Boosting Reliability and Energy-Efficiency in Indoor LoRa

Mahbubur Rahman
Queens College, City University of New York
Flushing, New York, USA
mdmahbub.rahman@qc.cuny.edu

Abusayeed Saifullah
Computer Science, Wayne State University
Detroit, Michigan, USA
saifullah@wayne.edu

ABSTRACT
LoRa (Long Range) is a promising communication technology for enabling the next-generation indoor Internet of Things applications. Very few studies, however, have analyzed its performance indoors. Besides, these indoor studies investigate mostly the RSSI (received signal strength indicator) and SNR (signal-to-noise ratio) of the received packets at the gateway, which, as we show, may not unfold the poor performance of LoRa and its MAC (medium access control) protocol – LoRaWAN – indoors in terms of reliability and energy-efficiency. In this paper, we evaluate the performance of LoRaWAN and use its key insights to boost the reliability and energy-efficiency in indoor environments by proposing LoRaIN (LoRa Indoor Network), a new link-layer protocol that can be effectively used for indoor deployments. The approach to boosting the reliability and energy-efficiency in LoRaIN is underpinned by enabling constructive interference with specific timing requirements for different pairs of channel bandwidth and spreading factor and relaying precious acknowledgments to the end-devices with the assistance of several booster nodes. The booster nodes do not need any special capability and can be a subset of the LoRa end-devices. To our knowledge, LoRaIN is the first protocol for boosting reliability and energy-efficiency in indoor LoRa networks. We evaluate its performance in an indoor testbed consisting of one LoRaWAN gateway and 20 LoRaWAN end-devices. Our extensive evaluation shows that when 15% of the end-devices operate as booster nodes, the reliability at the gateway increases from 62% to 95%, and the end-devices are approximately 2.5x energy-efficient.

CCS CONCEPTS
• Networks → Network protocol design. Network reliability;
• Computer systems organization → Sensor networks.

KEYWORDS
LPWAN, LoRaWAN, constructive interference, indoor LoRa

ACM Reference Format:

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

IoTDI ’23, May 9–12, 2023, San Antonio, USA
© 2023 Copyright held by the owner/authors. Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0037-8/23/05...$15.00
https://doi.org/10.1145/3576842.3582327

1 INTRODUCTION
The next-generation Internet of Things (IoT) envisions reliable, energy-efficient, and scalable indoor applications through wireless connectivity between living things, machines, and sensors. These applications may include continuous Ammonia monitoring for the animals in laboratories or barns [33], indoor localization [28, 41], industrial environment monitoring [27], patient monitoring [26, 42], and smart homes [7, 62]. While low-power wide-area network (LPWAN) such as LoRa (Long Range) is adopted mostly outdoors [13, 16, 32, 49, 61], its ubiquity, popularity, low-cost, and scalability have made it a promising technology for indoors as well. Additionally, LoRa may be adopted indoors in contrast to the traditional WiFi or Bluetooth technology because of the following. (i) There are typically many users in the WiFi/Bluetooth band and relatively few users/applications in the subGHz band used by LoRa. (ii) Being lower frequency, the LoRa band can better penetrate obstacles/walls than WiFi or Bluetooth. (iii) WiFi is not well suited for very low traffic or small packet sizes, which are the characteristics of LoRa applications. Even though Bluetooth uses low traffic and small packets, its range is overly short and has lower penetration capability than the LoRa band. Another motivating example is that LoRa and Comcast are soon to be conjoined in Comcast set-top boxes with the hope of proliferating smart home applications through macro, micro, and femto base stations/gateways [1, 14].

A LoRa system may involve one (or multiple) gateway(s) and numerous nodes (i.e., sensors) connected in a star (or star-of-stars) topology [32]. To achieve reliability and different datarates, LoRa may employ different channel bandwidths (Bandwidths (BW)) such as 125, 250, and 500kHz, spreading factors (SFs) such as 7, 9, 10, 11, and 12, and different coding rates (CRs) such as $\frac{4}{5}$, $\frac{4}{6}$, $\frac{4}{7}$, and $\frac{4}{8}$ with a maximum transmission (Tx) power of 14dBm. Despite its promises, only some studies have explored LoRa’s potential in indoor environments [25–28, 41, 42, 65], which show that it is not well-suited for the indoor applications. Also, these studies analyzed mostly the RSSI (received signal strength indicator) and SNR (signal-to-noise ratio) of the received packets at a gateway under the LoRaWAN (LoRa wide-area network) MAC (medium access control) protocol, which may not completely unfold the application scopes where reliability and energy-efficiency are very critical (e.g., in long-term monitoring applications). Furthermore, none of these works has focused on improving the communication reliability and energy-efficiency in an indoor LoRa network.

In this paper, we first evaluate the performance of LoRaWAN through experiments in an indoor environment, focusing primarily on the reliability and energy-efficiency at both the gateway and end-devices (i.e., nodes). In this experiment, we deploy 15 LoRa nodes and one gateway (capable of listening to 8 different channels simultaneously). We run this experiment for a week and find that the reliability at the LoRaWAN gateway can be as low as 62%.
Also, each node makes on average 5.2 transmission attempts to successfully deliver one packet to the gateway. Such low reliability at the gateway despite the high number of Tx attempts per packet is ill-suited for critical indoor applications. Specifically, LoRaWAN performs worse under moderate to heavy network traffic due to (1) the shadowing effect, higher path loss compared to outdoor, and interference between coexisting LoRa nodes and other devices operating in the same frequency band and (2) the LoRaWAN network server, which controls the gateway and acknowledges only the first reception of a packet by a node and never retransmits the acknowledgment (ACK) for any duplicate receptions to avoid the replay attack in the network [2]. Consequently, if a node misses that ACK, its subsequent Tx attempts are wasted.

To boost the reliability and energy-efficiency indoors, we leverage the key insights of our initial experiments on LoRaWAN and propose LoRaIN (LoRa Indoor Network), a novel link-layer protocol for LoRa. Our approach to boosting the reliability and energy-efficiency in LoRaIN is underpinned by creating constructive interference at the gateway and relaying precious ACKs to the nodes, respectively, with the assistance of booster nodes. The booster nodes (or simply boosters) may be a subset of the LoRa nodes, which perform the following. (1) For an ongoing Tx, the boosters may listen to the packet and create constructive interference (hence improving the RSSI) such that it may be decodable by the gateway. (2) They may listen to a one-shot ACK and relay to the node that misses it, thereby stopping the subsequent redundant Tx attempts of a packet. Note that boosting a signal far away may not result in a successful constructive interference due to the temporal displacement between a node and boosters, pathloss, and shadowing effect. The boosting in LoRaIN may thus be well-suited indoors only.

There are, however, a number of challenges to ensure the effective use of the boosters. A booster must send an identical packet both at the same time and on the same channel to create a constructive interference to a packet of a node. Otherwise, this may lead to a two-packet collision scenario. Also, if the booster fails to synchronize quickly with the packet, the node will suffer from high energy consumption due to many Tx attempts of the same packet. Moreover, the booster must ensure that it creates a constructive interference only if the node misses the one-shot ACK for the packet. While relaying an ACK, a booster must also synchronize with both the gateway and the node that expects it. ACK relaying should be fast to avoid energy waste at the nodes. Additionally, a booster must not relay a duplicate ACK to any node that has already received it. In this paper, we address the above challenges and make the following key contributions.

- To create constructive interference, a booster receives and synchronizes to the next Tx attempt of a packet using the carrier activity detection (CAD) feature of the LoRa chip (e.g., SX1276) and the receiving time window as well as an unused octet of the LoRaWAN frame, respectively. To the best of our knowledge, this is the first attempt to create constructive interference in LoRa.
- A booster synchronizes (in both time and frequency) with both the gateway and a corresponding node to receive and relay an ACK using the node’s receiving time window. Additionally, the booster compares the information on two customized octets of the LoRaWAN frame to suppress the duplicate ACKs.

- We derive the maximum allowable temporal displacement between two LoRa transmitters in Matlab simulations for a successful constructive interference for any pairs of BW and SF. As an example, our Matlab simulations show that for a BW of 125kHz and SF of 10, the maximum allowable temporal displacement between two LoRa transmitters may not exceed 6.8243μs, which is considerably less than the corresponding chip duration (1/125kHz = 8μs). To the best of our knowledge, this is the first attempt to derive the maximum allowable temporal displacement between two LoRa transmitters for a successful constructive interference.

- We implement LoRaIN on 20 Dragino LoRa Hat nodes, each running on a Raspberry Pi, and one RAK2245 Pi Hat LoRaWAN gateway. We customize the LMAC 1.6 LoRaWAN code base to facilitate communications between the gateway, boosters, and other nodes. We then deploy these 20 nodes and the gateway in an indoor area of approximately 600ft². Our one-week-long experimental results show that when 15% of the nodes act as boosters, the reliability in LoRa increases from 62% to 95%, and each node consumes 2.5x less energy, thus demonstrating the feasibility of LoRaIN with the commercial off-the-shelf devices.

In the rest of the paper, Section 2 overviews the related work. Section 3 overviews LoRa and its MAC protocol. Sections 4, 5, and 6 describe the system model, rationale, and the design of LoRaIN, respectively. Section 7 presents the implementation and experimental evaluation of LoRaIN. Finally, Section 8 concludes our paper.

2 RELATED WORK

LPWAN Technologies. Recently, many LPWAN technologies have been developed targeting licensed (e.g., cellular band), unlicensed (e.g., ISM band), and TV (e.g., white spaces) bands [32, 49]. LPWANs operating in the licensed band include LTE Cat M1, NB-IoT, and 5G. They require costly infrastructure and high service fees. LPWANs operating in the unlicensed band include LoRa, SigFox, RPMA, IQRF, DASH7, Telensa, Weightless-N/P, IEEE 802.11ah, IEEE 802.15.4k, and IEEE 802.15.4g. Similarly, SNOW has been developed to operate in the TV white spaces [46–48, 50–53]. Among these LPWANs, there is an increasing interest in LoRa from both the academic and industrial communities because of its wide adoption in an increasing number of IoT applications (e.g., smart city, smart farming, environmental monitoring, etc.) [13, 16, 21, 30, 36, 61]. Similarly, in this paper, we mainly focus on the LoRa technology.

Outdoor vs. Indoor LoRa. Most existing work on LoRa focuses on evaluating and improving its performance in outdoor environment and scenarios through modeling, experiments, and simulations [11, 20, 22, 23, 29, 31, 35, 37, 43, 44, 55, 56, 66]. Some measurement study in [25–28, 41, 42, 65] has recently showed the performance of LoRa in indoor environments in terms of RSSI and SNR at the gateway. To the best of our knowledge, no work has yet studied the reliability and energy-efficiency aspects of indoor LoRa at both the gateway and the LoRa nodes/sensors. In this paper, we show that only RSSI-based and SNR-based measurements (which are also the basis of outdoor measurements) may not unfold all the shortcomings that are responsible for the poor performance of LoRa gateway or nodes in indoor scenarios/applications. Additionally, through our indoor evaluation of LoRa, we propose LoRaIN to boost
its performance indoors in terms of reliability and energy-efficiency. Note that the adoption of boosters makes our work specific to indoor scenarios since boosting a signal (by creating constructive interference) in outdoor scenarios far away may not be feasible because of the time difference in signal propagation from the boosters and other LoRa nodes to the gateway, and vice versa. Additionally, outdoor pathloss and shadowing effects may reduce the chance of constructive interference in LoRaIN.

**Collision Recovery Techniques vs. LoRaIN.** Recently, several works have been proposed to recover the packets at the gateway from a collision of multiple LoRa packets [19, 58, 59, 63, 64]. Choir [19] utilizes the distinctive hardware imperfections of the LoRa nodes. FTrack [64] utilizes the distinctive signal edges of the LoRa nodes. mLoRa [63] applies successive interference cancellation. CoLoRa [59] transforms the time offsets between the collided packets to frequency domain information. NScale [58] scales up the FFT peaks of collided signals using a non-stationary signal to disentangle the collision. To the best of our knowledge, these techniques are applicable in both outdoor and indoor scenarios as long as the LoRa signal propagation is intact. However, they do not consider reliability at the LoRa nodes, which may be critical in many scenarios (e.g., confirmed messaging for control applications).

In fact, LoRaWAN may require a new operational class along with different spreading factors and spreading code rates. This is because the LoRa PHY layer becomes resilient (to some extent) to the Doppler effects and channel noise.

**Boosting Reliability and Energy-Efficiency in Indoor LoRa.** In LoRaIN, we reinforce the strengths of certain packets via constructive interference and then decode them. This enables the decoding of certain packets (e.g., packets of the nodes that suffer most) from many non-decodable concurrent ones with different signal strengths and temporal displacements.

![Figure 1: A typical LoRaWAN architecture.](image-url)

**3. AN OVERVIEW OF LORA AND LORAWAN**

**3.1 LoRa Physical Layer**

The LoRa PHY layer implements a *chirp spread spectrum* (CSS) modulation, where it encodes data using a linear frequency variation in a channel over time [45]. To encode 0’s (or chirp “0”) and 1’s (or chirp “1”), it differs the initial frequency in the chirps (a.k.a. symbols). In demodulation, a LoRa receiver multiplies an incoming chirp with a down-chirp (whose frequency changes linearly from BW/2 to -BW/2 over a channel of bandwidth BW) and then applies a Fast Fourier Transformation (FFT). The FFT leads to a peak in a frequency bin, revealing the delay of the received chirp. The LoRa receiver decodes the chirp by tracking the location of that frequency bin. To make it more robust, a LoRa transmitter may use different CRs such as $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, or $\frac{1}{32}$. To control the number of bits per chirp, it may also use different SFs between 7 and 12. Additionally, the LoRa PHY may choose different channel bandwidths (e.g., BWs such as 125, 250, or 500kHz). As the chirps fully utilize the channel bandwidth BW, the LoRa PHY layer becomes resilient (to some extent) to the Doppler and multi-path effects and channel noise.

**3.2 LoRaWAN Architecture and Basics**

As shown in Figure 1, a LoRaWAN network consists of end-devices (i.e., nodes/sensors), one or more gateways, a network server, and one or more application servers. The LoRaWAN frequency band is divided into two parts: multiple *uplink* and multiple *downlink* channels. The nodes send data to the gateways over the uplink channels. The gateways then pass the data to the network server. The network server deduplicates (if necessary) and sends the data to the application server, as necessary. On the other hand, the network server may send messages (e.g., for management or on behalf of the application server) through the gateways. The gateways communicate with the nodes using the downlink channels. LoRaWAN categorizes the nodes in three classes (class-A, class-B, and class-C) based on when they want to receive downlink messages. These classes directly determine the energy-efficiency of the nodes. In the following, we briefly discuss these classes.

**3.2.1 LoRaWAN Classes.** In LoRaWAN, all the nodes are required to support class-A (“Aloha”). A node may spend most of the time in
We consider the indoor applications that require high reliability. We describe the design rationale of our protocol: LoRa IN (LoRa IN). While the nodes (including the boosters) can be either wall-powered or battery-powered, it is beneficial to avoid or nullify interference in any network, which may be caused by redundant retransmissions by the nodes (as explained in Section 1). In LoRa IN, we achieve the above while providing energy-efficiency at the nodes using boosters who use the ultra-low-power class-C (“Continuous”) nodes never go to sleep. They constantly listen for downlink messages from the network server, except when they have their own data to transmit. As a result, they consume the most energy across all the classes.

3.2.2 Unconfirmed and Confirmed Messaging. A message from a node to the network server and vice versa may be confirmed or unconfirmed. In the case of confirmed messaging, the sender requests an ACK from the receiver. When a node sends a confirmed message to the network server, it makes up to 8 Tx attempts until it gets an ACK. In unconfirmed messaging, a sender does not request an ACK from the receiver.

4 LORAIN SYSTEM MODEL

We consider the indoor applications that require high reliability and energy-efficiency at the nodes. LoRa IN may have one or multiple gateways and numerous nodes in the network. We, however, evaluate LoRa IN’s performance with one gateway and a subset of the nodes working as the boosters. This setup is similar to the idea of having one Wi-Fi access point and a few range extenders (if needed) to improve the Wi-Fi network performance in a home or indoor scenario. Additionally, using multiple gateways instead of boosters may be cost prohibitive. A commercial gateway may cost, on average, US$250 (8-channel) to US$2,494 (16-channel) [8]. Having boosters instead of multiple gateways is thus an economic and favorable solution indoors since they are a subset of the nodes and incur no additional cost. In LoRa IN, the gateway is wall-powered, while the nodes (including the boosters) can be either wall-powered or battery-powered (which gives much freedom of installation and avoids wiring cost and complexity in the smart building use cases [34]). Wall-powered or battery-powered, it is beneficial to avoid or nullify interference in any network, which may be caused by redundant retransmissions by the nodes (as explained in Section 1). In LoRa IN, we achieve the above while providing energy-efficiency in the nodes using boosters who use the ultra-low-power CAD feature to participate in the boosting activities.

In LoRa IN, we adopt the class-A mode of LoRa W A N in the nodes because of its energy-efficiency. To ensure the reliability in data transfer, we adopt the confirmed uplink messaging of LoRa W A N. Note that a node makes up to 8 Tx attempts to get an ACK in confirmed uplink messaging. The network server acknowledges (via the gateway) only the first received Tx of a packet by a node and never retransmits it. Similar to LoRa W A N, we do not allow the gateway to send multiple ACKs for multiple Tx attempts of the same packet. The reasons for this are as follows. (1) If the first ACK is not received by the node, it is highly likely that the node will not receive the following ACKs as well. This may be because of the bad link quality between the gateway and the node. (2) Allowing multiple ACKs may result in a replay attack in the network. In the rest of the paper, we denote the messages from the network server (via the gateway) to the nodes simply as the messages from the gateway to nodes. Also, we denote the messages from the nodes to the network server (via the gateway) as the messages from the nodes to the gateway.

5 LORAIN DESIGN RATIONALE

We now analyze the performance of LoRa W A N indoor. Specifically, we analyze the reliability and energy requirements for both the gateway and the nodes and use these insights to design LoRa IN.

5.1 Reliability Analysis of LoRa W A N

First, we analyze the reliability at the LoRa W A N gateway and various number of nodes for confirmed uplink communication. We deploy a LoRa W A N network in an indoor area of approximately 600 ft², as shown in Figure 2. We use one LoRa W A N gateway (dark circle) and 15 nodes (labeled N1–N15). In this setup, the gateway runs a ChirpStack LoRa W A N network server [3] locally and the
nodes operate in LoRaWAN class-A mode. Our gateway is capable of receiving in eight 125kHz uplink channels. The gateway also uses eight 500kHz downlink channels to send the ACKs. The geographic location of our network does not have any duty cycle requirements in the channels and does not allow a SF of 12 as well. The adaptive data rate (ADR) feature of LoRaWAN is enabled at both the network server and the nodes. ADR adapts the SF and CR dynamically to improve the LoRaWAN signal/packet receptions. In our experiments, we find that the SF varies between 7 and 10 while the CR is fixed at 4/5. Also, the average RSSI and SNR at the gateway are approximately -44.7dBm and 9.5dB, respectively, when we transmit packets with a Tx power of 14dBm. The experimental data set has been made available online [10].

With the above setup, each node sends 100 confirmed uplink packets with an inter-packet interval of 1 minute. This interval is common in many indoor applications such as Ammonia monitoring for barn animals [33] and patient monitoring in hospitals. We send packets from 3 to 15 nodes with different payload lengths between 10 and 50 bytes. Figure 3 shows the reliability in the gateway and the nodes in the forms of packet reception rate (PRR) and packet delivery ratio (PDR), respectively. PRR at the gateway is defined as the ratio of the number of packets received at the gateway to the total number of packets sent by the nodes. On the other hand, PDR at the nodes is defined as the ratio of the number of ACKed packets to the total number of packets sent by the nodes. Since there is no idea of ACK in PRR, we use two different metrics to evaluate the reliability at the gateway and nodes.

5.1.1 Packet Reception Rate. As shown in Figure 3(a), the PRR at the LoRaWAN gateway is approximately 82.5% when 3 nodes transmit to the gateway with a payload of 10 bytes. As the number of nodes increases, the PRR at the gateway decreases significantly. For example, when 15 nodes transmit 10-byte payloads, the PRR goes down to approximately 69.3%. Figure 3(a) also shows that this decreasing trend in the PRR is steady for all packet sizes. When 15 nodes transmit 50-byte packets, the PRR at the gateway is as low as 62%. LoRa observes such low PRR at the gateway because of the interference due to severe multi-path and shadowing effects in indoor and packets (on the same channel) collisions, resulting in many packets being lost. Although the LoRa modulation allows the gateway to recover packets below the noise floor, the gateway may not be able to decode a packet residing within the above interference scenario. The reason is that the FFT at the gateway may not be able to distinguish between the up-chirps and down-chirps in the received signal because of the data availability (or unavailability) in the undesired (or desired) frequency bins.

5.1.2 Packet Delivery Ratio. As shown in Figure 3(b), when 3 nodes send 10-byte payloads, the average PDR is approximately 15.4% at the nodes. Also, as the number of nodes increases, they observe even lower PDRs. For example, the average PDR at the nodes is approximately 14% when 15 nodes send 10-byte payloads. Figure 3(b) also shows that as we vary the payload size, the packet delivery ratios at the nodes follow a similar pattern. When 15 nodes transmit 50-byte payloads, the average PDR is approximately 10.8%. The nodes observe such low PDRs due to the following two potential reasons. (1) The gateway cannot decode the received signals and thus does not send ACKs. (2) The ACK sent for a packet is not received/decoded at the corresponding node.

5.1.3 Discussion. Both PRR and PDR in this experiment are considerably very low, while the latter is much worse. This also means that a large number of correctly decoded packets at the gateway are redundantly retransmitted by the nodes. To this extent, we propose to boost the reliability of indoor LoRa by introducing LoRaIN.

5.2 Energy Requirement Analysis of LoRaWAN

We now analyze the energy requirements at the nodes. Equation (1) below estimates the relationship between the energy consumption and the Tx attempts for a packet.

\[
E_{\text{packet}} \approx N_{\text{attempt}} \times (E_{\text{air}} + E_{\text{Rx1}} + E_{\text{Rx2}})
\]

Here, \(E_{\text{packet}}\) is an estimation of the packet’s total energy consumption, \(N_{\text{attempt}}\) is the total attempts for a packet, \(E_{\text{air}}\) is the energy consumption for each Tx airtime, \(E_{\text{Rx1}}\) is energy consumption in the first receive (Rx) window, and \(E_{\text{Rx2}}\) is the energy consumption in the second Rx window. For simplicity, we do not include the energy consumption for the Tx–Rx radio switches and Rx-delays between the Tx window and Rx windows as their contributions are negligible. In the following, we analyze the Tx attempts by the nodes with the setup similar to the analysis in Section 5.1.

5.2.1 Considering All the Packets in Uplink. Here, we analyze the number of Tx attempts while considering all the packets scheduled from the nodes to the gateway.

All Transmission Attempts. Figure 4(a) shows the average number of Tx attempts per packet for various number of nodes. When 3 nodes send 100 packets each, each packet with a 10-byte payload, the average number of Tx attempts per packet is approximately 4.9.
This figure also shows that as the number of nodes increases, the average number of Tx attempts per packet also increases. When 15 nodes transmit packets with 10-byte payloads, the average number of Tx attempts is approximately 5.3. This increasing trend in the average number of Tx attempts remains steady for the packets of different payload sizes. For example, the average number of Tx attempts per packet is approximately 4.1 compared to 5.5 when 3 nodes and 15 nodes send packets, each with 50-byte payloads, respectively. The main reasons for such high numbers of Tx attempts by the nodes are twofold. (1) The gateway sends an ACK but the corresponding node is unable to receive the ACK. (2) The gateway does not send ACK for the subsequent Tx attempts by a node for which an ACK has already been sent for a prior Tx attempt. In the following, we further investigate the above two cases.

**Lost/No Acknowledgment Count.** In Figure 4(b), we show the analysis on the number of ACKs for all the packets in the uplink. Here, for any number of nodes between 3 and 15 that send packets with payload sizes between 10 to 50 bytes, the aggregate numbers of lost (at the nodes) and unsent (by the LoRaWAN server) ACKs vary approximately between 77% and 83%. Such a poor performance due to the design choices of the LoRaWAN network server can be considered a serious drawback for energy-constrained IoT nodes. In fact, lost/no ACKs is the most critical reason for the large number of unnecessary Tx attempts by the nodes.

**Discussion.** Figure 4 shows that it is critical to boost the performance of LoRaWAN in terms of the number of Tx attempts by the nodes to deliver a packet. Similarly, LoRaWAN needs a robust ACK mechanism so that it may enable high PDRs and reduce the average number of Tx attempts per packet, thereby not wasting node’s invaluable and limited energy budgets.

5.2.2 *Considering the Delivered Packets.* We now analyze the number of Tx attempts of the packets for which ACKs are received by the nodes (thus considered delivered). Figure 5(a) shows the cumulative distribution function (CDF) of the number of Tx attempts of all the packets for which ACKs are received by the nodes. When 3 to 15 nodes transmit with payloads of sizes between 10 and 50 bytes, approximately 20% of the packets are delivered to the gateway through single Tx attempts, and approximately 40% of the packets require 1 to 2 Tx attempts. The rest of the packets (approximately 60%) need up to 8 Tx attempts, which results in huge energy consumption at the nodes. Additionally, we take such 1000 packets with payload sizes varying between 10 and 50 bytes from the experiments and plot the numbers of Tx attempts by the nodes in Figure 5(b). As shown in this figure, when 3 to 15 nodes transmit packets with payloads of sizes between 10 and 50 bytes, on average 650 out of 1000 packets need 3 or more Tx attempts.

**Discussion.** As we analyze the number of transmission attempts for the delivered packets, we find that LoRaWAN gateway indeed misses a lot of packets, requiring the end-devices to make subsequent transmission attempts that are necessary to deliver the packets. Overall, our analysis (as shown in Figure 4 and Figure 5) reveals that the LoRaWAN network server performs poorly in indoor scenarios in terms of energy requirements at the end-devices and reliability at the gateway and end-devices. Hence, it is important to boost the energy-efficiency in indoor LoRaWAN.

6 DESIGN OF LORAIN

6.1 Design Principles

6.1.1 *Booster Selection.* In LoRaIN, we boost the reliability and energy-efficiency by introducing boosters in the network. Boosters are a subset of the LoRa nodes in the network. These nodes may be selected during the network deployment phase or later. Due to the uncertain noise or interference characteristics in the indoor environments, the number of boosters may be decided dynamically by the LoRaWAN network/application server from the set of LoRa nodes (e.g., nodes that observe better energy efficiency or that are application specific). For this purpose, the network/application server may retain an editable/configurable list of these boosters.

![Diagram of LoRaIN network with boosters](image)

Figure 6: Steps (denoted by numbered circles) of boosting the reliability in LoRaIN. "•" represents no/corrupted LoRa signal/packet, "single tick" represents a valid LoRa signal/packet, and "double tick" represents a superimposition of multiple LoRa signals/packets.

6.1.2 *Boosting Reliability.* The LoRaWAN gateway greatly suffers to decode packets in indoor environments. Although the LoRa modulation allows the gateway to recover packets residing below the noise floor, the gateway may not be able to decode a packet residing within the interference created by severe multi-path and shadowing.
effects, other LoRaWAN nodes and/or networks operating in the same frequency band. The FFT algorithm at the gateway cannot distinguish between bits 1s and 0s in the received signal because of the data availability in the undesired frequency bins or data un-availability (due to destructive interference or severe path loss) in the desired frequency bins. For a packet reception at the gateway, we boost the decoding by creating a constructive interference of the packet using the boosters. When a constructive interference of the packet is created, the energy levels in the desired FFT frequency bins will supersede the undesired energy levels in the other frequency bins, thereby improving the chances of decoding the chirp “0” and chirp “1” at the gateway. To be more specific, due to its capture effect capability [15, 18, 24], a LoRa receiver (e.g., gateway/node) locks to the signal/packet that is stronger compared to the others in the same (or nearby) frequency. In summary, we ensure that a packet has the highest signal strength and may be subject to the receiver’s capture effect by creating a constructive interference of the packet.

In Figure 6, we explain the steps (denoted by the numbered circles) for creating the reliability boost at the gateway in the uplink. For better understanding, we explain the steps involving one node, one booster, and the gateway. As shown in this figure, the booster first listens, decodes, and stores a Tx attempt of a packet by the node. If the gateway does not receive the packet, the node retransmits the packet (up to seven times) to the gateway. In the case of decoding error in reception, the gateway does not send an ACK, and hence the node knows that it has to retransmit the packet. Along with the retransmission attempts by the node, the booster also transmits the same packet at the same time and frequency (i.e., channel) to the gateway, thereby creating a constructive interference (hence a capture effect) and enhancing the packet reception at the gateway. In LoRaIN, multiple such boosters may transmit the same packet to create a stronger constructive interference-cum-capture effect at the LoRa gateway.

6.1.3 Boosting Energy-Efficiency. The LoRaWAN network server acknowledges only the first received Tx attempt of a packet (by a node) and never retransmits the ACK for the subsequent attempts of that packet. As LoRaWAN allows up to 8 Tx attempts (i.e., 7 retransmissions) of a packet, the number of wasted Tx attempts may be up to 7 if a node misses the ACK sent for the very first attempt that the gateway received. This situation causes a huge amount of energy wastage at the nodes (as per Equation (1)). As a result, the lifetime of the nodes may become significantly reduced. In LoRaIN, we enhance the energy-efficiency at the nodes by reducing the number of Tx attempts by them. Specifically, we utilize the boosters to enable ACK relay from the gateway to the nodes so that they do not waste energy by making unnecessary Tx attempts.

In Figure 7, we explain the steps of enabling energy-efficiency in LoRaIN using one node, one booster, and the gateway. Specifically, we achieve energy-efficiency by relaying the missing ACKs to the nodes via the boosters. As shown in this figure, the booster first listens to the ACK (along with the node that is expecting an ACK) that is sent by the gateway. If the node misses the ACK, the booster relays the ACK to the node (step 2 in this figure). Upon reception of the relayed ACK, the node refrains from making unnecessary retransmissions of the corresponding packet, thereby reducing the energy consumption. In LoRaIN, multiple boosters may relay the same ACK at the same time (not shown in this figure), thereby creating a constructive interference of the relayed-ACK that may boost the capture effect at the node.

6.1.4 Overall Workflow of the Boosters. As discussed above, a booster participates in enhancing both the reliability at the gateway and energy-efficiency at the nodes. Therefore, it has to have a non-conflicting workflow to accommodate both of these aspects. For this, a booster maintains the following workflow. At the beginning, it hops on to different LoRaWAN uplink channels and listens for the uplink packets. In each uplink channel, the booster listens for a fixed duration (will be discussed in Section 6.3). If it can decode any packet in any channel, it immediately starts listening for an ACK in the corresponding downlink channel for another fixed duration (will be discussed in Section 6.4). Later, depending on the status of the packet reception at the gateway/itself or the ACK reception by the node, the booster may transmit the packet to create constructive interference at the gateway and/or relay the ACK to the node. The booster may keep repeating this workflow in between its own packets Tx to the gateway.

6.2 Challenges in LoRaIN

The boosters face several critical challenges to boost the reliability and energy-efficiency in LoRaIN.

6.2.1 Challenges in Boosting Reliability. As shown in Figure 6, a booster helps to create a constructive interference at the gateway. For this, it must send the same physical layer frame to the gateway along with the node at the same time and on the same channel. LoRaWAN does not provide any mechanism such that the nodes may synchronize themselves to boost each other’s signals. The lack of synchronization between a booster and a node in terms of packet, time, and channel will result in severe performance degradation at the gateway due to the additional network traffic introduced by the boosters. Additionally, it is challenging for a booster to know if it really needs to transmit the packet (received from a node) to create a constructive interference. It is thus very crucial that we address these challenges in LoRaIN.

6.2.2 Challenges in Boosting Energy-Efficiency. To boost the energy-efficiency at the nodes, as shown in Figure 7, the boosters relay the ACKs from the gateway to the nodes. We need to address the following challenges in order to make the ACK relay beneficial for the nodes. (1) We must synchronize a booster (in terms of ACK packet, time, and channel frequency) with the ACK receive window of the desired node. (2) We must make sure that a booster does not
6.3 Creating Constructive Interference

In the following, we explain our techniques that create constructive interference at the gateway. Specifically, we synchronize the boosters and the nodes in terms of packet, time, and channel. Also, we discuss how a booster decides if it needs to transmit a packet for creating constructive interference or not.

6.3.1 Packet Synchronization. It is crucial that a booster sends the same physical layer frame along with a node to the gateway in order to create a constructive interference. Otherwise, it may lead to an effect similar to two packets collision at the gateway, despite having a tight time and frequency synchronization between the booster and the node. In the following, we describe how a booster receives a packet from a node, which it later transmits to the gateway.

![Figure 8: LoRaWAN message structure (PHDR: PHY header, MHDR: MAC header, M: message integrity code, FHDR: frame header, DevAddr: device address, FCtrl: frame control, FCnt: frame count, FOpts: frame options) [2].](image)

To receive a packet from a node, a booster first listens to the uplink Tx in the medium. The use of spread spectrum modulation in LoRa makes it impractical for the boosters to use an RSSI-based detection of signals in the medium. The reason is that the signal may reside below the noise floor. To this extent, we utilize the carrier activity detection (CAD) feature of the LoRa chips, which is not used in LoRaWAN [6]. In CAD mode, the booster first probes for a preamble of a packet (Figure 8) in the medium for a fixed duration. In a channel with spreading factor $SF$ and bandwidth BW, the duration for a CAD is $(2^{SF} + 32) \div BW$ seconds, which is approximately the duration of two LoRa symbols [6]. The booster may know about the SF and BW from the gateway when requested to operate in the boosting mode. Once the booster senses an activity in the channel, it looks for the start frame delimiter (SFD) of the preamble, which is 2.25 down-chirp symbols to synchronize and receive the rest of the packet (i.e., PHY header, Header CRC, payload, and CRC). In LoRaIN, we use a preamble length of 10.25 symbols, which is similar to the existing LMIC LoRaWAN implementation [5]. The booster may have to run the CAD several times with an interval that suits its own packet Tx. Additionally, it may have to hop to different channels (as per the gateway’s request) and run CAD to detect a preamble.

6.3.2 Time Synchronization. Once the booster has an identical copy of the packet of a node, it transmits the packet to the gateway along with the retransmission by the node. The booster, however, must synchronize in time with the node. Otherwise, the packet from the booster may create a destructive interference to the node’s retransmission. Below, we describe our technique to avoid the above scenarios and create the desired constructive interference.

![Figure 9: LoRaWAN node receive slot timing in Class-A mode of operation [2].](image)

To synchronize the time between a booster and a node, we utilize the node’s receive slot timing window. As shown in Figure 9, following an uplink Tx, the node opens two short receive windows Rx1 and Rx2. The end of transmit is the reference point for the start times of Rx1 and Rx2, which are Receive_Delay1 and Receive_Delay2, respectively. These delays are region specific and known to all the devices in the network [2]. Typically, Receive_Delay2 = Receive_Delay1 + 1 seconds. While datarate for Rx1 is fixed and identical to the transmit datarate, the datarate for Rx2 is region-specific and known to all the devices. If the node does not receive an ACK in any of these windows, it transmits the packet after Receive_Delay2 + r seconds, where $r$ is the required duration (region-specific) to detect and receive an ACK in Rx2. A booster also follows the same timing and transmits the packet.

6.3.3 Channel Synchronization. The boosters must transmit in the same channel (i.e., frequency) as a node to create the constructive interference. Otherwise, the packets from the boosters will interfere the ongoing Tx in the undesired channels. To synchronize the channel between a booster and a node, we follow the default channel increment procedure of LoRaWAN. In LoRaWAN, a node retransmits a packet on Channel_prev = (Channel_prev + 1) mod N, where Channel_prev is the channel in which the last Tx was lost and $N$ is the number of uplink channels the LoRaWAN gateway is capable of listening to. Since the booster knows Channel_prev during the packet synchronization, it calculates the desired channel based on the above equation and transmits to the gateway in order to create a constructive interference.

6.3.4 Decision on the Reliability Boost. It is critical for a booster to know if it needs to transmit a packet to create the constructive interference, despite being able to synchronize with the packet, time, and channel of a node. We achieve this in the boosters using the following technique. A node encodes 3 bits of additional information (to represent 8 Tx attempts) in the last octet (out of 15) of the FOpts field (Figure 8), which is unused in LoRaWAN [2]. Specifically, we use the flow of 3-bit natural binary numbers to represent the Tx attempt count. For example, we use 000 to denote the first Tx attempt, 001 to denote the second Tx attempt, and so on. Upon receiving a packet, a booster checks this information. Later, if it finds that no ACK is sent in Rx1 or Rx2 (Figure 9) for the node and the node has not exhausted all the attempts yet, it transmits the packet at the specified time and channel.
6.4 Enabling Acknowledgment Relay

In the following, we detail our techniques to overcome the challenges related to the ACK relay in LoRaIN. Specifically, we discuss how the boosters synchronize the time and channel to relay the ACKs as well as avoid relaying duplicate ACK or those that may introduce false negatives in the network.

6.4.1 ACK Packet, Time, and Channel Synchronization. It is crucial for a booster to correctly receive an ACK and relay to a node at the exact time and channel as expected. Otherwise, the relayed ACK may not be decoded correctly at the nodes and/or collide with other ACKs in the network, which will increase the number of Tx attempts by the nodes. For this, we again utilize the LoRaWAN node’s receive slot timing (Figure 9) to receive the ACK packet and synchronize the time of relay between a booster and a node. Since the booster is already synchronized with the transmit window (Section 6.3.1), it can receive the ACK in Rx1 or Rx2 from the gateway and relay it during the next Rx1 and Rx2 slots of the node. To synchronize the channel, we use the default channel mapping of the Tx-Rx operations in LoRaWAN, which is defined by Rx1 as ChannelRx1 = ChannelRx2 mod 8, where 8 is the number of downlink channels in LoRaWAN. For Rx2 Tx-Rx channel mapping, we use the first downlink channel that is also used in the existing LMIC LoRaWAN implementation [5]. Also, We use the respective datarates of Rx1 and Rx2, as discussed in Section 6.3.2. Upon receiving an ACK, a node checks for the DevAddr and MIC fields (Figure 8) to check if the ACK is intended for it or not. If multiple boosters relay the same ACK to a node, it will create a constructive interference, and hence the node has even higher chances of receiving the ACK due to its capture-effect capability.

6.4.2 Handling Duplicate ACKs. While relaying an ACK, a booster should avoid sending duplicate copies of the same ACK that has already been received by a node. Otherwise, that may collide with and destroy a legit ACK for another node on the same channel. Such duplication may be introduced because the node has already received the ACK from the gateway or a booster. To avoid this, a booster compares two packets received in the latest two consecutive Transmit windows (Figure 9). Specifically, the booster compares both the FCntUp and 3 bits of (that are encoded by the node) FOpts fields. A node increases FCntUp by one only for each packet with new payload data but keeps it the same for a retransmission attempt of an older packet. Thus, a booster relays an ACK only if the latest two FCntUp fields (of those packets) are the same and the Tx count (on FOpts) of the latest packet is greater than that in the second latest packet.

6.4.3 Handling Missing ACKs. There may be a few cases where the boosters may not receive any ACK from the gateway to relay to a node. This may happen when the gateway sends the ACK and the node misses it before any of the boosters come into action. In this case, we allow the node to increment FCntUp by one and retransmit the same packet. As per the LoRaWAN specification, the gateway sends a new ACK for an increase in the FCntUp field of the packet. The trade-offs of this technique include at most 7 wasted Tx attempts by the boosters of the packet with the old FCntUp.

6.5 LoRaIN Timing Requirements

In this section, we analyze the timing requirements for the constructive interference in LoRaIN. Specifically, we perform Matlab simulations to evaluate the maximum allowable temporal displacement (say, Δmax) between two LoRa transmitters (e.g., a node and a booster transmitting the same payload at the same frequency) such that they interfere constructively. In LoRaIN, even if a node and a booster transmit at the same time, their signals may reach the gateway with a temporal offset Δ due to the difference in their distances from the gateway. We use the Matlab LoRa simulator developed by the authors in [12]. In simulations, we analyze Δmax for different SFs and BWs while the CR is fixed at $\frac{25}{2}$ for a payload of 30 bytes from each transmitter with a Tx power of 14 dBm. We also add white Gaussian noise to the superimposed signal. The signals from both transmitters have the same amplitude, but one of them is delayed by a variable Δ with 10 ns granularity in the interval $[0, \frac{1}{8}]$, where $\frac{1}{8}$ is the chip duration in a LoRa symbol (e.g., for 125 kHz BW, 125000 chips/s or 8 ns/chip while the symbol is $2^7$ chips long).

In simulations, we analyze the behavior of Δmax for two cases: (1) only two transmitters are active and (2) an interferer is active as well with the two transmitters. For each pair of BW and SF in these cases, we run 100 simulations with different seeds for the noise.

Figure 10(a) shows the Δmax (averaged over 100 runs) for successful constructive interference (100% PRR) for BWs 125, 250, and 500 kHz with SFs 7–12 when only two transmitters are active in the same frequency. As shown in this figure, the Δmax for a given BW is almost constant regardless of the change in the SF. For 125 kHz BW, the average Δmax over SFs 7–12 is 6.85 μs with a standard deviation...
We use the Dragino LoRa Hat (SX1276 LoRa chip) on Raspberry Pi as the nodes that run on a Raspberry Pi and can simultaneously receive on 8 uplink channels [9]. We use the ChirpStack LoRaWAN network server that controls the gateway and runs locally on the gateway [3]. We use the Dragino LoRa Hat (SX1276 LoRa chip) on Raspberry Pi as the nodes [4, 6]. We also customize the LMIC 1.6 LoRaWAN development library and configure it with the parameters (e.g., frequency in 915MHz band, 125 uplink channels, 500kHz downlink channels, and maximum usable SF of 10) that are specific to our region [2, 5] and match the capability of our gateway (RAK2245 on a Raspberry Pi). In experiments, we choose between 5% and 15% of the nodes to act as the boosters while we vary the total number of nodes between 2 and 20. We calculate the actual number of boosters as \( F_{BN} \times M \), where \( F_{BN} \) is the percentage of nodes acting as boosters and \( M \) is the total nodes. For example, if 15 nodes are active in the network and we want 15% of them to act as boosters, then the actual number of boosters is \( \lceil 0.15 \times 15 \rceil = 3 \). We use this technique so that we may evaluate LoRaIN under varying number of nodes and depict easily. In each experiment, a node (including each booster) sends 100 confirmed packets with an inter-packet interval of 1 minute. Each packet contains a random payload of 30 bytes. Unless stated otherwise, these are our default parameter settings for all the experiments presented henceforth.

7.2 LoRaIN Carrier Activity Detection

In this section, we evaluate the performance of the activities related to LoRaIN CAD. Specifically, we look into the CAD detection accuracy, reception accuracy, and energy overhead at the boosters. CAD detection accuracy is defined as the ratio of the number of packets (including ACKs) whose preambles are detected and synchronized by the boosters to the total number of packets sent by the nodes or gateway. CAD reception accuracy is defined as the ratio of the number of correctly received packets (including ACKs) at the boosters to the total number of packets sent by the nodes or gateway. To calculate the energy overhead (in joule per bit unit) at the boosters, we take into account the total energy spent by the boosters in CAD detection, CAD reception, and one-shot forwarding of the packets (including ACKs) to the nodes or gateway.

7.2.1 CAD Detection Accuracy. Figure 11(a) shows the CAD detection accuracy of LoRaIN while the number of nodes and boosters are varied between 2 and 20 and between 5% and 15%, respectively. As shown in this figure, for 2 nodes and 5% boosters, the CAD detection accuracy is as high as 99%. Also, it increases with the increase in the number of boosters. For example, in the case of 20 nodes with 5%, 10%, and 15% boosters, the CAD detection accuracies are approximately 91%, 94.2%, and 97%, respectively. This experiment thus shows that the CAD detection accuracy is very high in LoRaIN, which confirms that boosters nodes are capable of detecting and synchronizing with almost all the packets in the network.

7.2.2 CAD Reception Accuracy. Figure 11(b) shows the CAD reception accuracy as we vary the number of nodes and boosters. For 2 nodes and 5% boosters, the CAD reception accuracy is 100%. Overall, as we increase the number of boosters, the CAD reception accuracy also increases. For example, in the case of 20 nodes with 5%, 10% and 15% boosters, the CAD reception accuracies are approximately 94%, 95%, and 97%, respectively. Such high CAD reception accuracy is very crucial in LoRaIN since constructive interferences are created by the boosters using these packets.

7.2.3 Energy Overhead at Boosters. Figure 11(c) depicts the energy consumption of the boosters in mJoule/bit unit when the number of nodes (between 2 and 20) and boosters (between 5% and 15%) is varied, which may be considered overhead if they are battery-powered. As shown in this figure, if we increase the number of boosters (for the same number of nodes), the average energy overhead tends to...
Boosting Reliability and Energy-Efficiency in Indoor LoRa

In this section, we evaluate the performance of the LoRaIN boosters in terms of creating successful constructive interferences at the gateway. Specifically, we calculate the PRR (packet reception rate) at the gateway. To recall, packet reception rate is the ratio of the number of packets received at the gateway to the total number of packets sent by the nodes.

**7.3.1 Results.** Figure 12 shows the PRR at the gateway as we vary the number of nodes from 2 to 20 and keep the number of boosters fixed at 15%. In this figure, we compare the performance of LoRaIN with LoRaWAN as well. In the case of 2 nodes, the PRR at the gateway is 100% in LoRaIN, compared to 83% in LoRaWAN. As we increase the number of nodes, the performance difference between LoRaIN and LoRaWAN becomes more prominent. Particularly, as the number of nodes increases, the PRR at the gateway in LoRaWAN goes down sharply, while it is still very high in LoRaIN. For example, in the case of 20 nodes, the PRR at the gateway is 95%, compared to only 62% in LoRaWAN. This experiments thus demonstrate that LoRaIN is much more reliable in indoor, compared to LoRaWAN.

**7.4 Experiments on Acknowledgment Relay**

In this section, we evaluate the performance of LoRaIN in terms of ACK relays from the boosters. We calculate the PDR and average number of Tx attempts per packet at the nodes. PDR is the ratio of the number of acknowledged packets to the number of total packets sent. If the ACK relays by the boosters work, the PDR should increase and the average number of Tx attempts should decrease at the nodes.

**7.4.1 Packet Delivery Ratio.** Figure 13(a) shows the PDR at the nodes as we vary the number of nodes between 2 and 20 and keep the boosters fixed at 15%. Additionally, we compare this performance with LoRaWAN. As shown in this figure, for all the cases,
the PDR in LoRaIN is very high, compared to those in LoRaWAN. For example, in the case of 20 nodes, the PDR in LoRaIN is approximately 90%, compared to only 13% in LoRaWAN. This experiment thus demonstrates that the ACK relays by the boosters are very effective in LoRaIN, which outperforms LoRaWAN significantly.

7.4.2 Transmission Attempts per Packet. Figure 13(b) shows the average number of Tx attempts per packet by the nodes in LoRaIN and also compares with LoRaWAN. In this experiment, the average number of Tx attempts per packet by the nodes almost stays the same in LoRaIN, compared to a noticeable increase in LoRaWAN, as we increase the number of nodes from 2 to 20 while keeping the number of boosters fixed at 15%. For example, in the case of 2 and 20 nodes, the average numbers of Tx attempts per packet in LoRaIN are 2.1 and 2.2, compared to 4.2 and 5.2 in LoRaWAN, respectively. In this experiment, LoRaIN performs more than twice as better as LoRaWAN, thereby showing the feasibility of its ACK relay.

7.5 Network Performance Analysis

In this section, we evaluate the network performance in terms of effective bitrate, average end-to-end (E2E) latency, and energy consumption. Effective bitrate is calculated based on the distinct packets received at the gateway. End-to-end latency per packet is defined as the time difference between the start of the first Tx attempt of the packet and end of its ACK reception. We calculate the average energy consumption per packet at the nodes considering the E2E latencies (energy/packet $\propto$ E2E/packet) of their packets.

7.5.1 Effective Bitrate. Figure 14(a) shows the effective bitrate at the gateway for all the packets sent from the nodes. In this experiment, we vary the number of nodes between 2 and 20 and compare the performances between LoRaIN and LoRaWAN. As shown in this figure, the effective bitrate at the gateway increases at a higher speed in LoRaIN, compared to the bitrate in LoRaWAN, as we increase the number of nodes and keep the number of boosters fixed at 15%. In the cases of 2 and 20 nodes, the effective bitrates at the gateway are 0.95kbps and 9.11kbps in LoRaIN, compared to 0.47kbps and 3.91kbps in LoRaWAN, respectively. Overall, such low bitrates at the LoRa gateways are due to the ADR feature of the LoRa nodes, which lets the nodes operate at higher SFs and on narrower channels. However, the bitrate at the gateway in LoRaIN is significantly higher than that in LoRaWAN. In the case of 20 nodes, LoRaIN has almost 3x higher bitrate than that in LoRaWAN.

7.5.2 E2E Latency and Energy Consumption. Figure 14(b) shows the average E2E latency per packet at the nodes in both LoRaIN and LoRaWAN. In general, the packets in LoRaIN observe much lower E2E latency than that in LoRaWAN for all the cases as we increase the number of nodes from 2 to 20 (boosters fixed at 15%). For 20 nodes, the average E2E latency per packet is approximately 4.5 seconds in LoRaIN, compared to 10 seconds in LoRaWAN. The ACK relays by the boosters play a vital role in the better performance of LoRaIN. Figure 14(c) shows the average energy consumption per packet at the nodes for both LoRaIN and LoRaWAN. The trend in the performances of LoRaIN and LoRaWAN in terms of average energy consumption per packet at the nodes is also similar to the performance trend in their average E2E latency per packet. For 20 nodes, the average energy consumption per packet is 250mJ, compared to 620mJ in LoRaWAN, thereby reducing the energy consumption per packet approximately 2.5x. Overall, all these experiments suggest that LoRaIN is much more suited than LoRaWAN in indoors.

8 CONCLUSION

In this paper, we have proposed a link-layer protocol called LoRaIN to boost the reliability at the gateway and energy-efficiency at the end-devices of indoor LoRa networks. To do this, we first extensively evaluated the performance of the LoRa MAC protocol – LoRaWAN – indoor and showed that the reliability at the gateway was as low as 62%. Also, the number of retransmissions per packet was as high as 4 to 5. In LoRaIN, we boosted the reliability and energy-efficiency in the network by creating constructive interference of the packets at the gateway with specific timing requirements and relaying missing acknowledgments to the end-devices. Our extensive experiments using LoRaIN showed that the reliability at the gateway in indoor LoRa networks increased from 62% to 95% while the end-devices operated 2.5x efficiently in terms of energy, thereby demonstrating the feasibility of LoRaIN.

ACKNOWLEDGEMENTS

This work was supported by NSF through grants CNS-2301757, CAREER- 2306486, CCF-2118202, CNS-2306745, and by ONR through grant N00014-23-1-2151.


