Handling Jamming Attacks in a LoRa Network

Md Ashikul Haque
Department of Computer Science
Wayne State University

Abusayeed Saifullah
Department of Computer Science
Wayne State University

***Abstract*—With the proliferation of Internet of Things (IoT) applications relying on LoRa to gather data from dispersed devices, LoRa communications become prone to jamming attacks, which can cause massive packet loss, reduced throughput, and depleting batteries. Existing work related to jamming in LoRa mainly considers the impact of jamming and does not provide any anti-jamming techniques. Mitigating jamming in a LoRa network is extremely challenging as the devices have low computation power and limited energy typically supplied by small batteries. In this paper, we present a jamming mitigation technique for LoRa that imposes no overhead for energy-constrained nodes and can decode packets even when the SNR is ultra-low. We do this by exploiting the temporal and spatial variations of jammed signals of the same packet at different locations. Our design requires no change in LoRa nodes' physical layer, making it usable with all commercial off-the-shelf (COTS) LoRa devices. It works even under a powerful jammer with variable jamming signals, length of jamming packets, and mobility. Our jamming mitigation technique can be combined with existing collision recovery techniques for multiple LoRa packets to recover packets that collide with either jamming signal or with other LoRa packets of the same network. Finally, we evaluate the effectiveness of our jamming mitigation technique through outdoor experiments. The results show that our technique mitigates jamming by improving packet reception rate and energy consumption per packet up to 83.91 and 115.65 times, respectively, compared to the traditional LoRa.**

***Index Terms*—Jamming, LPWAN, LoRa, IoT**

I. INTRODUCTION

Applications of the Internet of Things (IoT), such as smart agriculture [1], and smart cities [2], and healthcare [3] aim to enhance the quality of life, health, and safety in both rural and urban communities. The demand for these applications is rapidly increasing, leading to a projected count of approximately 29 billion IoT devices by 2030 [4]. Low-power wide-area networks (LPWANs) emerge as a new platform to connect IoT devices. LPWANs enable low-power and long-range communication. LoRa is commonly recognized as a leading technology in the LPWAN landscape [5] and is commercially accessible worldwide, boasting over 600 documented use cases and the deployment of over 50 million devices [6]. According to industry analysts at ABI Research, LoRa is anticipated to account for over 50% of all LPWAN connections by 2026, given its adaptability to both outdoor and indoor scenarios.

The number of connected LoRa devices is expected to exceed 50 billion within a year [7]. With Comcast recently announcing the addition of LoRa radios to set-top boxes, LoRa devices will become ubiquitous in the US [8]. The rapid growth of LoRa in the limited ISM spectrum, being widely dispersed along with its low power, brings forth the challenge of jamming. Any jamming can severely impact these ubiquitous LoRa devices, as they are not equipped to handle the impending challenge of jamming. LoRaWAN refers to a Wide-Area Network that employs LoRa in the physical layer. Mitigating jamming in a LoRaWAN is extremely challenging as the devices have low computation power and limited energy typically supplied by small batteries. Jamming can significantly affect LoRa communication by causing excessive packet loss, extended transmission delays, and depleting batteries.

In this paper, we propose a methodology to counteract jamming in a LoRaWAN. While a limited number of existing works have addressed jamming in LoRa or other LPWANs [9, 10, 11, 12, 13], all but [9] primarily examine the impact of jamming on these networks and provide minimal discussion on jamming mitigation techniques suitable for LPWANs. [9] is designed to mitigate jamming in SNOW [14, 15], and it is highly effective when the jammer follows their given game-theory model, but this is not always practical and cannot cope with a random or stealth jamming. To the best of our knowledge, our paper represents the first work in addressing the challenge of jamming mitigation specifically for LoRa [16].

We propose to mitigate jamming in LoRa by recovering interfered physical layer (PHY) samples, generally known as collision recovery in wireless networks. There are some existing research for collision recovery in wireless networks [17, 18, 19, 20, 21, 22, 23]. However, none of them are suitable to resolve collision caused by jamming, as they can only recover collisions caused by intra-network packets. There are a few works that handle collisions or improve Signal to Noise Ratio (SNR) to recover packets in LoRa [24, 25, 26, 27, 28, 29, 30]. These techniques for LoRa can only decode a LoRa symbol successfully if SNR at the gateway is better than -35 dBm. Note that with usual LoRa configuration the required SNR threshold becomes -20 dBm, implying they cannot decode even a single LoRa symbol correctly when the SNR at the gateway is below -20 dBm. [31] proposes a method that improves co-existence of a LoRaWAN by enhancing the timing of packet transmission through co-operative reinforcement learning. However, this is ineffective against jamming, as it is designed to handle the coexistence of legitimate networks/devices that transmit when needed and may have some patterns. On the other hand, jamming is intentional interference created by malicious users to disrupt communications and can change patterns based on the user's activity, making it hard to counteract.

979-8-3503-7025-6/24/$31.00 ©2024 IEEE
DOI 10.1109/IoTDI61053.2024.00017

In this paper, we present a jamming mitigation technique that imposes no overhead for energy-constrained LoRa nodes. It can decode packets even when the SNR is way below -35 dBm, which is required during jamming attacks. We achieve this by exploiting the temporal and spatial variations of jammed signals of the same packet at different locations. Specifically, we leverage multiple gateways to capture the temporal and spatial variations of LoRa signal and process them further to recover the jammed LoRa signal. Even though multiple gateways have been used in previous work [25], it is unsuitable for decoding a jammed (i.e., SNR $\leq$ -35 dBm) LoRa packet. Furthermore, our proposed technique can recover the original LoRa signal under severe jamming attacks commenced by any single reactive/proactive jammer.

We implemented our proposed approach and developed a jamming mitigation system for LoRa, consisting of four modules. This system processes the jammed LoRa signal through these four modules and feeds the recovered LoRa signal to the vanilla LoRa demodulation system. Consequently, the LoRa demodulation system does not need to be aware of our system and can decode the recovered LoRa signal in the usual manner. This design requires no change in LoRa nodes' physical layer, and thus, all COTS LoRa devices can use the approach. Our jamming mitigation technique can be combined with existing collision recovery techniques for multiple LoRa packets to recover packets that collide with either jamming signal or with other LoRa packets of the same network.

In summary, the paper makes the following contributions:

1) We introduce a new signal processing technique to recover the original LoRa signals from jammed LoRa signals, which works even under a powerful jammer with variable jamming signals, length of jamming packets, and mobility.
2) We develop a system based on this technique that works with COTS LoRa devices and is plug-and-play compatible with any LoRa demodulation system.
3) Our proposed system was evaluated through outdoor experiments using USRPs [32] as gateways and jammer, and Arduino [33] with a LoRa shield [34] as the LoRa node. The results demonstrate that our proposed technique mitigates jamming by improving packet reception rate and energy consumption per packet by up to 83.91 and 115.65 times, respectively, compared to the baseline.

In the rest of the paper, Section II gives an overview of LoRa. Section III describes related works. Section IV outlines the attack model of the jammer. Section V presents analyzing a jammed signal. Section V-E presents the challenges. Section VI details the system design. Section VII provides the experimental results. Finally, Section VIII concludes the paper.

II. BACKGROUND

LoRa [16], which stands for Long Range, is a wireless communication technology designed for long-distance, low-power applications. It was developed to enable Machine-to-Machine (M2M) communication for IoT applications. LoRa enables devices to communicate over substantial distances while consuming minimal power, making it well-suited for applications such as smart cities, agriculture, and industrial IoT.

A. Modulation

LoRa technology employs Chirp Spread Spectrum (CSS) modulation, which is based on the manipulation of chirp signals for data transmission. The fundamental chirp signal used is an up-chirp, where the frequency increases linearly over time. This up-chirp serves as the base signal from which various symbols are derived. The modulation process involves dividing the data to be transmitted into smaller chunks, each represented by a unique symbol. These symbols are created by shifting the frequency and duration of the base up-chirp. To create a symbol, the frequency of the up-chirp is shifted by a specific amount corresponding to the symbol's data content. A sequence of these frequency-shifted chirps represents the symbol sequence, encoding the original data.

The spreading factor (SF) in LoRa communication plays a pivotal role in balancing data rate, range, and sensitivity. A higher SF results in a longer symbol time, enabling more robust communication over extended distances, albeit at a lower data rate. This increased sensitivity makes higher SF suitable for scenarios with challenging signal conditions. Conversely, lower spreading factors offer higher data rates but are less resilient to interference, making them preferable for applications prioritizing data throughput over range. The choice of SF also influences channel occupancy, affecting overall capacity and power consumption.

B. Demodulation

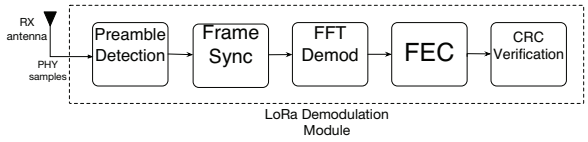


Fig. 1: LoRa demodulation module.

The demodulation process in LoRa technology involves precise signal processing to accurately detect the transmitted symbols. LoRa demodulation involves a series of key modules to accurately retrieve data from the received signal.

The demodulation begins with preamble detection, where the receiver identifies the synchronization sequence at the beginning of the LoRa frame. The preamble serves to synchronize the transmitter and receiver, allowing the latter to anticipate the start of the frame. After preamble detection, the frame synchronization module ensures alignment with the incoming data. It verifies the synchronization word or sequence in the frame, ensuring that subsequent data can be correctly interpreted.

Fast Fourier Transform (FFT) is employed for demodulation. The received signal, which is spread over a wide frequency range using chirp spread spectrum modulation, undergoes FFT to convert it from the frequency domain

to the time domain. This step is crucial for extracting the modulated data from the received signal. LoRa uses Forward Error Correction to enhance the reliability of data transmission. FEC adds redundant bits to the transmitted data, allowing the receiver to correct errors in the received information. This is particularly important in scenarios where the signal may be weakened or distorted.

The demodulation process includes a Cyclic Redundancy Check (CRC) verification step. CRC is a type of error-checking code that ensures the integrity of the received data. By comparing the calculated CRC value with the one included in the received frame, the receiver can determine whether the data has been accurately transmitted.

These demodulation modules collectively contribute to the accurate retrieval of information from LoRa frames. Preamble detection and frame sync establish the beginning of the frame, FFT demodulation decodes the spread spectrum signal, FEC enhances error correction capabilities, and CRC verification ensures the integrity of the received data, providing a robust demodulation process for LoRa communication.

III. RELATED WORK

Many existing works have studied techniques for handling severe interference or jamming for wireless networks [35, 36, 37, 38, 39, 40, 41, 42]. These works mainly rely on spread spectrum techniques, increased transmission (Tx) power, antenna polarization or directional transmission, and packet fragmentation [37, 38, 40]. Most of the above approaches are tailored for the IEEE 802.15.4 standard only [37, 38]. While Tx power control is a common technique that applies to LoRa, using only Tx power to combat jamming is often ineffective since the maximum Tx power of a LoRa node is quite low (typically $\leq$ 20dBm).

Several jamming mitigation approaches depend on regular message exchanging with a central coordinator node, causing considerable energy consumption and degrading the scalability [35, 36, 39, 40]. JAMMY [43] is an approach for WSN based on TDMA (Time Division Multiple Access) MAC protocol requiring strong time synchronization both in the network and between the network and the jammer. Thus, it may not be suitable for LoRa. Jamming mitigation through multipath routing has been studied as well [44, 45], which also may not apply to LoRa as the latter adopts M2M communication.

Some recent works have considered jamming in LPWAN [9, 10, 11, 12, 13]. The work in [9] is designed to mitigate jamming in SNOW [14, 15], and it is highly effective when the jammer follows their given game-theory model, but this is not always practical and can not cope with a random or stealth jamming. The work in [11, 12] design an efficient jammer to disrupt the communication in a LoRaWAN. The work in [10, 13] experimentally measures the impact of one or multiple jammers in a LoRaWAN. Thus, they mainly focus on the effect of jamming in these networks and hardly discuss jamming mitigation techniques suitable for LPWAN. None of these works proposes any technique that can mitigate jamming in LPWAN.

Some works have addressed collision recovery in wireless networks [17, 18, 19, 20, 21, 22, 23]. Interference alignment [18], Analog network coding [20], Full duplex [19], and XORs [21] rely on assumption that the receiver has one of the two collided packets. RnR [17] addresses the issues and resolves intra-network collisions in WSN. However, it cannot resolve collisions caused by other wireless signals (i.e., jamming). The work in [31] proposes a method that improves co-existence in a LoRaWAN by enhancing the timing of packet transmission through cooperative reinforcement learning. However, this is ineffective against jamming, as discussed in Section I.

Several studies address collision issues or aim to improve Signal-to-Noise Ratio (SNR) for packet recovery in LoRa [24, 25, 26, 27, 28, 29, 30]. mLoRa [24], FTrack [26], and CoLoRa [28] leverage either the temporal or spatial domain to mitigate collisions in LoRa. Charm [25] and Choir [27] utilize multiple gateways or nodes to enhance SNR in LoRa. NELoRa [29] employs deep learning to reduce the required SNR for decoding LoRa packets. Xcopy [30] employs retransmitted packets to decode all symbols. However, these LoRa techniques can decode packets successfully only when the SNR is at least -35 dBm. During jamming attacks, the SNR consistently falls below -35 dBm, rendering these techniques ineffective in handling jamming scenarios.

In this paper, we design a system at the PHY layer of the LoRa gateway that can recover a LoRa packet in the presence of a jammer even when the jamming power is very high (e.g., 40 dBm). To the best of our knowledge, this is the first paper that addresses the mitigation of jamming in LoRa.

IV. ATTACK MODEL

The jammer is a reactive device capable of transmitting any wireless signal within the same band to disrupt communication. Specifically, it monitors the airwaves for LoRa packets transmitted by the nodes. Upon packet detection, the jammer immediately transmits a disrupting signal. This approach aims to prevent the LoRa gateway from utilizing any temporal offset by aligning with the same sample of the LoRa packet. LoRa communications are mainly uplink (nodes to gateway) and most downlink LoRa packets are sent for acknowledgment purposes. The gateway can use much higher Tx power to combat jamming in downlink channel. Hence, in this paper, the downlink is considered not to be jammed. We also consider that, to jam a LoRa packet, the jammer uses the same channel on which the LoRa packet is transmitted. A jammer can use wider channels but our approach will not work in that case.

The jammer can operate stealthily during jamming. This is achieved by mimicking the preamble of LoRa packets (while the payload contains a disrupting jamming signal), leading the gateway to lock onto the jamming signal. In LoRa, when the gateway detects the preamble of two signals, it locks onto the one with higher signal strength—a phenomenon known as the *capture effect*. The jammer exploits this effect to remain stealthy. Furthermore, the jammer can use a packet with any payload size for jamming. For the purposes of this paper, we

assume that there will be at most one jammer operating at any given time.

In summary, our paper allows the jammer to be powerful with variable jamming signals, length of jamming packets, and reactive approach. Our approach can work even if a jammer occasionally moves its position after sending a jamming signal (currently we only have this sentence in the paper). However, if a jammer keeps moving while sending the jamming signal, decoding LoRa packets can be easier in few cases and much more difficult in most cases. Our anti-jamming approach can be less effective in such cases and hence we consider that a jammer does not move while sending a jamming signal. However, the jammer cannot continuously transmit at high Tx power as a practical limitation. That is, LoRa signals are not always fully buried by the jamming signal as it represents a scenario for which developing an anti-jamming technique can be infeasible in practice. This type of weak jammer's model (less constraint on jammer) creates a more challenging situation during the mitigation of jamming.

V. ANALYZING A JAMMED SIGNAL

In this section, we analyze different scenarios of a jammed or interfered signal in terms of temporal and spatial variation. We also present our fundamental ideas to develop our proposed anti-jamming technique based on the insights of this analysis.

The goal of our approach is to decode the LoRa packet at the gateway, even when the jamming power is excessively high. Existing works discussed in Section III can successfully decode signals only under low interference power, making them unsuitable for mitigating jamming. LoRa nodes are energy-constrained and must maximize their battery life. However, during jamming, these nodes cannot achieve successful transmissions due to insufficient transmission power to overcome the interference. Consequently, their only option is to attempt retransmission with a higher spreading factor (SF). Unfortunately, this approach also fails under severe jamming conditions and unnecessarily depletes the nodes' batteries.

To assess the impact of severe jamming on the LoRaWAN, we conducted a small experiment by varying the spreading factor and transmission power (results depicted in Figure 15). In this experiment, the jammer and LoRa node used transmission powers of 40 dBm and 0 dBm, respectively, with the jammer and LoRa node positioned at distances of 100 meters and 50 meters from the gateway.

A. Exploiting the Temporal and Spatial Variations of Collisions

As illustrated in Figure 15, the Packet Reception Rate (PRR) is nearly zero (below one percent) under jamming. This may imply that we lose all transmitted bits due to jamming. However, on closer examination, even when a LoRa packet is not received correctly, we may have captured some PHY samples of the LoRa signal that did not collide with the jammer's signal. Figure 3 shows three possible collision scenarios between a LoRa packet and a jamming packet. In cases (a) and (c), temporal offsets exist between the LoRa packet and the jamming packet, enabling the decoding of some PHY samples from both signals at the gateway. The amount of timing offset in both (a) and (c) can vary depending on the specific circumstances.

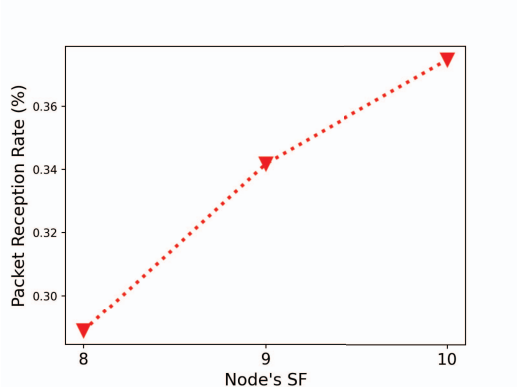
1) How can we leverage these offsets?: If we have information about a portion of the jammer's signal, we can subtract that known part from the collided LoRa packet corresponding to the exact time frame, as depicted in Figure 4. This approach allows us to recover the LoRa signal for that specific time frame. Similarly, the jammer's signal can be recovered if we possess knowledge of the collision-free LoRa signal for that precise time frame.

2) Can we use offsets to recover the entire packet?: With only one timing offset, constructing an entire packet is not feasible. However, if we have multiple different offsets like this, we can leverage them to reconstruct the complete packet. If we place two gateways at two different distances (from the LoRa node or the jammer) and both of them receive the same collided packet (one LoRa packet and a jamming packet), the timing offsets will be different at the two gateways. As illustrated in Figure 5, the recovery of the entire packet becomes possible when using two gateways with different timing offsets between the LoRa and jammer's packet. In Figure 5, Gateway B receives a collided LoRa packet with a higher timing offset, resulting in more non-collided PHY samples. We utilize this information in the initial step and obtain a new offset using the LoRa and jammer's signal at Gateway A. We iterate this process between Gateways A and B until the entire LoRa packet is reconstructed. It is important to note that each iteration provides new segments of both the LoRa and jammer packets, ensuring the successful decoding of the entire packet.

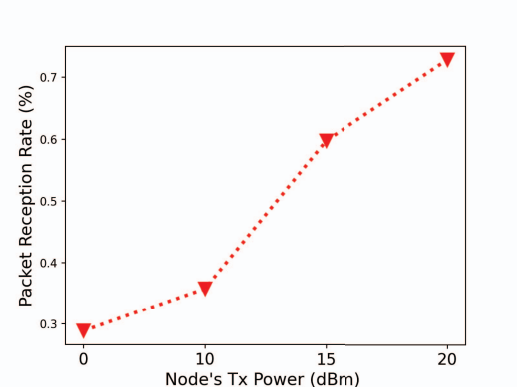
B. Exploiting Offsets of Jamming Mitigation

As discussed earlier, by generating distinct timing offsets for the LoRa and jamming packets across various gateways, we can successfully recover the entire LoRa packet. Importantly, the jamming power and signal quality become irrelevant (considering both packets' individual received signal strength was enough to be decoded) to our proposed recovery technique. Therefore, in cases of severe jamming that cannot be overcome by employing higher Spreading Factor (SF) or transmission (TX) power alone, utilizing multiple gateways strategically placed to ensure different timing offsets becomes a viable solution.

1) How many gateways are needed?: As evident from the depiction in Figure 5, having different timing offsets between the LoRa and jammer packets at two gateways is sufficient to decode a packet. Therefore, at a minimum, we require two gateways to successfully decode jammed LoRa packets. However, our observations indicate that two gateways may not be adequate for every jamming scenario. This is because the jammer can position itself to maintain the same distance from both gateways as depicted in Figure 6, significantly reducing the probability of successful decoding using the aforementioned technique. This issue with two gateways can



(a) Varying Spreading factor



(b) Varying Node's Tx

Fig. 2: Performance of LoRa under severe jamming.

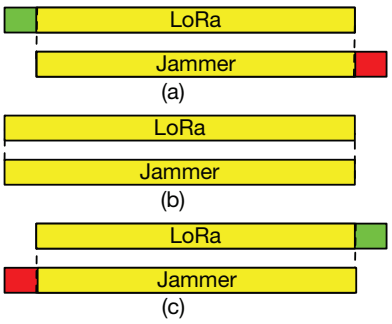


Fig. 3: Different types of packet collisions.

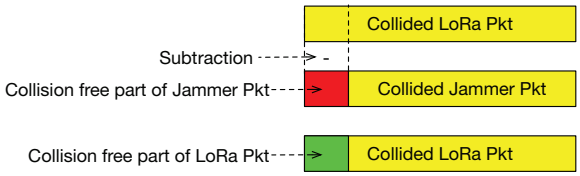


Fig. 4: Collision recovery.

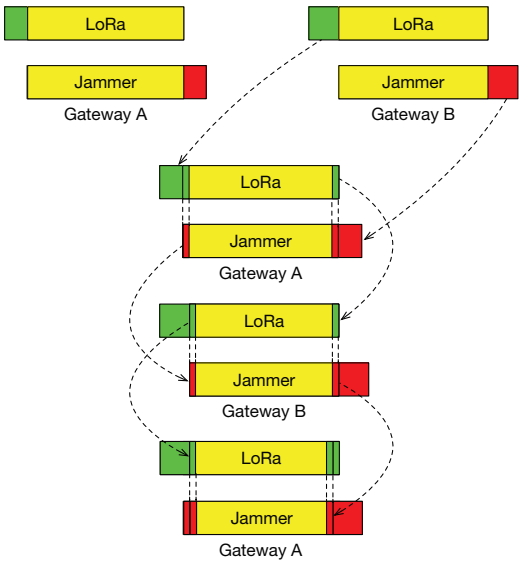


Fig. 5: Packet recovery.

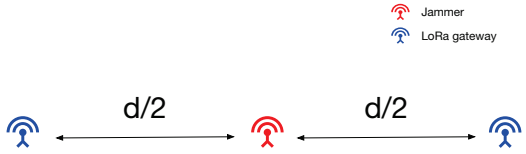


Fig. 6: Jammer having the same distance from both gateways.

be mitigated by using ***three gateways***. Placing three gateways in a straight line, with equal distances between every pair, prevents the jammer from positioning itself to ensure the same distance from all gateways as shown in Figure 7.

2) What should be the distance between gateways?: The necessity of having three gateways in a straight line, with the same distance between every pair, has been discussed above. Now, we need to determine the minimum distance between the gateways. We have observed that since LoRa typically utilizes narrow bandwidths (i.e., $\leq$ 500 kHz), the sampling interval for signal receiving at the LoRa gateway is relatively long (e.g., $\geq$ 2 μs). Consequently, signals arriving within 2 μs (equivalent to a wireless communication distance of 600 m) align with the same PHY sample. To address this, we propose maintaining a distance of at least $\frac{c}{2\times \text{BW}}$ meters between each gateway pair, where c is the speed of the electromagnetic wave, and BW is the bandwidth of LoRa channel. This ensures a minimum difference of at least one PHY sample between two gateways.

C. Special Scenarios

There are some special scenarios that differ from the usual scenario discussed above. For most of these scenarios, our proposed technique is highly suitable. Let us consider a

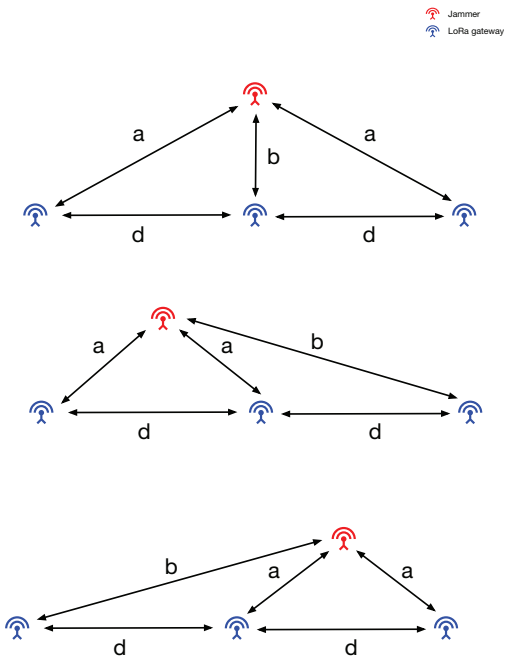


Fig. 7: Jammer not having the same distance from all three gateways.

scenario where the jammer's signal reaches all three gateways while the LoRa node's signal does not reach the third gateway. Now, the jammer tries to position itself to create zero timing offset among the two remaining gateways that receive the LoRa node's signal. Even though the scenario seems tricky, it can be easily handled with our technique. In this scenario, the third gateway is only receiving all the jamming samples, implying we can subtract them from the jammed samples and recover the entire LoRa signal.

The opposite scenario where the LoRa node's signal is received by all three gateways and the jammer's signal is received by two of the gateways can be handled in a straightforward manner, as the third gateway receives a LoRa signal that was not jammed. There is one scenario when there is zero timing offset at two gateways, but there is non-zero timing offset at the third gateway. Even though this scenario seems to be tricky for our proposed technique, the packet can be recovered in a straight forward manner as illustrated in Figure 8.

However, there is a tricky scenario where both the LoRa and jamming signals arrive at the same time or at the same time interval for all three gateways. In this scenario, there is zero temporal offset among all pairs of gateways.

D. Equal offset across gateways.

When the distance between a LoRa node and the jammer is $\leq \frac{c}{2\times BW}$ meters, there might be cases where no timing offset difference exists between any two gateways, as depicted in Figure 9. This occurs because the received LoRa and jamming signals align with the same PHY sample, resulting in zero

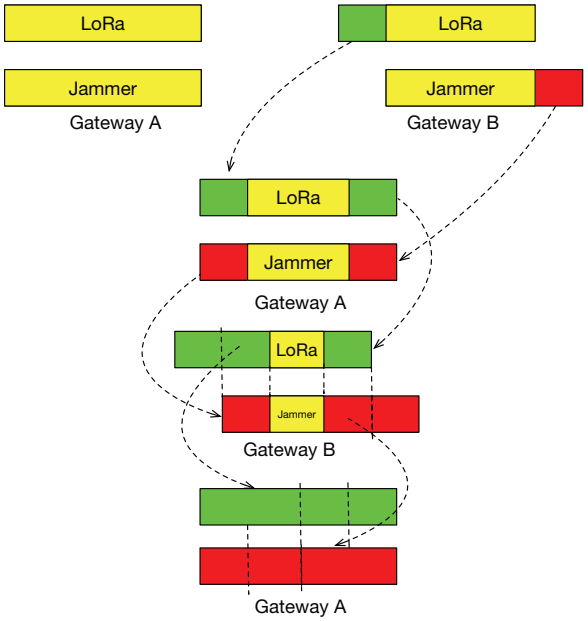


Fig. 8: Zero offset at two gateways.

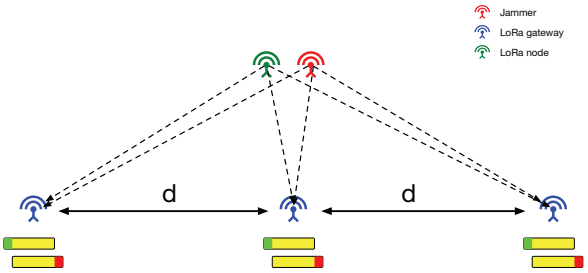


Fig. 9: Zero offset among any pair of gateways

timing offset across all gateways. The previously discussed approach for the usual scenario might not successfully decode the LoRa packet in this situation. It's worth noting that, in practice, the jammer, being illegitimate, would prefer to be physically far away from the users, making this scenario very rare. Moreover, existing approaches for collision recovery in the LoRa MAC layer might help alleviate this situation. The same issue arises when a jammer is located at the mirror position of the node. Nonetheless, this can be addressed by introducing a fourth gateway positioned non-linearly in relation to the other three gateways.

E. Discussion

While we have analyzed showing an equal length of the LoRa packet and the jammer packet in the figures, a similar analysis holds when their packet lengths are different. Specifically, for unequal lengths, in any collided scenario, there will be longer chunks that are collision-free chunks. After separating the collision-free chunks, the scenario reduces to the that of two equal length packets.

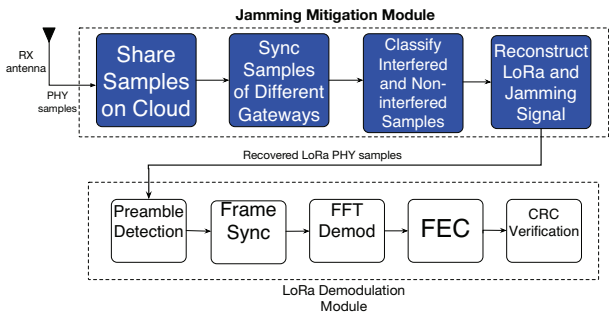


Fig. 10: LoRa Demodulation with Jamming Mitigation.

Developing an anti-jamming technique based on the above ideas and observations involves a number of challenges. First, synchronization of samples between different gateways. Second, distinguishing between interfered and non-interfered samples. Third, making it compatible with existing LoRa demodulation workflow. We describe our anti-jamming technique in the next section.

VI. JAMMING MITIGATION SYSTEM

This section describes the jamming mitigation system proposed in this paper. Note that our objective is to recover a LoRa signal from a jammed scenario (i.e., when a LoRa signal is interfered by a jamming signal). Therefore, we explain our jamming mitigation technique considering one LoRa signal and a jammer's signal. If multiple LoRa signals collide, it is an internal issue (not a jamming issue) of the LoRaWAN that can be mitigated through its link layer protocol or the collided packets can be recovered using any existing collision recovery technique for multiple LoRa packets (discussed in Section III). Our jamming mitigation technique can be combined with those existing collision recovery techniques for multiple LoRa packets to recover packets that collide with either jamming signal or with other LoRa packets of the same network.

A. System Overview

Our jamming mitigation system needs three LoRa gateways that should be placed in a straight line, one being placed at the middle of the other two. These gateways receive incoming LoRa packet PHY samples using their respective Rx antennas. They are configured not to use LoRa preamble detection; instead, they use the signal strength at the Rx antennas to determine whether a signal should be saved. This approach is proposed to ensure the gateways process the jamming samples to help reconstruct the LoRa signal later. There are four modules in the proposed LoRa jamming mitigation system. The modules of the system are as shown in Figure 10.

The first module is responsible for sharing samples with the LoRa Network Server (LNS). The three gateways need a more secure communication protocol to transmit their PHY samples and additional information associated with the samples to the LNS and assist in reconstructing the LoRa packet.

After receiving all the PHY samples, the LNS first synchronizes the samples of all the gateways. This synchronization among the PHY samples of different gateways is essential to ensure the reconstruction algorithm functions properly. This synchronization module leverages the timestamps of the samples along with correlation.

Following the synchronization of the PHY samples of all the gateways, the LNS classifies all the samples and labels them as interfered and non-interfered. This classification is a precursor to the reconstruction algorithm. Finally, the LNS utilizes the synchronized and labeled samples to recover all the LoRa samples using the algorithm described in Section V. After recovering all the LoRa samples, the samples are forwarded to the vanilla LoRa demodulation module.

B. Sharing Samples on Cloud

The first module plays a crucial role in sharing the collected PHY samples with the LNS. The three gateways involved in this process require a more robust and secure communication protocol to effectively transmit their PHY samples and associated information to the LNS. This LNS serves as a hub for aggregating the PHY samples from all gateways through a more secure communication over the Internet. Note that the LNS is already integrated into LoRaWANs. This module adds more security to the existing communication protocol and modifies the protocol to facilitate the transmission of additional information (i.e., timestamps for the samples).

To ensure the security and integrity of the transmitted data, the PHY samples are encrypted before being sent over the internet. This encryption adds an additional layer of protection, safeguarding the sensitive information contained in the PHY samples during the transmission process. Moreover, it ensures the samples are not tampered with by any malicious party.

Furthermore, to facilitate accurate synchronization during the reconstruction phase, the gateways employ a non-overlapping sliding window mechanism to set timestamps for the PHY samples. This allows us to avoid assigning timestamps to every individual PHY sample. This approach serves a dual purpose: it reduces the transmission of unnecessary data, optimizing bandwidth usage, while simultaneously ensuring that there is sufficient data for synchronization during subsequent stages of the process. The use of non-overlapping sliding windows aids in the efficient organization of the timestamped PHY samples, contributing to the overall effectiveness of the jamming mitigation system.

C. Synchronization of Samples

Upon receiving all PHY samples, the LNS initiates a crucial step by synchronizing samples across all gateways. This synchronization is essential for the proper functioning of the subsequent reconstruction algorithm. The synchronization module leverages the timestamps associated with each sample, integrating them with a correlation-based approach.

The timestamps assigned to each sample within the non-overlapping sliding windows play a pivotal role. Each sliding window forms a matrix specific to its respective gateway, organizing the PHY samples in a structured manner. We propose correlating the matrices derived from the timestamped

PHY samples of each gateway. Specifically, each matrix is correlated with the neighboring matrices (based on timestamp) of the other gateways. The neighboring samples with the highest correlation are aligned with each other. In this manner, we give priority to the timestamps associated with each window and then correlate to further refine synchronization.

To enhance synchronization, the sliding window size can be set to one, ensuring every sample has its own timestamp. Although this adds a burden to data transmission between the gateways, it ensures even better synchronization of samples from different gateways. By correlating matrices based on both timestamps and sample content, the synchronization module aims to maximize the accuracy of aligning PHY samples from different gateways. This approach considers not only the temporal aspect through timestamps but also the content of the samples in matrix form. The synergy of these techniques contributes to an effective synchronization process, ensuring that the subsequent stages of the system, particularly the reconstruction algorithm, can reliably and accurately process the synchronized PHY samples.

D. Classification of Samples

The LNS employs a straightforward yet effective method for classifying synchronized PHY samples based on their signal power. After synchronization, each sample undergoes a rapid assessment where its signal power is compared against a predetermined threshold. If the signal power exceeds the threshold, the sample is labeled as interfered; otherwise, it is categorized as non-interfered. This approach simplifies the classification process, relying on a single, easily interpretable criterion to distinguish between interference states. The threshold is determined empirically.

If the signal power is very close to the threshold, suggesting a potential for misclassification, further analysis is needed. The LNS then leverages machine learning to distinguish between interfered and non-interfered samples. This process begins by creating a comprehensive training dataset that includes a diverse range of interfered and non-interfered samples (both real and synthetic), capturing both temporal and spectral characteristics. The LNS adopts a Convolutional Neural Network (CNN) architecture with two hidden layers for the classification task.

CNNs are chosen for their ability to automatically learn features from the input data, making them well-suited for interference classification. Through iterative training, the CNN refines its internal parameters by adjusting weights and biases based on the provided training data. This enables the model to generalize and recognize intricate patterns within the synchronized PHY samples. Once trained, the CNN is applied to the synchronized samples, assigning each a label—*interfered* or *non-interfered*—based on the learned patterns. Note that the training is done in advance, and the trained model is loaded into the LNS for online classification.

The LNS combines both of these methods in two steps. First, signal power thresholds are applied to swiftly identify samples as interfered or not. If the signal power is very close to the threshold, a CNN with two hidden layers is used for classification. After classification, each sample is assigned a label (interfered or non-interfered) for the next step.

E. Reconstruction of LoRa Signal

In the concluding phase of the jamming mitigation process, the LNS utilizes synchronized and classified PHY samples to initiate the recovery of all interfered LoRa samples. This intricate recovery process is executed through our proposed technique detailed in Section V. It is designed to overcome the challenges posed by jamming, ensuring the restoration of the original LoRa signal despite the presence of jamming.

The recovery procedure begins with the LNS utilizing synchronized and labeled samples. Specifically, the algorithm employs a technique where jamming samples are iteratively recovered by subtracting non-interfered LoRa samples from interfered samples. With each jamming sample restored, it proceeds to recover the interfered LoRa sample by subtracting the recovered jamming sample from the initially interfered samples. Through this subtraction process, interference is effectively eliminated, revealing the true LoRa signal obscured by the jamming signal. Note that during this step, any pair of gateways' labeled and synchronized samples is used. If the chosen pair of gateways' samples do not have any offset, a different pair of gateways is chosen until an offset is found. If no pairs of gateways have any offset, the gateway drops that packet.

The recovery of LoRa samples is executed iteratively, continuously recovering LoRa samples until the entire LoRa signal is successfully reconstructed. Upon completion of the recovery process, the recovered LoRa samples are directed to the vanilla LoRa demodulation module. This approach ensures our jamming mitigation module is compatible with the existing demodulation module. The LoRa demodulation module, operating on the now reconstructed samples, can accurately demodulate and extract the original information encoded within the LoRa packets.

In this paper, our focus is on the recovery of LoRa packets affected by jamming signals, disregarding interference from other LoRa signals. Nevertheless, our proposed system remains compatible with existing LoRa MAC layer protocols with a minor (no-cost) modification and can be seamlessly integrated with other intra-network collision resolution techniques at both the PHY and MAC layers if necessary. The primary objective of our LoRa jamming mitigation module is to recover all LoRa samples even before preamble detection in LoRa demodulation, offering an orthogonal approach to current temporal and spectral methods. Our jamming mitigation technique can be combined with existing collision recovery techniques for multiple LoRa packets to recover packets that collide with either jamming signal or with other LoRa packets of the same network.

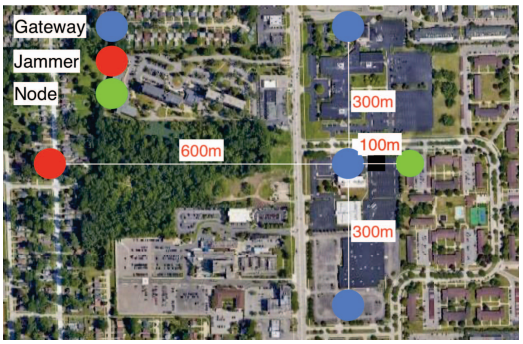


Fig. 11: Setup.

VII. EXPERIMENTS

A. Setup

We conduct our experiment outdoors in a suburban area of the Detroit Metro, utilizing four USRP B200s [32] that serve as three LoRa gateways and one jammer. We employ one Dragino LoRa shield [34] with Arduino Uno R3 [33] as the LoRa node. We use only one LoRa node because the main goal of this paper is to mitigate jamming, and handling interference from other LoRa nodes is out of scope for now. However, our approach can be combined with existing works that handle intra or inter-LoRaWAN interference.

We use a 915 MHz channel, spreading factor of 8, bandwidth of 500 kHz, and coding rate of 4/5 across all the setups. We employ 100 packets with a thirty-second interval for our evaluation. The packet size is 32 bytes for all the experiments. Other than varying distances, all other setups follow the configuration depicted in Figure 11 (jammer's position changes while maintaining the same distance). We consider the middle gateway as the central gateway for this experiment. The middle gateway serves as the gateway for the baseline. We use LoRa as the baseline for comparison, as other existing interference mitigation techniques [24, 25, 26, 27, 28, 29, 30] exhibit a Symbol Error Rate of 100% when SNR is below -35 dBm, implying they cannot decode even a single symbol correctly (to receive a packet, all symbols of that packet need to be decoded correctly). In our experiment, the SNR is consistently below -35 dBm at all the gateways.

B. Experimental Result

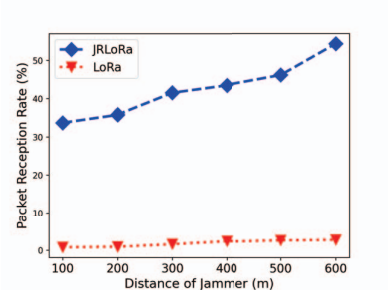
For evaluation, we employ two metrics, namely Packet Reception Rate (PRR) and Energy Per Packet (EPP).

- PRR is calculated as $\text{PRR} = \frac{\text{Total Received Packets}}{\text{Total Transmitted Packets}}$.
- EPP is calculated as $\text{EPP} = \frac{\text{Total Energy for Transmission + Total Energy for Retransmission}}{\text{Total Received Packets}}$.

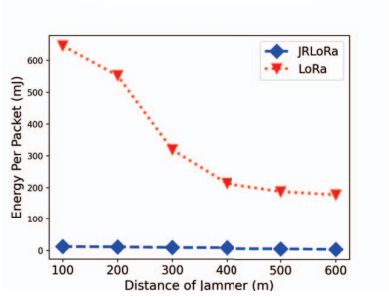
Note that a higher PRR and lower EPP indicate better performance. We name our system JRLoRa (Jamming Resilient LoRa) to present the experiment results and present them. The results are compared with vanilla LoRa (marked as 'LoRa' in the result figures in this section). In experiment, we vary the distance between the jammer and central gateway, the distance between the node and central gateway, the transmission power of the jammer, and the transmission power of the LoRa node.

1) Varying Distance Between Jammer and Central Gateway: In this setup, we vary the distance between the jammer and the central gateway from 100 to 600 meters. It is important to note that we position the jammer at different locations for each distance. Considering the distance as the radius of a circle, the jammer is placed at various points on the circle. We calculate averages for all positions to determine the values (i.e., PRR, EPP) for each distance. The transmission power is set to 13 dBm for the LoRa node and 30 dBm for the jammer. A 100-meter distance is maintained between the LoRa node and the central gateway.

As illustrated in Figure 12(a), the PRR improves for both JRLoRa and LoRa when the jammer is moved away from the central gateway. However, JRLoRa achieves a PRR of 54.34%, while LoRa achieves only 2.83% PRR when the jammer is at a 600-meter distance from the central gateway, indicating an improvement of 19.2 times. For severe jamming conditions when the jammer is at a 100-meter distance, the improvement is 42.74 times. The PRR of JRLoRa increases as the jammer is moved away from the central gateway.



(a) Packet Reception Rate



(b) Energy Per Packet

Fig. 12: Performance varying Jammer's Distance.

As shown in Figure 12(b), the EPP of JRLoRa and LoRa improves (decreases in value) as the jammer is moved away from the central gateway. However, JRLoRa has an EPP of 6.22 mJ, while LoRa incurs a very high EPP of 177.78 mJ when the jammer is at a 600-meter distance from the central gateway, indicating an improvement of around 28.57 times. For severe jamming conditions when the jammer is at a 100-meter distance, the improvement is 53.88 times. The EPP of JRLoRa decreases as the jammer is moved away from the central gateway.

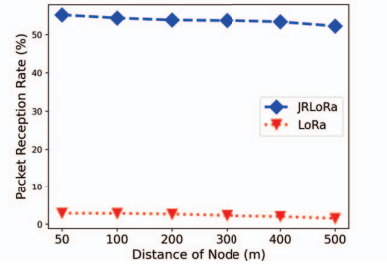
The PRR and EPP improvement with the jammer being far away is attributed to the increased probability of having a good temporal offset among gateways.

2) Varying Distance Between Nodes and Central Gateway: In this setup, we vary the distance between the LoRa node and the central gateway from 50 to 500 meters. It is essential

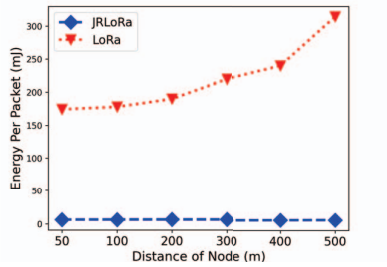
to note that we position the LoRa node at different locations for every distance. Considering the distance as the radius of a circle, the LoRa node is placed at various points on the circle. We calculate averages for all positions to determine the values (i.e., PRR, EPP) for each distance. The transmission power is set to 13 dBm for the LoRa node and 30 dBm for the jammer. A 100-meter distance is maintained between the jammer and the central gateway.

As depicted in Figure 13(a), the PRR declines for both JRLoRa and LoRa when the LoRa node is moved away from the central gateway. JRLoRa reaches a PRR of 55.16%, while LoRa achieves only 2.89% PRR when the LoRa node is at a 50-meter distance from the central gateway, indicating an improvement of 19.11 times. For worse conditions when the LoRa node is at a 500-meter distance, the improvement is 32.34 times. The PRR of JRLoRa decreases slightly as the LoRa node is moved away from the central gateway.

As shown in Figure 13(b), the EPP of JRLoRa and LoRa increases in value when the LoRa node is moved away from the central gateway. JRLoRa has an EPP of 6.08 mJ, while LoRa incurs a very high EPP of 174.15 mJ when the LoRa node is at a 50-meter distance from the central gateway, indicating an improvement of around 28.63 times. For worse conditions when the LoRa node is at a 500-meter distance, the improvement is 47.51 times. The EPP of JRLoRa increases slightly as the LoRa node is moved away from the central gateway.



(a) Packet Reception Rate



(b) Energy Per Packet

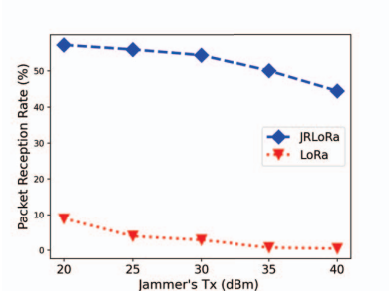
Fig. 13: Performance varying Node's Distance.

The slight deterioration in PRR and EPP with the LoRa node being far away is caused by the slightly decreased probability of having a good temporal offset among gateways.

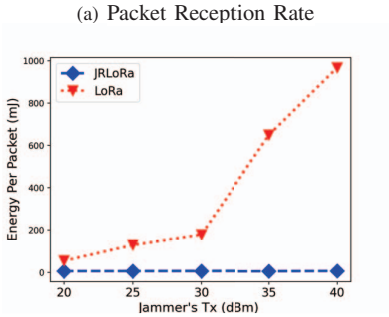
3) Varying Transmission Power of Jammer: For this setup, we vary the transmission power of the jammer from 20 dBm to 40 dBm, using 13 dBm transmission power at the LoRa node. The distance between the jammer and central gateway is set to 600 meters, and the distance between the LoRa node and central gateway is 100 meters. We iterate this experiment multiple times by changing the jammer's position, keeping it on a circle with a radius of 600 meters. We calculate the average to determine the final results (i.e., PRR, EPP).

As depicted in Figure 14(a), the PRR declines for both JRLoRa and LoRa as the transmission power of the jammer is increased. JRLoRa reaches a PRR of 44.3%, while LoRa achieves only 0.528% PRR when the jammer's transmission power is 40 dBm, indicating an improvement of 83.91 times. Under better conditions, when the jammer's transmission power is 20 dBm, the improvement is 6.47 times. The PRR of JRLoRa decreases as the transmission power of the jammer is increased.

As shown in Figure 14(b), the EPP of JRLoRa and LoRa increases as the transmission power of the jammer increases. JRLoRa has an EPP of 8.36 mJ, while LoRa incurs a very high EPP of 966.49 mJ when the jammer's transmission power is 40 dBm, indicating an improvement of 115.65 times. Under better conditions when the jammer's transmission power is 20 dBm, the improvement is 9.5 times. The EPP of JRLoRa increases as the transmission power of the jammer increases.



(a) Packet Reception Rate



(b) Energy Per Packet

Fig. 14: Performance varying Jammer's Tx.

The PRR and EPP deteriorate with the increasing jammer's transmission power. This occurs due to decreasing SNR when the jammer's transmission power increases.

4) Varying Transmission Power of Nodes: For this setup, we vary the transmission power of the LoRa node from 0 dBm to 20 dBm, using 30 dBm transmission power at the jammer. The distance between the jammer and central gateway is set to 600 meters, and the distance between the LoRa node and central gateway is 100 meters. We iterate this experiment multiple times by changing the jammer's position, keeping it on a circle with a radius of 600 meters. We calculate the average to determine the final results (i.e., PRR, EPP).

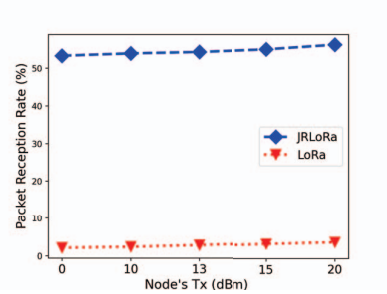
As depicted in Figure 15(a), the PRR increases for both JRLoRa and LoRa as the transmission power of the LoRa node is increased. JRLoRa reaches a PRR of 56.29%, while LoRa achieves only 3.808% PRR when the LoRa node's transmission power is 20 dBm, indicating an improvement of 14.78 times. Under worse conditions, when the LoRa node's transmission power is 0 dBm, the improvement is 25.05 times. The PRR of JRLoRa increases as the transmission power of the LoRa node is increased.

As shown in Figure 15(b), the EPP of JRLoRa and LoRa improves (decreases in value) as the transmission power of the LoRa node increases. JRLoRa has an EPP of 5.89 mJ, while

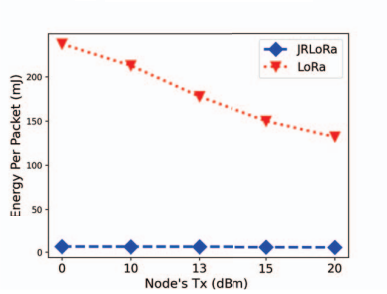
LoRa incurs a high EPP of 131.25 mJ when the LoRa node's transmission power is 20 dBm, indicating an improvement of 22.26 times. Under worse conditions when the LoRa node's transmission power is 0 dBm, the improvement is 37.08 times. The EPP of JRLoRa decreases as the transmission power of the LoRa node increases.

The PRR and EPP improve with the increasing LoRa node's transmission power, attributed to the increasing SNR when the LoRa node's transmission power increases.

From this experiment, it is evident that our proposed system improves the PRR and EPP by up to 83.91 and 115.65 times, respectively, compared to the baseline across different jamming scenarios. Moreover, our system outperforms the baseline in all jamming scenarios.



(a) Packet Reception Rate



(b) Energy Per Packet

Fig. 15: Performance varying Node's Tx.

VIII. Conclusion

In this paper, we have proposed a jamming mitigation system suitable for energy-constraint LoRa devices. With the proliferation of IoT applications relying on LoRa to gather data from dispersed devices and the devices being power-constrained, they become prone to jamming attacks, which can cause massive packet loss, extensive transmission delays, and depleting batteries. Existing work related to jamming in LoRa mainly considers the impact of jamming and does not provide any anti-jamming techniques.

Our proposed jamming mitigation technique for LoRa that imposes no overhead for energy-constrained LoRa nodes and can decode packets even when the SNR is ultra-low. We achieve this by exploiting the temporal and spatial variations of jammed signals of the same packet at different locations. Our design requires no change in LoRa nodes' physical layer, making it usable with all commercial off-the-shelf (COTS) LoRa devices. Our system can be integrated with existing techniques to resolve intra-network collisions as well. Finally, we evaluate the effectiveness of our jamming mitigation technique through outdoor experiments. The results show that our technique mitigates jamming by improving packet reception rate and energy consumption per packet up to 83.91 and 115.65 times, respectively, compared to the baseline.

Acknowledgment

The work was supported by NSF through grants CNS-2301757, CAREER- 2306486, CNS-2306745, and by ONR through grant N00014-23-1-2151.

References

[1] Business Insider, "Smart farming in 2020: How iot sensors are creating a more efficient precision agriculture industry," https://www.businessinsider.com/smart-farming-iot-agriculture, 2020.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE IoT journal*, 2014.

[3] Semtech, "Smart healthcare," https://www.semtech.com/lora/lora-applications/smart-healthcare.

[4] Statista, "Number of Internet of Things (IoT) connected devices," https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, 2021.

[5] https://www.i-scoop.eu/internet-of-things-guide/iot-network-lora-lorawan/.

[6] "LoRaWAN," https://www.lora-alliance.org.

[7] K. Mekkia, E. Bajica, F. Chaxela, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," in *ICT Express*, 2018.

[8] S. Marek, "Comcast will test lorawan iot networks in two markets," 2016, https://www.sdxcentral.com/articles/news/comcast-will-test-lora-iot-network-two-markets/2016/10/.

[9] M. A. Haque and A. Saifullah, "A game-theoretic approach for mitigating jamming attacks in lpwan," *EWSN*, 2023.

[10] C.-Y. Huang, C.-W. Lin, R.-G. Cheng, S. J. Yang, and S.-T. Sheu, "Experimental evaluation of jamming threat in lorawan," in *VTC2019-Spring*. IEEE, 2019, pp. 1–6.

[11] E. Aras, N. Small, G. S. Ramachandran, S. Delbruel, W. Joosen, and D. Hughes, "Selective jamming of lorawan using commodity hardware," in *MobiQuitous*, 2017, pp. 363–372.

[12] N. Hou, X. Xia, and Y. Zheng, "Jamming of lora phy and countermeasure," in *INFOCOM*. IEEE, 2021, pp. 1–10.

[13] K. Mikhaylov, R. Fujdiak, A. Pouttu, V. Miroslav, L. Malina, and P. Mlynek, "Energy attack in lorawan: Experimental validation," in *ARES*, 2019, pp. 1–6.

[14] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra, "Low-power wide-area network over white spaces," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1893–1906, Aug 2018.

[15] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu, "Snow: Sensor network over white spaces," in *SenSys*, 2016.

[16] "Lora alliance," https://lora-alliance.org/.

[17] D. Ismail, M. Rahman, A. Saifullah, and S. Madria, "Rnr: Reverse & replace decoding for collision recovery in

wireless sensor networks," in *SECON*. IEEE, 2017, pp. 1–9.
[18] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *SIGCOMM*, 2009, pp. 159–170.
[19] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *MobiCom*, 2011, pp. 301–312.
[20] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," *SIGCOMM*, vol. 37, no. 4, pp. 397–408, 2007.
[21] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *SIGCOMM*, 2006, pp. 243–254.
[22] L. Kong and X. Liu, "mzig: Enabling multi-packet reception in zigbee," in *MobiCom*, 2015, pp. 552–565.
[23] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *SIGCOMM*, 2008, pp. 159–170.
[24] X. Wang, L. Kong, L. He, and G. Chen, "mlora: A multi-packet reception protocol in lora networks," in *ICNP*. IEEE, 2019, pp. 1–11.
[25] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in *IPSN*. IEEE, 2018, pp. 60–71.
[26] X. Xia, Y. Zheng, and T. Gu, "Ftrack: Parallel decoding for lora transmissions," in *SenSys*, 2019, pp. 192–204.
[27] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *SIGCOMM*, 2017, pp. 309–321.
[28] S. Tong, Z. Xu, and J. Wang, "Colora: Enabling multi-packet reception in lora," in *INFOCOM*. IEEE, 2020, pp. 2303–2311.
[29] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, "Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation," in *SenSys*, 2021, pp. 56–68.
[30] X. Xia, Q. Chen, N. Hou, Y. Zheng, and M. Li, "Xcopy: Boosting weak links for reliable lora communication," *MobiCom*, 2023.
[31] S. Fahmida, V. P. Modekurthy, M. Rahman, and A. Saifullah, "Handling coexistence of lora with other networks through embedded reinforcement learning," in *IoTDI*, 2023, p. 410–423.
[32] "Ettus research," https://www.ettus.com/product/.
[33] "Arduino uno rev3," https://store-usa.arduino.cc/products/arduino-uno-rev3.
[34] "Dragino gps/lora shield," https://www.dragino.com/products/lora/item/102-lora-shield.html.
[35] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *IEEE Communications Surveys Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.
[36] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, 2008.
[37] A. D. Wood, J. A. Stankovic, and G. Zhou, "Deejam: Defeating energy-efficient jamming in ieee 802.15. 4-based wireless networks," in *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2007, pp. 60–69.
[38] A. Proano and L. Lazos, "Packet-hiding methods for preventing selective jamming attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 101–114, 2012.
[39] R. Daidone, G. Dini, and M. Tiloca, "A solution to the gts-based selective jamming attack on ieee 802.15.4 networks," *Wirel. Netw.*, vol. 20, no. 5, pp. 1223–1235, Jul. 2014.
[40] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *WiSec*.
[41] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang, "Coping with a smart jammer in wireless networks: A stackelberg game approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4038–4047, 2013.
[42] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems?part ii: Unknown channel statistics," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 274–283, 2011.
[43] M. Tiloca, D. D. Guglielmo, G. Dini, G. Anastasi, and S. K. Das, "Jammy: A distributed and dynamic solution to selective jamming attack in tdma wsns," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 392 –405, August 2017.
[44] "WirelessHART," 2007, https://www.fieldcommgroup.org/technologies/hart.
[45] H. Mustafa, X. Zhang, Z. Liu, W. Xu, and A. Perrig, "Jamming-resilient multipath routing," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 852–864, 2012.