Pearl: A More Reliable LoRaWAN

Matthew Pana

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2025

Committee:
Akshay Gadre
Jim Ritcey

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Pearl: A More Reliable LoRaWAN

Matthew Pana

Chair of the Supervisory Committee:
Akshay Gadre
Electrical and Computer Engineering

LoRaWAN is the dominant unlicensed low-power wide-area networking (LP-WAN) technology for large-scale IoT deployments, celebrated for its potential to offer decade-long battery life. However, in real-world urban environments, highly dynamic wireless channels degrade this promise as their high variance lead to significant packet loses. These losses are often caused by just a few incorrectly decoded bits, despite the presence of error-correcting codes, resulting in disproportionate retransmission overhead and wasted energy.

This thesis introduces *Pearl*[1], a new decoding architecture that reduces packet failure and improves communication efficiency in LoRaWAN. *Pearl*estimates the posterior probability that each received symbol is correct by modeling the channel as AWGN and using the Rician distribution, then converts these symbol-level probabilities into bit-level likelihoods. Using these, *Pearl*probabilistically selects the most likely transmitted codeword, enabling correction of multiple bit errors beyond what traditional forward error correction alone allows. Evaluated on a campus-scale deployment covering 0.18 km^2, *Pearl*improves client battery life by 1.3×, achieves 2.045× higher throughput, and reduces latency by 52.7% compared to standard LoRaWAN implementations.

---

[1]Probabilistic Error-Aware Repair of LoRaWAN

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

LoRaWAN is the leading unlicensed low-power wide-area networking (LP-WAN) technology, widely deployed across domains such as agriculture, logistics, environmental sensing, and the oil and gas industry. Despite its broad adoption, LoRaWAN is often considered an unreliable backhaul for latency- and reliability-sensitive applications. In such scenarios, users tend to prefer licensed cellular alternatives like LTE-M and NB-IoT. This perception largely stems from the high temporal variability of wireless channel quality in LoRaWAN, which undermines its reliability.

Most LoRaWAN deployments use class A devices that transmit data intermittently using fixed parameters—spreading factor and coding rate—chosen based on outdated information or hardcoded configurations. Since the wireless channel may degrade between transmissions, this often results in failed packet delivery and the need for retransmission. In contrast, cellular technologies like NB-IoT allocate transmission resources just before sending, ensuring more accurate channel estimation and higher decoding success. Moreover, LoRaWAN's inflexible modulation and coding scheme (MCS) limits its adaptability to dynamic channel conditions. Our motivation study in Section 3.2 highlights how these configuration choices impact the reliability-throughput trade-off in LoRaWAN devices.

Prior work has sought to address the effects of signal variability by enhancing packet detection and decoding through neural receivers [14] or multi-base-station signal fusion [6, 3, 16, 18, 7, 25]. Other efforts have used opportunistic frequency configuration [10] to improve performance. However, these methods often require dense base station deployments, extensive environment-specific data, or explicit feedback from the infrastructure. Crucially, none of them tackle the core issue: the temporal variability of channel conditions experi-

enced by the LoRaWAN transceiver. In contrast, this thesis proposes a more pragmatic solution—one that reduces packet failure rates under low channel quality without modifying the base station's physical layer processing.

This thesis introduces *Pearl*, a probabilistic error repair technique for LoRaWAN that significantly reduces packet failures in low signal-to-noise ratio environments. Central to *Pearl*'s approach is its ability to estimate the error likelihood of each bit within a packet. By integrating this probabilistic model with existing forward error correction (FEC) mechanisms, *Pearl* enhances the ability to recover from multiple bit errors per symbol, thereby boosting both reliability and throughput. I evaluated *Pearl* in a campus-scale deployment covering 0.18 square kilometers and show that it improves throughput by 104.5%, reduces latency by half, and cuts packet failure rates by 52.7%.

The key insight behind *Pearl* is that not all decoded symbols are equally likely to be erroneous. The signal processing pipeline inherently assigns varying levels of confidence to decoded symbols. By translating this symbol-level confidence into bit-level error likelihoods, *Pearl* enables a new form of probabilistic error repair. Leveraging these likelihoods in conjunction with existing Hamming code FEC, our system can correct more errors than previously possible. The remainder of this thesis details how bit-level error likelihoods are computed and how they are used for enhanced error repair.

Bit-Level Likelihood for Chirp Spread Spectrum (CSS) Symbols: To start, it is necessary to have a bit-level likelihood metric while decoding. A simple approach would be something akin to measuring the SNR of every symbol and propagating that to every bit encoded in the symbol. However, I have found that calculating the likelihood based on a rician model demonstrates significant improvement in not only estimation of the symbol, but also in the ability for decoding to correct bit errors compared to the naive and standard approaches.

Probabilistic Error Repair in LoRaWAN: I leverage the bit-level error likelihoods to correct for 1-bit, 2-bit and 3+-bit errors in every encoded symbol. I further take into account the Hamming Code constraints to reduce the search space for efficient error correction. My

approach leverages the probabilistically generated bit-flips and performs the error correction aware of these confidence values. Finally, I evaluate the packet checksum to verify successful reception of packet.

I evaluate *Pearl* both theoretically and empirically on a campus scale testbed spanning 300,000 sq. m. The testbed contains various hills, trees, buildings and water bodies as well as frequent pedestrian traffic I evaluate the improvement in throughput and reliability across time by collecting 52610 measurements across 120 locations. My results show:

- a 27.28% improved reliability for LoRaWAN clients in real world environments
- a 1.378× increased throughput in good SNR conditions for LoRaWAN clients
- an average of 52.7% lower latency for retrieving sensor data using LoRaWAN clients.

Contributions: This thesis's contributions are as follows:

- Bit-level error likelihood metric for data encoded in chirp spread spectrum symbols.
- A probabilistic Hamming decoder that can reduce packet failure rate significantly in low SNR conditions
- Extensive theoretical and empirical evaluation over a campus-scale testbed demonstrating improvements in reliability, throughput and latency of LoRaWAN.

Chapter 2

RELATED WORK

Enhancing LoRaWAN Throughput: Extensive research has been conducted to enhance the throughput, energy efficiency, and latency of LoRaWAN clients. A significant portion of this work targets the physical layer, including multi-gateway signal combining [6, 3, 18, 7, 25], optimized client configurations [10, 22, 29], and improved interference resilience [8, 19, 26]. Additionally, machine learning techniques have been employed to detect and decode LoRaWAN packets at lower SNRs [14]. Alternative, non-standard approaches aimed at increasing concurrency have explored the use of non-linear chirps [17, 15], Doppler-based techniques [20, 9], and intentional packet collisions [11]. Recent work also includes machine learning-driven belief propagation and SVM-based symbol error-likelihood estimation [28, 24]. In contrast, *Pearl* is ***complementary*** to these physical layer innovations, as it operates primarily before modulation and after symbol demodulation. Notably, *Pearl* introduces the first data-agnostic, **bit-level** error likelihood estimation for chirp-spread spectrum symbols, along with a protocol-aware, error-sensitive decoding scheme tailored for LoRaWAN.

Forward Error Correction (FEC) in Wireless: FEC is a foundational technique in wireless communication, widely used to mitigate the effects of noise, interference, and other impairments that increase the likelihood of bit errors. For example, 5G New Radio (NR) employs polar codes for control channels [21, 4], while Wi-Fi 6 utilizes LDPC codes [5, 30] to improve reliability. LoRaWAN also incorporates FEC through the use of Hamming codes, which allow for the correction of a single bit error within each byte, typically using 4/7 or 4/8 coding rates [27]. As expected, these error correction capabilities come at the expense of reduced throughput due to added redundancy.

Traditional FEC schemes generally assume that all bits are equally likely to be corrupted.

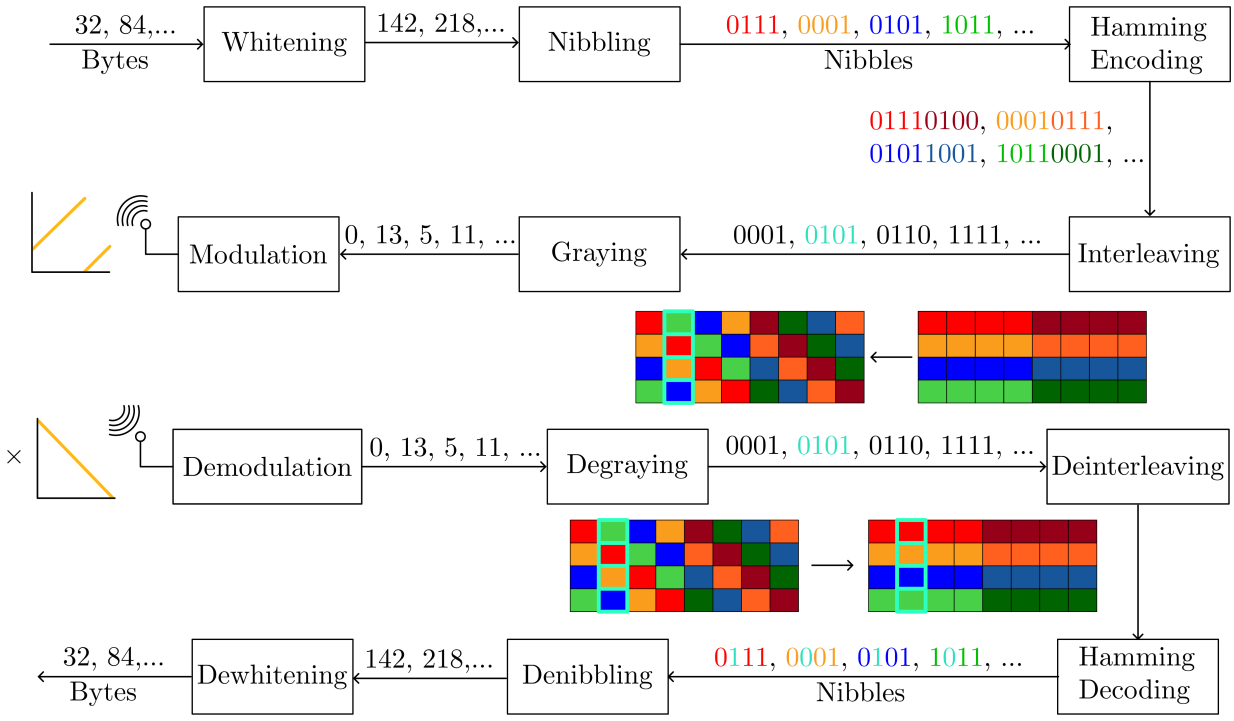

Figure 2.1: LoRaWAN in a nutshell: The incoming bits are whitened and converted to nibbles. These nibbles are hamming encoded with CR bits and passed on to an interleaver before transmission. The receiver decodes these symbols and deinterleaves them. Notice how a single symbol error is spread across multiple coded bytes. The hamming decoder then corrects the bits and dewhitens them to output the bytes. A checksum also operates to verify the received bytes are correct.

In contrast, *Pearl* introduces a key insight: in chirp spread spectrum (CSS) modulation, the likelihood of bit errors varies across the symbol. Building on this, *Pearl* proposes a probabilistic error correction method that estimates bit-level error likelihoods and uses this information to extend the correction capability of LoRaWAN's existing Hamming codes, enabling the recovery of multiple erroneous bits per symbol

Dealing with Channel Variance: Channel variability is a well-recognized challenge across nearly all wireless technologies, and significant research has focused on addressing it [1, 12, 31]. Most existing solutions rely on timely channel feedback from the base station to

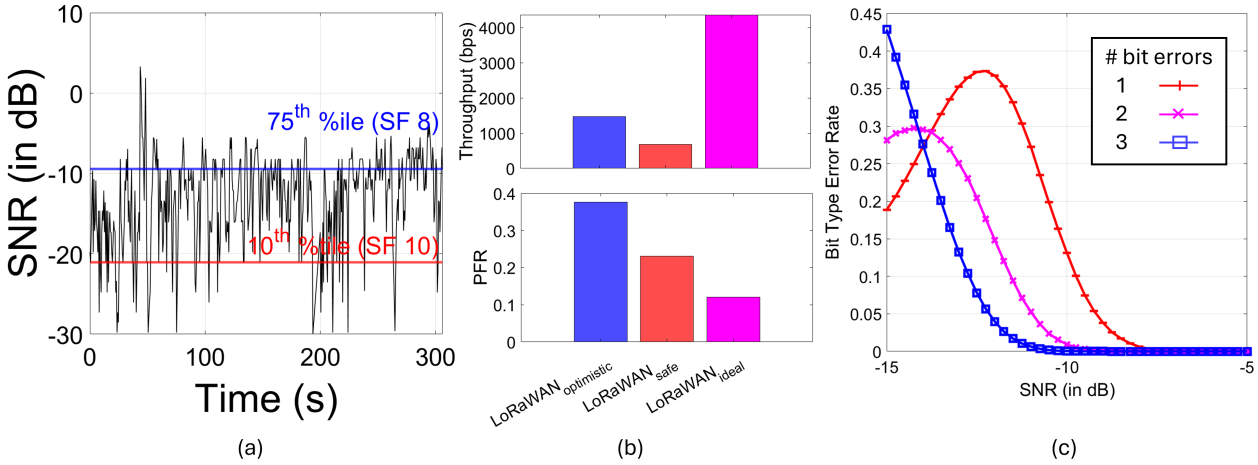

Figure 2.2: Motivation Study: Our motivation study demonstrates the drastic temporal variation of signal power across time. This makes both conservative and optimistic spreading factor choices operate suboptimally. Further, at low SNRs, LoRaWAN faces multi-bit errors that lead to packet failures.

dynamically adjust the modulation and coding scheme. For example, Wi-Fi transmitters use acknowledgment packets to continuously update their modulation rate based on current channel conditions [13]. Likewise, cellular systems assess link quality through association exchanges before establishing a connection with a base station [2, 23].

LoRaWAN, however, operates on a fundamentally different timescale. The interval between a transmitted packet and its corresponding acknowledgment—and even longer until the next transmission—can span several minutes. As a result, any channel quality estimate quickly becomes outdated, limiting the system's ability to adapt to changing conditions in real time. Rather than relying on rapid adaptation, *Pearl* addresses this challenge by enhancing LoRaWAN's resilience to channel variability. It does so by improving robustness to errors at the receiver, enabling better performance even when channel conditions degrade unpredictably.

Chapter 3

OVERVIEW

3.1 A brief primer on LoRaWAN

As shown in Fig.2.1, LoRaWAN uses chirp-spread spectrum modulation to transmit data. The transmitter breaks data bytes into nibbles which are then encoded using the following when the Hamming code is set to Coding Rate (CR) 4/8:

$$c_1 = d_1 \oplus d_2 \oplus d_3$$

$$c_2 = d_2 \oplus d_3 \oplus d_4$$

$$c_3 = d_1 \oplus d_2 \oplus d_4$$

$$c_4 = d_1 \oplus d_3 \oplus d_4$$

These 8-bit coded symbols are grouped according to the spreading factor (SF) and interleaved to generate SF-bit symbols. These SF-bit symbols are transmitted over the air.

The SF-bit symbols are first decoded and then deinterleaved to produce 8-bit symbols. Importantly, if there a symbol errors during decoding, this results in SF single-bit errors after deinterleaving. Standard Hamming decoding is then applied, which can correct all single-bit errors in the encoded symbols. The corrected nibbles are subsequently reassembled into bytes and forwarded to the LoRaWAN server. *Pearl* enhances this process by targeting the Hamming decoding stage—leveraging additional information extracted during CSS demodulation, particularly under low SNR conditions, to improve error correction effectiveness.

3.2 Motivation Study

To highlight the need for improved resilience to temporal fluctuations in channel quality, I examined how LoRaWAN throughput is affected in a static deployment within a campus-

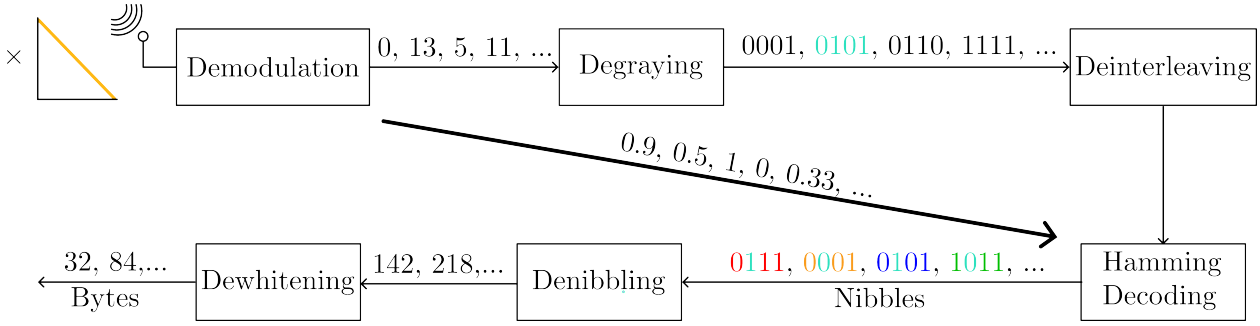

Figure 3.1: *Pearl* Overview: Our system makes three key improvements to the LoRaWAN pipeline - (a) a chirp spread spectrum demodulator that can estimate the bit error likelihood of bits; and (b) a probabilistic Hamming decoder that can recover multi-bit errors.

scale testbed. While prior work [10, 11] has documented the variability in channel conditions using public datasets, our motivation study specifically quantifies its impact on throughput and packet failure rate.

Setup: I collected data from a static LoRaWAN client deployed at 120 locations over a five-minute interval on a campus-scale testbed (detailed in Section 7). The primary objective of this study is to observe how the signal-to-noise ratio (SNR) varies over time at each location. Based on these SNR traces, I estimate the expected throughput and packet failure rate using two representative strategies:

- $LoRaWAN_{safe}$: selects the spreading factor (SF) based on the 10th percentile SNR value.
- $LoRaWAN_{optimistic}$: selects the SF based on the median SNR value.

These two strategies model typical real-world behavior, where LoRaWAN nodes choose SFs based on prior SNR observations. This allows me to examine how such choices perform under real, time-varying channel conditions.

Key Insights: Figure 2.2(a) illustrates the substantial temporal variation in SNR at a representative location. The horizontal lines indicate the SNR thresholds used to select

SF_{safe} and $SF_{optimistic}$. Figure 2.2(b) compares the throughput and packet failure rate (PFR) for three strategies: an *oracle* SF selector that hypothetically chooses the optimal SF for each packet (an ideal but unattainable baseline), and the two practical approaches—SF_{safe} and $SF_{optimistic}$. As expected, SF_{safe} significantly lowers the PFR but at the cost of reduced throughput. Conversely, $SF_{optimistic}$ delivers higher throughput but suffers from increased packet failures due to its aggressive rate selection.

Figure 2.2(c) further reveals that many of the failed packets—especially under varying SNR conditions—are due to 2-bit or 3-bit errors, which exceed the correction capability of LoRaWAN's native Hamming codes. The core objective of *Pearl* is twofold: to reduce packet failures under $SF_{optimistic}$ in high-variance environments, and to enhance the throughput achievable with SF_{safe}, thereby narrowing the performance gap between practical strategies and the ideal SF_{oracle}.

Note: *Pearl* will only be compared to $SF_{optimistic}$. Against SF_{safe} *Pearl* does not show much improvement, as it shines in low SNR conditions, but SF_{safe} would rather move to a higher SF for lower throughput than risk low SNR conditions.

3.3 Pearl *Overview*

In this section, I present a detailed analysis of *Pearl*'s operation. Upon receiving a LoRaWAN packet, the standard demodulation procedure involves correlating the signal with a sequence of down chirps, taking the Discrete Fourier Transform (DFT) of each chirp-aligned segment, and identifying the index corresponding to the maximum magnitude component. Traditionally, only this maximum component is retained as the decoded symbol, but this approach discards useful probabilistic information embedded in the remaining DFT bins. In contrast, *Pearl* retains all DFT magnitudes and models each as the realization of a random variable drawn from a Rician distribution. Specifically, exactly one DFT bin contains the true symbol, which under ideal noiseless conditions corresponds to a perfectly coherent addition of sinusoidal components. The remaining bins contain only noise and are thus expected

to follow a Rayleigh or zero-mean Rician distribution. To justify this model rigorously, I begin by analyzing the structure of a received LoRa chirp under additive white Gaussian noise (AWGN).

Let the sampling frequency be F_S, bandwidth be B, and spreading factor be SF, such that $M = 2^{SF}$ is the number of DFT bins per chirp, and the chirp duration is $T = \frac{2^{SF}}{B}$. The frequency of the h-th bin is given by $f_h = h \cdot \frac{B}{2^{SF}}$, and I define the baseband upchirp as $\text{chirp}(t; f_n, A) = Ae^{j2\pi\left(f_n + \frac{B}{2T}t\right)t}$. In practice, the receiver observes a superposition of a transmitted chirp and complex white Gaussian noise: $\text{chirp}(t; f_0, A) + \eta(t)$, where $\eta(t) \sim \mathcal{CN}(0, \sigma^2)$. After dechirping (multiplying by a conjugate upchirp), the receiver performs a DFT on the dechirped signal to extract frequency components.

Calculating the magnitude of the DFT directly is difficult, so instead I started with the real part and worked from there.

$$\Re\left[\text{DFT}\left[(\text{chirp}(t; f_0, A) + \eta(t)) \cdot e^{-j2\pi\frac{B}{2T}t^2}\right]\Big|_{\omega=f_1}\right].$$

Substituting and simplifying, this becomes:

$$\sum_{t\in[0,T)} A\cos\left(2\pi(f_0 - f_1)t\right) + \eta_r(t)\cos\left(2\pi(f_1 + \tfrac{B}{2T}t)t\right) + \eta_i(t)\sin\left(2\pi(f_1 + \tfrac{B}{2T}t)t\right),$$

where $\eta(t) = \eta_r(t) + j\eta_i(t)$, and $\eta_r(t), \eta_i(t) \sim \mathcal{N}(0, \frac{\sigma^2}{2})$ are IID real Gaussian components. This expression is thus a sum of independent Gaussian random variables, each with mean $A\cos(2\pi(f_0 - f_1)t)$ and variance $\sigma^2/2$. Over the entire interval, if $f_1 \neq f_0$, the sum of $\cos(2\pi(f_0 - f_1)t)$ over M samples is zero, implying the DFT output is distributed as $\mathcal{N}(0, M\frac{\sigma^2}{2})$. If $f_1 = f_0$, the mean term coherently sums to AM, leading to $\mathcal{N}(AM, M\frac{\sigma^2}{2})$.

Applying the same reasoning to the imaginary part, which has mean zero regardless of f_1, it's observed that the full DFT output S_{f_1} is a complex Gaussian random variable:

$$S_{f_1} \sim \begin{cases} \mathcal{CN}(AM, M\frac{\sigma^2}{2}) & \text{if } f_1 = f_0, \\ \mathcal{CN}(0, M\frac{\sigma^2}{2}) & \text{otherwise.} \end{cases}$$

Therefore, the magnitude $|S_{f_1}|$ follows a Rician distribution:

$$|S_{f_1}| \sim \begin{cases} \text{Rice}(\nu = AM, \sigma^2 = M\frac{\sigma^2}{2}) & \text{if } f_1 = f_0, \\ \text{Rice}(\nu = 0, \sigma^2 = M\frac{\sigma^2}{2}) & \text{otherwise.} \end{cases}$$

To further justify the probabilistic model used in *Pearl*, I must confirm that the DFT magnitudes are statistically independent. This is non-trivial because each DFT bin arises from overlapping time-domain noise. However, since the DFT is a linear operation and noise samples $\eta(t)$ are IID, it follows that $\text{Cov}[S_{f_1}, S_{f_2}] = 0$ for $f_1 \neq f_2$, and because the underlying components are Gaussian. While nin general, zero covariance does not imply statistical independence for arbitrary RVs, for Gaussians having a zero covariance is equivalent to independance. As a result, the DFT magnitudes $\{|S_{f_i}|\}_{i=1}^{M}$ are independent Rician random variables: one with a nonzero mean and the rest zero-centered.

While the derivation above demonstrates why the magnitude of each DFT bin of the received chirp signal corrupted by complex Gaussian noise follows a Rician distribution, it is important to note that the use of Ricians is a well-established method in communications theory rather than a novel contribution. The Rician distribution frequently arises in modeling the envelope of complex Gaussian random variables, particularly in the context of wireless communications involving fading channels with a strong line-of-sight component (known as Rician fading).

Furthermore, the independence of the random variables corresponding to different DFT bins is not merely a simplifying assumption but directly follows from the orthogonality properties inherent in the design of the chirp and the DFT transform. Specifically, the transmitted chirp signals and the DFT basis functions form an orthogonal set over the symbol duration. This orthogonality implies that the noise projected onto distinct frequency bins — which are linear combinations of temporally independent Gaussian noise samples

weighted by these orthogonal basis functions — results in uncorrelated Gaussian random variables across bins.

Having established the statistical structure, this is how *Pearl* uses this information. Suppose the magnitude spectrum is observed to be $\{a_0, \ldots, a_{M-1}\}$, and assume exactly one of these values corresponds to the true symbol (i.e., drawn from the Rician distribution with non-centrality $\nu = AM$), and all others from the noise-only (central) Rician. The posterior probability that bin i contains the true symbol using Bayes' rule is:

$$P(S_i = Z \mid s_0, \ldots, s_{M-1}) = \frac{P_Z(s_i) \prod_{j \neq i} P_Q(s_j)}{\sum_{k=0}^{M-1} P_Z(s_k) \prod_{j \neq k} P_Q(s_j)},$$

where P_Z and P_Q are the probability density functions of the Rician distributions corresponding to the signal and noise hypotheses, respectively. These posterior probabilities form the foundation for symbol-wise likelihood estimation.

For practical decoding, these symbol-wise probabilities need to be converted into bitwise probabilities. More discussion on this is in Chapter 4. For a coding rate CR, let $\{p_1, p_2, \cdots, p_{4+CR}\}$ be the bitwise probabilities for $4 + CR$ bits where $p_i = P(\text{bit}_i = 1)$. Next, I enumerate the $2^4 = 16$ candidate codewords and compute the likelihood of each as:

$$P(\mathbf{b} = b_1 b_2 \cdots b_{4+CR} \mid \{p_i\}) = \prod_{i=1}^{4+CR} p_i^{b_i} (1 - p_i)^{1-b_i},$$

The codeword with the largest likelihood is what *Pearl* repairs the $4+CR$ bits into. This allows *Pearl* to support decoding even at low coding rates such as $CR = 1$, thereby enabling higher throughput under favorable channel conditions. Moreover, by leveraging knowledge of the channel statistics, *Pearl* can modulate the redundancy introduced by the Hamming code at the transmitter, transmitting more information bits when the channel is stable and thus opportunistically improving spectral efficiency.

Chapter 4

SYMBOL LIKELIHOOD TO BIT-LEVEL SOFT ESTIMATION

4.1 From Rician Likelihoods to Symbol Probabilities

While the previous section talked about how the values in the DFT are modeled by Rician distributions, it didn't go much into detail about how this relates to bitwise probabilities. This section will explain the process of going from the Rician to the corrected codeword.

Rician Modeling of DFT Magnitude

After dechirping and applying the Discrete Fourier Transform (DFT) to the received baseband signal, the demodulator gets 2^{SF} complex-valued frequency bins, each corresponding to a potential symbol in the LoRa constellation. Let us denote the magnitudes of these bins as:

$$\{a_0, a_1, \ldots, a_{2^{SF}-1}\},$$

where each $a_k = |\hat{y}_k|$ is the magnitude of the DFT coefficient at index k. In traditional LoRa demodulation, the decoded symbol is selected as:

$$\hat{s} = \arg\max_k a_k,$$

which implies complete confidence in the largest amplitude being the correct symbol. What the values of the other magnitudes are not considered at all.

As discussed in Chapter **??**, under AWGN conditions, the complex DFT coefficient at each frequency bin can be modeled as a complex random variable $Z_k = X_k + jY_k$ where X_k and Y_k are independent Gaussian random variables. When the frequency bin corresponds

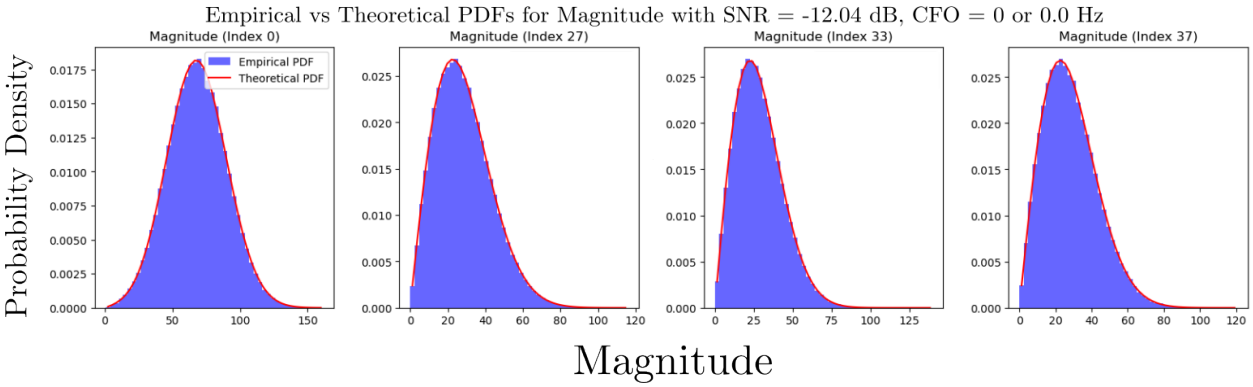

Figure 4.1: Distribution of Magnitude assuming AWGN: The empirical and calculated PDFs of the magnitude of the DFT after de-chirping. First panel shows the distribution for the correct indexes, while the other panels show the distributions of 19, 30, and 42 indices away from the correct index.

to the correct symbol (i.e., when k equals the transmitted symbol index), the magnitude $a_k = |Z_k|$ follows a non-central Rician distribution $a_k \sim \text{Rice}(AM, \sigma)$ while every other index follows a central Rician $a_k \sim \text{Rice}(0, \sigma)$.

The corresponding probability density function for a Rician distribution is:

$$f_{\text{Rician}}(a; \nu, \sigma^2) = \frac{a}{\sigma^2} \exp\left(-\frac{a^2 + \nu^2}{2\sigma^2}\right) I_0\left(\frac{a\nu}{\sigma^2}\right),$$

where $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind.

Fig. 4.1 demonstrates how this distribution manifests in simulated LoRa signals by showing the empirical and theoretical distributions for various frequency bin offsets relative to the correct peak.

Computing Symbol Likelihoods

Rather than relying on the maximum-amplitude heuristic, *Pearl*instead computes the posterior probability that each frequency bin k corresponds to the transmitted symbol. Let $\mathbf{a} = (a_0, a_1, \ldots, a_{2^{SF}-1})$ denote the observed magnitudes. Specifically:

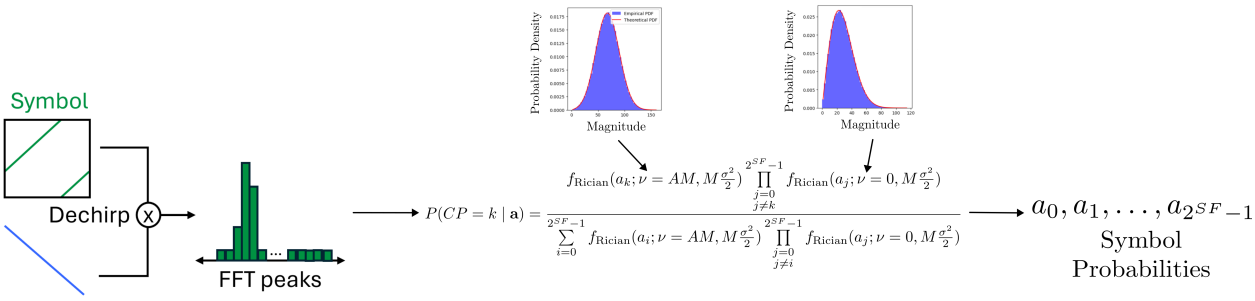

Figure 4.2: *Pearl* Symbol Likelihood Prediction: *Pearl* takes a posteriors approach to estimate the likelihood of each symbol to be the correct symbol using the magnitudes of the FFT values as inputs.

$$P(CP = k \mid \mathbf{a}) = \frac{f_{\text{Rician}}(a_k; \nu = AM, M\frac{\sigma^2}{2}) \prod_{\substack{j=0 \\ j\neq k}}^{2^{SF}-1} f_{\text{Rician}}(a_j; \nu = 0, M\frac{\sigma^2}{2})}{\sum_{i=0}^{2^{SF}-1} f_{\text{Rician}}(a_i; \nu = AM, M\frac{\sigma^2}{2}) \prod_{\substack{j=0 \\ j\neq i}}^{2^{SF}-1} f_{\text{Rician}}(a_j; \nu = 0, M\frac{\sigma^2}{2})}$$

This formula follows directly from Bayes' rule, under the assumption that the noise across frequency bins is independent and identically distributed. Intuitively, it's assuming that index k is the correct index, calculates the likelihood of that, then normalize it with the likelihood of every other index. The output is a softmax-style distribution over possible symbols, denoted:

$$\mathbf{p} = (p_0, p_1, \ldots, p_{2^{SF}-1}),$$

where $p_k = P(CP = k \mid \mathbf{a})$.

This probabilistic formulation captures the uncertainty in decoding and prepares the symbol space for further soft-decision decoding at the bit level. Fig. 4.2 demonstrates this whole process.

4.1.1 Limitations of Phase-Based Symbol Inference

In addition to magnitude, the complex DFT outputs also yield phase values:

$$\phi_k = \arg(Z_k) = \arg(a_k e^{j\theta_k}),$$

which, in theory, can provide further discriminatory power between symbols. Indeed, in AWGN channels with known timing, the phase distribution also follows a known closed-form that depends on the carrier frequency offset (CFO), symbol index, and the value of time at the start of the chirp t_0. Specifically, the phase ϕ of a Rician-distributed complex variable follows the distribution:

$$f(\phi;\mu,\theta,\sigma) = \frac{1}{2\pi}\exp\left(-\frac{\mu^2}{2\sigma^2}\right)\cdot\left[1+\sqrt{\frac{\pi}{2}}\cdot\frac{\mu\cos(\phi-\theta)}{\sigma}\right.$$
$$\left.\cdot\exp\left(\frac{\mu^2\cos^2(\phi-\theta)}{2\sigma^2}\right)\cdot\left(1+\operatorname{erf}\left(\frac{\mu\cos(\phi-\theta)}{\sqrt{2}\sigma}\right)\right)\right],$$

where the parameters μ, θ, and σ are determined by the CFO and timing as:
- $\mu = \frac{\sin(\pi\cdot\mathrm{CFO})}{\sin(\pi\cdot\frac{\mathrm{CFO}}{M})}$, - $\theta = 2\pi\cdot\frac{\mathrm{CFO}}{M}\left(T_0+\frac{M-1}{2}\right)$, - $\sigma = \sigma_0$.

Fig. 4.3 shows how this distribution manifests across frequency bins, including the correct peak and several incorrect offsets.

How do I know t_0? The issue with using the formula above is that it is time-dependent. While some sort of method may exist for calculating the relative time between the chirp and the preamble, instead I would like to propose a work around. Instead of working with the phase of the DFT, instead work with the difference of phases between the last chirp's correct phase and the potential phases of this chirp. The pdf of this new distribution is approximately $f(\phi)$ where $T_0 = 0.5$ and $\sigma = \sigma_0\sqrt{2}$ as shown by Fig. 4.4. This works as the CFO phase term is dependent on both the CFO and t_0 which are the same across the chirp (other than t_0 increasing by a constant amount equal to the duration of the chirp) so a difference of phases removes its influence.

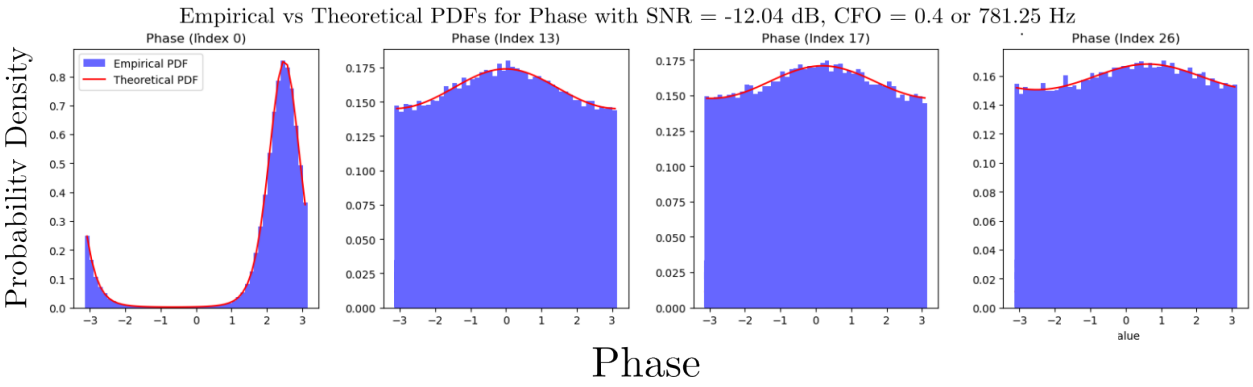

Figure 4.3: Distribution of Magnitude assuming AWGN: The empirical and calculated PDFs of the phase of the DFT after de-chirping. First panel shows the distribution for the correct indexes, while the other panels show the distributions of 19, 30, and 42 indices away from the correct index.

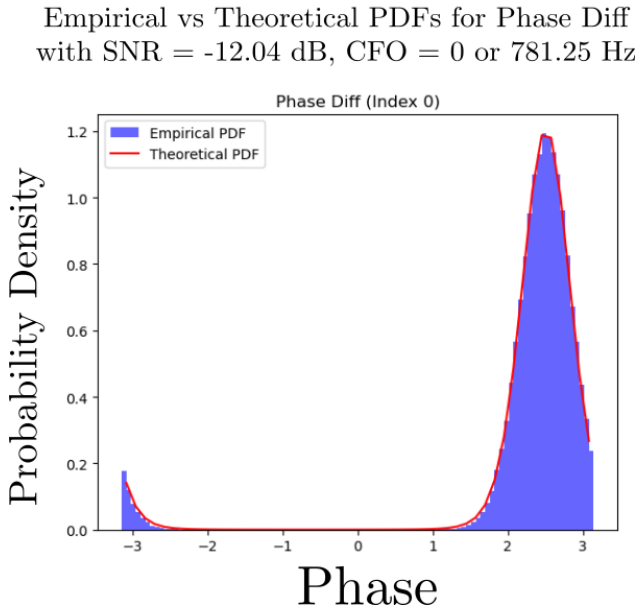

Figure 4.4: Distribution of Difference of Phase assuming AWGN: The empirical and calculated PDFs of the difference in phase of the DFT after de-chirping for two consecutive chirps. Only 1 panel is shown as approximation only works, and is relevant for, the correct index.

Empirical Challenges with Phase Use

Despite the elegant theory, our empirical investigations reveal a critical flaw in relying on phase information in real-world LoRaWAN deployments. Due to various imperfections in synchronization—especially in the context of imperfect packet detection—the starting index of the chirp is often misaligned. This misalignment adds a term to the phase calculation that is dependant on both how much misalignment there is, and the actual value of the frequency that is in consideration. Therefore the difference trick no longer works.

Therefore, phase-based inference was excluded from all practical components of *Pearl*. The reason I talked about it is because it serves as an example of how the probability calculation is flexible and indepdendant of the bit-probability repair section of *Pearl*. Meaning that if more a method is more reliable in calculating the posterior probabilities, such as channel information is know or an ML is used, then the symbol probability calculation part modified or subbed out for better results.

4.2 Mapping Symbol Probabilities to Soft Bits

Once *Pearl*has the full posterior over symbols $\mathbf{p} = (p_0, \ldots, p_{2^{SF}-1})$, the next step is to translate this into soft-bit likelihoods for the individual bits within the symbol. Each symbol x_i can be mapped to bits via:

1. Gray decoding: $x_i \mapsto y_i$ to minimize bit flips between adjacent symbols.
2. Binary expansion: $y_i \mapsto \mathbf{B}_i = (b_{i1}, b_{i2}, \ldots, b_{iSF})$.
3. Interleaving: $\mathbf{B}_i \mapsto \mathbf{b}_i = (b_{i1}, b_{i2}, \ldots, b_{i(4+CR)})$.

Gray decoding is a simple bijective map, so there is mainly nothing interesting to say. The one thing to note is that the LoRaWAN protocol has the first $4+CR$ symbols modulated with less possible symbols, meaning nearby symbols will actually map to the same symbol after Gray decoding. For those symbols, *Pearl*just adds their probabilities when they go through the map.

Then, the soft likelihood for each bit j is computed as a weighted sum over all symbols:

$$b_j = \sum_{i=0}^{2^{SF}-1} p_i \cdot b_{ij},$$

where $b_j \in [0, 1]$ represents the probability that bit b_j is 1.

This can be vectorized using matrix multiplication as:

$$\mathbf{b} = \mathbf{p} \cdot B,$$

where B is a $(2^{SF} \times SF)$ binary matrix of symbol-to-bit mappings.

Practical Approximation

While computing the full sum over all 2^{SF} symbols is ideal, it is computationally expensive. In practice, an approximation can be used where only the most likely symbol $\hat{x}$ with probability $\hat{p}$ is considered. The rest of the probabilities are assumed to contribute uniformly to either bit. This yields the bitwise confidence as:

$$b_j^* = \hat{p} \cdot b_j + \frac{1}{2}(1 - \hat{p}),$$

where b_j is the j-th bit of the most likely symbol. This approximation simplifies the decoder and means the 2^{SF} likelihoods calculated before an only needed to normalize $\hat{x}$ into a probability and don't need to be stored after that.

4.3 Discussions

Applicability of Probabilistic Decoding in LoRaWAN

In many traditional wireless systems, bit error rates are typically so low that the added complexity of probabilistic decoding provides little practical benefit. In contrast, LoRaWAN operates under conditions where symbol error rates can be significantly higher—often exceeding 30–40% in low-SNR or interference-prone environments. In these scenarios, the

determinism of conventional decoding becomes a liability, as hard decisions made under uncertainty can easily lead to incorrect decoding.

Probabilistic decoding, by contrast, naturally incorporates uncertainty into the decoding process. Rather than making binary decisions on symbol correctness, it allows the receiver to reason about the full likelihood distribution across possible values. This flexibility enables more robust downstream decoding, particularly in systems like LoRaWAN where individual symbols differ in reliability due to spreading factor modulation and frequency-selective fading. By preserving soft information throughout the pipeline, *Pearl* increases the likelihood of recovering the correct bits, even when the raw symbol error rate is high.

Interoperability with Existing Physical Layer Techniques

Pearl is designed to be modular and decoupled from the specifics of the physical layer. This design choice ensures that it can be seamlessly integrated with a wide range of prior physical-layer enhancements for LoRaWAN. For instance, techniques that exploit geographic diversity across multiple gateways [10, 6] or apply advanced baseband signal processing at a single gateway [14] operate independently of the bit-level error correction scheme.

Because *Pearl* interfaces cleanly at the decoding layer—after symbol demodulation and deinterleaving—it can serve as a drop-in replacement for the standard Hamming-based decoding logic. In this role, it acts as a probabilistic refinement stage that improves decoding performance without requiring changes to modulation, synchronization, or radio hardware. This compatibility allows *Pearl* to function as a safety net: it augments standard decoding paths with a probabilistic backup that improves robustness in adverse channel conditions while remaining backward-compatible with the broader LoRaWAN protocol stack.

Chapter 5

IMPLEMENTATION

I implemented *Pearl* both in simulation and through a comprehensive real-world deployment using off-the-shelf SemTech SX1262 transceivers. Our Python-based simulation framework extends the state-of-the-art LoRaWAN decoder [27] incorporating *Pearl*'s probabilistic error-aware repair capabilities.

In simulation, I compare *Pearl* against standard LoRaWAN under a wide range of conditions—spanning spreading factors $\{7, 8, 9, 10\}$ and varying SNR levels—conducting over 10^7 trials. I evaluated throughput, latency, and packet failure rate (PFR) for *Pearl* variants. Section 6 presents a detailed empirical analysis highlighting the information-theoretic gains that *Pearl* introduces.

For real-world evaluation, I deployed *Pearl*using SemTech SX1262 transceivers across a 300,000 m^2 campus-scale testbed. The base station was positioned at the NEWT Lab window in the ECE building, at an elevation of 10.7 meters. Fig. 5.1 illustrates our deployment.[1] In total, I collected 62,600 channel traces across diverse locations and time windows, capturing wireless impairments from buildings, foliage, terrain, and pedestrian interference. Using *Pearl*, I decoded received packets and evaluate the gains in throughput, reliability, battery longevity, and PFR, as described in Section 7.

Baselines: For both simulations and deployments, I compare *Pearl* against standard LoRaWAN configured with the same spreading factor (SF) and signal-to-noise ratio (SNR), and a fixed coding rate (CR) of 4/8.. In real-world tests, I benchmarked *Pearl* against three baselines:

[1]Our deployment scale and methodology are comparable to prior LoRaWAN studies.

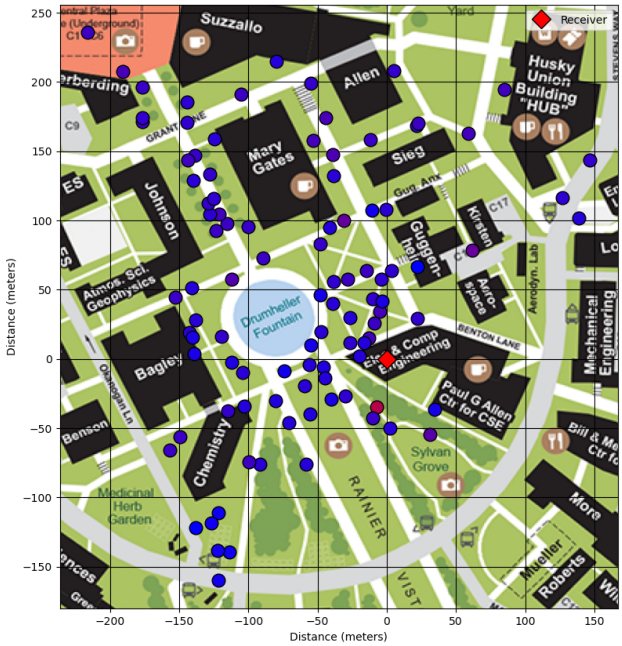

Figure 5.1: Deployment Map: At each dot I recorded 5 minutes of channel data using a SemTech SX1262 transceiver. About 70 points were low variance channels, and 50 were high variance. A total of 10 hours of data was recorded.

1. **LoRaWAN$_{\text{optimistic}}$** — SNR-based SF selection using the 75$^{\text{th}}$ percentile.
2. **LoRaWAN$_{\text{safe}}$** — SF selection based on the 10$^{\text{th}}$ percentile.

Unless specified, I use SF=7 and CR=4/8 to maintain consistency with typical LoRaWAN settings.

Chapter 6

MICROBENCHMARKING

We now present a detailed evaluation of *Pearl*'s performance through simulations conducted over at least 10^6 trials per configuration using an AWGN channel model.

Improvement in Throughput: Fig. 6.1(a) shows the throughput for *Pearl* under varying SNRs with SF=7. At lower SNRs, standard LoRaWAN suffers from significant packet losses. In contrast, *Pearl* remains highly resilient, achieving 2–40 $\times$ throughput improvements thanks to its soft-decision decoding and robust bit-level error correction.

Improvement in Reliability: Fig. 6.1(b) quantifies PFR across spreading factors. *Pearl* consistently achieves lower PFRs in poor SNR environments. It corrects 72.0% of 2-bit errors and 22.1% of 3-bit errors under favorable SNR, highlighting the strength of its probabilistic decoding framework.

Reduction in Latency: Fig. 6.1(c) presents end-to-end latency. *Pearl* proves robust under deteriorating SNR, offering consistently lower latency than LoRaWAN due to fewer retransmissions.

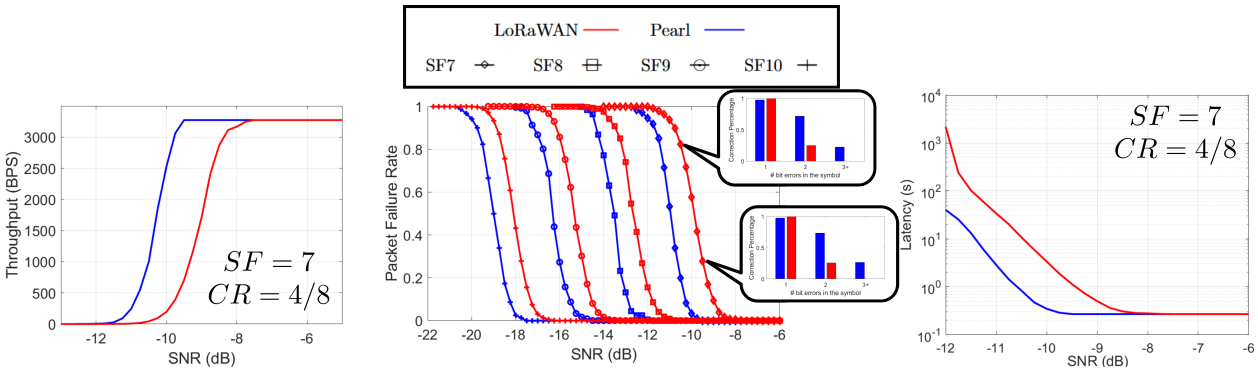

Figure 6.1: *Pearl* Microbenchmarks. (a) *Pearl* outperforms standard LoRaWAN in throughput due to opportunistic encoding and probabilistic decoding. (b) This benefit holds across spreading factors, especially with *Pearl*ś ability to correct up to 3-bit errors. (c) Throughput improvements yield lower latency by reducing retransmissions.

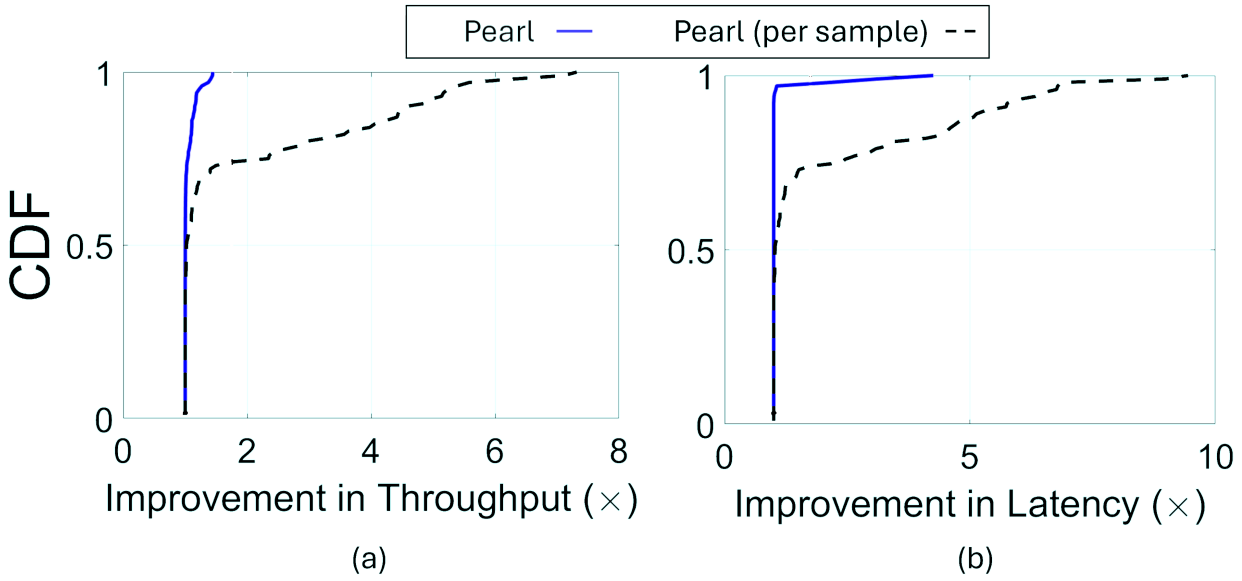

Figure 6.2: *Pearl* Real-World Results. (a) *Pearl* improves throughput by 2.045× over LoRaWAN. (b) Latency is reduced by 52.7% in volatile conditions.

Chapter 7

RESULTS

I now present empirical results from our real-world deployment, evaluating the temporal, spatial, and spectral benefits of *Pearl* using 62,600 channel traces across a 120-node, campus-scale testbed.

7.0.1 Temporal Analysis and Benefits

The primary aim of *Pearl* is to make LoRaWAN more robust to temporal channel variations. The results in this section validate that claim in the real world.

Setup: I gathered temporal traces from 120 locations, capturing channel behavior over several minutes. Of these, 70 were low-variance (e.g., elevated, unobstructed), and 50 exhibited high variance (e.g., obstructed or pedestrian-heavy). A location is deemed "high variance" if its standard deviation exceeds 3.5 dB. Baselines LoRaWAN∗optimistic set its spreading factors based on the 75%ile SNRs.

Note: *Pearl* is not designed to outperform LoRaWAN in static/good-channel conditions, but rather to ensure robust performance under adverse and dynamic environments.

Throughput Benefit: Fig. 6.2(a) *Pearl* improves median throughput by 1.742× over LoRaWAN∗optimistic. *Pearl* achieves up to 7.11× improvement under volatile channels.

Reduction in Latency: Fig. 6.2(b) indicates a latency reduction of up to 4.252× for *Pearl*Ṫhe median improvement approximately 1.

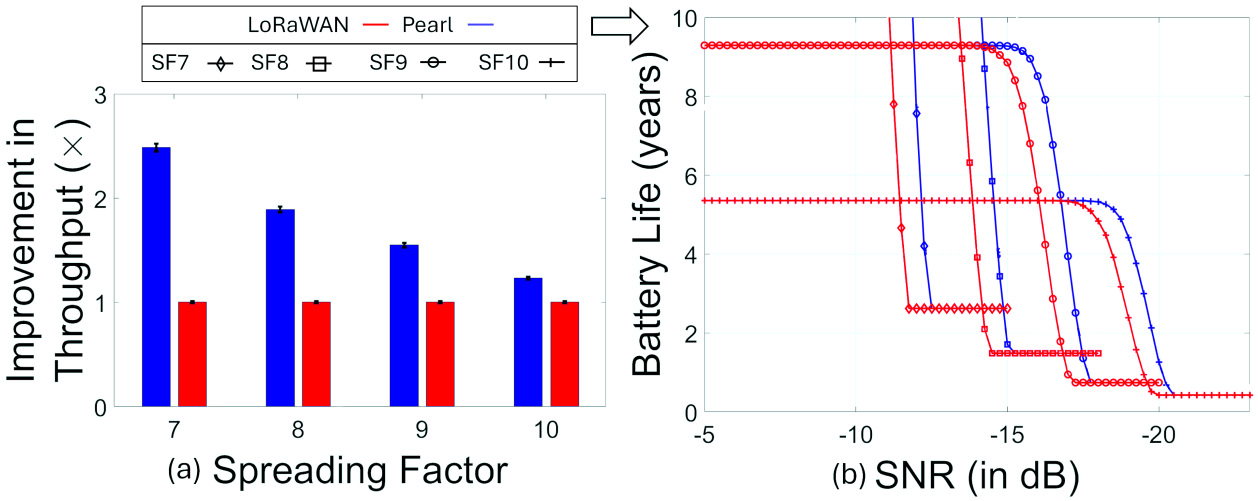

Figure 7.1: *Pearl* Spatial and Energy Efficiency Gains. (a) Across 120 nodes, *Pearl* improves network throughput by 2.045 × on average. (b) Battery life is extended by 1.3 × due to reduced retransmissions.

7.0.2 Spatial and Battery Life Benefits

I collected channel traces from 120 unique geographic locations across the campus. Fig. 7.1(a) shows throughput improvements using *Pearl* across all SFs are 2.48×, 1.89×, 1.55×, 1.23× for $SF = \{7, 8, 9, 10\}$ respectively

Battery Life Analysis: Based on transmission-only battery models [6], *Pearl* achieves an average improvement of 1.316× in battery life due to fewer retransmissions. As shown in Fig. 7.1(b), this translates to multiple years of additional operational lifetime—capped at 10 years due to typical battery lifespan constraints.

Chapter 8

CONCLUSION AND FUTURE WORK

This thesis introduces *Pearl*, a probabilistic, error-aware repair technique for LoRaWAN that achieves higher throughput, improved reliability, and reduced latency in the presence of wireless channel variability. Importantly, *Pearl*operates on commodity hardware and remains compatible with physical-layer enhancements developed by the broader community.

*Pearl*demonstrates that LoRaWAN reliability can be significantly enhanced without protocol overhauls. Looking forward, future work will explore:

1. Integration of side-channel context (e.g., phase, RSSI)
2. Machine-learned priors for symbol likelihoods
3. Multi-base-station inference to enhance decoding accuracy across spatial diversity

My results suggest that probabilistic decoding is a promising pathway to making LoRaWAN competitive with cellular LP-WAN solutions in reliability, longevity, and scalability.

BIBLIOGRAPHY

[1] Maqsood Ahamed Abdul Careem and Aveek Dutta. Real-time prediction of non-stationary wireless channels. *IEEE Transactions on Wireless Communications*, 19(12):7836–7850, 2020.

[2] Ahmed Alkhateeb, Omar El Ayach, Geert Leus, and Robert W. Heath. Channel estimation and hybrid precoding for millimeter wave cellular systems. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):831–846, 2014.

[3] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. A cloud-optimized link layer for low-power wide-area networks. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 247–259, 2020.

[4] Valerio Bioglio, Carlo Condo, and Ingmar Land. Design of polar codes in 5g new radio. *IEEE Communications Surveys & Tutorials*, 23(1):29–40, 2020.

[5] Torben Brack, Matthias Alles, Frank Kienle, and Norbert Wehn. A synthesizable ip core for wimax 802.16 e ldpc code decoding. In *2006 IEEE 17th international symposium on personal, indoor and mobile radio communications*, pages 1–5. IEEE, 2006.

[6] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks. In *ACM/IEEE IPSN*, pages 60–71, 2018.

[7] Jialuo Du, Yidong Ren, Zhui Zhu, Chenning Li, Zhichao Cao, Qiang Ma, and Yunhao Liu. Srlora: Neural-enhanced lora weak signal decoding with multi-gateway super resolution. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 270–279, 2023.

[8] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 309–321, 2017.

[9] Akshay Gadre, Zachary Machester, and Swarun Kumar. Adapting lora ground stations for low-latency imaging and inference from lora-enabled cubesats. *ACM Transactions on Sensor Networks*, 20(5):1–30, 2024.

[10] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. Frequency configuration for low-power wide-area networks in a heartbeat. In *NSDI*, pages 339–352, 2020.

[11] Akshay Gadre, Fan Yi, Anthony Rowe, Bob Iannucci, and Swarun Kumar. Quick (and dirty) aggregate queries on low-power wans. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 277–288. IEEE, 2020.

[12] Kais Hassan, Mohammad Masarra, Marie Zwingelstein, and Iyad Dayoub. Channel estimation techniques for millimeter-wave communication systems: Achievements and challenges. *IEEE Open Journal of the Communications Society*, 1:1336–1363, 2020.

[13] Muhammad Owais Khan and Lili Qiu. Accurate wifi packet delivery rate estimation and applications. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[14] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 56–68, 2021.

[15] Chenning Li, Xiuzhen Guo, Longfei Shangguan, Zhichao Cao, and Kyle Jamieson. {CurvingLoRa} to boost {LoRa} network throughput via concurrent transmission. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 879–895, 2022.

[16] Manan Mishra, Daniel Koch, Muhammad Osama Shahid, Bhuvana Krishnaswamy, Krishna Chintalapudi, and Suman Banerjee. {OpenLoRa}: Validating {LoRa} implementations through an extensible and open-sourced framework. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1165–1183, 2023.

[17] Yidong Ren, Puyu Cai, Jinyan Jiang, Jialuo Du, and Zhichao Cao. Prism: High-throughput lora backscatter with non-linear chirps. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[18] Muhammad Osama Shahid, Daniel Koch, Jayaram Raghuram, Bhuvana Krishnaswamy, Krishna Chintalapudi, and Suman Banerjee. {Cloud-LoRa}: Enabling cloud radio access {LoRa} networks using reinforcement learning based {Bandwidth-Adaptive} compression. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 1959–1976, 2024.

[19] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. Concurrent interference cancellation: Decoding multi-packet collisions in lora. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 503–515, 2021.

[20] Jayanth Shenoy, Om Chabra, Tusher Chakraborty, Suraj Jog, Deepak Vasisht, and Ranveer Chandra. Cosmac: Constellation-aware medium access and scheduling for iot satellites. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 724–739, 2024.

[21] Ido Tal and Alexander Vardy. How to construct polar codes. *IEEE Transactions on Information Theory*, 59(10):6562–6582, 2013.

[22] Shuai Tong, Zilin Shen, Yunhao Liu, and Jiliang Wang. Combating link dynamics for reliable lora connection in urban settings. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 642–655, 2021.

[23] Deepak Vasisht, Swarun Kumar, Hariharan Rahul, and Dina Katabi. Eliminating channel feedback in next-generation cellular networks. In *Proceedings of the 2016 ACM SIGCOMM Conference*, page 398–411, New York, NY, USA, 2016. Association for Computing Machinery.

[24] Ruidongxue Wang and Jianhua Bao. Ellr-bp: An enhanced llr bp algorithm based on ldpc coding for lora physical layer. *IEEE Transactions on Green Communications and Networking*, 2024.

[25] Yuting Wang, Fanhao Zhang, Xiaolong Zheng, Liang Liu, and Huadong Ma. Decoding lora collisions via parallel alignment. *ACM Transactions on Sensor Networks*, 19(3):1–25, 2023.

[26] Xianjin Xia, Yuanqing Zheng, and Tao Gu. Ftrack: Parallel decoding for lora transmissions. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 192–204, 2019.

[27] Zhenqiang Xu, Shuai Tong, Pengjin Xie, and Jiliang Wang. From demodulation to decoding: Toward complete lora phy understanding and implementation. *ACM Transactions on Sensor Networks*, 18(4):1–27, 2023.

[28] Kang Yang and Wan Du. A low-density parity-check coding scheme for lora networking. *ACM Transactions on Sensor Networks*, 2024.

[29] Kang Yang, Miaomiao Liu, and Wan Du. : Rateless-enabled link adaptation for lora networking. *IEEE/ACM Transactions on Networking*, 2024.

[30] Wenchao Zhang, Song Chen, Xuefei Bai, and Dajiang Zhou. A full layer parallel qc-ldpc decoder for wimax and wi-fi. In *2015 IEEE 11th International Conference on ASIC (ASICON)*, pages 1–4. IEEE, 2015.

[31] Zhibin Zou, Maqsood Careem, Aveek Dutta, and Ngwe Thawdar. Joint spatio-temporal precoding for practical non-stationary wireless channels. *IEEE Transactions on Communications*, 71(4):2396–2409, 2023.