2024 21st Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) | 979-8-3315-1918-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/SECON64284.2024.10934871

ZChirp: Speeding Up LPWANs by Combining the Chirp with Binary Sequences

Zhenghao Zhang
Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA

***Abstract*—Low Power Wide Area Networks (LPWAN), such as LoRa, support long range wireless connections for a vast number of low power IoT devices. One of the main challenges in LPWANs is to provide high network capacity. In this paper, a novel modulation technique, referred to as ZChirp, is proposed. ZChirp augments the chirp modulation in LoRa by multiplying the chirp with a binary sequence called the Z sequence, which serves dual purposes. First, chirps multiplied with different Z sequences have very low cross-correlations, which enables simultaneous transmissions of multiple nodes and increases the network capacity. Second, additional data can be modulated by using different Z sequences for different data values, which proves to be an effective option to increase the data rate. Experiments and simulations show that ZChirp does not increase bandwidth usage nor reduce the communication range comparing to the original LoRa. Experiments in the POWDER platform demonstrate that the ZChirp gateway successfully decoded packets from multiple nodes transmitted simultaneously at random times. Simulations further show that the capacity gain of ZChirp over the state-of-the-art solution that also modifies the LoRa modulation is $2.0\times$.**

I. Introduction

Low Power Wide Area Networks (LPWAN) support a large number of connections from IoT devices to *gateways* over long distances. Among the existing LPWAN technologies, LoRa [3] has attracted significant interest in recent years. To serve a wide area, the main challenge of LoRa today is to provide high network capacity, i.e., to allow the gateway to receive as many packets as possible from the nodes simultaneously. As LoRa was not originally designed with strong capabilities to resolve packet collisions, many solutions have been proposed to decode collided packets without modifying the nodes, which have demonstrated impressive performance, such as [9], [22], [21], [8]. A clean-slate alternative is to redesign the transmitted signal, such as CurvingLoRa [15], which holds the potential of achieving even higher performance.

In this paper, *ZChirp* is proposed, which is a clean-slate redesign of LoRa focusing on the uplink communications from nodes to the gateway. As the original LoRa, ZChirp is also based on *chirp*, which is a complex vector with constant amplitude and linearly increasing frequency. The key novelty of ZChirp is to further element-wisely multiply the chirp with a vector, called the *Z sequence*, which consists of '1's and '-1's. The name ZChirp reflects the combination of the chirp with the Z sequence. The Z sequence serves dual purposes. First, chirps multiplied with different Z sequences exhibit very low cross-correlations, so that nodes cause low interference to each other, enabling multiple simultaneous transmissions and higher overall network capacity. Second, additional data can be modulated with the *Z modulation*, i.e., using different Z sequences for different data values, which is a new dimension to increase the data rate and proves to be very effective.

While the multiplication with the Z sequence fundamentally changes the chirp, compared to the original LoRa, ZChirp does not suffer undesirable side effects, i.e., ZChirp does not occupy more bandwidth, reduce the communication range, or increase the transmission power and complexity of the nodes. ZChirp occupies the same bandwidth as the original LoRa because multiplying the chirp with a Z sequence basically creates a random vector with energy evenly distributed within the same bandwidth of the chirp. Multiplying the Z sequence also does not reduce communication range because the receiver can still add all samples constructively. In fact, experiments show that ZChirp reaches slightly longer communication range than the traditional chirp modulation due to the high efficiency of the Z modulation. Lastly, ZChirp does not change the transmission power nor significantly increase the complexity of the nodes because the Z sequence only changes the signs of the transmitted samples.

The key challenge in ZChirp is the tradeoff between the receiver complexity and the data rate. That is, to modulate more data, the pool of the Z sequences should be larger, which means that the receiver must test more Z sequences. This challenge is addressed by designing the Z sequence with a layered structure, as well as further optimizations in implementation, so that the demodulation complexity of a ZChirp symbol is significantly reduced. To be more exact, with the current design, the number of complex multiplications and additions are no more than $9N$ and $N^{2.5}/2^{0.5}$, respectively, where N is the length of the chirp. In comparison, to demodulate a traditional chirp with the Fast Fourier Transform (FFT), the number of complex multiplications and additions are both $O(N \log N)$. As addition is significantly simpler than multiplication, the demodulation complexity of ZChirp is comparable to that of the traditional chirp modulation when N is not large. ZChirp is still applicable to longer chirps by reducing the number of Z sequences.

ZChirp has been demonstrated with real-world experiments in POWDER [6], which is an open platform in the University of Utah with radios that can be controlled remotely. In the experiment, the gateway could successfully decode most packets transmitted by multiple nodes at random times. Simulations further show that the network capacity of ZChirp is $2.0\times$ that of CurvingLoRa [15], which is the state of the art, and at least

an order of magnitude higher than the original LoRa.

The rest of the paper is organized as follows. Section II discusses related work. Section III gives the background of chirp modulation. Section IV gives an overview of ZChirp. Section V explains details of the Z sequence. Section VI describes techniques to further reduce the computation complexity. Section VII explains the feasibility of ZChirp. Section VIII describes the evaluation. Section IX concludes the paper.

II. Related Work

LoRa [3] has attracted significant interest in recent years because of its simplicity, long communication range, and existing global footprint. LoRa faces challenges because its Medium Access Control (MAC) protocol allows nodes to transmit packets at any time, while its Physical Layer (PHY) was not designed to decode collided packets with the same Spreading Factor (SF). As a result, the uplink capacity could be severely limited due to collisions. Many solutions have been proposed to decode collided packets without modifying the LoRa nodes, such as Choir [9], FTrack [22], CIC [21], TnB [20], and Hi^{2}LoRa [8]. However, the current chirp modulation in LoRa makes it very challenging to effectively identify and separate signals from different nodes.

An alternative is to rethink the chirp modulation [13], [17], [10], [12], [15], [4]. A good representative that aiming at improving the network capacity is CurvingLoRa [15], which modifies the chirp by increasing the frequency non-linearly. Due to the non-linearity, interference is spread out, rather than focusing at a single point, leading to a capacity increase of more than 5-fold over the original LoRa. ZChirp also modifies the chirp but in a completely different manner because ZChirp multiplies a binary sequence with the chirp which has not been attempted before.

Although ZChirp uses the Z sequence which resembles the spreading sequences in Code Division Multiple Access (CDMA) [18], ZChirp is fundamentally different from CDMA because the Z sequences in ZChirp can be used by any node while disjoint spreading sequences are assigned to different users in CDMA. ZChirp is specifically designed for LPWANs to support asynchronous transmissions from the nodes and tolerates power difference as large as 20 dB as shown in Section VIII. CDMA is optimized for cellular networks and needs tight power control to alleviate the near-far effect.

ZChirp is a physical layer solution and therefore complements those in the higher layers, such as adding own application layer codes [16] or enhancing the MAC layer [11]. Lastly, LR-FHSS has been recently proposed as a new modulation option for LoRa; however, accroding to [7], the network capacity is around 3 kbps with 137 kHz channel bandwidth which is significantly lower than ZChirp.

III. Background of Chirp Modulation

Spreading Factor (SF) is a small positive integer, e.g., 6. The chirp, denoted as C, is a complex vector of length $N = 2^{SF}$. C is defined as

$$C_t = \exp\left[i\pi(t - N/2)^2/N\right], \quad t = 0, 1, \ldots, N-1, \qquad (1)$$

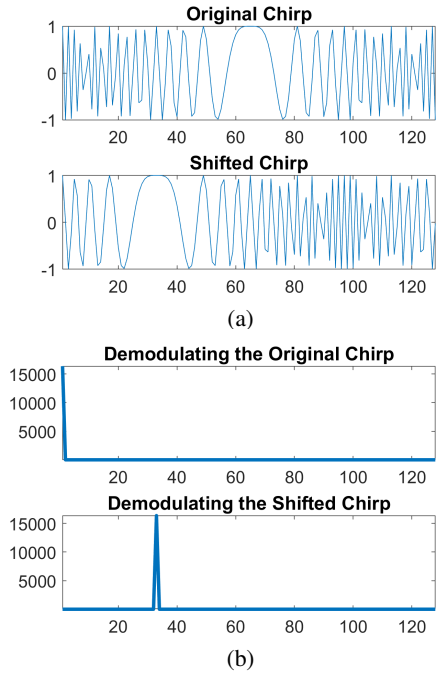


Fig. 1. Chirps with the traditional shift modulation. (a) Modulation. (b) Demodulation.

where C_t denotes element t in C. For example, the top of Fig. 1(a) shows the real part of C when the SF is 7. C is also called the *upchirp*. The conjugate of C is called the *downchirp* and is denoted as C'.

The traditional chirp modulation is cyclic shift and modulates SF bits with a chirp. To modulate a value k, where $0 \leq k < N$, C is cyclically shifted to the left k times, where the shifted chirp is denoted as C^k. For example, the bottom of Fig. 1(a) shows C^{32}. Let Y be the received symbol and "$\odot$" the element-wise multiplication of two vectors. Let $Q = Y \odot C'$, where multiplying Y with C' is called *dechirping*. The *signal vector*, denoted as S, is computed with FFT, i.e., $S = FFT(Q)$. If $Y = C^k$, Q is a sinusoid with k cycles, so that S has a peak at location k. The power of the signal vectors when $Y = C$ and $Y = C^{32}$ are shown in Fig. 1(b) with peaks at locations 0 and 32, respectively.

IV. Overview of ZChirp

ZChirp supports uplink communications from nodes to the gateway and is motivated by the possibility of improving the capacity of LoRa while preserving its known advantages due to the use of chirp.

A. Data Modulation and Demodulation

A data *symbol* in ZChirp is a transformed version of the chirp. To increase modulation density, the transformation is multi-dimensional, that is, in addition to the traditional shift modulation, two additional modulations are adopted, namely, the *phase modulation* and the *Z modulation*. The phase modulation is the standard QPSK, i.e., the chirp is multiplied with one of the values in $\{e^0, e^{\frac{i\pi}{2}}, e^{i\pi}, e^{\frac{i3\pi}{2}}\}$, and therefore

modulates 2 bits. The Z modulation multiplies the chirp with a selected Z sequence element-wisely. The total number of Z sequences is denoted as G; therefore, the Z modulation modulates $\log_2 G$ bits. The generation of Z sequences is discussed in Section V; for now, a Z sequence can be thought of as a random binary vector consisting of '1's and '-1's.

To be more specific, let the values to be modulated with the phase, shift, and Z modulations be t, k, and h, respectively, where $0 \leq t < 4$, $0 \leq k < N$, and $0 \leq h < G$. Let Z sequence h be Z^h. The modulated symbol is

$$\Omega_{t,k,h} = e^{\frac{it\pi}{2}} C^k \odot Z^h. \tag{2}$$

Let the received symbol be Y and let $Q = Y \odot C'$. The most straightforward approach of demodulation is to calculate

$$\Xi^l = FFT(Q \odot Z^l) \tag{3}$$

for every $0 \leq l < G$. Clearly, if $Y = \Omega_{t,k,h}$, when $l = h$, $\Xi^l = FFT(e^{\frac{it\pi}{2}} W^k)$, where W^k is a sinusoid that completes k cycles in N samples, i.e., $W_t^k = e^{i2\pi kt/N}$ for $0 \leq t < N$. As a result, Ξ^l has a peak at k with phase $\frac{t\pi}{2}$. On the other hand, as different Z sequences have low cross-correlations, Ξ^l has no peaks when $l \neq h$. Therefore, the receiver can calculate a matrix denoted as Ξ, where row h is Ξ^h. If the highest peak is at row h and column k, the demodulated data for the Z and shift modulations are h and k, respectively. The phase value can then be found according to the phase of the peak. It should be noted that the demodulation complexity can be significantly reduced, which is discussed in Sections V and VI.

For example, Fig. 2(a) shows Z^{11} multiplied with C^{32}, which is the shifted chirp in Fig. 1(a). Fig. 2(b) shows the power of Ξ, which has a peak at row 11 and column 32, matching the modulated data. It can be seen that the power at any other point is very low, confirming the low cross-correlation between different ZChirp symbols.

The data to be transmitted is split into three parts to be modulated by the phase, shift, and Z modulations, respectively. Each employs independent error correction codes, which is the Turbo code [5] with coding rate 1/2 for the phase modulation, and the Reed-Solomon (RS) code with coding rate 1/2 for the shift and Z modulations. The PHY layer data rate can be calculated as follows with $SF = 7$ and $G = N$ as an example. On each symbol, a total of $2/2 + 7/2 + 7/2 = 8$ coded bits are modulated. With 125 kHz bandwidth, each symbol is 1.024 ms, therefore the data rate is 7.81 kbps.

In this paper, ZChirp is illustrated for SFs 6 to 9 when $G = N$. Other options, e.g., using a smaller G to reduce the computation complexity, can be further explored and are left to future work.

B. Node-Side Operations and Complexity

Node-side operations with ZChirp are very simple. When a node has data to transmit, it picks an SF depending on the channel condition, and transmits the packet. A packet consists of the preamble, the PHY header, and the data. The preamble allows the gateway to perform *packet acquisition*, i.e., to detect the packet and estimate key parameters, such as the symbol boundary and Carrier Frequency Offset (CFO). ZChirp adopts *ChirpPair* as its preamble, which consists of an upchirp followed by a downchirp both with spreading factors $SF+2$ where SF is the spreading factor of the data symbols. The PHY header allows the gateway to learn information such as the length of the packet. In ZChirp, the numbers of symbols in the PHY header are 8, 8, 6, and 6 for SFs 6, 7, 8, and 9, respectively. The details of the packet acquisition method with ChirpPair can be found in [23].

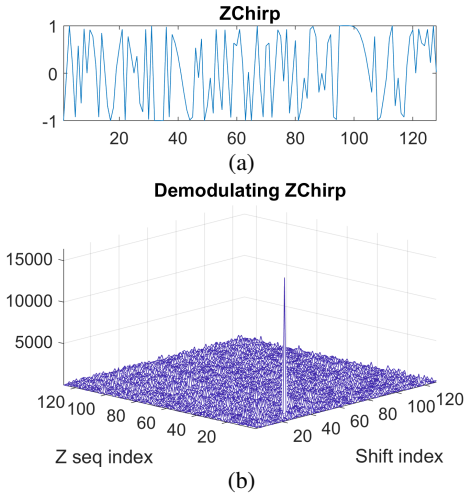


Fig. 2. (a) Z^{11} multiplied with C^{32}. (b). The power of Ξ, which has a peak at row 11 and column 32.

The node complexity is low because a node only needs to encode the error correction code and modulate the data. The heavy lifting, i.e., packet acquisition, coherent demodulation, and decoding the error correction code, are carried out by the gateway. The encoding of the Turbo and RS codes involve only simple linear calculations. The Z and phase modulations are both very simple, because they only change the phases of elements in the chirp by a multiple of 90 degrees. The transmission power is not affected because the magnitude of the elements are not changed.

C. Gateway-Side Operations

The gateway continuously search for packets for every SF. Once a packet is detected, the gateway attempts to decode the packet. If the decoding is successful, Successive Interference Cancellation (SIC) is performed, i.e., the signal of the packet is reconstructed and removed from the time-domain signal. A simple method is used for signal reconstruction, i.e., the peak in the signal vector, such as the peak in Fig. 2(b), is "carved out" to reconstruct the time-domain signal by reversing the demodulation process. After removing the signals of the strong nodes, more packets can be detected. Therefore, the gateway runs the packet detection process for a total of 3 times and decodes any newly detected packets.

V. Z Sequence Design

The core of ZChirp is the Z sequence design, which is discussed in this section.

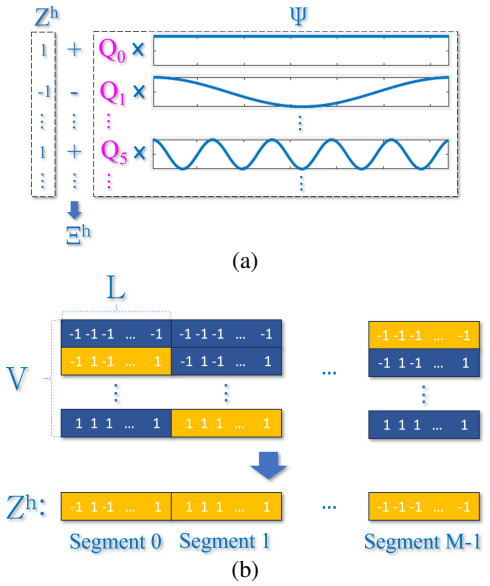


Fig. 3. (a). Obtaining Ξ^h by adding or subtracting rows in Ψ. (b). The construction of the Z sequence.

A. Layers in the Z Sequence

As explained earlier, the main challenge in ZChirp is the tradeoff between demodulation complexity and data rate. It could be expensive to demodulate Q according to Eq. 3 which requires G FFTs each on N points. To reduce the computation complexity, the key idea is to construct Z sequences with multiple *layers*, because computations in the lower layer can be considered as finding *summaries* of the signal, which need not be repeated in the higher layer.

Owing to the binary nature of the Z sequence, the layered design is possible because the conversion from Q to Ξ can be carried out by adding or subtracting rows in a matrix rather than performing FFTs. To elaborate, let W^{-k} be a sinusoid that completes k cycles in N samples with a negative frequency, i.e., $W_t^{-k} = e^{-i2\pi kt/N}$ for $0 \le t < N$. According to the definition of DFT,

$$\Xi_k^h = \sum_{t=0}^{N-1} W_t^{-k} Q_t Z_t^h = \sum_{t \in Z^{h+}} W_t^{-k} Q_t - \sum_{t \in Z^{h-}} W_t^{-k} Q_t \quad (4)$$

where Z^{h+} and Z^{h-} denote the indices of the positive and negative elements in Z^h, respectively. Let Ψ be a N by N matrix where row t, denoted as Ψ^t, is W^{-t} multiplied by Q_t. As a result, Ξ^h can be obtained by adding and subtracting rows in Ψ, as shown in Fig. 3(a).

Clearly, the complexity is still high if every row of Ψ is directly involved in the computation of every Z sequence. Therefore, *layers* are introduced. That is, Ψ is evenly partitioned into M submatrices, denoted as ψ^0, ψ^1, ..., ψ^{M-1}, where each submatrix contains a certain number of rows. The computations involve those in the *lower layer* and the *higher layer*, where the former refers to evaluating the submatrices to find their *summaries*, while the latter refers to combining the summaries into the final result. To find a summary of a submatrix is to add or subtract rows in the submatrix according to a binary vector, which is a *segment* in the Z sequence. A total of V summaries are found for each submatrix, which is simple because the submatrices are much smaller than Ψ and V is much smaller than N. In the higher layer, combining M summaries is also simple because M is much less than N.

The details of the layered design are explained in the following. To match the submatrices, a Z sequence is partitioned into M *segments*, where segment m corresponds to ψ^m. The number of bits in segment m is the same as the number of rows in ψ^m. Let $L = \lfloor N/M \rfloor$. If N is a multiple of M, each segment contains L bits; otherwise, each segment contains L or $L+1$ bits. For simplicity, in the following, Z sequence design is explained for the former case because the latter is a simple extension. Each segment may take one of V values, where each value is a binary vector of length L and is called a *segment pattern*. ψ^m is basically evaluated V times according to the segment patterns of segment m. For simplicity, all segments share the same pool of segment patterns. Segment pattern u is denoted as η^u for $0 \le u < V$. Clearly, a Z sequence, say, Z^h, is the concatenation of segment patterns where different segments may pick different patterns. The related concepts are shown in Fig. 3(b), where segments 0, 1, and $M-1$ pick η^1, η^{V-1}, and η^0, respectively.

The choice of segment patterns for segments in Z^h is determined by a vector of length M, which is called the *codeword* and is denoted as Θ^h. Suppose $\Theta^h = [u_0^h, u_1^h, ..., u_{M-1}^h]$, where each element is an integer within $[0, V-1]$. Let $\Pi^m()$ be a permutation of V numbers for segment m. The permutations are different for different segments to further reduce the similarities of the Z sequences. Let $\Pi^m(u)$ be value u in $\Pi^m()$. Formally, Z^h is given by

$$Z^h = \eta^{\Pi^0(u_0^h)} || \eta^{\Pi^1(u_1^h)} || ... || \eta^{\Pi^{M-1}(u_{M-1}^h)}, \quad (5)$$

where "||" denotes vector concatenation.

Let $\lambda_{m,u}$ be the summary of ψ^m evaluated according to $\eta^{\Pi^m(u)}$. $\lambda_{m,u}$ is a vector of length N and is given by

$$\lambda_{m,u} = \sum_{t \in \eta^{\Pi^m(u)+}} \psi^{m,t} - \sum_{t \in \eta^{\Pi^m(u)-}} \psi^{m,t}, \quad (6)$$

where $\psi^{m,t}$ denotes row t of ψ^m, while $\eta^{\Pi^m(u)+}$ and $\eta^{\Pi^m(u)-}$ denote the indices of the positive and negative elements in $\eta^{\Pi^m(u)}$, respectively. Clearly,

$$\Xi^h = \sum_{m=0}^{M-1} \lambda_{m,u_m^h}. \quad (7)$$

B. Computation Complexity

To demodulate a ZChirp symbol, the computation includes obtaining Ψ, finding summaries of the submatrices, and combining the summaries according to the codewords. To obtain Ψ is to multiply Q with the sinusoids, which, naively, would need N^2 complex multiplications. However, as explained in Section VI-A, the number of complex multiplications can be reduced to $9N$. The rest of the computation involves only complex additions. To obtain all summaries of all submatrices, no more than $MVLN$ additions are needed. Each row vector

in Ξ is the summation of the summaries, which can be obtained with no more than MN additions. Therefore, the total number of additions is no more than $MVLN + MN^2$, which can be simplified to $(V + M)N^2$ because $ML = N$. As will be explained shortly in Section V-C and Section VI-B, $V > M$ and $V \leq \sqrt{2N}$. With the fast addition method discussed in Section VI-B, the number of additions is further reduced by a factor of 4. The number of complex additions is therefore no more than $N^{2.5}/2^{0.5}$. In contrast, a naive implementation will need $N^2(N-1)$ additions. The savings factor related to additions is therefore more than $\sqrt{N}$.

C. Codeword Generation with the RS code

The codewords should be as different from each other as possible, because sharing a common element between two codewords means sharing L common bits between two Z sequences. Therefore, the Reed-Solomon (RS) code is used to generate the codewords, because the RS code guarantees a minimum distance between the codewords. With the RS code, a codeword is the concatenation of *data symbols* and *parity checking symbols*, where the latter is generated by transforming the former in a finite field. To minimize the number of common elements between two codewords, currently, the number of data symbols is 2, so that the RS code generates V^2 unique codewords and guarantees that any two codewords share at most one common symbol. The constraints of the parameters include the following. First, V must be a power of 2, i.e., $V = 2^v$ for some integer v, because RS code is typically defined in a finite field where the number of elements is a power of 2. Second, $V > M$, because the maximum length of the codeword with the RS code is $V - 1$. Third, $V^2 \geq N$, because the number of codewords must be no less than the number of Z sequences.

D. Choice of Parameters

The key parameters of ZChirp are M, L, and V. Fewer segments lead to less computations but also a larger L and a higher cost of sharing an identical segment. Currently, M is chosen according to a simple rule that leads to provable theoretical guarantees. That is, when SF is even, $M = 2^{\frac{SF}{2}} - 1$; otherwise, $M = 2^{\frac{SF-1}{2}}$. As a result, it can be verified that when SF is even, $L = 2^{\frac{SF}{2}} + 1$; otherwise, $L = 2^{\frac{SF+1}{2}}$. Lastly, as V should be greater than M but no less than $\sqrt{N}$, when SF is even, $V = 2^{\frac{SF}{2}}$; otherwise, $V = 2^{\frac{SF+1}{2}}$.

E. Segment Pattern Generation

As the basic building blocks of Z sequence, segment patterns should differ from each other as much as possible. Therefore, for any $u_1 \neq u_2$, the Hamming distance between η^{u_1} and η^{u_2} should be maximized. In addition, as the elements are either 1 or -1 and the receiver chooses the point with the highest power, the Hamming distance between η^{u_1} and $-\eta^{u_2}$, called the *negative Hamming distance*, should also be minimized. For example, even if η^{u_1} and η^{u_2} differ in every location, the demodulation results have the same magnitude and just opposite signs. Therefore, the quality of the segment patterns should be measured by the *double minimum Hamming distance (DMHD)*, which is the minimum of the original Hamming distance and the negative Hamming distance.

The segment patterns are generated according to a linear binary code then replacing the '0's with '-1's. A generator matrix with v rows and L columns generates 2^v segment patterns each with length L. The construction of the generator matrix is very simple. Let the *base matrix*, denoted as B_v, be a v by 2^v binary matrix, in which column a is the binary representation of a for $0 \leq a < 2^v$. For example,

$$B_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

A generator matrix with v rows and L columns is obtained by concatenating B_v horizontally $\lceil L/2^v \rceil$ times and taking the first L columns. The DMHD of the base matrix is proved in the following.

Theorem 1. *$D(B_v) = 2^{v-1}$, where $D(B_v)$ denotes the DMHD of segment patterns generated according to B_v.*

Proof. Based on induction. The claim is clearly true for B_2, because the segment patterns are 0000, 0011, 0101, and 0110, which are obtained by multiplying row vectors [00], [10], [01], and [11] with B_2, respectively. Suppose the claim is still true till $v-1$. Note that B_v can be obtained by concatenating two B_{v-1} horizontally then adding a new row in which the first half is '0' and the second '1'. The difference between any two segment patterns is a linear combination of rows in B_v. If the new row is not involved, due to the induction hypothesis, there are exactly 2^{v-2} elements that are '1' in both the first and second half, therefore the claim is true. If the new row is involved, the claim is still true, because the '1' in the second half are all flipped to '0' and '0's to '1'. □

The generator matrices are analyzed further in the following, where the maximum DMHD for given L and v is denoted as $\Gamma_{L,v}$.

Lemma 1. *If $L = 2d + 1$ for some integer d, $\Gamma_{L,v} \leq d$ for any v.*

Proof. Suppose the claim is not true. Without Loss of Generality (WLOG), consider the all-0 segment pattern and an arbitrary segment pattern denoted as η. If the minimum Hamming distance is $d + 1$, η should have at least $d + 1$ '1's; similarly, if the negative minimum Hamming distance is $d + 1$, η should have at least $d + 1$ '0's. Clearly, this is not possible with only $2d + 1$ bits. □

Lemma 2. *If $L = 2d$ for some odd integer d, $\Gamma_{L,v} \leq d - 1$ for any $v > 1$.*

Proof. Suppose the claim is not true. Therefore, any non-zero segment pattern must have exactly d '1's and d '0's. WLOG, consider a segment pattern η_1 in which the first half of the bits are '1' and the second half '0'. As $v > 1$, there is another segment pattern η_2. Suppose the number of '1's in the first half of η_2 is x. As a result, the Hamming and negative Hamming

distances between η_1 and η_2 are $2d-2x$ and $2x$, respectively, which is a contradiction because $\min\{2d-2x, 2x\} < d$. □

Theorem 2. *The generator matrices used for segment pattern generation are theoretically optimum in the sense that they achieve the maximum DMHD.*

Proof. When SF is even, the generator matrices have $\frac{SF}{2}$ rows but L or $L+1$ columns, where $L = 2^{\frac{SF}{2}}+1$. The DMHDs of which are both types of matrices are at least $2^{\frac{SF}{2}-1}$, as padding additional columns does not reduce the DMHD. The DMHD of the first type of matrices is optimal based on Lemma 1. The DMHD of the second type of matrices is optimal based on Lemma 2. When SF is odd, there is only one type of generator matrix with $\frac{SF+1}{2}$ rows and $2^{\frac{SF+1}{2}}$ columns. The DMHD of the matrix is $2^{\frac{SF-1}{2}}$, which is optimal based on similar arguments as those in the proof of Lemma 1. □

F. DMHD of the Z Sequences

The following theorem gives the DMHD of the Z sequences, which is defined as the minimum of the original Hamming distance and the negative Hamming distance among all pairs of Z sequences.

Theorem 3. *The DMHD of the generated Z sequences is at least $\frac{N}{2}-\sqrt{N}$ when SF is even and $\frac{N}{2}-\sqrt{\frac{N}{2}}$ when SF is odd, where N is the length of the chirp.*

Proof. With the RS code, two codewords differ in at least $(M-1)$ elements. Clearly, the DMHD of the Z sequences is at least $(M-1)$ times the DMHD of segment patterns, which is at least $2^{v-1}\lfloor L/2^v \rfloor$. As $L = \lfloor N/M \rfloor$, the bound is

$$(M-1)2^{v-1}\lfloor \frac{N}{M2^v} \rfloor. \tag{8}$$

When SF is even, $M = 2^{\frac{SF}{2}}-1$ and $v = \frac{SF}{2}$. It can be verified that $\lfloor \frac{N}{M2^v} \rfloor = 1$. Therefore, the bound is

$$(M-1)2^{v-1} = (2^{\frac{SF}{2}}-1-1)2^{\frac{SF}{2}-1} = \frac{N}{2}-\sqrt{N}. \tag{9}$$

When SF is odd, $M = 2^{\frac{SF-1}{2}}$ and $v = \frac{SF+1}{2}$. It can be verified that $\lfloor \frac{N}{M2^v} \rfloor = 1$. Therefore, the bound is

$$(M-1)2^{v-1} = (2^{\frac{SF-1}{2}}-1)2^{\frac{SF+1}{2}-1} = \frac{N}{2}-\sqrt{\frac{N}{2}}. \tag{10}$$

□

The ideal DMHDs is $\frac{N}{2}$ when the Z sequences are orthogonal. Based on Theorem 3, the generated Z sequences are at most $\sqrt{N}$ away from the optimal, which means that the power of any peak other than the actual transmitted peak is at most a fraction of $4/N$ of the latter, because most samples are canceled with each other and no more than $2\sqrt{N}$ samples add up constructively. The actual performance is better because the bound considers the worst case.

G. Mapping of Segment Patterns

It was found that some codewords generated by the RS code exhibit regular differences, leading to relatively high correlations between a small number of Z sequences. A simple but effective solution is to change the mapping of segment indices to segment patterns by applying different permutations to different segments, which has been described earlier. Currently, $\Pi^m(u) = \mod[(2m+1)u(u+1)/2, V]$.

VI. Further Simplifications

The demodulation complexity can be further reduced with some tricks in implementation, as is explained in this section.

A. Obtaining Ψ

To obtain Ψ, naively, one will need N complex multiplications for each row of Ψ and N^2 in total. Fortunately, owing to the simplicity of Ψ, the number of multiplications can be reduced to 9 for each row and $9N$ in total, if a very slight loss of accuracy can be accepted. This is explained in the following for a generic row denoted as Ψ^t.

First, let $\Phi = [1, e^{\frac{i2\pi}{N}}, e^{\frac{i4\pi}{N}}, \ldots, e^{\frac{i2(N-1)\pi}{N}}]$, and consider multiplying $|Q_t|$ with Φ. Actual multiplications are needed only for $1/8$ of values in Φ, i.e., those with phases in $[0, \pi/4]$, because the rest can be obtained by performing one or two very simple additional operations, namely, swapping the real and imaginary parts and flipping the sign. Second, after approximating the phase of Q_t as a multiple of $\frac{2\pi}{N}$, $Q_t\Phi$ can be obtained by cyclically shifting $|Q_t|\Phi$ according to the phase of Q_t. Lastly, Ψ^t can be obtained by relocating elements in $Q_t\Phi$ because unique values in W^{-t} are a subset of Φ.

At this point, $N/8+1$ multiplications are still needed to obtain Ψ^t. However, as $[1, e^{\frac{i2\pi}{N}}, e^{\frac{i4\pi}{N}}, \ldots, e^{\frac{i\pi}{4}}]$ are evenly-spaced constants, at a very slight loss of accuracy, multiplications of only 9 points, i.e., $[1, e^{\frac{i\pi}{32}}, e^{\frac{i\pi}{16}}, \ldots, e^{\frac{i\pi}{4}}]$, are needed, because the rest can be approximated by simple linear interpolation. Division in the interpolation is simply shifting because the number of points to be interpolated between two known points is a power of 2 minus 1.

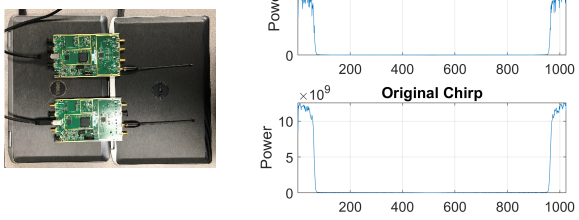
B. Fast Summation

When computing Ξ, fast summation is possible by exploiting the nature of values in Ψ with a saving factor of 4. Let $t' = \mod(t, 4)$, which is called the *type* of Ψ^t. Consider Ψ^t_k, where $0 \le k < N/4$. For $0 \le a < 4$,

$$\Psi^t_{k+aN/4} = Q_t e^{-i2\pi kt/N} e^{-iat'\pi/2} = \Psi^t_k e^{-iat'\pi/2}. \tag{11}$$

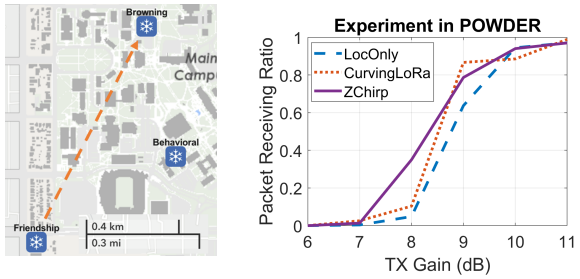
Consider 4 cases depending on t':

- $t' = 0$: for all $0 \le a < 4$ and $0 \le k < N/4$, $\Psi^t_{k+aN/4} = \Psi^t_k$;
- $t' = 1$: for $0 \le k < N/4$, $\Psi^t_{k+N/4} = \Im[\Psi^t_k] - i\Re[\Psi^t_k]$, while for $0 \le k < N/2$, $\Psi^t_{k+N/2} = -\Psi^t_k$;
- $t' = 2$: for $0 \le k < N/4$, $\Psi^t_{k+N/4} = -\Psi^t_k$, while for $0 \le k < N/2$, $\Psi^t_{k+N/2} = \Psi^t_k$;
- $t' = 3$: for $0 \le k < N/4$, $\Psi^t_{k+N/4} = -\Im[\Psi^t_k] + i\Re[\Psi^t_k]$, while for $0 \le k < N/2$, $\Psi^t_{k+N/2} = -\Psi^t_k$.



(a) (b)

Fig. 4. Bandwidth test. (a) USRP used in the test. (b) Frequency usage.



(a) (b)

Fig. 5. Communication performance test. (a) Node locations. (b) PRRs of the compared schemes.

Therefore, Ψ can be partitioned into 4 submatrices, one for each type. For each submatrix, summations are needed only for the first quarter of the columns because the rest can be obtained by performing one or two very simple additional operations, namely, swapping the real and imaginary parts and flipping the sign. The number of additions is therefore reduced by a factor of 4. To combine fast summation with the layered design, a Z sequence segment should be assigned rows of the same type. Summaries of the same type can also be combined with fast summation. There are up to 4 segments that are assigned rows of two types, where each type can be processed separately. Overall, the saving factor is still close to 4.

VII. Feasibility of ZChirp

The feasibility of ZChirp related to bandwidth usage and communication distance are discussed in this section.

A. Bandwidth Usage

Bandwidth is a key parameter in communication networks that limits how fast the signal may change. As the Z sequence flips the signs of elements in the chirp, one concern is if ZChirp uses more bandwidth than the original LoRa. Fortunately, *if the Z sequence approximates a random binary vector, multiplying a Z sequence with the chirp basically creates a random vector of length N, the spectrum of which resembles that of noise with energy evenly distributed in N FFT coefficients*. Therefore, ZChirp uses the same bandwidth as the original LoRa, because the spectrum of the chirp is also evenly distributed in the same range.

Fig. 4 shows an experimental validation of the claim above. In the experiments, signal was transmitted and received by 2 USRPs shown in Fig. 4(a), where the carrier frequency was 910 MHz, the bandwidth was 250 kHz, the SF was 7, and the Over Sampling Factor (OSF) was 8, i.e., the receiver takes 8 samples for each transmitted sample. Fig. 4(b) shows the average of the FFTs of 50 symbols, each with 1024 samples. When the OSF is 8, the signal should occupy $1/8$ of the spectrum, which is true for both the original chirp and the chirp multiplied with the Z sequence.

B. Communication Performance

Another concern is likely about the communication performance, i.e., the ability to decode weak signals, because the chirp has been completely modified. Fortunately, *ZChirp does not degrade the communication performance because weak signals are decoded correctly by adding samples constructively rather than relying on the special structure of the chirp*.

To verify the claim above, a head-to-head comparison was conducted between ZChirp and a scheme called *LocOnly* which represents the traditional shift modulation. LocOnly shares the same code base as ZChirp and was implemented by keeping only the shift modulation and disabling the phase and Z modulations of ZChirp. The comparison therefore reveals the differences only related to data modulation and is not affected by other issues such as error correction, packet acquisition, etc. CurvingLoRa was also compared and was implemented in a similar manner as explained in Section VIII-A. Experiments were conducted in the POWDER platform [6]. As shown in Fig. 5(a), packets were transmitted from rooftop radio "Friendship" to rooftop radio "Browning." For all schemes, the packet size was 64 bytes, the carrier frequency was 3.515 GHz, the bandwidth was 125 kHz, and the OSF was 8. The SF of ZChirp was 7, while those for LocOnly and CurvingLoRa were both 6, because the communication performance should be measured with similar data rates. Note that ZChirp modulates $2SF + 2$ bits per symbol while both LocOnly and CurvingLoRa modulate SF bits per symbol. Therefore, using the same SF is clearly not fair. Even when ZChirp uses SF 7, its data rate is still $1.33\times$ of LocOnly and CurvingLoRa with SF 6. Fig. 5(b) shows the Packet Receiving Ratios (PRR) as a function of the transmitter gain. It can be seen that the PRR of ZChirp is slightly better than both LocOnly and CurvingLoRa. Therefore, ZChirp should achieve a slightly longer communication range with a higher data rate than both the traditional shift modulation and CurvingLoRa. This proves that the Z modulation is an effective option to increase the data rate which is due to the low cross-correlations between Z sequences.

VIII. Evaluations

ZChirp has been evaluated with both over-the-air experiments and simulations. The performance of a scheme is measured by the *network capacity*, which is defined as the network throughput when the PRR is above 0.9.

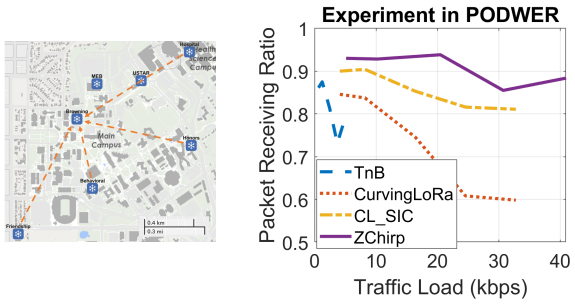


Fig. 6. Network capacity test. (a). Node locations. (b). PRRs of the compared schemes.

A. Experiments in the POWDER Platform

Experiments were conducted in the POWDER platform [6]. As shown in Fig. 6, rooftop radio "Browning" acted as the gateway, while "Behavioral (Bes)," "Friendship (FM)," "Hospital" and "Honors" acted as nodes with transmission gains of 23 dB, 20 dB, 17 dB, and 31.5 dB, respectively. In each experiment, each node transmitted the same number of packets for 10 seconds. The number of packets depends on the traffic load; at the highest load, each node transmitted 200 packets. The packet transmission times were random. All packets transmitted by the same node had the same power but with CFO randomly selected from $[-4.88, 4.88]$ kHz. Each experiment was repeated 3 times. The carrier frequency was 3.515 GHz, the bandwidth was 125 kHz, the OSF was 8, and the packet size was 64 bytes.

For comparison, TnB [20] and CurvingLoRa [15] were also tested, which represent solutions that do not modify and modify the LoRa nodes, respectively. For TnB, the original implementation at [19] was used. CurvingLoRa was implemented by combining its original code at [14] with the same code base of ZChirp. That is, the original code were used to generate data symbols, while the rest, e.g., packet acquisition, error correction, were the same as those in ZChirp. This is because CurvingLoRa proposes no changes to LoRa except modifying the chirp. ZChirp, on the other hand, enjoys improvements due to other modifications such as better error correction. Incorporating the chirp design of CurvingLoRa into the ZChirp framework therefore levels the playing field and allows for a more fair comparison. As the original CurvingLoRa does not support Successive Interference Cancellation (SIC), SIC was designed and CurvingLoRa with SIC is referred to as *CL_SIC*.

To maximize the traffic load, in the experiments, the SF of ZChirp was 6. The SF of CurvingLoRa was 5 to match the data rate of ZChirp. The SF of TnB was 7 because it was the smallest SF supported by TnB. The PRR of the schemes are shown in Fig. 6(b), where the gain of ZChirp over TnB and CurvingLoRa can be clearly seen. The capacity of ZChirp in this network 22.7 kbps and its PRR is still close to 0.9 when the load is over 40 kbps. On the other hand, the PRR of CurvingLoRa drops below 0.7 when the load is beyond 20 kbps. CurvingLoRa does not perform well because a strong node can still cause severe interference even after its energy

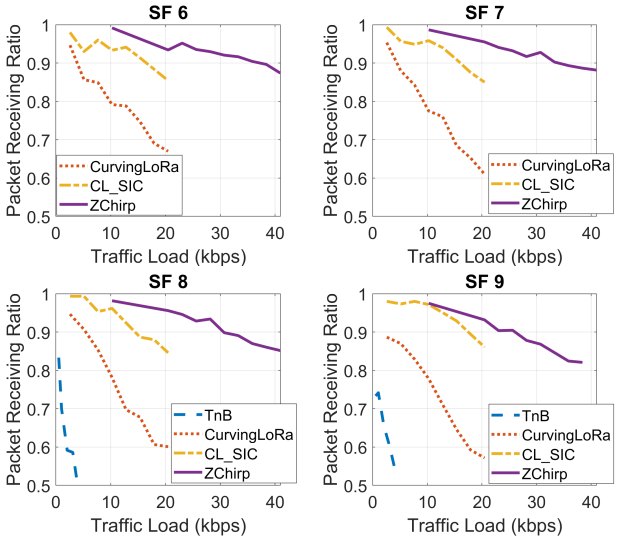


Fig. 7. Single SF simulation results.

has been spread with its non-linear chirp. The benefit of SIC can be clearly seen from CL_SIC, which is much better than CurvingLoRa. Still, the capacity of ZChirp is 2.84× that of CL_SIC in this network.

The experiments serve the important purpose of demonstrating ZChirp in a real-world scenario. The main limitation, however, is the available physical radios that can communicate with each other in a star topology. Therefore, ZChirp was also evaluated with simulations, as explained in the following.

B. Simulations

Simulations were conducted in the LTE ETU channel which has strong multipath components and larger delay spread [1], [2]. The duration of simulation was 10 seconds, the bandwidth was 125 kHz, the packet size was 64 bytes, the OSF was 8, and the CFOs of packets were randomly selected from $[-4.88, 4.88]$ kHz. The number of receiving antennas was 2 in the simulation. For ZChirp, SFs 6, 7, 8, and 9 of were tested. The Signal to Noise Ratio (SNR) of a received packet was randomly selected in [-2, 18] dB, [-4, 16] dB, [-6, 14] dB, and [-8, 12] dB for ZChirp packets with SFs 6, 7, 8, and 9, respectively. The range of SNR was 20 dB to simulate errors in transmission power control. As mentioned earlier in Section VII-B, each ZChirp symbol modulates $2SF + 2$ bits while each LoRa or CurvingLoRa symbol modulates SF bits. Therefore, to best match the data rates, the SFs of LoRa and CurvingLoRa were one less than that of ZChirp. The SFs shown in Fig. 7 and Fig. 8 are the SF of ZChirp.

1) Single-SF Tests: In a single-SF test, all packets were transmitted with the same SF. The results in Fig. 7 show that ZChirp outperforms the compared schemes significantly. As TnB could not achieve PRR over 0.9, its capacity is not defined. However, as the PRR of TnB drops below 0.6 when the traffic load is around 3 kbps while the PRR of ZChirp stays above 0.8 when the traffic load is around 40 kbps, ZChirp is clearly capable of decoding over 10 times the number of packets than the orignal LoRa. As the original CurvingLoRa

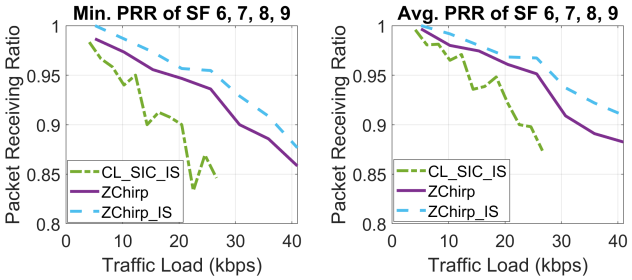


Fig. 8. Simulation results when an equal number of packets were transmitted with SFs 6, 7, 8, and 9.

also could not achieve PRR over 0.9, in the following, the comparison is only between ZChirp and CL_SIC. Specifically, the capacities of ZChirp are 33.5 kbps, 30.6 kbps, 27.5 kbps, and 23.4 kbps for SFs 6, 7, 8, and 9, respectively, while those of CL_SIC are 14.9 kbps, 14.4 kbps, 13.0 kbps, and 15.7 kbps, respectively. Therefore, the gains of ZChirp are 2.25×, 2.12×, 2.12×, and 1.49×, the average of which is 2.0×, which is used to represent the gain of ZChirp.

2) Multiple-SF Tests: In a multiple-SF test, an equal number of packets were transmitted with SFs 6, 7, 8, and 9, respectively. To stay better focused, TnB and the original CurvingLoRa were not included. A new challenge was also encountered for CurvingLoRa, because packets with different SFs cause strong interference to packet acquisition, which is likely due to the non-linear frequency change with CurvingLoRa. Therefore, CL_SIC was further enhanced with ideal packet acquisition and is referred to as CL_SIC_IS; that is, the receiver was provided with the start time and CFO of all packets. ZChirp, on the other hand, does not suffer this problem and need not be changed. However, for a fair comparison, a version of ZChirp, denoted as ZChirp_IS, was tested, in which ZChirp was also enhanced with ideal packet acquisition. Fig. 8 shows the minimum and average PRRs of all SFs. It can be seen that ZChirp still significantly outperforms CurvingLoRa. For example, when measured by the minimum PRR, the capacities of ZChirp_IS and CL_SIC_IS are 33.5 kbps and 18.4 kbps, respectively, with a gain of 1.82×. The gap between ZChirp and ZChirp_IS is small, confirming the effectiveness of the packet acquisition method employed by ZChirp.

IX. Conclusions

ZChirp is a novel wireless data modulation scheme for LPWANs. ZChirp modulates data by combining the chirp with the Z sequence, which enables both simultaneous packet transmissions and higher data rates. In this paper, the complete physical layer of ZChirp is designed, implemented, and tested. Experiments in the POWDER platform confirm that ZChirp packets from multiple nodes can be received correctly over real-world wireless channels. Simulations under the LTE ETU channel model show that the network capacity of ZChirp is 2.0× that of the state of the art.

Acknowledgment

This research work was supported by the US National Science Foundation under Grant 1910268 and Grant 2312113.

References

[1] 3GPP TS 36.101. User Equipment (UE) Radio Transmission and Reception. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network. Evolved Universal Terrestrial Radio Access (E-UTRA).

[2] 3GPP TS 36.104. Base Station (BS) radio transmission and reception. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network. Evolved Universal Terrestrial Radio Access (E-UTRA).

[3] LoRaWAN 1.1 specification. https://www.lora-alliance.org/ resource-hub/lorawantm-specification-v11.

[4] Ali Waqar Azim, Ahmad Bazzi, Roberto Bomfin, Raed Shubair, and Marwa Chafii. Layered chirp spread spectrum modulations for lpwans. *IEEE Transactions on Communications*, 72(3):1671–1687, 2024.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error correcting code and decoding – Turbo Codes (1). In *IEE ICC*, pages 1064–1070, 1993.

[6] Joe Breen and et al. POWDER: Platform for open wireless data-driven experimental research. In *WiNTECH*, September 2020.

[7] Jumana Bukhari and Zhenghao Zhang. Understanding long range-frequency hopping spread spectrum (lr-fhss) with real-world packet traces. *ACM Trans. Sen. Netw.*, 20(6), October 2024.

[8] Weiwei Chen, Shuai Wang, Tian He, Xianjin Xia, and Shuai Wang. Hi2lora: Exploring highly dimensional and highly accurate features to push lorawan concurrency limits with low implementation cost. In *IEEE ICNP*, pages 1–11, 2023.

[9] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yagan. Empowering low-power wide area networks in urban settings. In *ACM SIGCOMM*, pages 309–321. ACM, 2017.

[10] Guillaume Ferré and Mohamed Amine Ben Temim. A dual waveform differential chirp spread spectrum transceiver for leo satellite communications. In *IEEE ICC*, pages 1–6, 2021.

[11] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. LMAC: efficient carrier-sense multiple access for lora. In *ACM Mobicom*, pages 43:1–43:13. ACM, 2020.

[12] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. Netscatter: Enabling large-scale backscatter networks. In *USENIX NSDI*, pages 271–284. USENIX Association, 2019.

[13] Muhammad Ajmal Khan, Raveendra K. Rao, and Xianbin Wang. Performance of quadratic and exponential multiuser chirp spread spectrum communication systems. In *SPECTS*, pages 58–63, 2013.

[14] Chenning Li, Xiuzhen Guo, Longfei Shuangguan, Zhichao Cao, and Kyle Jamieson. CurvingLoRa Implementation. https://github.com/liecn/CurvingLoRa_NSDI22.

[15] Chenning Li, Xiuzhen Guo, Longfei Shuangguan, Zhichao Cao, and Kyle Jamieson. Curvinglora to boost lora network throughput via concurrent transmission. In *Proceedings of USENIX NSDI*, 2022.

[16] Paul J. Marcelis, Nikolaos Kouvelas, Vijay S. Rao, and R. Venkatesha Prasad. Dare: Data recovery through application layer coding for lorawan. *IEEE Transactions on Mobile Computing*, 21(3):895–910, 2022.

[17] Yubi Qian, Lu Ma, and Xuwen Liang. The performance of chirp signal used in leo satellite internet of things. *IEEE Communications Letters*, 23(8):1319–1322, 2019.

[18] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2nd edition, 2002.

[19] Raghav Rathi and Zhenghao Zhang. TnB Implementation. https://github.com/raghavrathi10/TnB.

[20] Raghav Rathi and Zhenghao Zhang. Tnb: resolving collisions in lora based on the peak matching cost and block error correction. In *ACM CoNEXT 2022 Conference, Roma, Italy, December 6-9, 2022*, pages 401–416. ACM, 2022.

[21] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. Concurrent interference cancellation: decoding multi-packet collisions in LoRa. In *ACM SIGCOMM 2021 Conference, Virtual Event, USA, August 23-27, 2021*, pages 503–515. ACM, 2021.

[22] Xianjin Xia, Yuanqing Zheng, and Tao Gu. FTrack: Parallel decoding for LoRa transmissions. *IEEE/ACM Transactions on Networking*, 28(6):2573–2586, 2020.

[23] Zhenghao Zhang. Chirppair: packet acquisition in uncoordinated access channels of low earth orbit (LEO) satellite networks. *EURASIP J. Wirel. Commun. Netw.*, 2024(1):47, 2024.