ChirpTransformer: Versatile LoRa Encoding for Low-power Wide-area IoT

Chenning Li[▶1], Yidong Ren[▶1], Shuai Tong[2], Shakhrul Iman Siam[3]
Mi Zhang[3], Yunhao Liu[2], Jiliang Wang[2], Zhichao Cao[1*]
▶ Co-primary Authors
[1]Michigan State University [2]Tsinghua University [3]The Ohio State University

ABSTRACT

This paper introduces ChirpTransformer, a versatile LoRa encoding framework that harnesses broad chirp features to dynamically modulate data, enhancing network coverage, throughput, and energy efficiency. Unlike the standard LoRa encoder that offers only single configurable chirp feature, our framework introduces four distinct chirp features, expanding the spectrum of methods available for data modulation. To implement these features on commercial off-the-shelf (COTS) LoRa nodes, we utilize a combination of a software design and a hardware interrupt. ChirpTransformer serves as the foundation for optimizing encoding and decoding in three specific case studies: weak signal decoding for extended network coverage, concurrent transmission for heightened network throughput, and data rate adaptation for improved network energy efficiency. Each case study involves the development of an end-to-end system to comprehensively evaluate its performance. The evaluation results demonstrate remarkable enhancements compared to the standard LoRa. Specifically, ChirpTransformer achieves a 2.38 × increase in network coverage, a 3.14 × boost in network throughput, and a 3.93 × of battery lifetime.

CCS CONCEPTS

• **Networks** → **Network reliability**; **Wide area networks**; **Physical links**.

KEYWORDS

LoRa, Network Encoding, IoT

ACM Reference Format:
Chenning Li[▶1], Yidong Ren[▶1], Shuai Tong[2], Shakhrul Iman Siam[3], Mi Zhang[3], Yunhao Liu[2], Jiliang Wang[2], Zhichao Cao[1]. 2024. ChirpTransformer: Versatile LoRa Encoding for Low-power Wide-area IoT. In *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3643832.3661861

1 INTRODUCTION

Long Range (LoRa) technology has emerged as a highly promising solution for seamlessly connecting unattended Internet-of-Things (IoT) devices on a large scale [19, 39]. As of March 2023, the global presence of LoRa networks has expanded significantly, with 181 public operators worldwide facilitating connectivity for over 300 million IoT devices [2]. This wide-area IoT technology has found application in a multitude of scenarios. For instance, Amazon Sidewalk [1] employs LoRa to bridge connections with smart IoT devices located beyond the range of conventional home Wi-Fi networks. In the agricultural sphere, Microsoft FarmBeats [3, 7] utilizes LoRa to efficiently gather data from remote sensors deployed across vast farmlands. Additionally, LoSee [24, 52, 53] leverages LoRa to track shared bicycles within urban environments.

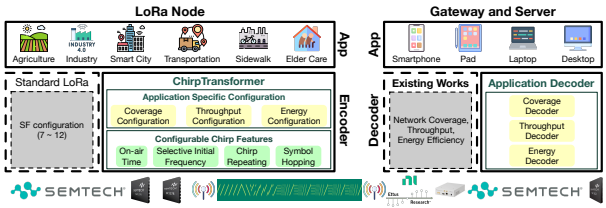


Figure 1: An illustration of the ChirpTransformer encoding framework in LoRa systems.

Figure 1 shows the network architecture underlying LoRa systems for IoT data collection [19]. LoRa nodes are responsible for encoding sensory data and transmitting data packets to LoRa gateways. These gateways serve as crucial intermediaries; upon receiving the packets, they decode the data and subsequently relay it to network and application servers through backhaul networks. It is important to note that LoRa nodes, equipped with low-cost commercial-off-the-shelf (COTS) Semtech LoRa radios [30, 35], operate under constrained resources. In stark contrast, LoRa gateways and servers utilize potent COTS Semtech radio [31, 32], some even employing Software Defined Radio (SDR) [29], to manage energy-intensive and computationally demanding tasks. Furthermore, distinct LoRa applications exhibit diverse performance considerations. For example, agricultural and urban deployments prioritize ubiquitous coverage [7, 28, 42, 50, 53], whereas industrial dense deployments contend with potential packet collisions necessitating scalable throughput [17, 22, 36]; In mobile and semi-outdoor applications (e.g., transportation, elder-care), the dynamic environmental conditions can swiftly deplete the energy of LoRa nodes, underscoring the need for sustainable adaptation strategies [13, 25, 49]. Consequently, existing works predominantly focuses on optimizing network coverage [6, 8, 10, 13, 18, 21, 40, 45, 55], throughput [11, 36, 41, 43, 44, 46, 47, 54], and energy efficiency [16, 23, 25, 38, 51] at the gateway and server side, often leaving LoRa nodes untouched due to their resource limitations.

However, the efficacy of decoding designs in these systems heavily relies on how LoRa nodes encode data. LoRa employs Chirp

*Zhichao Cao is the corresponding author.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
MOBISYS '24, June 3–7, 2024, Minato-ku, Tokyo, Japan
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0581-6/24/06…$15.00
https://doi.org/10.1145/3643832.3661861

Spread Spectrum (CSS) modulation, where data is encoded by linear chirps. The standard LoRa encoder uses a configurable parameter known as the spreading factor (SF) to determine the data rate of a chirp symbol. Notably, all chirp symbols within a LoRa packet adhere to the same SF setting. With COTS LoRa nodes offering only six available SFs ranging from 7 to 12 for configuring LoRa encoding, the standard LoRa encoder's reliance solely on packet-level SF configuration presents limitations in optimizing end-to-end performance across diverse application scenarios concerning network coverage, throughput, and energy efficiency.

In this work, we present an innovative approach to LoRa encoder design aimed at enhancing the versatility and overall performance of LoRa networks. Our proposed solution, ChirpTransformer, reimagines the conventional LoRa encoding framework by introducing a set of four configurable chirp features. These features enable a more extensive range of encoding options at LoRa nodes, as depicted in Figure 1. Through the utilization of these configurable parameters, a diverse array of encoding techniques can be devised and tailored to meet the specific coverage, throughput, and energy efficiency requirements of diverse applications. Furthermore, our approach extends beyond node-side enhancements; we have developed corresponding decoding methodologies for gateways and servers. These decoding methods leverage the pre-configured chirp features implemented by the encoders, aiming to optimize the end-to-end performance of the LoRa network comprehensively.

Challenge #1: Chirp Feature Design. Our objective is to construct a set of specific chirp features that collectively create an informative feature space. However, the pursuit of achieving this goal does not involve indiscriminately adding numerous chirp features, as this would unnecessarily complicate the LoRa network stack. To tackle this challenge, ChirpTransformer goes beyond solely incorporating time and frequency domain data; it strategically designs four distinct chirp features aimed at enhancing the informativeness of the feature space. First, ChirpTransformer controls the on-air time of a symbol, which encompasses one or more chirps. Second, ChirpTransformer selects a collection of initial frequencies that a chirp can employ to encode data. Third, ChirpTransformer involves a novel intra-symbol chirp pattern. Departing from the conventional approach of comprising a symbol with just one chirp, a ChirpTransformer symbol may consist of repeated chirps sharing the same initial frequency offset. Fourth, ChirpTransformer incorporates an innovative inter-symbol chirp pattern. In contrast to using identical types of chirps, ChirpTransformer employs diverse chirp patterns across different symbols within a LoRa packet.

Challenge #2: Implementation on COTS LoRa nodes. Numerous chirp features, such as non-linearity [20, 22, 27] and interleaving [12], pose implementation challenges on COTS LoRa nodes, necessitating costly node replacements, especially in large-scale deployments. To circumvent this issue, the generation of these designed chirp features must not require hardware modification or additional energy consumption. To address this challenge, we have devised a lightweight symbol converter that transforms the chirp features – such as on-air time, selective initial frequency, and chirp repeating – into corresponding chirps configured for SF. Furthermore, we leverage an existing hardware interrupt specifically designed for frequency hopping to incorporate our symbol-hopping chirp feature without incurring extra overhead.

Challenge #3: Application-specific Encoder-Decoder Co-design. With our suite of four chirp features, we unlock the potential for multiple encoder configurations to enhance application-specific performance. To maximize these performance gains, our focus lies in optimizing performance through an end-to-end approach, encompassing encoder configuration and decoder co-design. To address this challenge, we have undertaken three case studies targeting network coverage, throughput, and energy efficiency:

1) *Network Coverage:* Addressing weak signals is crucial [21, 53]. The standard LoRa encoder extends on-air time to ensure adequate energy reception, countering noise interference. As the SF increases, the required signal-to-noise-ratio (SNR) for successful decoding decreases. ChirpTransformer takes a different approach, enhancing noise tolerance by widening the feature distance between adjacent symbols and leveraging a neural-enhanced decoder to maximize SNR gain (§4).

2) *Network Throughput:* Enabling concurrent transmission [11, 22, 36] is a primary concern. While LoRa permits simultaneous transmissions with different SFs, the varying on-air times across these SFs result in imbalanced SNR tolerance among concurrent transmitters. ChirpTransformer utilizes intra-symbol chirp repeating to create novel orthogonal encoding configurations and employs template-based decoding to resolve collisions (§5).

3) *Network Energy Efficiency:* Fine-tuning encoding configurations for optimized data rates under various noise levels is crucial [23, 25]. The standard LoRa encoder supports only six data rates, whereas ChirpTransformer achieves 23 configurations by adjusting the number of available initial frequencies using a selective peak searching algorithm for decoding. (§6).

We have implemented end-to-end ChirpTransformer systems for each case study, utilizing COTS LoRa nodes and a USRP N210 SDR. Using our expansive campus-scale testbed spanning 2800 m × 1700 m, we have conducted extensive experiments to evaluate ChirpTransformer's performance against both standard LoRa and a state-of-the-art benchmark. The results confirm ChirpTransformer's superiority across all three case studies.

In summary, our contributions are listed as follows:

- We propose ChirpTransformer, an innovative LoRa encoding framework incorporating four chirp features. This framework significantly enhances encoding methods, enabling efficient adaptation to diverse applications.
- ChirpTransformer seamlessly integrates with COTS LoRa nodes without introducing additional overhead. Through three comprehensive case studies, we showcase its ability to optimize end-to-end network performance by adopting an encoder-decoder co-design.
- We implement ChirpTransformer on COTS LoRa nodes and conduct extensive experiments using a campus-scale testbed. The results demonstrate ChirpTransformer's exceptional performance, achieving a 2.38 × increase in network coverage, a 3.14 × improvement in network throughput, and a 3.93 × battery lifetime compared to the standard LoRa.

2 BACKGROUND AND MOTIVATION

2.1 Standard Encoding and Decoding

Figure 2 illustrates the encoding and decoding method of the standard LoRa. As shown in Figure 2(a), the standard LoRa encoder

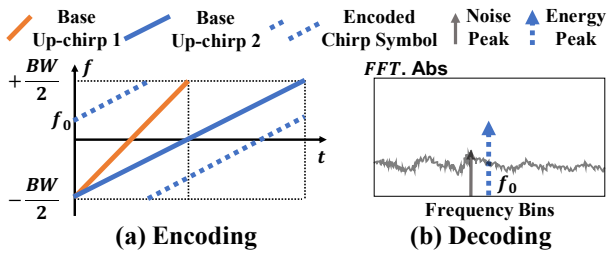


Figure 2: The illustration of (a) shifted initial frequency encoding and (b) energy peak detection decoding.

uses bandwidth (BW) to configure a base up-chirp (e.g., the orange chirp), whose frequency increases linearly from $-\frac{BW}{2}$ to $\frac{BW}{2}$ over time. Notably, the on-air time of a base up-chirp is adjustable. For example, the blue base up-chirp 2 spends more time swiping the whole BW than the orange base up-chirp 1. LoRa uses SF configuration, ranging from 7 to 12, to control the on-air time. Given the pre-configured BW and SF, the on-air time is $\frac{2^{SF}}{BW}$. Thus, the on-air time is doubled when SF increases by 1.

Given the base up-chirp, data bits are encoded by shifting the initial frequency of a base up-chirp to f_0. For example, in Figure 2(a), we shift the initial frequency of the blue base up-chirp 2 to f_0. When the chirp frequency meets BW/2, the rest of the chirp will be shifted down and restarts from −BW/2. As a result, the blue dashed chirp forms a typical chirp symbol. Moreover, The range of f_0 is $[0, BW)$. The number of data bits encoded by a chirp symbol is also determined by the frequency chip, indicating the distance between two adjacent initial frequency offsets. On COTS LoRa nodes, the frequency chip is $\frac{BW}{2^{SF}}$. Thus, in $[0, BW)$, we have 2^{SF} initial frequencies to represent data. Given an SF configuration, a chirp symbol encodes SF data bits.

At the decoder side, the dechirp [11, 21] is the standard decoding process in LoRa. First, a received chirp symbol is multiplied with a base down-chirp (i.e., the conjugate of a base up-chirp). The Fast Fourier Transform (FFT) is then used to focus on the energy of the chirp symbol at a single tone, which corresponds to the initial frequency offset of the chirp symbol, on the spectrum [11, 41]. For example, in Figure 2(b), the blue dashed arrow indicates the energy peak by applying FFT on the encoded chirp symbol shown in Figure 2(a). Then, we find the frequency bin where the highest spectral energy peak appears to determine the initial frequency offset f_0. Once we obtain the initial frequency offset, data bits can be decoded. On the other hand, to decode a received chirp symbol successfully, the energy peak of the received chirp symbol should be higher than the highest energy peak derived from noises on the spectrum. For example, in Figure 2(b), the blue dashed arrow is higher than the grey arrow, indicating the noise peak, to guarantee f_0 can be found correctly.

2.2 SF-configured LoRa Performance

We can see that the standard LoRa encoder is purely controlled by SF configuration. We have six SF configurations, ranging from 7 to 12, on COTS LoRa nodes [30, 35]. In a LoRa packet, all chirp symbols employ the same SF configuration for simplicity. As such, LoRa's performance highly depends on SF configuration. We analyze the influence of SF configurations for SNR tolerance and data rate as examples.

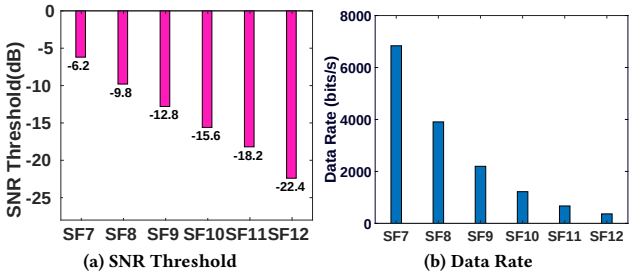


Figure 3: The influence of SF configurations on (a) SNR tolerance and (b) data rate.

SNR Tolerance: SNR tolerance indicates the SNR threshold, above which a chirp symbol can be successfully decoded. SNR tolerance determines the capability of network coverage. As the on-air time of a chirp symbol increases, more energy can be combined to form a higher energy peak on the spectrum, making it easy to detect the energy peak under strong noises. Thus, a chirp symbol with larger SFs can achieve better SNR tolerance. We set BW as 125 kHz and empirically measure the required SNR of the dechirp under different SFs with a synthesis dataset [9, 21]. Figure 3a shows the SNR threshold of different SF configurations, taking 1% symbol error rate (SER) as the criteria of successful decoding. The minimum SNR threshold reaches -22.4 dB.

Data Rate: Data rate indicates how many data bits can be successfully transmitted per second without collision. Given a certain SNR level, a higher data rate leads to higher energy efficiency. Given the settings of SF and BW, a chirp symbol encodes SF data bits, and its on-air time is $\frac{2^{SF}}{BW}$. Thus, the data rate is $\frac{SF \cdot BW}{2^{SF}}$. A chirp symbol with a larger SF has a longer on-air time, lowering its data rate. When BW is set as 125 kHz, the data rate under different SFs is shown in Figure 3b. We can see that the data rate is reduced by 42.9%-45.7% with an SF increment. As a tradeoff, Figure 3a shows that increasing SF by one can achieve a 2.6-4.2 dB gain of the SNR threshold.

2.3 Motivation

The purely SF-controlled LoRa encoder is simple, providing SNR tolerance as low as -22.4 dB and six data rate options under six SNR thresholds. However, with SF-12 configuration, the LoSee measurement study [53] has shown that although the longest communication range can reach 3.2 km - 3.5 km, a gateway can only cover about an irregular 11 km^2 - 12 km^2 area in an urban environment, which is far from needed to achieve ubiquitous wide-area coverage. On the other hand, in those environments with dynamic link budgets [25, 49], only six SF options are too coarse-grained to achieve sustainable energy efficiency. This motivates us to rethink the LoRa encoder design, particularly on COTS LoRa nodes, for inherently supporting various LoRaWAN applications.

3 DESIGN OF CHIRPTRANSFORMER

3.1 Chirp Feature Design

Figure 4 illustrates the four chirp features, representing the information from four domains, that are used to define a symbol in ChirpTransformer.

First, we define a time-domain feature, on-air time, which defines the propagation time of a symbol. A symbol contains one or multiple

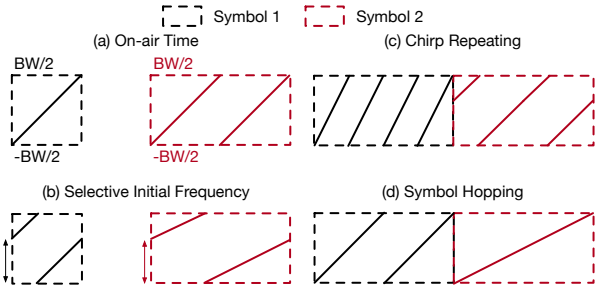


Figure 4: The illustration of the four chirp features in ChirpTransformer.

chirps, and the total propagation time is its on-air time. As shown in Figure 4(a), Symbol 1 contains one base up-chirp, and Symbol 2 has longer on-air time with two base up-chirps.

Second, we select a set of frequencies in the range $[0, BW)$ to define the available initial frequency offsets. In Figure 4(b), an initial frequency offset indicates the shape of a chirp compared to the base up-chirp. Therefore, the selective initial frequencies are frequency-domain features to determine the shapes of those chirps used to encode data.

Third, given the on-air time of a symbol, we design an intra-symbol chirp pattern, called *Chirp Repeating*, to depict the repeated identical chirps in a symbol. As shown in Figure 4(c), Symbol 1 consists of four repetitive base up-chirps, and Symbol 2 contains two repetitive chirps with the same initial frequency offsets. The chirp repetition is a time-domain pattern, while the same initial frequency offset of those chirps reflects the frequency characteristic.

Lastly, given a fixed on-air time for all symbols in a packet, we design an inter-symbol chirp operation, called *Symbol Hopping*, to create a new pattern domain for encoding. The basic idea is to hop different chirp repeating patterns among the symbols in a packet instead of using the same configuration. As shown in Figure 4(d), Assume Symbol 1 and Symbol 2 are in the same packet. Symbol 1 has two repetitive base up-chirps, but Symbol 2 revises the chirp repeating pattern with only one base up-chirp. Based on the four chirp features, we abstract four parameters to configure encoders versatilely: 1) on-air time (OT), 2) available initial frequency offsets (IFO), 3) chirp repeating times (CRT) in a symbol, and 4) the number of available chirp patterns in symbol hopping (SH). ChirpTransformer adjusts the four-configuration-tuple (OT, IFO, CRT, SH) to cope with a specific performance demand.

3.2 Chirp Feature Configuration

On COTS LoRa nodes, we have six types of SF-configured chirps. Based on these chirps, we illustrate the supported configurations of our four chirp features.

OT indicates the on-air time of a symbol. On COTS LoRa nodes, the on-air time of a chirp is determined by six SFs from 7 to 12. Its on-air time is $\frac{2^{SF}}{BW}$, where BW is the bandwidth. The minimum on-air time is with SF-7. Similar to the way of using SF to determine on-air time, our OT is in the range of $[7, +\infty)$. When OT is k, the on-air time of a symbol is $\frac{2^k}{BW}$. In ChirpTransformer, the on-air time of a symbol could be extended flexibly. Since the maximum SF is 12, a symbol has to contain multiple chirps when OT is larger than 12.

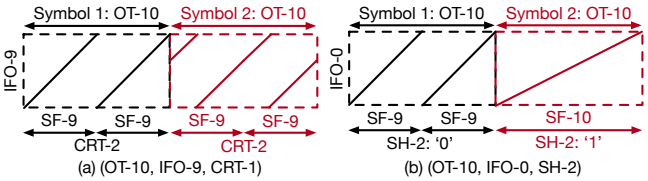


Figure 5: Two examples to illustrate the concepts of CRT and SH. (a) parameter tuple (OT-10, IFO-9, CRT-2); (b) parameter tuple (OT-10, IFO-1, SH-2).

IFO indicates the available initial frequency offsets for encoding. In SF-configured encoding, an SF-k chirp is used to encode k data bits. In ChirpTransformer, given OT-k chirps, the value of IFO can be 2^i, where i is in the range of $[1, k]$. Thus, the configuration (OT-k, IFO-i) means that we use OT-k chirps to encode i data bits by shrinking the available initial frequency offsets from 2^k to 2^i. For example, the configuration (OT-7, IFO-6) indicates using OT-7 chirps to encode 6 data bits. Half of the initial frequency offsets in OT-7 chirps will no longer be used for encoding. When OT is larger than 12, the maximum IFO can be 2^{12} on COTS LoRa nodes. In such cases, the IFO values are from IFO-1 to IFO-12.

CRT indicates the number of repetitive identical chirps in a symbol. We know that the on-air time of an OT-10 symbol equals eight SF-7 chirps, four SF-8 chirps, two SF-9 chirps, and one SF-10 chirp. Similarly, when the on-air time of a symbol is OT-k, we use CRT-j repetitive SF-$(k-j)$ chirps with the same initial frequency to fill the symbol, where j is in the range of $[\max\{k-12, 0\}, k-7]$. For example, as shown in Figure 5(a), (OT-10, IFO-9, CRT-1) means the on-air time of each symbol is OT-10. Each symbol consists of two (i.e., 2^1) identical SF-9 (i.e., $10-1$) chirps, which have 2^9 available initial frequency offsets for encoding.

SH represents the available chirp repeating patterns of symbol hopping. SH has three values: 0, 2, and 4. If SH is 0, the symbols in a packet follow the same pattern without symbol hopping. The data bits encoded by a symbol purely rely on the IFO settings. If SH is 2 or 4, we have 2 or 4 different patterns. We can use the different patterns to encode 1 data bit or 2 data bits. When SH is not 0, CRT will be invalid. Given the on-air time of a symbol OT-k ($k \leq 12$) and the available chirp patterns SH-t, the t chirp repeating patterns of a symbol can be represented by one SF-k base up-chirp, or two SF-$(k-1)$ repetitive base up-chirps, ..., or 2^t SF-$(k-t)$ repetitive base up-chirps. We must keep $k - t \geq 7$. And when k is larger than 12, the SH-t chirp repeating patterns include 2^{k-12} SF-12 base up-chirps, 2^{k-11} SF-12 base up-chirps, ..., or 2^{k-13+t} SF-$(13-t)$ base up-chirps. For example, as shown in Figure 5(b), (OT-10, IFO-0, SH-2) means the on-air time of a symbol is OT-10. IFO-0 indicates we do not use initial frequency offsets to encode data. In this case, with SH-2, we have two symbols. One is two repetitive SF-9 base up-chirps, indicating bit '0'. The other is one SF-10 base up-chirp, indicating bit '1'.

3.3 COTS Feature Implementation

A ChirpTransformer packet consists of four parts: preamble, start frame delimiter (SFD), header, and payload. The format of the preamble, SFD, and header are the same as the standard LoRa. We put the values of (OT, IFO, CRT, SH) in the header. The symbols in the payload are encoded by the encoding method determined by (OT,

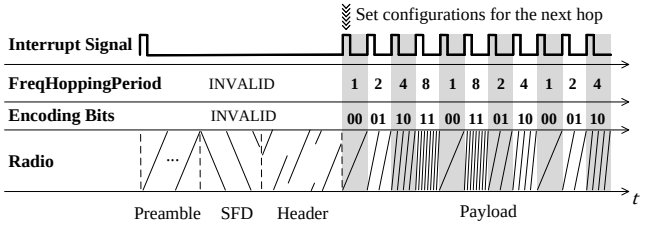


Figure 6: Timeline of the symbol hopping feature, leveraging a hardware interrupt.

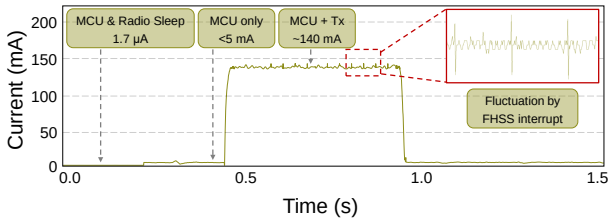


Figure 7: Current profile for a ChirpTransformer transmission with symbol hopping.

IFO, CRT, SH). By default, COTS LoRa nodes support SF-configured packets in which all chirp symbols follow the same SF configuration. However, if symbol hopping is enabled, a ChirpTransformer packet consists of symbols with different SF chirps. Thus, without symbol hopping, we design a symbol converter to translate the payload of a ChirpTransformer packet to SF-configured chirp symbols. On the other hand, we leverage a hardware interrupt to implement symbol hopping.

Given OT-k, IFO-i, and CRT-j configurations, the symbol converter includes three steps. In the first step, it determines the corresponding SF configuration. If only one chirp exists in a symbol, the SF configuration corresponds to OT-k. Otherwise, the symbol contains a chirp repeating pattern CRT-t. Then, the SF configuration is $k - t$ corresponding to the SF of each repetitive chirp. In step two, we calculate the number of SF chirps to form a symbol. If CRT is not applied, only one SF chirp exists. For CRT-j, the number corresponds to 2^j. In the last step, we set the initial frequency offset for each SF chirp. Since IFO-i must be a subset of IFO-SF, we directly assign the initial frequency offset of each chirp in a ChirpTransformer symbol to the corresponding chirp generated in step two. In this way, a ChirpTransformer symbol can be translated to a series of SF-configured chirp symbols.

We implement the symbol hopping feature based on the frequency hopping capability of COTS LoRa nodes [30]. The LoRa standard requires LoRa nodes to support frequency hopping, enabling long-duration packet transmission without violating the maximum permissible channel dwell time. The key principle behind the frequency hopping scheme is hardware interrupt, named ChangeChannelFhss, that enables LoRa nodes to select and switch to a new channel during packet transmission. After a predetermined hopping period, the transmitter and receiver change to the next channel in a predefined list of hopping frequencies to continue transmission and reception of the next portion of the packet. Our key observation is that a LoRa node can modify not just the channel but all configurations every time it triggers the ChangeChannelFhss interrupt. Thus, we can implement symbol hopping on a COTS LoRa node without adding extra hardware by making the node periodically trigger the ChangeChannelFhss interrupt and change its SF configuration during packet transmission. Moreover, we keep the transmission on a single channel by setting the list of hopping frequencies as identical frequencies.

For example, we use (OT, IFO-0, SH-4) to encode 2-bit data with four different chirp repeating patterns. Figure 6 shows the timeline of the encoder for transmitting a packet. The transmission starts with the preamble, SFD, and header. At the beginning of each transmission, an interrupt signal ChangeChannelFhss is generated, where the interrupt handler programs the frequency, SF, and hopping period for the first hop of the payload. The interrupt signal is cleared after all configurations have been settled. Then, during payload transmission, the transmitter periodically triggers the ChangeChannelFhss interrupt to modify the SF configuration for ChirpTransformer data modulation. We use our symbol converter to determine the SF for each chirp repeating pattern. The time that each hop of transmission will dwell is determined by FreqHoppingPeriod, which is an integer multiple of symbol periods. As illustrated in data modulation, the periods of all symbols should be identical to the OT configuration. Thus, we determine the FreqHoppingPeriod based on the SF configuration of chirps in the corresponding hop, i.e., $FreqHoppingPeriod = 2^{OT-SF}$. The new configurations are programmed within the current hopping period to ensure it has been set when the next hop begins. The interrupt computation is much shorter than the symbol on-air time, causing no extra latency.

3.4 Energy Profiling

To demonstrate the energy overhead brought by the hardware interrupt processing, we estimate the energy profile of a ChirpTransformer transmission with symbol hopping using the Monsoon HV Power Monitor [4]. The node is powered by 3.6 V. Figure 7 shows the current profile of the transmission, including the instant current for all radio access phases. The LoRa node stays in sleep mode when it does not transmit data. The radio transmission consumes the highest amount of energy by a large margin. The power consumption of the MCU is much less than that of the radio circuit. Thus, the ChangeChannelFhss interrupt processed by the MCU only introduces a small current fluctuation during the packet transmission, which has a neglectable impact on the whole energy consumption of the LoRa node.

4 CASE I: NETWORK COVERAGE

In the first case study, we focus on reliable weak signal decoding to extend LoRaWAN coverage [21]. We are seeking a design to tolerate the SNR lower than the -22.4 dB SNR threshold of the standard LoRa (§ 2.2). With ChirpTransformer, the key idea is to maximize the feature distance among different symbols during encoding beyond SF-12 with SH configurations.

4.1 SH Encoder-Decoder Co-design

Encoding Method: Our design utilizes the chirp pattern domain created by symbol hopping and prohibits using initial frequency offsets (i.e., IFO-0) for data encoding. We use four chirp patterns to represent each 2-bit data (i.e., SH-4). To obtain four chirp repeating patterns, the minimum OT configuration is OT-10 so that we have the SF-7, SF-8, SF-9, and SF-10 chirps to construct four different

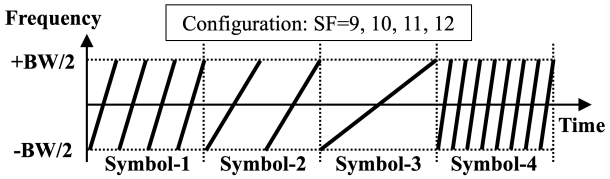


Figure 8: The illustration of symbol hopping encoding, using four chirp repeating patterns (e.g., SF-9 – SF-12) to represent 2-bit data.

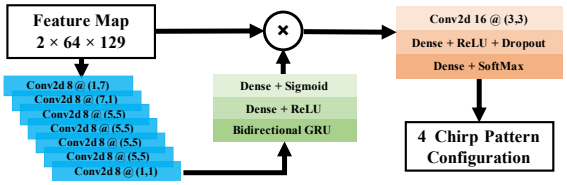


Figure 9: The structure of the neural-enhanced decoder.

patterns. When OT increases, the SNR tolerance will be enhanced while the data rate reduces. Figure 8 shows an example of (OT-12, IFO-0, SH-4), which uses four chirp repeating patterns of SF-9 – SF-12 to define 4 symbols. We use SH-[9-12], SH-[8-11], and SH-[7-10] to represent (OT-12, IFO-0, SH-4), (OT-11, IFO-0, SH-4), and (OT-10, IFO-0, SH-4), separately.

Neural-enhanced Decoding Method: Our decoding problem is to extract a symbol's chirp patterns (i.e., 4-class classification). Inspired by the neural-enhanced decoder in NELoRa [21], ChirpTransformer also aims to decode the symbols in a neural-enhanced manner to maximize the SNR gain from both the encoder and decoder side. Like NELoRa, our neural-enhanced decoder converts symbols into time-frequency spectrograms as input feature maps. We simplify NELoRa's architecture by using a lightweight first Conv2d module with fewer filters and replacing the LSTM layer with a bidirectional GRU layer, which is more computation-efficient for temporal feature extraction. Figure 9 shows the concise network structure, which consists of seven modules in total. Specifically, the first four modules aim to generate a filter mask to be multiplied with the input spectrogram. Then, we feed the masked spectrogram into a three-module classifier for chirp pattern recognition.

4.2 IFO-2 based Encoder-Decoder Co-design

Encoding Method: With the similar design principle of our symbol hopping encoder, we can set the available initial frequency offsets as 4 (i.e., IFO-2) to encode 2-bit data. Specifically, as shown in Figure 10, we select four SF-12 chirps with deterministic initial frequency offsets (e.g., f_0, f_1, f_2, f_3) to encode 2-bit data. We set the initial frequency offsets as $-\frac{BW}{2}$, $-\frac{BW}{4}$, 0 and $\frac{BW}{4}$ to distinguish these symbols as much as possible in the feature space. To keep a similar data rate to the symbol hopping encoding, IFO based encoding has three configurations, namely IFO-2-10, IFO-2-11, and IFO-2-12, which use SF-10, SF-11, and SF-12 chirps to form symbols matching SH-[7-10], SH-[8-11], and SH-[9-12] symbols.

Neural-enhanced Decoding Method: The decoding problem becomes classifying the initial frequency offsets, which is still a four-class classification. We adopt the same deep neural network

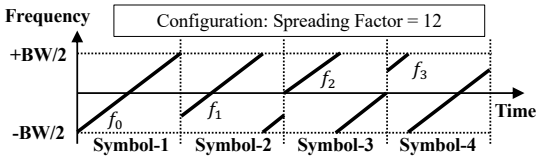


Figure 10: The illustration of the IFO-2 based LoRa encoder, using four initial frequency offsets (e.g., f_0, f_1, f_2, f_3) to represent 2-bit data.

(DNN) structure in Figure 9 to classify IFO-2 based symbols for decoding.

4.3 Weak Signal Packet Detection

After a LoRa node transmits a packet through the air, the packet should be reliably detected at the decoder side. Then, its payload is divided into multiple symbols and fed into the DNN-based decoder for chirp pattern recognition. Since the preamble part of a ChirpTransformer packet remains the same as the standard LoRa encoder, we need a reliable way to detect the LoRa preamble under ultra-low SNR. Given a period of received signals, we identify whether they contain a LoRa preamble, which consists of a series of base up-chirps, as an indicator of whether a packet is coming. We divide the received signals into N symbol-length signal segments, where N is the number of base up-chirp symbols in a LoRa preamble. Then, we combine the N segments as a superposed signal segment. All the base up-chirp symbols are constructively superposed if a preamble is in the received signals [21]. Considering the random initial phase of each base up-chirp symbol due to carrier frequency offset (CFO) and sampling frequency offsets (SFO), along with the quadratic distribution of the phase shift related to frequency bias and chirp symbol index, we heuristically determine the optimal phase compensation [21, 41]. After phase compensation, we coherently sum up the N symbol-length signal segments of the received signals. Then, we use a standard base up-chirp symbol to calculate its cross-correlation with the superposed signal segment. We treat the detection of a significant correlation peak as a sign of a successful preamble detection. The threshold for peak detection is based on channel estimation, which is six standard deviations of the mean noise correlation. The index of the correlation peak also indicates the boundary of the received LoRa preamble symbols. Therefore, we can align the timing of the demodulation window and extract the aligned symbols from the payload part using the detected preamble.

4.4 Implementation

We have implemented ChirpTransformer on COTS LoRa nodes and use an SDR as a gateway, shown at the bottom of Figure 11. Specifically, the USRP N210 SDR platform captures over-the-air LoRa signals by operating on a UBX daughter board with a sampling rate of 1 MS/s for both ChirpTransformer and standard LoRa, which is a widely used sampling rate setting [10, 21, 22, 41]. Since the maximum BW is 500 kHz in LoRa, COTS LoRa gateways should support a sampling rate of at least 1 MS/s according to the Nyquist–Shannon sampling theorem. The captured signal samples are then delivered to a back-end host, preprocessed, and demodulated by the decoder algorithms. The COTS SX1278 [30] based LoRa nodes transmit



Figure 11: USRP N210 based gateway and COTS LoRa nodes deployed in a campus environment.

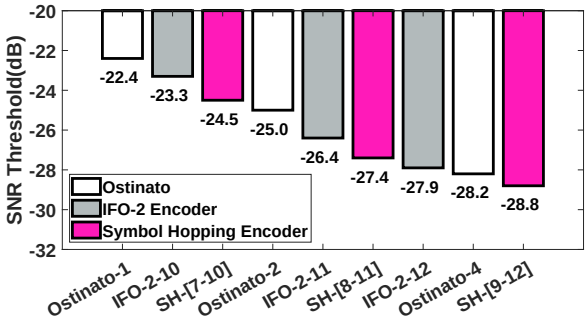


Figure 12: The comparison of SNR threshold between ChirpTransformer and Ostinato.

ChirpTransformer packets with random payloads. They are deployed in a campus environment, including both indoor and outdoor scenarios, as shown at the top of Figure 11. ChirpTransformer implements the DNN model on a Raspberry PI 4 [26] with an average inference time of 0.26 s over 100 runs and a memory requirement of 17.19 MB. In comparison, the compressed NELoRa model under the SF-10 setting reaches 152.9 MB in memory and 0.97 s for inference. This makes NELoRa 3.7× slower and 8.9× larger in memory than ChirpTransformer. This can further help reduce the overhead of the DNN model deployment at the gateway.

4.5 Baseline Methods and Metrics

Besides the standard LoRa, which uses SF-12 chirps for encoding and the dechirp for decoding, we choose Ostinato [48], a state-of-the-art encoder-decoder co-design for weak signal decoding beyond SF-12, as one of our baselines.

Ostinato: repetitive SF-12 chirps based encoder + chirp coherent-combining based decoder. Ostinato [48] can be regarded as a special case of ChirpTransformer's chirp repeating encoding. Specifically, Ostinato uses repeated SF-12 chirps, having the identical initial frequency offset, to encode the same data as a single SF-12 chirp does. Then, the decoder coherently combines the multiple SF-12 chirps to obtain a more obvious energy peak during dechirp than any individual SF-12 chirp. We implement the encoder by using our chirp repeating feature to generate the repeated SF-12 chirps and adopt the phase calibration method to fine-tune the chirp coherent combining for reliable decoding.

We use two metrics to indicate the performance of weak signal decoding.

• *Symbol Error Rate (SER)* is widely used to demonstrate the channel noise resilience of a physical layer design given different SNR [21, 41, 43]. A low SER is desirable.

• *SNR Threshold* is the lowest SNR that a physical layer design can keep the SER under a predetermined value. By default, we set the predetermined SER as 1%.

4.6 Communication Reliability Evaluation

Setup: We evaluate the SNR threshold of ChirpTransformer under SH-[7-10], SH-[8-11], SH-[9-12], IFO-2-10, IFO-2-11, and IFO-2-12 configurations. We further evaluate the SNR threshold of **Ostinato** with three comparable configurations, using single, two, and four repetitive SF-12 chirps to compose a symbol, indicated as **Ostinato-1**, **Ostinato-2**, and **Ostinato-4**. In the experiments, we collect high-SNR symbols on our campus testbed. Then, we inject random noises to generate weak signals with arbitrary SNR levels using the same method of NELoRa [21]. We use the synthesis symbols to calculate the SNR threshold, indicating communication reliability. We generate the same number of synthesis symbols to train the DNN decoder for our symbol hopping and IFO-2-based encoders.

Results: The results are shown in Figure 12.

General SNR Gain: The SNR thresholds of symbol hopping encoder are -24.5 dB, -27.4 dB and -28.8 dB using SH-[7-10], SH-[8-11], and SH-[9-12] encoding settings, separately. Compared with -22.4 dB SNR threshold of the standard LoRa under SF-12, we achieve the maximum 6.4 dB SNR gain, which is much larger than the approximate 2 dB SNR gain of NELoRa [21]. This indicates that ChirpTransformer significantly improves the communication reliability of LoRa. LoSee [53], an urban LoRa measurement study, introduces a link model to predict the packet delivery ratio (PDR) within a square coverage area based on link SNR. The coverage area is defined where the overall PDR value exceeds 70%. With the link model of LoSee and 6.4 dB SNR gain, ChirpTransformer can achieve approximately 2.38× coverage area compared with the standard LoRa in the urban environment.

SNR Gain from symbol hopping: the IFO-2 based encoder adopts the same DNN-based decoder with the symbol hopping encoder but encodes data with the same type of chirps. We can see that the SNR thresholds of the IFO-2 based encoder are 1.2 dB, 1.0 dB, and 0.9 dB higher than the symbol hopping encoder with the identical on-air time settings of OT-10, OT-11, and OT-12. On average, the symbol hopping encoder can tolerate 1.03 dB lower SNR than the IFO-2 based encoder. This verifies our symbol hopping feature provides a larger feature space than IFO-2 feature, which makes the different symbols can be easily distinguished with a DNN model. The SNR thresholds of SH-[7,10], SH-[8,11], and SH-[9-12] are 2.1 dB, 2.4 dB, and 0.6 dB lower than **Ostinato**-1, **Ostinato**-2, and **Ostinato**-4, separately. On average, ChirpTransformer achieves 1.7 dB SNR gain compared to **Ostinato**. This indicates our symbol hopping feature outperforms the chirp repeating pattern in **Ostinato** with complicated phase calibration for chirp coherent-combining.

4.7 Campus-scale Evaluation

Setup: SH is adopted by those LoRa nodes experiencing weak links to improve network coverage. The SH encoder extends its range and reduces the SNR threshold by lowering the data rate and neural-enhanced decoder. We conduct campus-scale experiments to verify

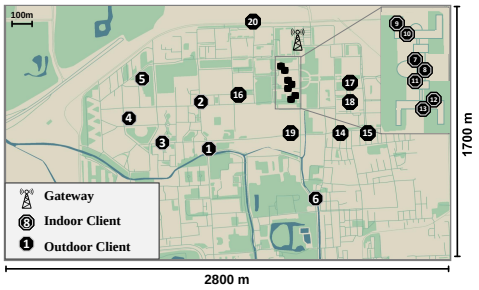
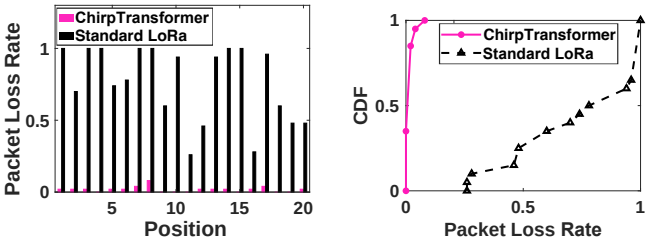


Figure 13: The campus-scale testbed with 20 LoRa nodes at 20 indoor and outdoor locations.

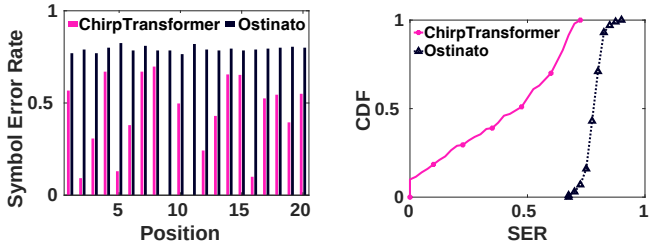


(a) Cross-position Performance (b) CDF Performance

Figure 14: The packet detection accuracy on our campus-scale testbed with extremely low SNR.

the gain of SH encoder-decoder co-design with extremely low SNR conditions. Figure 13 illustrates the deployment of our outdoor testbed at a campus ($2800m \times 1700m$). We randomly deploy 20 COTS LoRa nodes at 20 different NLOS positions covering indoor and outdoor scenarios. For indoor nodes, concrete walls are the main obstacles. Buildings and trees are the obstacles in the outdoor environment. For each LoRa node at a position, we collect tens of packets, and each contains 40 payload symbols. We repeat the data collection with three encoding methods of the standard LoRa, ChirpTransformer using SH-[9,12] configuration, and Ostinato using **Ostinato-4** configuration, respectively. We first evaluate the preamble detection accuracy of our design compared to the standard LoRa. In addition, for those detected packets, we compare the decoding SER between ChirpTransformer and Ostinato. For our SH-[9,12], we apply the DNN-based decoder pre-trained with our synthesis dataset collected in § 4.6 on the weak signal packets.

Results for packet detection: To understand whether a ChirpTransformer packet can be successfully detected at extremely low SNR conditions, we evaluate ChirpTransformer's performance on packet detection for all 20 positions. Figure 14a shows the comparison of the packet loss rate between ChirpTransformer and the standard LoRa. Figure 14b shows the CDFs of the packet loss rate for ChirpTransformer and the standard LoRa. We can see that the standard LoRa suffers a high packet loss rate at all positions, where more than 90% of packets are undetected at 10 out of 20 positions. In comparison, ChirpTransformer achieves a lower than 6% packet loss rate at all positions. This is because the standard LoRa detects LoRa preambles by searching for continuous identical frequency-domain energy peaks by applying the dechirp to the continuously received signals. Thus, it requires the energy peak of each chirp symbol in the preamble to be detectable. However, the real SNR at most of the positions is much lower than the SF-12 SNR threshold (e.g.,



(a) Cross-position Performance (b) CDF Performance

Figure 15: The comparison of communication reliability on our campus-scale testbed.

-22.4 dB), leading to undetectable energy peaks. After mitigating the CFO and SFO in a preamble, ChirpTransformer concentrates the energy of all preamble chirp symbols to detect a LoRa packet. The results verify that ChirpTransformer can reliably detect any LoRa packets at the extremely low SNR.

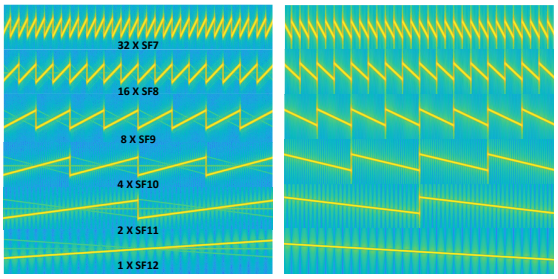
Results for decoding SER: We utilize our preamble detection design to detect Ostinato packets. Then, we compare ChirpTransformer with Ostinato by computing the decoding SER for the detected symbols at all 20 positions. As shown in Figure 15b, we can see the median SER of Ostinato is about 79.25%. In comparison, ChirpTransformer can achieve a much lower median SER of 40.54%, indicating the boundary of the communication range will be enlarged greatly. The reason is that Ostinato suffers from severe noises in the wild environment, leading to the signals not being coherently combined. At the same time, our neural-enhanced decoder can tolerate it by exploring multi-dimensional features. The specific SER across 20 positions is shown in Figure 15a. Our SER is lower than Ostinato at all 20 positions. In addition, we observe that ChirpTransformer has a higher SER in some positions, such as positions 4, 7, 8, 14, and 15, than others. The reason is that some coexisting wireless interference also brings new noise patterns that the pre-trained DNN model does not see, degrading the SER. We can involve an online fine-tuning process to deal with those new noise patterns [21].

5 CASE II: NETWORK THROUGHPUT

In our second case study, we target to enable efficient concurrent transmission, allowing multiple LoRa nodes to transmit their packet simultaneously to enhance the network throughput and scalability [11, 22, 36]. Specifically, in standard LoRa, concurrent transmission is infeasible for multiple packets with the same SF configuration [22]. The six SFs create six quasi-orthogonal logic channels to enable concurrent transmission [54]. However, since the SNR threshold increases when SF decreases (§ 2.2), we cannot assign arbitrary SF to a LoRa node without noticing its SNR requirement, thus degrading the efficiency of concurrent efficiency. With ChirpTransformer, given the same SNR tolerance, we aim to design multiple orthogonal logic channels to achieve efficient concurrent transmission. Our key idea is to use different repeating chirp patterns with CRT configurations to create orthogonal logic channels.

5.1 CRT Encoder-Decoder Co-design

Encoding Method: For an orthogonal logic channel, the encoding method can be indicated as (OT-k, IFO-i, CRT-$(k-i)$, SH-0). OT-k is the pre-configured on-air time. IFO-i indicates that we use



(a) Chirp Repeating (b) Template Downchirp

Figure 16: Six orthogonal logic channels with OT-12. (a) Chirp repeating based encoding. (b) Template down-chirp based decoding.

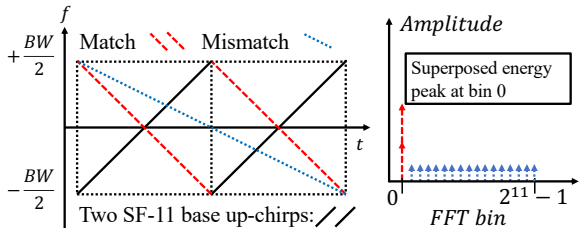
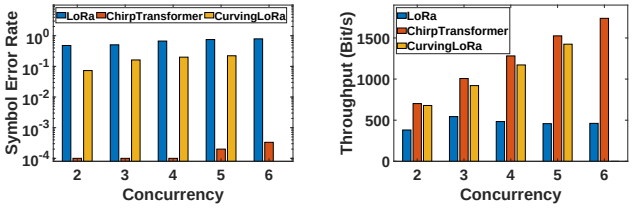


Figure 17: After multiplying the matched template down-chirp, a superposed energy peak appears.

SF-i chirps to construct the chirp repeating pattern and encode i data bits. The chirp repeating times are 2^{k-i}. i is in the range of $[7, \max\{k, 12\}]$, indicating k must be larger than i to create a diverse chirp repeating pattern, and i is not less than 7, the minimum SF chirp we have. Therefore, given the pre-configured on-air time OT-k, we have $k-7$ orthogonal logic channels if $k \leq 12$. Otherwise, we have six orthogonal logic channels using SF-7 to SF-12 chirp repeating patterns. The larger the k is, the more orthogonal logic channels we can have. Figure 16a shows an example of the on-air time OT-12. We have six orthogonal logic channels with 32 SF-7 chirp repeating (OT-12, IFO-7, CRT-5, SH-0), 16 SF-8 chirp repeating (OT-12, IFO-8, CRT-4, SH-0), 8 SF-9 chirp repeating (OT-12, IFO-9, CRT-3, SH-0), 4 SF-10 chirp repeating (OT-12, IFO-10, CRT-2, SH-0), 2 SF-11 chirp repeating (OT-12, IFO-11, CRT-1, SH-0), and one SF-12 chirp (OT-12, IFO-12, CRT-0, SH-0).

Template Down-chirp based Decoding: Given the (OT-k, IFO-i, CRT-$(k-i)$, SH-0), we use a template down-chirp to find the energy peak in the FFT spectrum. The template down-chirp consists of 2^{k-i} SF-i base down-chirps. Figure 16b shows the template down-chirps for our six logic channels with OT-12. The basic observation is illustrated in Figure 17. Given an (OT-12, IFO-11, CRT-1, SH-0) symbol with two SF-11 base up-chirps, we multiply two matched base down-chirps (called *template*) and coherently combine the derived signals in the two SF-11 windows together. After applying FFT, a superposed energy peak appears at bin 0, which is equivalent to the energy peak of an SF-12 base up-chirp due to the same on-air, thus supporting the same SNR tolerance ability. When using a mismatched template like an SF-12 base down-chirp, the energy is dispersed at all bins.

With this observation, the decoder first initializes $\max\{k-7, 6\}$ down-chirp templates with different SF configurations. Then, for each template, due to the CFO and SFO among the repetitive chirps



(a) SER (b) Throughput

Figure 18: Collision performance of ChirpTransformer, CurvingLoRa and LoRa. (a) SER. (b) Throughput.

in the symbol, we compensate various phase offsets calculated by the CFO and SFO estimated during packet detection for alleviating the random initial phase problem [21], then apply the template to obtain the corresponding spectrum with FFT. With the phase compensation, we can make the obtained energy peak at frequency bin f_0 as accurate as possible. Suppose the derived energy peak with a template is larger than the current maximum energy peak. In that case, we update the expected initial frequency offset f_0 and IFO configuration of the symbol and the latest maximum energy peak. When traversing all templates, the maximum energy peak is selected, and the f_0 and IFO configuration are converted to data bits accordingly.

5.2 Implementation and Baseline

We implement the encoder and decoder on our campus testbed. Besides the standard LoRa, we select CurvingLoRa [22] as the state-of-the-art baseline from the perspective of LoRa encoder design. CurvingLoRa utilizes the orthogonal coding space created by different non-linear chirps to enable concurrent LoRa transmissions. The non-linear chirps have similar SNR tolerance compared to the linear chirp with the same on-air time. Moreover, besides SER, we use another metric *Throughput*, which indicates the amount of data transmitted over a network in a unit of time.

5.3 Concurrent Transmission Evaluation

Setup: We collect high-SNR symbols from our campus testbed. We conduct trace-driven experiments to evaluate the SER and network throughput of ChirpTransformer, CurvingLoRa [22], and the standard LoRa during concurrent transmission. As the same operation in CurvingLoRa [22] to emulate signal collisions, we collect individual symbols in real environments separately and then add the symbols together to generate the overlapping patterns. To simulate diverse temporal patterns of symbol collisions, we randomly assign a symbol offset from [0,1] times the symbol on-air time. To evaluate the performance under near-far issues, we add a signal-to-interference ratio (SIR) ranging from -20 to 0 dB to concurrently transmitted symbols. The frequency bandwidth is 125 kHz. For the standard LoRa, we use SF-12 chirps for all concurrent transmitters. For CurvingLoRa, we adopt five types of chirps: quadratic1, quartic1, quadratic1, quartic1, and linear with on-air time OT-12. Since we assign concurrent transmitters to different orthogonal logical channels, the five logic channels only support five concurrencies at most. For ChirpTransformer, we use the six orthogonal chirp repeating logic channels (OT-12, IFO-12, CRT-0, SH-0), (OT-12, IFO-11, CRT-1, SH-0), (OT-12, IFO-10, CRT-2, SH-0), (OT-12, IFO-9, CRT-3,

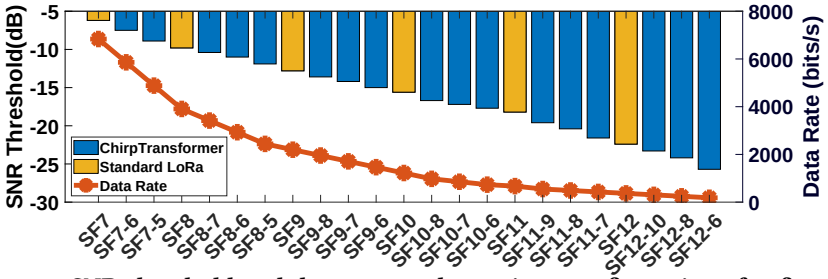


Figure 19: SNR threshold and data rate under various configurations for fine-grained data rate adaption.

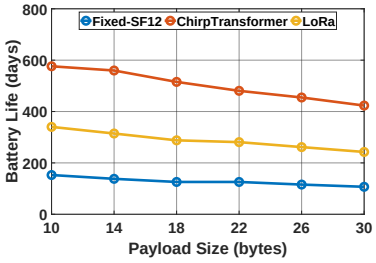


Figure 20: Battery life measurement with LoSee vehicle tracking traces.

SH-0), (OT-12, IFO-8, CRT-4, SH-0), and (OT-12, IFO-7, CRT-5, SH-0). To calculate the average network throughput and SER. We repeat experiments for each concurrency configuration 2,000 times.

Results for Symbol Error Rate Performance: The SER for different concurrency values from 2 to 6 is shown in Figure 18a. 10^{-4} represents no error symbols. When concurrency is 2, LoRa fails to decode half of the symbols correctly. In contrast, the SER for CurvingLoRa is less than 10%, and there is no error symbol for ChirpTransformer. As concurrency increases, the SER for CurvingLoRa and ChirpTransformer also increases. However, CurvingLoRa exhibits significantly higher symbol decoding errors compared to ChirpTransformer. Despite a SIR range of [-20,0] dB and a concurrency of 6, ChirpTransformer can still maintain a SER less than 0.2%.

Results for Network Throughput Performance: As shown in Figure 18b, LoRa consistently has the lowest throughput due to its high SER, unaffected by concurrency values ranging from 2 to 6. In contrast, ChirpTransformer consistently outperforms CurvingLoRa in throughput due to our lower SER. Specifically, at a concurrency of 5, the throughput of ChirpTransformer is 3.33× that of LoRa and 1.07× that of CurvingLoRa, demonstrating ChirpTransformer's superior performance in concurrent transmissions.

6 CASE III: NETWORK LIFETIME

In the standard LoRa, we configure SF to balance between SNR tolerance and data rate. After a LoRa node is deployed or its surrounding environment changes, a LoRa gateway will check the node's SNR and choose an SF configuration that can reach the highest data rate to enhance the energy efficiency while its SNR threshold is lower than the observed SNR to keep reliable packet delivery. Only six SFs are too coarse-grained to optimize network energy efficiency in complex environments [25, 49]. As such, we aim to develop fine-grained data rate adaptation with IFO configurations to achieve energy efficiency.

6.1 IFO based Encoder-Decoder Co-Design

Encoding Method: We adjust the available initial frequency offsets to obtain more encoding configurations. The encoding method can be indicated as (OT-k, IFO-i, CRT-0, SH-0). For energy efficiency, the on-air time is not larger than that of an SF-12 chirp. Thus, we have $k \leq 12$ and $i \leq k$. Given the on-air time OT-k, we encode i bits of data. The data rate is $\frac{i \times BW}{2^k}$. To encode i bits data, we only need 2^i initial frequency offsets. Given the total 2^k initial frequency offsets, we uniformly remove unused offsets and select the rest as the available initial frequency offsets for encoding. For example, for OT-12, the range of initial frequency offsets is $[0, \frac{2^{12}-1}{2^{12}}BW]$. For IFO-10, we uniformly select a quarter of these offsets as $\{0, \frac{4}{2^{12}}BW, \frac{8}{2^{12}}BW, \frac{12}{2^{12}}BW, ..., \frac{2^{12}-4}{2^{12}}\}$. When k is from 7 to 11, i is in the range of $[\lceil \frac{k+1}{2} \rceil, k]$. When k is 12, i is in the range of $[6, 12]$. For example, when k is 7, the available i is 7, 6, and 5. If i is 4, the data rate is $\frac{4BW}{2^7}$ which is the same with $\frac{8BW}{2^8}$ represented by (OT-8, IFO-8, CRT-0, SH-0). In this way, we avoid the redundant data rate among different chirps with different OTs, resulting total 23 data rate configurations.

Selective Energy Peak Searching Decoding Method: According to the set of available initial frequency offsets with the configuration (OT-k, IFO-i, CRT-0, SH-0), we apply the standard dechirp for decoding and search the energy peak at specific FFT bins corresponding to the available initial frequency offsets. Since noises are random across the entire spectrum, the fewer the FFT bins we search for the energy peak, the less the energy peak of noises can influence our energy peak searching. Thus, when we limit the range of the searching space, we can lower the SNR threshold for successful decoding. Hence, we provide the tradeoff between noise tolerance and data rate.

6.2 Implementation and Evaluation

Setup: We collect a full set of standard LoRa chirp symbols with all available initial frequency offsets across SF-7 to SF-12 at high SNR levels (>30 dB) on our campus testbed. Then, we generate chirp symbols at multiple low SNR levels by injecting random Gaussian noise to simulate real-world scenarios [21]. We use the SNR threshold and data rate as metrics to compare the performance of different settings. The encoding configuration (OT-k, IFO-i, CRT-0, SH-0) is indicate as SFk-i. For example, SF7-6 indicates using SF-7 chirps to encode 6-bit data.

Moreover, to verify the battery life gain (BLG) of ChirpTransformer compared to other methods, we use real-world LoSee traces [53] for vehicle tracking in an urban environment. A mobile LoRa node on a car periodically travels across the roads between office and home, ranging from hundreds of meters to 3.2 kilometers in an urban area. The LoRa node transmitted packets with SF-12 configuration. With the idea of data rate adaption, we reassign the data rate according to the SNR to reduce transmission time and shorten energy consumption. One baseline method is the default LoRa data rate adaptation, which uses SF-7 to SF-11 configurations to replace SF-12 in some cases with better SNR conditions. However, ChirpTransformer offers a more precise data rate adaptation scheme with

23 configurations. We estimate the battery life assuming that the LoRa node is powered by two AA batteries (2200mAh) with 3.6 V voltage and sending packets of different lengths (from 10 to 30 bytes) 240 times per day, as the LoRa node would do in the morning and evening vehicle tracking. Based energy profiling on Figure 7, we calculate the energy consumption during two consecutive transmission beginning times by multiplying the currents across different phases with input voltages. The extra energy consumption caused by ChangeChannelFhss interruption on the node is neglectable. Then we accumulate the energy over time and calculate the battery life.

Results for SNR Threshold v.s. Data Rate: Figure 19 displays the SNR threshold and data rate in ChirpTransformer with various encoder configurations. The orange bars (e.g., SF-7, SF-8, SF-9, SF-10, SF-11, and SF-12) represent the SNR threshold of standard LoRa, while the blue bars show the complementary SNR threshold of ChirpTransformer. As the data rate (e.g., brown lines) gradually decreases from $\frac{7 \cdot BW}{2^7}$ to $\frac{6 \cdot BW}{2^{12}}$, the SNR threshold slowly decreases from -6.2 dB to -25.7 dB. The mean difference between adjacent data rate schemes is about 0.89 dB. Compared with the standard LoRa, we can see that ChirpTransformer provides a more flexible data rate scheme with a more diverse SNR threshold.

Results for Battery Life Gain (BLG): Figure 20 shows that LoRa data rate adaptation averagely extends battery life by 2.26× compared to the fixed-SF-12. On average, ChirpTransformer data rate adaption provides a battery life of 3.93× that of LoRaWAN with fixed-SF-12 and 1.74× that of LoRa data rate adaptation. The reason for this improvement is that ChirpTransformer reassigns a better data rate for the positions where the measured SNR is between the SNR threshold values of two standard LoRa SF configurations, whereas conventional rate adaptation in LoRa wastes the BLG improvement space. Moreover, the BLG trends remain stable as payload size increases.

7 RELATED WORK

LoRa Reliable Decoding: By utilizing either multiple gateways and LoRa nodes, recent studies [6, 8, 11, 18, 45, 48] bring extra SNR gains for LoRa transmissions. Charm [8] coordinates multiple gateways to decode weak signals undecodable at any gateway by detecting the combined energy peak in the spectrum. Choir [11] exploits the correlation across co-located LoRa nodes, enabling a larger communication range than an individual one. NELoRa [21] develops a DNN decoder capturing multi-dimension features to obtain 1.84-2.35 dB SNR gains. Ostinato [48] uses repeated SF-12 chirps to achieve weak signal decoding beyond SF-12 configuration. XCopy [45] enhances signal strength by coherently combining retransmitted packets over weak links to boost weak signal decoding. In contrast, instead of developing a decoder at the gateway and server side, we obtain SNR gains from the encoder side. ChirpTransformer is parallel with these works.

Packet Collision Resolving: Previous research has focused on identifying collisions in the time or frequency domain. For instance, Choir[11] matches bits to each LoRa node by detecting the frequency changes caused by oscillator deficiencies. FTrack[46] identifies collisions by exploring distinct tracks on the spectrum and

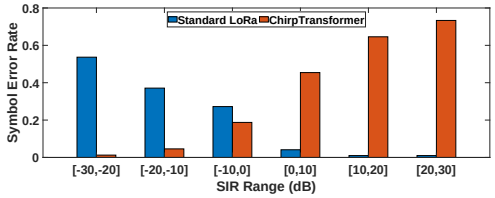


Figure 21: Concurrent decoding SER of collided ChirpTransformer and standard LoRa signals under various SIR settings.

symbol edges in the time domain. Due to the near-far problem, existing work cannot implement concurrent transmission with high SIR among different devices in large-scale deployment. CIC[36] decodes signals by combining the spectrum from different parts of a single symbol to cancel interference signals at gateways and support concurrent transmission. CurvingLoRa [22] enables efficient concurrent transmission with non-linear chirps, but it is not compatible with COTS LoRa nodes. LMAC [14] utilizes Channel Activity Detection (CAD) to implement carrier-sense multiple access protocol to avoid collision. Compared to these works, we propose orthogonal encoding space at the encoder side to enable concurrent transmission while it can be realized on COTS LoRa nodes. The decoder design (e.g., CIC, Choir, FTrack) can be further combined with ChirpTransformer to further improve network performance.

Rate Adaptation in LoRa: To achieve an adaptive data rate [33], LoRa adjusts the data rate by using various SFs based on received SNR levels. FLoRa [37] focuses on dynamically managing link parameters to improve network scalability and energy efficiency. DyLoRa [23] establishes an energy model that associates link properties with transmission parameters. AdapLoRa [16] is another approach that periodically adjusts resource allocation based on a linear regression process estimating network lifetime. Current works rely on the standard LoRa encoding scheme and are unable to implement precise data rates necessary to adapt to diverse environments to extend battery life. Beyond the standard LoRa, ChirpTransformer provides fine-grained data rate adaption by selecting different sets of initial frequency to encode data bits with different on-air times.

8 DISCUSSION

8.1 System Co-existence

ChirpTransformer and the standard LoRa encoder share the same group of basic chirps (e.g., SF-7 to SF-12), raising the concern of the co-existence issues when we apply ChirpTransformer in existing LoRa deployments. For example, when we utilize SH-[9-12] encoder to enhance communication reliability, a ChirpTransformer symbol with the pattern of 8 repeated SF-9 chirps may interfere with other transmissions with SF-9 setting in co-existent standard LoRa deployment.

To measure the co-existence issues, we conduct experiments to emulate real-world signal collision between ChirpTransformer and the standard LoRa signals. First, we generate a ChirpTransformer symbol by selecting from one of three SH settings (i.e., SH-[7-10], SH-[8-11], SH-[9-12]). Then, the ChirpTransformer symbol is superposed with several standard LoRa symbols, characterized by one of the four different SF settings in the corresponding SH setting, with a random initial frequency offset. The on-air time of

these standard LoRa symbols is the same as the ChirpTransformer symbol. We set a random time offset between the two superposed signals ranging from 0 to 1 of ChirpTransformer symbol on-air time. Moreover, the signal-to-interference ratio (SIR) [20, 22] of standard LoRa to ChirpTransformer indicates the signal strength of standard LoRa signals compared to ChirpTransformer signals. The higher the SIR is, the stronger the standard LoRa signals are. The lower the SIR is, the stronger the ChirpTransformer signals are. We divide the SIR values into 6 ranges: [-30, -20], [-20, -10], [-10, 0], [0, 10], [10, 20], and [20, 30]. With these emulation settings, we mimic the diverse collision conditions of ChirpTransformer and standard LoRa signals in various real-world scenarios. Finally, we apply our neural-enhanced decoder and the standard dechirp to decode ChirpTransformer and standard LoRa symbols from the collided signals. In each SIR range, we uniformly generate 1,000 collided signals across different SIR settings, SH settings in ChirpTransformer, and SF settings in the standard LoRa to calculate the average SER of decoding ChirpTransformer and standard LoRa symbols.

The impact of ChirpTransformer to existing LoRa deployments. As illustrated in Figure 21, it is evident that standard LoRa signals, when stronger than ChirpTransformer signals in the SIR ranges of [0,10], [10,20], and [20,30], can maintain a SER below 5%. However, when ChirpTransformer signals are stronger than standard LoRa signals as [-10,0] SIR, the SER of standard LoRa signals increases to 27.2%. This increase in SER is more pronounced as the SIR range decreases, highlighting interference from much stronger ChirpTransformer signals on existing LoRa.

The impact of the standard LoRa to ChirpTransformer. In Figure 21, we can also observe that when standard LoRa signals are weaker than ChirpTransformer signals when SIR ranges are [-30,-20] and [-20,-10], the SER of decoding ChirpTransformer signals remains less than 5%. The SER rises up to about 20% at the SIR range [-10,0]. When standard LoRa signals are stronger than ChirpTransformer, the SER of ChirpTransformer increases quickly to 73% at the SIR range [20,30].

Remarks: The results imply a limitation of applying ChirpTransformer in practice that the co-existence issues are non-negligible. We could have three ways to alleviate the co-existence issues. Firstly, we can leverage the near-far effect. Since our SH encoder is more noise-resilient than the standard LoRa for weak signal decoding, ChirpTransformer is primarily adopted by LoRa nodes located beyond the reach of standard LoRa gateways. This ensures the high SIR of standard LoRa to ChirpTransformer and standard LoRa signals can be successfully decoded. Secondly, we can leverage LoRa carrier sense [14, 15] to avoid packet collision with transmission backoff. Thirdly, we can develop a new collision resolving method at the decoding side to enable successful decoding under low SIR conditions by borrowing the features from different LoRa packets [36, 41, 46]. For example, using multiple sub-symbol temporal windows of varying lengths [36] allows solving collision of ChirpTransformer and standard LoRa packets. By merging the spectral results from these windows, we retain the consistently appearing desired packet and eliminate intermittent interfering frequency peaks.

8.2 Efficient Adaptive Date Rate

ChirpTransformer supports a fine-grained adaptive data rate (ADR) with 23 configurations, allowing a LoRa nodes to adapt to its deployment scenario optimally. On the other hand, we need an efficient protocol to enable agile and accurate ADR with low control overhead. A feasible method is to inherit the current LoRaWAN ADR framework [5, 33, 34] in MAC commands. Specifically, LoRa gateways and network servers run an ADR algorithm that adjusts the encoding configuration of a LoRa node according to the SNR level of recently received packets. The LoRa node will open a receiving window after a packet transmission to receive the potential configuration request from LoRa gateways and network servers. When an encoding configuration change is necessitated, the LoRa gateways will initiate a request for data rate adjustment using *LinkADRReq* in the receiving window of the LoRa node. Then, the LoRa node will change its data rate correspondingly and respond with *LinkADRAns* for ADR acknowledgment.

By adopting our fine-grained ADR in this framework, a LoRa node will receive more requests from LoRa gateways and network servers and transmit more ADR acknowledgments. However, since the LoRa node keeps opening its receiving window after each packet transmission no matter whether a feedback packet is coming or not, receiving more requests will not bring extra energy consumption. Moreover, an ADR acknowledgment only contains two bytes of frame payload [33, 34], which is much shorter than a normal LoRa packet. Thus, the extra energy cost of sending more ADR acknowledgments is affordable regarding the benefits of our fine-grained SDR.

9 CONCLUSION

In this paper, we propose ChirpTransformer, a versatile LoRa encoding framework to enable the reliable connection between a LoRa node and gateways at extremely low SNR, achieve high network throughput with collision resolving, and improve energy efficiency in complex environments. Instead of using only one parameter, SF, to adjust the encoding method in the standard LoRa, ChirpTransformer develop four chirp features: 1) on-air time of a symbol; 1) available initial frequency offsets; 2) intra-symbol chirp repeating; 3) inter-symbol chirp pattern hopping, to enlarge the encoding feature space so that we can design different encoders to meet various LoRa deployment requirements. Different encoding methods can be represented with a four-factor tuple (OT, IFO, CRT, SH). Specifically, we have designed a symbol hopping based encoder for weak signal decoding, a chirp repeating based encoder for collision resolving, and a selective initial frequency offset based encoder for fine-grained data rate adaption. We implement ChirpTransformer with COTS LoRa nodes and SDR. Then, we conduct extensive experiments in both a campus testbed and real-world trace-driven studies to evaluate the performance of ChirpTransformer. The results show a 2.38 × network coverage, 3.14 × network throughput, and 3.93 × battery lifetime compared with the standard LoRa.

ACKNOWLEDGEMENT

We sincerely thank the anonymous reviewers and our shepherd for their valuable feedback. This work was partially supported by NSF Award NeTS-2312674, 2312675.

REFERENCES

[1] [n. d.]. Amazon Sidewalk. https://www.amazon.com/Amazon-Sidewalk/b?ie=UTF8&node=21328123011. Retrieved by Nov 29 2023.

[2] [n. d.]. LoRa by the Numbers. https://www.semtech.com/lora. Retrieved by Nov 12 2023.

[3] [n. d.]. Microsoft FarmBeats. https://www.microsoft.com/en-us/research/project/farmbeats-iot-agriculture/. Retrieved by Nov 29 2023.

[4] [n. d.]. Monsoon Solutions Inc. High voltage power monitor. https://www.msoon.com/high-voltage-power-monitor. Retrieved by Nov 29 2023.

[5] LoRa Alliance. Retrieved by Nov 19th 2020. A technical overview of LoRa and LoRaWAN. In *https://lora-alliance.org/resource-hub/what-lorawanr*.

[6] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. 2020. A cloud-optimized link layer for low-power wide-area networks. In *Proceedings of ACM MobiSys*.

[7] Tusher Chakraborty, Heping Shi, Zerina Kapetanovic, Bodhi Priyantha, Deepak Vasisht, Binh Vu, Parag Pandit, Prasad Pillai, Yaswant Chabria, Andrew Nelson, et al. 2022. Whisper: IoT in the TV White Space Spectrum. In *Proceedings of USENIX NSDI*.

[8] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Charm: Exploiting Geographical Diversity through Coherent Combining in Low-Power Wide-Area Networks. In *Proceedings of ACM/IEEE IPSN*.

[9] Jialuo Du, Yidong Ren, Mi Zhang, Yunhao Liu, and Zhichao Cao. 2023. NELoRa-Bench: A Benchmark for Neural-enhanced LoRa Demodulation. *International Conference on Learning Representations (ICLR) Workshop on Machine Learning for IoT* (2023).

[10] Jialuo Du, Yidong Ren, Zhui Zhu, Chenning Li, Zhichao Cao, Qiang Ma, and Yunhao Liu. 2023. SRLoRa: Neural-enhanced LoRa Weak Signal Decoding with Multi-gateway Super Resolution. In *Proceedings of ACM MobiHoc*.

[11] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering Low-Power Wide Area Networks in Urban Settings. In *Proceedings of ACM SIGCOMM*.

[12] Tallal Elshabrawy and Joerg Robert. 2019. Interleaved chirp spreading LoRa-based modulation. *IEEE Internet of Things Journal* 6, 2 (2019), 3855–3863.

[13] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. 2020. Frequency Configuration for Low-Power Wide-Area Networks in Heartbeat. In *Proceedings of USENIX NSDI*.

[14] Amalinda Gamage, Christian Jansen Liando, Chaojie Gu, Rui Tan, and Mo Li. 2020. LMAC:Efficient Carrier-Sense Multiple Access for LoRa. In *Proceedings of ACM MobiCom*.

[15] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. 2020. Lmac: Efficient carrier-sense multiple access for lora. In *Proceedings of ACM MobiCom*.

[16] Weifeng Gao, Zhiwei Zhao, and Geyong Min. 2020. AdapLoRa: Resource Adaptation for Maximizing Network Lifetime in LoRa networks. In *Proceedings of IEEE ICNP*.

[17] Branden Ghena, Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. 2019. Challenge: Unlicensed lpwans are not yet the path to ubiquitous connectivity. In *Proceedings of ACM MobiCom*.

[18] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. 2022. Don't Miss Weak Packets: Boosting LoRa Reception with Antenna Diversities. In *Proceedings of IEEE INFOCOM*.

[19] Chenning Li and Zhichao Cao. 2022. Lora networking techniques for large-scale and long-term iot: A down-to-top survey. *Comput. Surveys* 55, 3 (2022), 1–36.

[20] Chenning Li, Zhichao Cao, and Li Xiao. 2022. Curvealoha: Non-linear chirps enabled high throughput random channel access for lora. In *Proceedings of IEEE INFOCOM*.

[21] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. 2021. NELoRa: Towards Ultra-low SNR LoRa Communication with Neural-enhanced Demodulation. In *Proceedings of ACM Sensys*.

[22] Chenning Li, Xiuzhen Guo, Longfei Shangguan, Zhichao Cao, and Kyle Jamieson. 2022. CurvingLoRa to Boost LoRa Network Throughput via Concurrent Transmission. In *Proceedings of USENIX NSDI*.

[23] Yinghui Li, Jing Yang, and Jiliang Wang. 2020. DyLoRa: Towards Energy Efficient Dynamic LoRa Transmission Control. In *Proceedings of IEEE INFOCOM*.

[24] Li Liu, Yuguang Yao, Zhichao Cao, and Mi Zhang. 2021. DeepLoRa: Learning Accurate Path Loss Model for Long Distance Links in LPWAN. In *Proceedings of IEEE INFOCOM*.

[25] Di Mu, Yitian Chen, Junyang Shi, and Mo Sha. 2020. Runtime control of LoRa spreading factor for campus shuttle monitoring. In *Proceedings of IEEE ICNP*.

[26] Raspberry PI. Retrieved by Mar 28 2024. Raspberry PI 4. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[27] Yidong Ren, Puyu Cai, Jinyan Jiang, Jialuo Du, and Zhichao Cao. 2023. Prism: High-throughput LoRa backscatter with non-linear chirps. In *Proceedings of IEEE INFOCOM*.

[28] Yidong Ren, Wei Sun, Jialuo Du, Huaili Zeng, Younsuk Dong, Mi Zhang, Shigang Chen, Yunhao Liu, Tianxing Li, and Zhichao Cao. 2024. Demeter: Reliable Cross-soil LPWAN with Low-cost Signal Polarization Alignment. In *Proceedings of ACM MobiCom*.

[29] Ettus Research. [n. d.]. USRP N210 datasheet. https://www.ettus.com/all-products/un210-kit/. Retrieved by Nov 15 2023.

[30] Semtech. [n. d.]. SX1278/77/78/79 Datasheet. https://www.semtech.com/products/wireless-rf/lora-core/sx1278. Retrieved by Nov 25 2023.

[31] Semtech. [n. d.]. SX1302. https://www.semtech.com/products/wireless-rf/lora-core/sx1302. Retrieved by Nov 13 2023.

[32] Semtech. [n. d.]. SX1303. https://www.semtech.com/products/wireless-rf/lora-core/sx1303. Retrieved by Nov 13 2023.

[33] Semtech. [n. d.]. Understanding LoRaWAN ADR. https://lora-developers.semtech.com/library/tech-papers-and-guides/understanding-adr/. Retrieved by Nov 25 2023.

[34] Semtech. [n. d.]. Understanding LoRaWAN ADR. https://lora-developers.semtech.com/documentation/tech-papers-and-guides/implementing-adaptive-data-rate-adr/implementing-adaptive-data-rate/#inform-the-network-server-that-the-end-device-is-ready-to-use-adr. Retrieved by Mar 29 2023.

[35] Semtech. Retrieved by Nov 13 2023. Semtech SX1276. https://www.semtech.com/products/wireless-rf/lora-connect/sx1276.

[36] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. 2021. Concurrent interference cancellation: Decoding multi-packet collisions in LoRa. In *Proceedings of ACM SIGCOMM*.

[37] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. 2018. Adaptive configuration of LoRa networks for dense IoT deployments. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–9.

[38] Zehua Sun, Tao Ni, Huanqi Yang, Kai Liu, Yu Zhang, Tao Gu, and Weitao Xu. 2023. FLoRa: Energy-Efficient, Reliable, and Beamforming-Assisted Over-The-Air Firmware Update in LoRa Networks. In *Proceedings of ACM/IEEE IPSN*.

[39] Zehua Sun, Huanqi Yang, Kai Liu, Zhimeng Yin, Zhenjiang Li, and Weitao Xu. 2022. Recent advances in LoRa: A comprehensive survey. *ACM Transactions on Sensor Networks* 18, 4 (2022), 1–44.

[40] Shuai Tong, Zilin Shen, Yunhao Liu, and Jiliang Wang. 2021. Combating link dynamics for reliable lora connection in urban settings. In *Proceedings of ACM MobiCom*.

[41] Shuai Tong, Jiliang Wang, and Yunhao Liu. 2020. Combating packet collisions using non-stationary signal scaling in LPWANs. In *Proceedings of ACM MobiSys*.

[42] Shuai Tong, Jiliang Wang, Jing Yang, Yunhao Liu, and Jun Zhang. 2023. City-wide LoRa Network Deployment and Operation: Measurements, Analysis, and Implications. In *Proceedings of ACM SenSys*.

[43] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. 2020. CoLoRa: Enabling Multi-Packet Reception in LoRa. In *Proceedings of IEEE INFOCOM*.

[44] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. 2019. mLoRa: A Multi-Packet Reception Protocol in LoRa networks. In *Proceedings of IEEE ICNP*.

[45] Xianjin Xia, Qianwu Chen, Ningning Hou, Yuanqing Zheng, and Mo Li. 2023. XCopy: Boosting Weak Links for Reliable LoRa Communication. In *Proceedings of ACM MobiCom*.

[46] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: parallel decoding for LoRa transmissions. In *Proceedings of ACM SenSys*.

[47] Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. 2021. Pyramid: Real-time lora collision decoding with peak tracking. In *Proceedings of IEEE ICNP*.

[48] Zhenqiang Xu, Pengjin Xie, Jiliang Wang, and Yunhao Liu. 2022. Ostinato: Combating lora weak links in real deployments. In *Proceedings of IEEE ICNP*.

[49] Jing Yang, Zhenqiang Xu, and Jiliang Wang. 2021. Ferrylink: Combating link degradation for practical lpwan deployments. In *Proceedings of IEEE ICPADS*.

[50] Kang Yang, Yuning Chen, Xuanren Chen, and Wan Du. 2023. Link quality modeling for lora networks in orchards. In *Proceedings of ACM/IEEE IPSN*.

[51] Kang Yang and Wan Du. 2022. LLDPC: A low-density parity-check coding scheme for LoRa networks. In *Proceedings of ACM Sensys*.

[52] Yuguang Yao, Zijun Ma, and Zhichao Cao. 2019. LoSee: Long-Range Shared Bike Communication System Based on LoRaWAN Protocol.. In *Proceedings of EWSN*. 407–412.

[53] Ren Yidong, Liu Li, Li Chenning, Cao Zhichao, and Chen Shigang. 2022. Is LoRaWAN Really Wide? Fine-grained LoRa Link-level Measurement in An Urban Environment. In *Proceedings of IEEE ICNP*.

[54] Fu Yu, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2023. Enabling Concurrency for Non-orthogonal LoRa Channels. In *Proceedings of ACM MobiCom*.

[55] Shiming Yu, Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. 2024. Revolutionizing LoRa Gateway with XGate: Scalable Concurrent Transmission across Massive Logical Channels. In *Proceedings of ACM MobiCom*.