single tone at the frequency of f_{sym}, i.e.,

$$C^{-1}(t) \times C(t)e^{j2\pi f_{sym}t} = e^{j2\pi f_{sym}t}$$

Then the receiver applies the Fast Fourier Transform (FFT) on the multiplication result, translating the time-domain signal into an energy peak in the frequency domain, as shown in Figure 2(b) and Figure 2(d). The index of the energy peak (i.e., frequency) indicates the encoded data of the received chirp symbol.

LoRa follows a unique packet structure 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 followed by a variable number of payload symbols [18]. The preambles are identical base up-chirps, and the SFDs are identical base down-chirps. The payload symbols are all shifted up-chirps.

2.2 Limitations & Challenges

The main advantage of LoRa design is that it can concentrate time domain energy into a single tone frequency peak by dechirping [19]. CSS is inherently robust against channel noise. Thus, LoRa signals can be detected and decoded even under extremely low SNR (e.g. SNR as low as -20 dB) [20], enabling low power and long range communications [21, 22].

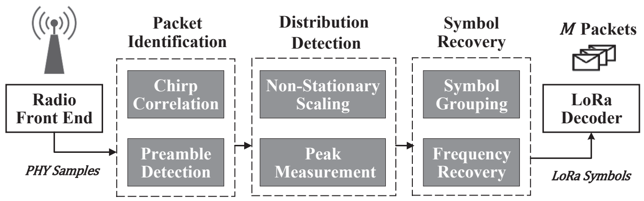


Figure 3: Main workflow of NScale design.

When multiple LoRa nodes transmit simultaneously, their signals will collide at the receiver. As shown in Figure 1, multiple chirp segments overlap in the same demodulation window, each of which corresponds to an energy peak in the FFT result. The LoRa demodulator cannot map FFT peaks to the correct transmitters, and thus it fails to decode the collided signals. The key to decode collisions is to correspond multiple peaks in each demodulation window to different transmitters.

Existing collision decoding approaches do not thoroughly exploit the LoRa encoding properties and cannot work well under low SNR LoRa signals. For example, SIC usually focuses on time-domain signal decoding and cancellation and does not leverage the LoRa properties. As a result, existing collision decoding approaches [13, 15] cannot work for low SNR LoRa signals. They mainly provide experiment results on high SNR decoding, like $SNR > 0$ in [13] and [15]. We argue this significantly removes the major advantages of LoRa, which is supposed to provide long range and low power communications with very low SNR of even -20 dB.

2.3 Motivation

We leverage the fact that collided LoRa packets are likely to be misaligned in time. As shown in Figure 1, the collided signal is divided into consecutive demodulation windows of L, where L is the symbol length. When a packet is not exactly aligned with the window, there will be two LoRa segments in each window. Assuming the length of the first LoRa symbol and second LoRa symbol 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 in-window segment distribution, i.e., γ_1 and γ_2, are the same across all consecutive windows. (2) For two unaligned LoRa packets, their in-window segment distributions should be different. The in-window segment distribution can be applied to disentangle different packets in a collision.

3 NSCALE DESIGN

3.1 Design Overview

Design Goals. Overall, NScale has the following main design goals:

- NScale should work for low SNR LoRa collisions and incurs very small SNR loss to LoRa decoding.

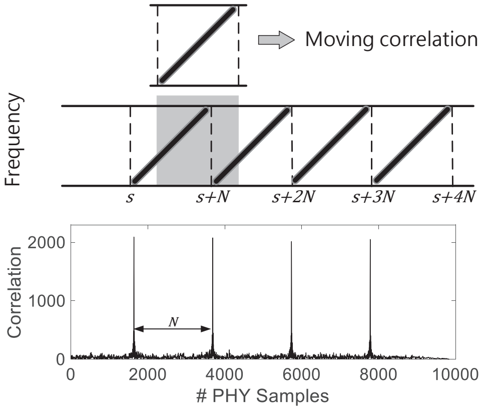


Figure 4: Detecting LoRa packets by correlation with a single base up-chirp.

- NScale should incur no modification to LoRa node and thus can be applied to COTS nodes and existing LoRa network deployments.
- NScale should incur small computation overhead compared with typical LoRa decoding.

Figure 3 illustrates the main flow of NScale design:

Packet identification. For a received signal sequence, NScale first detects the existence of LoRa packets by correlating with preambles. When there is no collision, the signal is sent directly to a standard LoRa decoder. Otherwise, NScale utilizes up-chirps of preambles to identify the coarse beginning of each collided packet. Then the collided signal is divided into multiple consecutive demodulation windows of chirp length.

In-window distribution detection. For each demodulation window, NScale transforms the in-window distribution of each low-SNR symbol into robust FFT peak features by non-stationary scaling. Usually, in a demodulation window, there will be multiple chirp segments when the window and the packet are misaligned. As long as two packets are not aligned, the in-window distribution of chirp segments for those two packets are different. Consequently, we can infer which packet the corresponding chirp segments belongs to according to the segment distribution. NScale accurately recovers the features of peaks in the presence of distortion due to the phase rotation property of the Fourier transform.

Symbol recovery. Based on the estimated in-window distribution information, NScale classifies symbols into multiple clusters, each corresponding to a collided LoRa packet. Before decoding, we estimate the CFO and the packet-window time offsets utilizing up-chirps and down-chirps from the preamble and SFD of the corresponding LoRa packet. Finally, NScale combines each pair of chirp segments into packet chirps, and the output chirps are fed to the standard LoRa decoder for packet decoding.

3.2 Packet Identification

Upon receiving a signal sequence, NScale first detects the existence of LoRa packets. One intuitive idea for packet identification is to correlate the received digital samples with a standard LoRa preamble [23], which consists of N_c consecutive base up-chirps. The correlation is expected to create a peak when the standard preamble aligns with a received packet. This, however, has some practical limitations. Due to the CFO between the transmitter and the receiver, up-chirps in the received preamble have different initial phases. Thus, the up-chirps in the standard LoRa preamble may have different initial phases compared to the received up-chirps. After correlating the received samples with a standard LoRa preamble, the resulting correlation peak of the entire preamble becomes low or disappears, even when the correlated sequence accurately aligns with the received packet.

We propose an enhanced two-step preamble detection. We leverage the fact that CFO influences the correlation of the entire preamble, but has less effect on correlating a single up-chirp. Thus, we use a single moving base up-chirp with constant amplitude to correlate with the incoming signal, resulting in N_c correlation peaks, as shown in Figure 4. The intervals between each two adjacent correlation peaks are identical, equal to the number of samples within a chirp, i.e., N. Denote $C[i]$ as the ith correlation output, our packet identification works as

$$find\ s, \quad s.t. \quad |C[s+kN]| > \delta, \quad \forall k \in [0, N_c - 1]$$

where s is the start of packet preamble and δ represents the minimum correlation requirements, which can be determined by channel estimation.

3.3 In-Window Distribution Detection

NScale separates LoRa collisions according to the in-window distribution of each chirp symbol. A practical challenge is precisely extracting the time domain segment distribution under low SNR. We address this from the following aspects: (1) translating the time-domain feature to robust FFT peak features in the frequency domain; (2) concentrating the energy in the time domain to frequency features by dechirping, which preserves the merits of LoRa decoding; (3) using the non-stationary signal scaling to further amplify the features. The algorithm chiefly involves the following three steps. (1) For signals in each window, we multiply it with a base down-chirp, and perform the Fourier transform on the multiplication. This translates time-domain chirp segments to energy peaks in the frequency domain. (2) We further multiply the received signal with a non-stationary scaled down-chirp with varying amplitude over time. We transform the result of multiplication to energy peaks in the frequency domain, extracting each peak's index and height. (3) We pair the energy peaks from the above two steps according to peak indexes, and calculate peak scaling factors as the height ratios of each pair of peaks. Given the scaling function of the non-stationary scaled down-chirp, we can derive the in-window distribution of each chirp segment from the peak scaling factors.

Suppose n LoRa packets collide at the receiver. We detect the start time of the received signal and divide the collision into consecutive demodulation windows, each having the same length as a chirp. For signals in each demodulation window, we illustrate how to extract the in-window distribution information of two chirp segments: a left segment to the start of the window (Figure 5(a)) and a right segment adjacent to the end of the window (Figure 5(d)). Figure 5(a) shows an example of the left segment with a symbol-window time

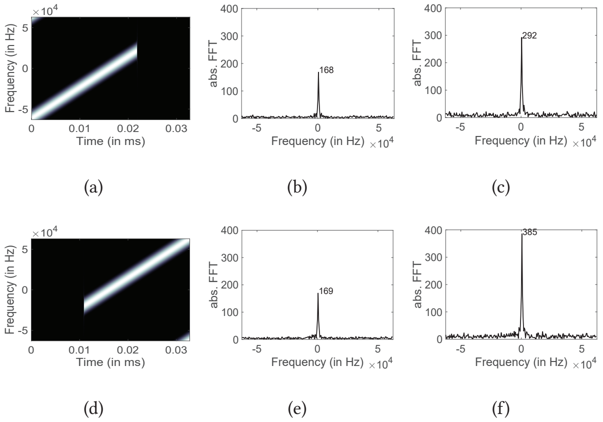


Figure 5: In-Window Distribution Detection: (a)(d) Symbol segments with different in-window distributions. (b)(e) FFT after multiplying with a base down-chirp. (c)(f) FFT after multiplying with a non-stationary scaled down-chirp. In-window distributions of multiple collided segments can be detected simultaneously through a single non-stationary dechirping.

offset of Δt, i.e.,

$$C_L(t) = He^{j2\pi ft}C(t+\Delta t) \qquad 0 \le t < T - \Delta t \tag{2}$$

where $C(t)$ is the base up-chirp, f and H denotes the initial frequency and amplitude of the received chirp. As the time offsets of chirps can be translated to frequency shifts, we have $C(t+\Delta t) = e^{j2\pi(k\Delta t)t}C(t)$, where k is the increasing rate of the chirp frequency. We multiply $C_L(t)$ by a base down-chirp (i.e., $C^{-1}(t)$) with stationary amplitude throughout the whole symbol duration. This multiplication dechirps the chirp segment into a single tone, with the frequency of $f + k\Delta t$ and time range of $[0, T - \Delta t)$. After the multiplication, we perform FFT to aggregate the energy of the chirp segment to an energy peak in the frequency domain, as shown in Figure 5(b). We perform zero-padding to the original signal before the Fourier transform to improve the frequency granularity. Suppose the energy peak of the chirp segment appears at the mth FFT bin of transformation result. The height of that peak can be calculated as:

$$h_1 = |X_1[m]| = \left| \sum_{n=0}^{N_0-1} C^{-1}[n] \times C_L[n]e^{-j2\pi\frac{nm}{N}} \right|$$

where $C_L[n]$ is the nth discrete sample of the chirp segment, N_0 and N represent the number of samples for the chirp segment and the whole demodulation window, respectively. Substituting $C_L[n]$ with Eq. 2, we have the peak height as $h_1 = H \times N_0$.

Beside the base down-chirp, we design a non-stationary scaled down-chirp $A(t)C^{-1}(t)$, whose amplitude changes over time with a known scaling function $A(t)$. We multiply the left segment $C_L(t)$ with the non-stationary scaled down-chirp. After the Fourier transformation on the multiplication result, we obtain an energy peak in the frequency domain, as shown in Figure 5(c), with the height:

$$h_2 = |X_2[m]| = \left| \sum_{n=0}^{N_0-1} A[n]C^{-1}[n] \times C_L[n]e^{-j2\pi\frac{nm}{N}} \right|$$

where $A[n]$ is the discrete sample of scaling function. Substituting $C_L[n]$ with Eq. 2, the height of the energy peak is simplified as $h_2 = H\sum_{n=0}^{N_0-1} A[n]$.

Note that the non-stationary scaling on the down-chirp does not affect the frequency of the FFT result. The energy peaks of both h_1 and h_2 locate at the same frequency, i.e., $f + k\Delta t$. The above procedures also translate the right segment in Figure 5(d) into two FFT peaks with the same frequency, as shown in Figure 5(e) and (f). In the presence of collisions, multiple overlapped chirp segments can be translated into FFT peaks simultaneously through a single dechirping. For each chirp segment, we pair its energy peaks from the multiplication of base down-chirp and non-stationary down-chip according to the peak frequencies. For paired peaks of the same chirp segment, we calculate the peak scaling factor P as the ratio of peak heights

$$P = \frac{h_2}{h_1} = \frac{\sum_{n=0}^{N_0-1} A[n]}{N_0} \tag{3}$$

As the scaling function $A[n]$ is already known, we can infer the in-window distribution for each chirp segment directly from the peak scaling factor P.

At this point, we can calculate the in-window distribution of chirp segments. Then, two consecutive chirp segments for the same chirp are paired and merged by searching peaks with the same frequency in consecutive windows. Finally, we can obtain all chirp symbols extracted from the collision, each with the estimated in-window distribution information.

3.4 Peak Estimation in Practice

The above in-window distribution calculation relies on accurately estimating FFT peaks from the LoRa collisions, including both the peak frequency and the peak height. However, accurate peak estimation is challenging due to peak distortion caused by phase rotation property of the Fourier transform. In this subsection, we discuss the reason of peak distortion and further show how to improve the estimation accuracy in practice.

In practice, the phase rotation property of the Fourier transform distorts frequency peaks. A shift in the time domain can be translated into a phase rotation in the frequency domain [24]:

$$\begin{aligned} \mathcal{F}\{r(t)\} &= R(f) \\ \mathcal{F}\{r(t+\tau)\} &= R(f) \cdot e^{j2\pi f\tau} \end{aligned} \tag{4}$$

where $r(t)$ is the signal in the time domain, and $R(f)$ is the corresponding frequency-domain representation. LoRa conveys data by cyclically shifting the frequency of base up-chirps. After dechirping, a LoRa symbol generates two frequencies, i.e., f and $f - BW$, respectively. When the sampling rate is equal to the chirp bandwidth BW, the Fourier transform of the two frequencies will result in two peaks, denoted as $R_1(f)$ and $R_2(f)$, at the same location. If the LoRa chirp accurately aligns with the demodulation window, $R_1(f)$ and $R_2(f)$ add up constructively, resulting in an ideal peak as shown in Figure 6(a). However, when the LoRa symbol and the

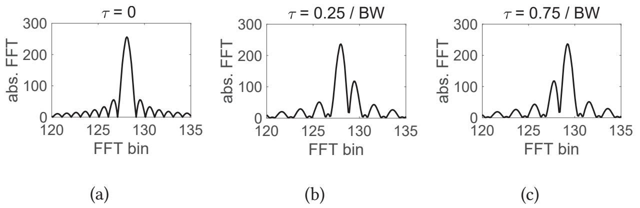


Figure 6: Peak Distortion Example: (a) Ideal peak when a chirp is aligned with the demodulation window. (b)(c) Distorted peaks when a chirp and demodulation window are misaligned by a time offset τ.

demodulation window are misaligned (suppose the time offset is τ), the Fourier transform of the two frequencies will rotate by different phases. Recall the phase rotation property in Eq. 4, with the same time offset τ, peaks of different frequencies have different phase shifts, i.e., $R_1(f) \cdot e^{j2\pi f \tau}$ and $R_2(f) \cdot e^{j2\pi(f-BW)\tau}$. Those two peaks with different phase shifts add up destructively, resulting in peak distortions. As shown in Figure 6(b) and (c), with a different time offset τ, peaks in the frequency domain are distorted differently, impacting the estimation of accurate peak frequency and peak height in practice.

To solve this problem, NScale recovers FFT peaks from distortion by compensating the phase rotation of the two frequencies, i.e., f and $f - BW$, which relies on the sampling rate of off-the-shelf ADCs being much higher than the chirp bandwidth BW. When sampling the received signal at a high rate (e.g., higher than $2BW$ according to the Nyquist-Shannon theorem [25]), the Fourier transform of the received signal will result in two separate peaks, $R(f)$ and $R(f - BW)$, respectively. To compensate the phase rotation, NScale searches the phase difference between $R(f)$ and $R(f - BW)$ via:

$$\phi = \arg \max_{0<\phi\leq 2\pi} R(f) \cdot e^{j\phi} + R(f - BW) \tag{5}$$

The maximum can be obtained only when the phase rotation effect is compensated. Then we can estimate the peak accurately. It is worth noting that we can apply stochastic gradient-descent algorithms on Eq. 5 with randomly chosen initial points that are likely to converge to the global maximum.

3.5 Symbol Recovery

To decode the collisions, we further need to group symbols into different packets and then recover the precise information of each symbol.

NScale utilizes a constrained k-means based approach to group the symbols into k clusters (i.e., the k collided packets). The in-window distribution, which is identical for symbols of the same packet but distinct for symbols of different packets, is selected as the characteristic value for clustering. The chirp symbols with different in-window distribution are fed into the clustering algorithm for symbol grouping. The grouping method further applies constraints from the following aspects. (1) Symbols are grouped to the clusters in time order. (2) A new cluster can only emerge and start gathering symbols after detecting the start of a packet. (3) Each cluster has one and only one symbol in each demodulation window. Based on the constraints, we obtain k groups of symbols, each of which corresponds to a collided packet. Finally, we send the recovered symbols to a standard LoRa decoder for decoding.

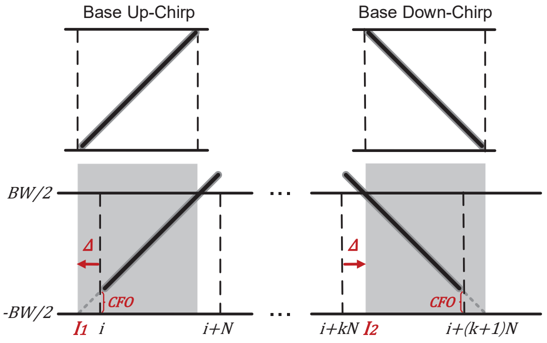


Figure 7: Detecting accurate packet start by eliminating the impact of CFO.

Before packet decoding, we need to (1) eliminate the impact of CFO to recover the accurate frequency of each chirp, and (2) find the accurate start of each packet to compensate for the time offset of each chirp. A practical challenge is that the CFO and the packet time offset are cross dependent, impacting the estimation of each other. We leverage the unique structure of LoRa packets in accurately calculating CFO and packet time offsets. One key finding is that for the same amount of CFO, the resulted correlation peaks of up-chirps and down-chirps shift oppositely. As shown in Figure 7, assuming a LoRa packet is received with a positive CFO, we correlate the received signal with a base up-chirp followed by a base down-chirp, respectively. The shift of peak frequency of the up-chirp is $-N \cdot CFO/BW$, while the shift for the down-chirp is $N \cdot CFO/BW$. Denote $\Delta = N \cdot CFO/BW$. For the base up-chirp, the correlation peak appears at $I_1 = i - \Delta$, where i is the actual start of the packet. While for the base down-chirp, the correlation peak appears at $I_2 = (i + kN) + \Delta$. We calculate the shift of correlation peaks as

$$\Delta = MOD(I_2 - I_1, N)/2 \tag{6}$$

Then we can calculate CFO as

$$CFO = \Delta \times BW/N \tag{7}$$

Therefore, we can eliminate the impact of CFO and time offset for each chirp, and finally recover the accurate frequency for packet decoding.

4 NSCALE ANALYSIS

4.1 SNR Loss

We present an analysis of the SNR requirements of NScale and show that the non-stationary amplitude scaling introduces a very small SNR loss, allowing NScale to decode collisions under extremely low SNR.

NScale processes the signal in each demodulation window. Let $y = y_A + w$ be the received signal within a demodulation window, where y_A is the chirp segment and w is the channel noise. By multiplying with a base down-chirp, y_A is dechirped into a single tone, while w still follows the compound Gaussian distribution [26].

NScale then performs the FFT on the multiplication as

$$\begin{aligned}\mathcal{F}(y \cdot C^{-1}) &= \sum_{n=0}^{N-1} e^{-j2\pi\frac{nk}{N}} \cdot (y[n] \cdot C^{-1}[n]) \\ &= \sum_{n=0}^{N-1} e^{-j2\pi\frac{nk}{N}} \cdot (\hat{y}_A[n] + \hat{w}[n])\end{aligned}$$

For the target segment $\hat{y}_A$, after the transformation, the energy of $\hat{y}_A$ is aggregated, resulting in an energy peak at the corresponding FFT bin. The height of the energy peak is the product of the amplitude and the length of the target segment, i.e.,

$$H = \lim_{k/T \to f_{sym}} \left| \sum_{n=0}^{L-1} h e^{j2\pi(f_{sym}\frac{nT}{N} - \frac{nk}{N})} \right| = h \times L$$

where f_{sym}, h and L are the frequency, amplitude and length of $\hat{y}_A$, respectively.

We also apply dechirping and the FFT with a non-stationary scaled down-chirp. With a non-stationary scaling function $A[n]$, the peak height of the resulted signal is

$$H = \sum_{n=0}^{L-1} A[n]h = h \sum_{n=0}^{L-1} A[n] \tag{8}$$

For the noise $\hat{w}$, generally, it follows the compound Gaussian distribution, where $\Re(\hat{w}) \sim \mathcal{N}(0, \sigma^2)$ and $\Im(\hat{w}) \sim \mathcal{N}(0, \sigma^2)$. Thus for signal in a demodulation window, the total energy of noise can be calculated as:

$$C_w = \sum_{n=0}^{N-1} |\hat{w}[n]|^2 = N \times E(|\hat{w}[n]|^2) = 2\sigma^2 N$$

where $E(|\hat{w}[n]|^2) = 2\sigma^2$ is the expectation of the instant noise energy density [27]. After applying the FFT on the received signal, the noise $\hat{w}$ also transforms from the time domain to the frequency domain. Based on the Parseval's theorem, the total energy of the signal in the time domain is equal to the energy of the frequency domain, i.e., $C_w = 2\sigma^2 N = \frac{1}{N}\sum_{k=0}^{N-1} |X[k]|^2$, where $X[k]$ is the power spectral density (PSD) of $\hat{w}$ at the frequency of k/T. As the PSD of Gaussian white noise is subject to uniform distribution, the energy strength of $\hat{w}$ at each FFT bin can be expressed as $|X[k]|^2 = 2\sigma^2 N$.

Now considering the non-stationary scaling on the received signal, both the target symbols and the noise within the window are amplified non-stationarily. The noise amplified by factor $A[n]$ follows the distribution of $\mathcal{N}(0, \sigma^2 A[n]^2)$. Thus, the total energy of the noise within a demodulation window is $C'_w = 2\sigma^2 \sum_{n=0}^{N-1} |A[n]|^2$ and the energy strength at any frequency is:

$$|X'[k]|^2 = 2\sigma^2 \sum_{n=0}^{N-1} |A[n]|^2$$

Consequently, in the frequency domain, the energy output of noise follows the Rayleigh distribution with the parameter of $|X'[k]|$. According to [28], for a sequence of N random values, each of which follows the Rayleigh distribution with the parameter of δ, the maximum of the sequence approximates to $\sqrt{\delta^2 \times H_N}$, where $H_N = \sum_{n=1}^{N} \frac{1}{n}$ is the Nth value of the *harmonic series*. Therefore, after performing the FFT on the noise, the maximum output in the frequency domain is:

$$M = \max_{k, k \neq l} |X'[k]| \approx \sqrt{2\sigma^2 \sum_{n=0}^{N-1} |A[n]|^2 \times H_{2^{sf}-1}} \tag{9}$$

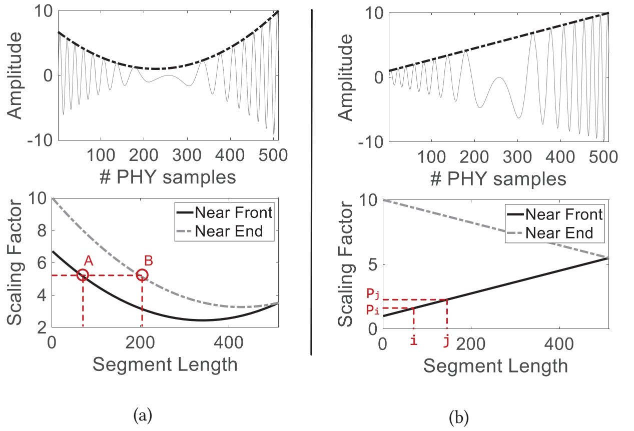


Figure 8: Designing non-stationary scaling: (a) A non-monotonous scaling function leads to two different chirp segments corresponding to the same peak scaling factor. (b) A linear scaling function maximize the mean inter-object distances.

To decode signal from noise, the energy peak in Eq. 8 should be higher than the maximum noise output in Eq. 9, i.e., $H > M$. Thus, the minimal SNR requirement for NScale is:

$$R_1 = 10\lg\left(\frac{h^2}{2\sigma^2}\right) = 10\lg\left(\frac{\sum_{n=0}^{N-1} |A[n]|^2 \times H_{2^{sf}-1}}{(\sum_{n=0}^{L-1} A[n])^2}\right)$$

Ideally, the SNR requirement for original LoRa in the case of no collision is $R_2 = 10\lg(H_{2^{sf}-1}/N)$. Thus, the SNR loss introduced by NScale's non-stationary amplitude scaling is

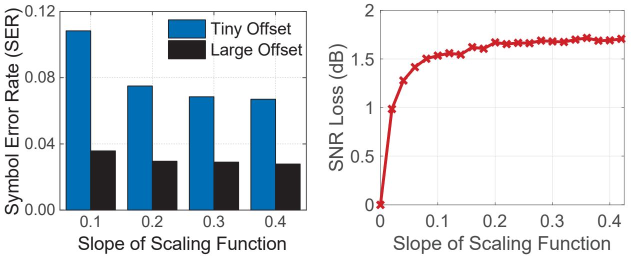
$$Loss = R_1 - R_2 = 10\lg\left(\frac{N\sum_{n=0}^{N-1} |A[n]|^2}{(\sum_{n=0}^{L-1} A[n])^2}\right) \tag{10}$$

We can see that the SNR loss is related to both the segment length L and the scaling function $A[n]$. We further show how to design non-stationary scaled down-chirps, and show that the SNR loss by a well-designed scaling function can be less than 1.7 dB in Sec. 4.2.

4.2 Designing Non-Stationary Scaling

NScale calculates the in-window distributions of collided symbols using non-stationary signal scaling. Therefore, the design of the scaling function impacts the decoding performance. We show the effect of the non-stationary scaling function on the decoding performance of NScale, and present strategies on designing an effective function.

The first rule for non-stationary scaling function is being monotonous. Through non-stationary amplitude scaling, the in-window distributions of chirps are translated to peak scaling factors in the



(a) (b)

Figure 9: Performance comparison of linear scaling functions with different slopes (i.e., change of the amplitude, starting at 1, over the whole symbol duration): (a) Averaged SER. (b) SNR Loss.

frequency domain (i.e., P_i in Eq.3). As shown in Figure 8(b), for a monotonous scaling function, the chirp segments with different in-window distributions can generate different peak heights, leading to different peak scaling factors. Otherwise, for a non-monotonous scaling function, symbols with different in-window time distributions may result in the same peak scaling factor, leading to ambiguity. We show an example in Figure 8(a) with a down-chirp of non-monotonous scaling function. The bottom of this figure is the relationship between peak scaling factors and in-window distributions, calculated according to Eq. 2. It is possible for two different in-window distributions to result in the same peak scaling factor. As shown in Figure 8(a), the points A and B in this figure represent two chirp segments of different in-window distributions. Due to the non-monotonicity of the scaling function, these two segments with distinct in-window distributions generate the same peak scaling factor, making it difficult to distinguish them.

The scaling function should also be linear. Here we describe why linear scaling functions are required for separating collided symbols. As illustrated in Sec. 3.5, NScale groups collided symbols into different packets according to the symbol in-window distribution, which is reflected by the peak scaling factor in Eq. 3. For accurate distribution calculation, the peak scaling factors for symbols of different in-window distributions should be different. Thus, the goal of our non-stationary scaling design is to maximize the difference in peak scaling factors of different in-window distributions. For two different in-window distributions i and j, we define the distance between their peak scaling factors as

$$d(i,j) = \frac{|P_i - P_j|}{\max\{\mathbb{P}\} - \min\{\mathbb{P}\}}$$

where P_i and P_j are the scaling factors for i and j, as shown in Figure 8(b); and $\mathbb{P}$ is a set of all possible peak scaling factors. For a demodulation window with N sample points, there are $2N$ possible in-window distributions corresponding to $2N$ different peak scaling factors, i.e., $|\mathbb{P}| = 2N$. All scaling factors are normalized to the range of $(0, 1]$ by dividing $\max\{\mathbb{P}\} - \min\{\mathbb{P}\}$.

Given a specific scaling function, the set of all possible peak scaling factors, i.e., $\mathbb{P}$, can be derived according to Eq. 2. Each object in $\mathbb{P}$ corresponds to a different in-window distribution. For object $i \in \mathbb{P}$, let

$$a(i) = \frac{1}{|\mathbb{P}| - 1} \sum_{j \in \mathbb{P}, j \neq i} |d(i,j)|^2$$

be the mean distance between i and all other objects. We can interpret $a(i)$ as a measure of how well i is separated from other in-window distributions (the bigger the value, the better the separation). Thus, the problem of designing non-stationary scaling is to finding a collection of $\mathbb{P}$ to maximize the mean inter-object distances, i.e.,

$$\mathbb{P}_{opt} = \arg\max_{\mathbb{P}} \frac{1}{|\mathbb{P}|} \sum_{i \in \mathbb{P}} a(i) \tag{11}$$

Given a monotonous scaling function and the number of j greater than i, as shown in Figure 8(b), we have $d(i,j) = \sum_{k=i}^{j-1} d(k,k+1)$. The mean inter-object distances in Eq.11 achieves maximum only when the distances between each pair of neighboring objects are identical, i.e., $d(k,k+1) = \frac{1}{|\mathbb{P}|}, \forall k \in \mathbb{P}$, indicating that the scaling function should be linear.

We finally evaluate the NScale's decoding performance under different linear scaling functions. We use the slope of a linear scaling function to represent the change of its amplitude (starting at 1) over the whole chirp duration. We use linear scaling functions with different slopes to decode the same set of two-packet collision. As shown in Figure 9(a), for collisions with small inter-packet offsets (< 10% symbol duration), the SER decreases as the slope increases. This is because scaling functions of large slope magnify the small time offsets. While for collisions with large time offsets (> 35% symbol duration), the slopes of scaling functions have less impact on the SERs, as the packets can already be distinguished. We also examine the SNR loss for scaling functions of different slopes. We vary the SNR of the received LoRa signal by manually adding white Gaussian noise, and evaluate the minimum decoding SNR requirement for both original LoRa and NScale with different linear scaling slope. The results of the experiment are shown in Figure 9(b). We can see from the results that NScale introduces less than 1.7 dB SNR loss compared with the original LoRa.

Summary. Linear function is a good design for the non-stationary scaling. The slope of the linear scaling function can be determined according to the inter-packet offset and SNR of the received collisions. Compared with the original LoRa, NScale introduces less than 1.7 dB SNR loss.

5 EVALUATION

We implement NScale on the software defined radios (SDRs) and evaluate its performance with commercial LoRa devices. The prototype of NScale is shown in Figure 10, which is composed of a USRP N210 along with a UBX daughter board, operating at the 470MHz bands. Decoding algorithms of NScale are hardware-independent, so it can be implemented on any other commercial LoRa gateways as long as the physical samples can be obtained. Note that LoRa gateways are usually deployed with tethered power supplies, and thus we do not consider energy consumption at the gateway. We use the UHD+GNU-Radio library [29] for developing our own LoRa demodulator, and implement NScale in MATLAB to process the PHY samples offline. By default, our experiment uses the spreading factor SF = 10, coding rate CR = 4/5 and bandwidth BW = 125 kHz. The sampling rate of NScale in our experiment is set to 1 MS/s.

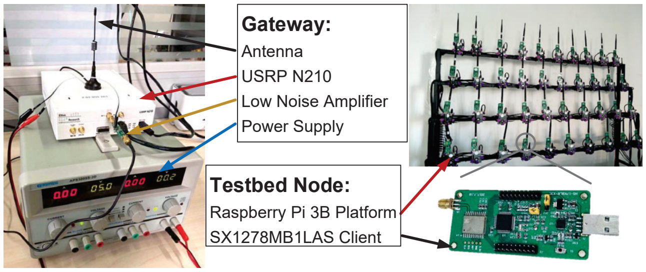


Figure 10: LoRa Gateway and Testbed Setup: Depicts NScale's USRP N210 based gateway and commodity client based LoRa testbed.

5.1 Evaluation Methodology

5.1.1 Experiment environments. We evaluated NScale's performance in both indoor and outdoor environments.

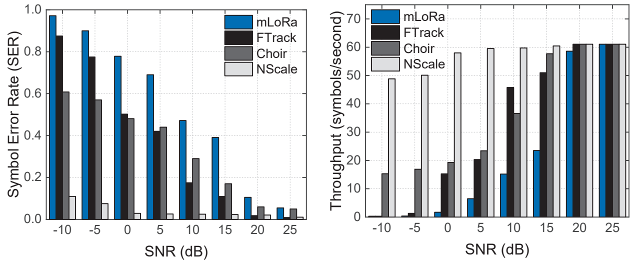
- As shown in Figure 10, the indoor testbed (*LoRaNet*) consists of 40 LoRa end nodes, each of which uses an SX1278 radio chip [30] and works at the frequency of 470 MHz. Each node is connected to a Raspberry Pi and placed at a fixed position on a shelf. All the Raspberry Pis are connected to a backbone network and thus all the LoRa nodes can be efficiently and accurately controlled to facilitate precise collision generation and measurement.
- The outdoor LoRa testbed is composed of enclosed nodes shown in Figure 15, each of which can sense the temperature and humidity of the environment and transmits the sensed data to the gateway via an SX1268 LoRa radio chip. The nodes of the outdoor testbed can harvest energy from solar power, making them easy to deploy on different locations such as roads, roofs, and parking lots.

5.1.2 Compared methods. We compare NScale with three recent works for LoRa collision decoding.

- Choir [14]: a collision decoding method for LoRa using hardware imperfection.
- FTrack [15]: a collision decoding method for LoRa based on time domain signal analysis.
- mLoRa [13]: a collision decoding method for LoRa based on SIC.

5.2 Comparing with Existing Works

We first compare NScale's performance with three existing works. Three LoRa nodes are used for generating LoRa collisions. We configure one node to send beacons every 3 seconds. Upon receiving a beacon, the other two nodes each replies with a LoRa packet, to generate collisions. We configure an additional processing delay (smaller than a packet duration) for each transmitter to generate different misalignment. Thus, packets from transmitters collide at different parts with different time offsets (e.g., preambles, sync words, SFDs and payloads). We vary the transmitting power of the transmitters to generate collisions with different SNRs. For fine-grained SNR control, we add white Gaussian noise with controlled amplitudes to the collected I and Q traces.



(a)

(b)

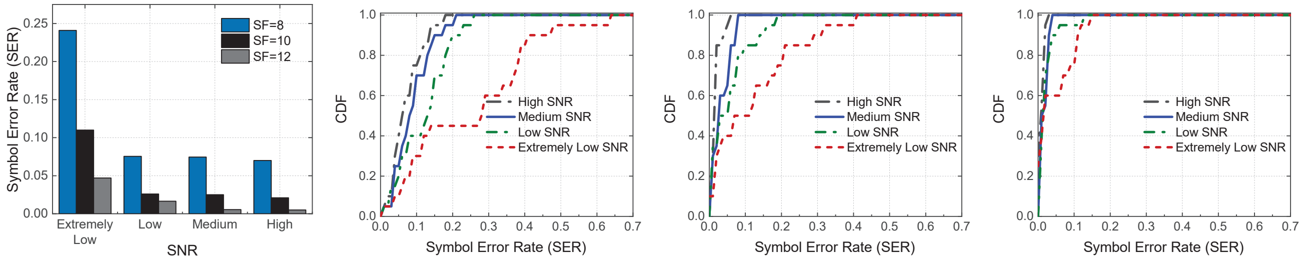
Figure 11: Performance comparison of four methods under different SNRs: (a) Averaged Symbol Error Rate (SER). (b) Network throughput.

We compare the performance of NScale with the other three LoRa demodulation schemes, in terms of SNR. Figure 11 shows the results of the experiment. When packets collide with a relatively high SNR (> 20 dB), both NScale and FTrack experience a low symbol error rate (SER $<$ 0.01) as well as a high network throughput. Choir and mLoRa fail to decode some of the concurrent transmissions even under such high SNR conditions. Choir uses the fraction of the FFT bin to distinguish collided symbols, which has errors due to the frequency offsets of low-cost LoRa nodes drift over time. Thus, collided symbols may be classified incorrectly, resulting in decoding errors. mLoRa uses an SIC based approach for decoding packet collisions, which suffers from error propagation. This leads to symbol recovering errors for mLoRa, especially when the packet length is long or the concurrency increases. As the SNR decreases, the SER of mLoRa and FTrack increase rapidly, because both of these two methods have fundamental limitations in decoding low SNR LoRa signals. For collisions under extremely low SNR (< -5 dB), both mLoRa and FTrack even turn to be invalid, resulting in a network throughput close to zero. The performance of Choir also decreases for low SNR situations, as the tiny hardware offsets are vulnerable to noise interference. NScale performs much better than the other three methods. When packets collide under extremely low SNR (-10 dB), the network throughput of NScale (49 symbols per second, sps) is about 3.3× of Choir (15 sps).

5.3 Basic Performance

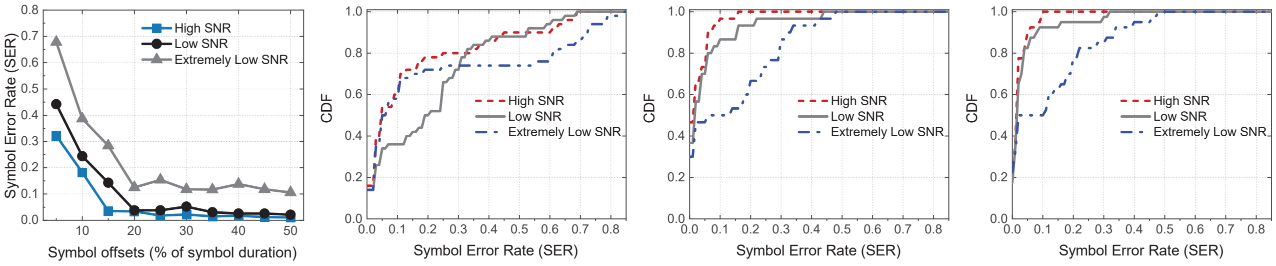
In this subsection, we examine NScale's basic performance for separating LoRa collisions regarding spreading factors (SFs), SNRs, and inter-packet offsets.

First, we evaluate the impact of SNR on the performance of NScale. The decoding performance is evaluate under four SNR regimes: high (>20 dB), medium (5~20 dB), low (-5~5 dB) and extremely low (< -5 dB). High channel noise cause peaks of chirp segments suffering from distortion, which further disturbs the detection of symbol in-window distribution. Figure 12 shows the SER for NScale under different levels of SNRs with different SFs. NScale can decode collisions even with extremely low SNR as it concentrates the energy of a chirp and translate the time domain information to robust peak features in the frequency domain. As



(a) Overall SER performance (b) CDF of SERs (SF=8) (c) CDF of SERs (SF=10) (d) CDF of SERs (SF=12)

Figure 12: In-depth study of SNR and SF on NScale's performance: (a) Overall performance of averaged SER. (b-d) CDF of the SER with different SNRs and SFs.



(a) Overall SER performance (b) Small offsets (c) Medium offsets (d) Large offsets

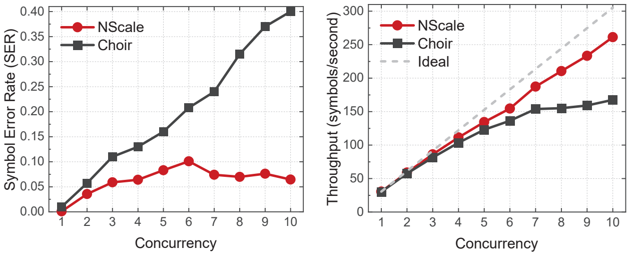
Figure 13: Relationship between the SER and symbol offsets of collisions: (a) Overall performance of averaged SER. (b-d) CDF of the SERs when symbol offset is small ($< 20\%$), medium ($20\% \sim 35\%$) and large ($> 35\%$), respectively.

shown in Figure 12(a), the SER is low under all four different SNR levels. The performance slightly degrades for extremely low SNR scenarios. However, those errors can mostly be recovered by the Forward Error Correction strategy of LoRa. Figure 12(b-d) shows the relationships between SERs and SNR levels regarding three different SFs. We observe that in high, medium and low SNR conditions, the SERs of 100% with high SF (SF12 and SF10) and 90% with small SF (SF8) are lower than 20%. This is because NScale reduces some of the noise interference, making its SER performance robust against channel noise. In the situation of extremely low SNR, the median SERs for SF8, SF10, and SF12 are 0.28, 0.07, and 0.02, respectively. In practice, small SFs in LoRa are used for near-range high data rate transmission. Therefore, we can increase SF for the scenario of low SNRs to improve the decoding performance to a very low SER of 0.02.

We also explore the impact of inter-symbol time offsets, i.e., packet misalignment. It has been shown that the NScale leverages the time offset information to separate collided packets. We examine how the inter-symbol time offsets affect NScale's performance. Figure 13(a) shows the averaged SER of NScale in terms of the inter-symbol time offsets and SNR levels. The SER performance of NScale decreases when packets collide with smaller inter-symbol time offsets. This is because a small offset leads to a small difference between in-window distributions, which further makes it difficult to distinguish collided packets. While in practice, nodes in LoRa transmit packets in random time, where the inter-symbol offset follows a uniform distribution within a symbol duration. Thus, the inter-symbol offsets vary in practice, and NScale can successfully separate collisions in most cases. We can further solve the decoding failures by using the retransmission mechanism of LoRaWAN protocol. Figure 13(b-d) present the detailed SER performance regarding SNRs and inter-symbol time offsets. For collisions with small offsets, decoding errors are mainly due to the ambiguity of in-window distributions. And the influence of the SNR is not that obvious, as high SNR collisions may also fail to decode due to ambiguous in-window distribution clustered incorrectly. The SER in Figure 13(b) is higher than that of medium offsets (Figure 13(c)) and the large offsets (Figure 13(d)) under all three SNR levels. While for collisions with median and large offsets, the median SERs of NScale for both high SNR and low SNR scenarios are below 0.02, indicating that most collided packets are correctly decoded.

5.4 Impact of Concurrency

In this experiment, we examine the scalability of NScale by decoding LoRa collisions with different number of concurrent transmissions. As mLoRa and FTrack cannot work for $SNR < 0$, we only show the performance of NScale and Choir. We use the indoor *LoRaNet* testbed to efficiently generate multi-packet collisions. To produce a collision with m overlapped packets, we use a beacon to synchronize transmissions for m different end nodes. At the gateway, we use



(a) (b)

Figure 14: Decoding collided transmissions with different concurrency. (a) Averaged SER. (b) Network throughput.

NScale and Choir to decompose the collided packets. The packets sent by each end node is known in prior. Thus, we can calculate the SER and network throughput in this experiment.

Figure 14(a) shows the SER for NScale and Choir. As the number of concurrent nodes increases from 1 to 10, the SERs of both NScale and Choir grow up. We can see that the SER of NScale increases much more slowly than that of Choir, because NScale extracts more efficient features to separate packets while Choir uses hardware imperfection, which is less stable and difficult to detect, especially under inter-chirp interference and channel noise. We further investigated the performance of NScale and found that a symbol error usually happens when the in-window distribution of a symbol is incorrectly detected or a symbol is incorrectly clustered to a packet.

We also show the overall network throughput in Figure 14(b). The network throughput of both NScale and Choir increase as the number of concurrent transmitters increases, due to the benefit of multi-packet reception from collision resolution. Meanwhile, the network throughput of Choir is much lower than that of NScale, especially for the large concurrency situations. That is because the hardware offsets in Choir inevitably resemble each other as the number of LoRa nodes increases, which leads to symbol clustering errors. We will further show the performance of NScale in the outdoor real deployed LoRa networks in Sec.5.5.

5.5 Performance in a Real Network

We evaluate the performance of NScale in a real deployed LoRa network. As illustrated in Figure 15, the outdoor LoRa network consists of 30 LoRa sensor nodes, each of which can sense the temperature and humidity information from the surrounding environment. We deploy the sensor nodes across various environments with different distance from the gateway, and the SNR of the received signal varies from −15dB to 10dB. With an integrated solar panel, each sensor node collects and transmits sensed data by harvesting solar power. The LoRa sensors transmit to the LoRa gateway in a duty-cycled manner, where we set the duty cycle ratio of each node to 10%. In the experiment, we change the number of active nodes in the network, and evaluate NScale's performance regarding to different network sizes.

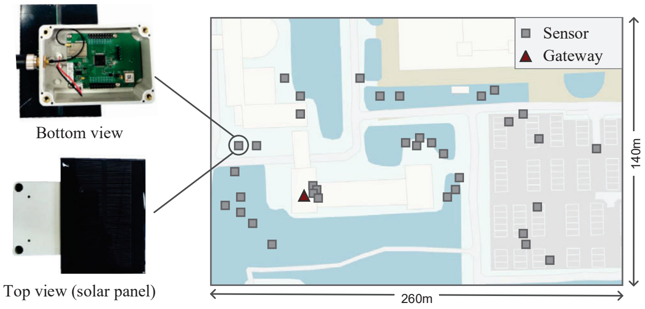
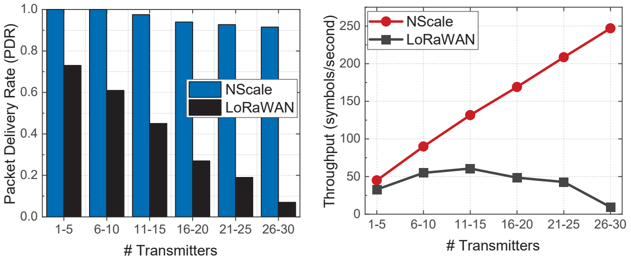


Figure 15: Outdoor Real LoRa Network: LoRa nodes with temperature and humidity sensors are placed around the campus, which consist of SX1268 radio chip and operate by harvesting solar power.



(a) (b)

Figure 16: Performance in real outdoor deployed LoRa networks: (a) Packet Delivery Rate. (b) Network throughput.

Figure 16 shows the performance of NScale and original LoRaWAN in the outdoor LoRa network. For the LoRaWAN receiver without collision resolution, the Packet Delivery Rate (PDR) decreases rapidly as 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 the channel utilization. However, when the network scales, frequent collisions occur, which significantly degrades the performance of the LoRaWAN receiver.

The performance of NScale is much better than the original LoRaWAN due to NScale's decoding advantage. As the network scales, NScale shows a relatively high PDR, where more than 90% packets are successfully delivered even when the size of the network reaches 30. The overall network throughput of NScale increases as the number of transmitters goes up. As shown in Figure 16(b), when 30 sensor nodes operate simultaneously, the network throughput of NScale (247 sps) is about 27× than that of the original LoRaWAN (9 sps).

6 RELATED WORK

Orthogonal chirp division multiplexing: The idea of orthogonal chirp division multiplexing has received much interest in radar and communication systems. Some FMCW radars use orthogonal chirp waveforms for simultaneously transmitting and receiving radar signals in multiple paths, thereby increasing the diversity or dimension of the information [31–37]. The radar systems proposed in [31–33] introduce the orthogonality of the transmitted signals by frequency multiplexing, where any two sources are separated by a carrier frequency offset. In [34] and [35], the chirp rate division is adopted for the orthogonality between simultaneous transmissions. In [36] and [37], the researchers designed multiple orthogonal chirp waveforms (including nonlinear chirps) to enable multi-user radar systems. Multi-user orthogonal chirp division is also exploited in communication systems for improving the spectral efficiency [38–42]. In [38] and [39], constraints are derived for assigning different users with different frequency-modulated rates, so that the cross-correlation between different users minimizes. The work [40] presents the orthogonal chirp spread spectrum (OCSS) based on the Fresnel transform and its convolution theorem. This work assumes perfect synchronization between all transmitters and receivers, and shows that OCSS outperforms the conventional OFDM by exhibiting higher resilience to inter-symbol interference. In [41] and [42], nonlinear time-frequency functions are designed to enable orthogonal multiple access with the assumption of synchronous or quasi-synchronous transmitters. The main feature that distinguishes NScale from those prior works is that NScale resolves chirp collisions without requiring any scheduling, power controlling, and synchronization assumptions.

Resolve collisions in traditional wireless: Combating signal collision is a traditional problem in wireless. There are many excellent works in this area. Some works aim to avoid collisions by using MIMO/MU-MIMO on wireless devices [43–46]. They synchronize signal phases from different transmitters, enabling concurrent transmissions without inter-packet interference. However, such solutions have high demands on hardware overheads, and thus is not appropriate for LPWAN devices. Successive Interference Cancellation (SIC) eliminates signal collisions by estimating and extracting decoded symbols iteratively [47–50]. ZigZag[51] combats signal collisions in 802.11, where the collision free chunk, due to the misalignment between collided packets, are leveraged for separating overlapped signals. For decoding an m-packet collision, ZigZag requires each end node retransmitting *m* times to generate *m* repeated collisions. Similar to ZigZag, mZig [52] also leverages the packet misalignment to decode collisions in ZigBee networks. mZig can decompose *m* concurrent ZigBee packets from one collision directly. Both ZigZag and mZig decode collisions based on the signal in the time domain, and they cannot work well for low SNR LoRa signal.

Parallel transmissions in LoRa: This work is inspired by some recent advances for concurrent transmission and collision resolution in LoRa. Netscatter [53] migrates LoRa encoding mechanism to backscatter devices and enables hundreds of concurrent transmissions for backscatter systems. Transmissions in Netscatter are strictly synchronized. Thus, we cannot apply Netscatter in current LoRa networks, where end nodes transmit to the gateway in the manner of ALOHA. DeepSense[54] enables random access and coexistence for LPWANs by identifying collided frames using neural networks. It can support carrier sense across different LPWAN protocols. However, in the emergence of packet collisions, DeepSense only identifies the existence of each frame, without recovering the encoded data bits.

The most related works to ours are Choir [14], mLoRa [13], and FTrack [15]. Choir [14] exploits the hardware imperfection of low-cost LoRa devices to decompose overlapped signals. However, as demonstrated in [53], this approach does not scale well as the tiny frequency offset is difficult to extract under low SNR. More recently, mLoRa [13] and FTrack [15] exploit the misaligned edges of LoRa symbols to separate collisions. mLoRa [13] uses a collision-free chunk to boot up the decoding, and iteratively reproduces and extracts collided symbols based on the known patten of LoRa chirps. FTrack [15] recovers collisions by detecting the edge of each LoRa symbol and then removing interference based on the continuity of each symbol's frequency. Both of the two approaches have fundamental limitations in processing low SNR signals.

7 CONCLUSION

We present NScale, a novel protocol to resolve the low SNR LoRa packet collisions, whereby NScale utilizes the subtle packet time offset to decompose multiple collided packets. To deal with collisions with extremely low SNR, NScale translates the packet time offset, which is vulnerable to noise, to more robust frequency features through non-stationary signal scaling. We propose several novel techniques to address practical challenges in NScale design. To accurately measure frequency peaks, we propose a noise resistant iterative peak recovery algorithm to combat peak distortion in low SNR LoRa signal. Further, we remove the impact of the CFO and symbol-window time offset to decode each separated packet. We implement NScale on USRP N210 and thoroughly evaluate its performance in both indoor and outdoor networks. NScale is completely implemented in software at the gateway, without requiring any modifications to the end nodes; thus we can apply NScale to current LoRa networks with little overhead. The evaluation results show that NScale improves the network throughput by **3.3×** for low SNR collided signals compared with other state-of-the-art methods.

ACKNOWLEDGEMENTS

We thank anonymous shepherd and reviewers for their helpful comments on the paper. This work is in part supported by National Key R&D Program of China 2018YFB1004800 , National Natural Science Fund for Excellent Young Scholars (No. 61722210), National Natural Science Foundation of China (No. 61932013, 61532012).

REFERENCES

[1] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873, January 2017.

[2] Nur Azmi Ali and Nurul Adilah Abdul Latiff. Environmental monitoring system based on lora technology in island. In *Proceedings of IEEE ICSigSys*, Bandung, Indonesia, July 16-18, 2019.

[3] Jithu G. Panicker, Mohamed Azman, and R. Kashyap. A lora wireless mesh network for wide-area animal tracking. In *Proceedings of IEEE ICECCT*, Tamil Nadu, India, February 20-22, 2019.

[4] 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*, New York, NY, USA, November 10-13, 2019.

[5] R. Jedermann, M. Borysov, N. Hartgenbusch, S. Jaeger, M. Sellwig, and W. Lang. Testing lora for food applications - example application for airflow measurements inside cooled warehouses with apples. *Procedia Manufacturing*, 24(5):284–289, February 2018.

[6] Ghena Branden, Adkins Joshua, Shangguan Longfei, Jamieson Kyle, Levis Phil, and Dutta Prabal. Challenge: Unlicensed lpwans are not yet the path to ubiquitous connectivity. In *Proceedings of ACM Mobicom*, Los Cabos, Mexico, October 21-25, 2019.

[7] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. Do lora low-power wide-area networks scale? In *Proceedings of ACM MSWiM*, Malta, Novenber 13-17, 2016.

[8] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao. A survey on lora networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials*, 22(1):371–388, October 2019.

[9] Jaco Morné Marais, Adnan M Abu-Mahfouz, and Gerhard P Hancke. A survey on the viability of confirmed traffic in a lorawan. *IEEE Access*, 8(1):9296–9311, January 2020.

[10] Jaco Morne Marais, Reza Malekian, and Adnan M Abu-Mahfouz. Evaluating the lorawan protocol using a permanent outdoor testbed. *IEEE Sensors Journal*, 19(12):4726–4733, February 2019.

[11] Shyamnath Gollakota, Samuel David Perli, and Dina Katabi. Interference alignment and cancellation. In *Proceedings of ACM SIGCOMM*, Barcelona, Spain, August 17-21, 2009.

[12] Mohsen Mollanoori and Majid Ghaderi. Uplink scheduling in wireless networks with successive interference cancellation. *IEEE Transactions on Mobile Computing*, 13(5):1132–1144, May 2013.

[13] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. mlora: A multi-packet reception protocol for lora communications. In *Proceedings of IEEE ICNP*, Chicago, Illinois, USA, October 7-10, 2019.

[14] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of ACM SIGCOMM*, Los Angeles, CA, USA, August 21-25, 2017.

[15] Xia Xianjin, Zheng Yuanqing, and Gu Tao. Ftrack: Parallel decoding for lora transmissions. In *Proceedings of ACM SenSys*, New York, NY, USA, November 10-13, 2019.

[16] Albert Berni and WO Gregg. On the utility of chirp modulation for digital signaling. *IEEE Transactions on Communications*, 21(6):748–751, June 1973.

[17] 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*, Hawaii, USA, September 11-15, 2017.

[18] 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*, 15(2):1–35, February 2019.

[19] 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 *Processings of ACM/IEEE IPSN*, Porto, Portugal, April 11-13, 2018.

[20] Aloys Augustin, Jiazi Yi, Thomas Clausen, and William Mark Townsley. A study of lora: Long range & low power networks for the internet of things. *MDPI Sensors*, 16(9):1–18, September 2016.

[21] Brecht Reynders and Sofie Pollin. Chirp spread spectrum as a modulation technique for long range communication. In *IEEE Symposium on Communications and Vehicular Technologies*, Mons, Belgium, November 22, 2016.

[22] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. Automated estimation of link quality for lora: A remote sensing approach. In *Proceedings of IEEE IPSN*, Montreal, Canada, April 16-18, 2019.

[23] 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*, Budapest, Hungary, August 20-25, 2018.

[24] Omid Abari, Deepak Vasisht, Dina Katabi, and Anantha Chandrakasan. Caraoke : An e-toll transponder network for smart cities. In *Proceedings of ACM SIGCOMM*, London, UK, August 17-21, 2015.

[25] Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, April 1928.

[26] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge Press, 2005.

[27] Tallal Elshabrawy and Joerg Robert. Closed-form approximation of lora modulation ber performance. *IEEE Communications Letters*, 22(9):1778–1781, June 2018.

[28] Dilip Roy. Discrete rayleigh distribution. *IEEE Transactions on Reliability*, 53(2):255–260, June 2004.

[29] GNU Free Software Foundation Project. Gnu radio. *Available: http://gnuradio.org*.

[30] Semtch. Sx1276/77/78/79 datasheet. *Available: https://www.semtech.com/*.

[31] Se-Yeon Jeon, Min-Ho Ka, Seungha Shin, Munsung Kim, Seok Nyeon Kim, Sunwook Kim, Jeongbae Kim, Aulia Dewantari, Jaeheung Kim, and Hansup Chung. W-band mimo fmcw radar system with simultaneous transmission of orthogonal waveforms for high-resolution imaging. *IEEE Transactions on Microwave Theory and Techniques*, 66(11):5051–5064, September 2018.

[32] Matthias Steinhauer, H.-O. Ruo, Hans Irion, and W. Menzel. Millimeter-wave-radar sensor based on a transceiver array for automotive applications. *IEEE Transactions on Microwave Theory and Techniques*, 56(2):261–269, February 2008.

[33] Junghyo Kim, Marwan Younis, Alberto Moreira, and Werner Wiesbeck. A novel ofdm chirp waveform scheme for use of multiple transmitters in sar. *IEEE Geoscience and Remote Sensing Letters*, 10(3):568–572, September 2013.

[34] Jos De Wit, W. L. van Rossum, and A. J. de Jong. Orthogonal waveforms for fmcw mimo radar. In *Proceedings of IEEE RadarConf*, Kansas City, MO, USA, May 23-27 2011.

[35] Y. Makino, T. Nozawa, M. Umehira, X. Wang, S. Takeda, and H. Kuroda. Inter-radar interference analysis of fmcw radars with different chirp rates. *The Journal of Engineering*, 2019(19):5634–5638, October 2019.

[36] Arijit Roy, Debasish Deb, Harshal B. Nemade, and Ratnajit Bhattacharjee. Design of discrete frequency-coding waveforms using phase-coded linear chirp for multiuser and mimo radar systems. Karnataka, India, February 20-23, 2019.

[37] Mohammad Omar Khyam, Li Xin-de, Shuzhi Sam Ge, and Mark R. Pickering. Multiple access chirp-based ultrasonic positioning. *IEEE Transactions on Instrumentation and Measurement*, 66(12):3126–3137, August 2017.

[38] Hao Shen and Antonia Papandreou-Suppappola. Diversity and channel estimation using time-varying signals and time-frequency techniques. *IEEE Transactions on Signal Processing*, 54(9):3400–3413, August 2006.

[39] Yingtuo Ju and Braham Barkat. A new efficient chirp modulation technique for multi-user access communications systems. In *Proceedings of IEEE ICASSP*, Quebec, Canada, May 17-21, 2004.

[40] Xing Ouyang and Jian Zhao. Orthogonal chirp division multiplexing. *IEEE Transactions on Communications*, 64(9):3946–3957, July 2016.

[41] Nozhan Hosseini and David W. Matolak. Nonlinear quasi-synchronous multi user chirp spread spectrum signaling. In *ArXiv*, 2019.

[42] Muhammad Ajmal Khan, Raveendra Rao, and Xianbin Wang. Non-linear trigonometric and hyperbolic chirps in multiuser spread spectrum communication systems. In *Proceedings of IEEE ICET*, Islamabad, Pakistan, December 9-10, 2013.

[43] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. In *Proceedings of ACM SIGCOMM*, Toronto, Canada, August 15-19, 2011.

[44] Ezzeldin Hamed, Hariharan Rahul, Mohammed A. Abdelghany, and Dina Katabi. Real-time distributed mimo systems. In *SIGCOMM*, Florianopolis, Brazil, August 22-26, 2016.

[45] Ezzeldin Hamed, Hariharan Rahul, and Bahar Partov. Chorus: truly distributed distributed-mimo. In *SIGCOMM*, Budapest, Hungary, August 20-25, 2018.

[46] Hariharan Rahul, Swarun Kumar, and Dina Katabi. Jmb: scaling wireless capacity with user demands. In *SIGCOMM*, Helsinki, Finland, August 13-17, 2012.

[47] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Successive interference cancellation: Carving out mac layer opportunities. *IEEE Transactions on Mobile Computing*, 12(2):346–357, January 2013.

[48] Daniel Halperin, Thomas Anderson, and David Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *Proceedings of ACM MobiCom*, San Francisco, CA, USA, September 14-19, 2008.

[49] Pulin Patel and Jack Holtzman. Analysis of a simple successive interference cancellation scheme in a ds/cdma system. *IEEE journal on selected areas in communications*, 12(5):796–807, June 1994.

[50] G. Xing, M. Sha, J. Huang, G. Zhou, X. Wang, and S. Liu. Multi-channel interference measurement and modeling in low-power wireless networks. In *Proceedings of IEEE RTSS*, Washington, D.C., USA, December 1-4, 2009.

[51] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. In *Proceedings of ACM SIGCOMM*, Seattle, WA, USA, August 17-22, 2008.

[52] Linghe Kong and Xue Liu. mzig: Enabling multi-packet reception in zigbee linghe. In *Proceedings of ACM MobiCom*, Paris, France, September 7-11, 2015.

[53] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. Netscatter: Enabling large-scale backscatter networks. In *Proceedings of USENIX NSDI*, Boston, MA, USA, February 26-28, 2019.

[54] Justin Chan, Anran Wang, Arvind Krishnamurthy, and Shyamnath Gollakota. Deepsense: Enabling carrier sense in low-power wide area networks using deep learning. In *ArXiv*, 2019.