# A Cloud-Optimized Link Layer for Low-Power Wide-Area Networks

Artur Balanuta
Carnegie Mellon University
Instituto Superior Técnico
artur@cmu.edu

Nuno Pereira
Carnegie Mellon University
npereira@cmu.edu

Swarun Kumar
Carnegie Mellon University
swarun@cmu.edu

Anthony Rowe
Carnegie Mellon University
agr@ece.cmu.edu

## ABSTRACT

Conventional wireless communication systems are typically designed assuming a single transmitter-receiver pair for each link. In Low-Power Wide-Area Networks (LP-WANs), this one-to-one design paradigm is often overly pessimistic in terms of link budget because client packets are frequently detected by multiple gateways (i.e. one-to-many). Prior work has shown massive improvement in performance when specialized hardware is used to coherently combine signals at the physical layer.

In this paper, we explore the potential of using multiple receivers at the MAC and link layer where these performance gains are often neglected. We present an approach called Opportunistic Packet Recovery (OPR) that targets the most likely corrupt bits across a set of packets that suffered failed CRCs at multiple LoRa LP-WAN basestations. We see that bit errors are often disjoint across receivers, which aids in collaborative error detection. OPR leverages this to provide increasing gain in error recovery as a function of the number of receiving gateways. Since LP-WAN networks can easily offload packet processing to the cloud, there is ample compute time per packet (order of seconds) to search for bit permutations that would restore packet integrity. Link layer corrections have the advantage of being immediately applicable to the millions of already deployed LP-WAN systems without additional hardware or expensive RF front-ends. We experimentally demonstrate that OPR can correct up to 72% of packets that would normally have failed, when they are captured by multiple gateways.

## CCS CONCEPTS

• **Networks** → **Sensor networks**.

## KEYWORDS

Low-Power Wide-Area Network (LPWAN), Cloud Computing, Interference Mitigation, Co-existence


*MobiSys '20, June 15–19, 2020, Toronto, ON, Canada*

ACM ISBN 978-1-4503-7954-0/20/06…$15.00
https://doi.org/10.1145/3386901.3388915

**ACM Reference Format:**
Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. 2020. A Cloud-Optimized Link Layer for Low-Power Wide-Area Networks. In *The 18th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '20), June 15–19, 2020, Toronto, ON, Canada.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3386901.3388915

## 1 INTRODUCTION

Low-Power Wide-Area Networks (LP-WANs) are increasingly seen as a vital wireless connectivity framework for low data-rate Internet of Things applications. LP-WANs offer large areas of coverage (several kilometers from the base station), long battery lives (up to ten years), and low cost (a few dollars per radio). Among the most popular LP-WAN technologies is LoRa, which uses unlicensed sub-GHz bands (915 MHz in the United States). LoRa is soon to be integrated with Comcast set-top boxes [31] to enable smart-home applications, creating a proliferation of both macro- and femto-base stations. While we are seeing this trend of increasing gateway density, it becomes inevitable that multiple networks will become co-located, causing interference in unlicensed spectrum. This co-existence problem is compounded by the multitude of recent Low-Power wireless standards – 802.11ah (HaLow), LoRa SF5, 802.15.4g (Wi-Sun) – that share the same sub-GHz unlicensed bands in indoor smart-home deployments.

While there is rich recent literature on tackling interference in LoRa [9, 12, 16, 41], much of these have focused on solutions at the physical and MAC layers. These include clean-slate solutions to re-engineer or synchronize client transmissions [16, 41] or advanced collision detection at software-radio base stations [12, 41], all of which would require re-designing LoRa clients and/or base stations. Our efforts in this paper differs in two important ways: (1) First, we expand interference to include cross-technology collisions from indoor users of unlicensed bands, with whom LoRa shares spectrum. These impact the LoRa performance at the link layer. (2) Second, we seek solutions that are compatible with existing commercial LoRa base stations and clients.

This paper presents OPR (Opportunistic Packet Recovery), the first software-only solution to recover packets by pooling information from multiple gateways at the link layer that would otherwise be lost due to short-lived interference to LoRa. OPR works by disabling the packet rejection that normally takes place at LoRa gateways when packets are subject to CRC and/or FEC errors. Instead of being discarded, these corrupt packets are collected by a network service that groups them based on geographic proximity

and reception time. One of our key insights was that even though these packets might fail integrity checks, they often fail in a disjoint manner due to the spatial diversity in the receivers. In the case of two corrupt packets, it is possible to XOR their bit patterns to target the most likely candidate locations of bit errors. When three or more gateways receive a corrupt packet, it is possible to not only detect potential error locations, but we can employ majority voting to estimate the most likely bit combination. Once OPR has generated a candidate set of corrupt bits, we use the ample processing available in the Cloud to search through all possible bit combinations that yield a valid CRC. This typically results in a small number of packets which can then be passed to the LoRaWAN application layer that performs a final filtering step when decrypting the packet in order to verify its Message Integrity Check (MIC).

OPR pushes link layer management from the low-power device to the much more capable gateways and Cloud infrastructure, which can improve client battery-life by removing the need for costly retransmissions and potentially allowing the system to operate at higher spreading factors or with less Forward Error Correction. Our approach is fully compatible with current commercial LoRa client and base station hardware. It explicitly accounts for interference in the unlicensed sub-GHz spectrum from indoor radio sources, including from non-LoRa technologies. We implement and evaluate OPR on a large-scale university-led testbed located on Carnegie Mellon University's campus that uses commercial base stations and on a client testbed spanning ten square kilometers. While some concepts apply to both upstream and downstream traffic, we focus on upstream packets, which represent the majority of LP-WAN traffic where you have redundancy across gateways and accessibility to Cloud compute resources.

The rest of this paper addresses two key challenges in designing OPR: (1) First, how do we detect the segment of a LoRa packet that has experienced short-lived interference?; and (2) Second, how do we correct the erroneous segment without increasing latency or causing client battery drain?

**Error Detection:** The traditional approach to addressing the first challenge of detecting erroneous bit segments is to adaptively code transmissions with forward error correcting (FEC) codes to be robust to a small number of bit flips. However, doing so increases the duration of packets, the resulting client battery drain, and client complexity [6, 23]. Ideally, such codes could be greatly simplified or even avoided if the precise locations of the bit flips were known. In the case of LoRa, FEC is achieved using a limited set of block Hamming Codes which are known to be sub-optimal [22], a design decision that trades off performance to ensure short packet duration and therefore less client battery drain. To address this problem, OPR leverages the spatial diversity of multiple gateways, which has the benefit of being additive given the number of gateways with no impact on the duration of the packet itself. With a critical mass of gateways, we find it is potentially more efficient to entirely disable FEC. In the case where the spatial diversity is not significant (i.e. a single receiving gateway), we leverage a unique side-channel that is available with commercial LoRa hardware. Specifically, LoRa devices report the Received Signal Strength Indicator (RSSI) levels throughout the duration of a packet at a sample-granularity. Contrast this with other radio technologies (e.g. Wi-Fi) whose commodity chips only expose one RSSI value per-packet to higher layers. OPR develops a classifier that can both combine information from multiple gateways and also track RSSI values through a packet and detect bursts that resemble interference.

**Error Correction:** Having detected the location of the bit errors, OPR seeks to actively correct them. Importantly, it aims to do this without proactively seeking re-transmissions from the clients or additional coded bits, which would introduce battery and latency overhead. Our approach is to reuse the structure of the LoRa packet at the MAC and application layer. Specifically, every LoRa packet has an existing Cyclic Redundancy Check (CRC), static MAC layer fields, and a Message Integrity Check (MIC) that are well-defined functions, traditionally designed to detect bit errors. OPR instead seeks to reuse these error-detection and integrity fields as error-correcting codes. Intuitively, we cycle through different possibilities of the bits within the damaged portion of the received message and identify which sequence(s) of bits match in the error-detection fields. We further fine-tune our system to leverage spatial diversity, combining observations of the same message across multiple gateways. We benefit from the fact that statistically, different gateways are likely to see bit errors in different segments of their message, depending on the locations of the gateways relative to interfering sources. Sec. 5 elaborates on the approach and performance limits of our MAC-layer solution to packet correction without soliciting re-transmissions.

We evaluate Opportunistic Packet Recovery (OPR) in both indoor and outdoor environments using two test-beds within and around Carnegie Mellon University in the heart of Pittsburgh, PA. Four rooftop gateways support the deployment of Long-Range (LoRa) Low-Power Wide-Area Network (LP-WAN) which services a large 10 sq. km area. Indoor test-beds with up to six LoRa gateways were used for proof-of-concept and micro benchmarks. Our results reveal the following:

- We see that real-world LoRaWAN gateways suffer from as much as 57% packet-loss.
- The vast majority of packet drops (>90%) are caused by CRC failures that result in different corrupt packets across a set of nearby gateways.
- RSSI detection alone can correctly identify 83% of the bits that were corrupted with 17% false positives.
- We are able to correct up to 30 bits worth of corruption in the 1-second required to reply to a LoRaWAN ACK message using commodity hardware.
- In bursty interference micro-benchmarks, we correct as many as 72% of CRC errors that normally would be dropped (when received by multiple gateways) while never generating a false packet.

**Contributions:** We develop to our knowledge the first software-only solution to mitigate inter- and intra-technology interference to LoRa from short-lived transmissions compatible with existing LoRa base stations and client hardware. Our specific contributions include:

- An approach to detect interference on commodity LoRa hardware by using a combination of multiple gateways and sample-level RSSI measurements as a side-channel.

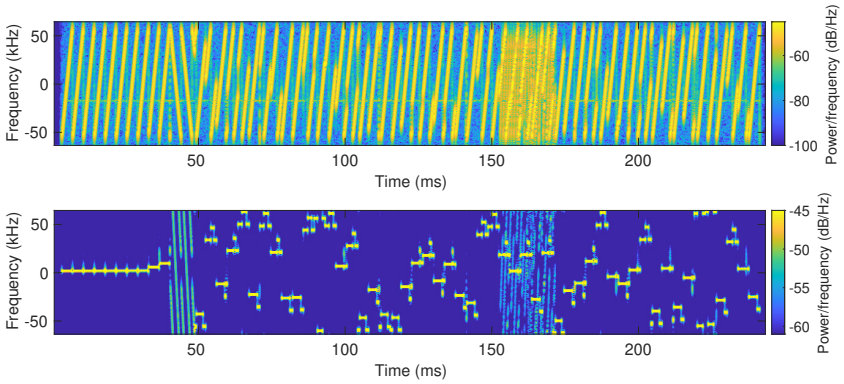


**Figure 1: in Phase/in Quadrature (I/Q) of a LoRa packet with Collision captured with a software defined Radio (Top). Convolution of the same signal with a down-chirp, an intermediate step in demodulating LoRa packets (Bottom)**

- A bit-level consensus approach for detecting and correcting bits across LoRa gateways.
- A system design to recover bit errors caused by short-lived interference to LoRa by leveraging the message integrity fields of the LoRa MAC as an error correcting code.
- A detailed deployment study of the impact of interference in wide-area LoRa deployments and the effectiveness of our approach on a 10 sq. km. test-bed in Pittsburgh, PA.

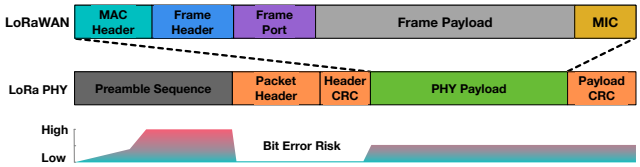


**Figure 2: LoRa PHY and LoRaWAN frame components. Packet Header and Header CRC are encoded with more robust 1/2 FEC coding to protect against interference**

## 2 BACKGROUND

**LoRa PHY:** The LoRa physical layer is based on Chirp spread spectrum (CSS) modulation. As shown in Figure 1, a packet is encoded into chirp signals that are composed of linear frequency sweeps, where each sweep has a symbol duration and encodes multiple data bits as a function of the spreading factor – a quantity that dictates data rate. For example, at spreading factor $N$, each chirp encodes N bits into one of $2^N$ possible uniformly separated initial frequency offsets. A higher spreading factor, e.g. $N+1$, improves resilience to noise by encoding one more bit per chirp in double the transmission time, effectively halving the data rate.

Figure 1 shows an I/Q capture of a LoRa packet with a collision at around $t + 160ms$. To demodulate the signal, it is first convoluted with a down-chirp resulting in symbols shown in the bottom figure. These are then mapped to one of the frequency offset bin indexes for that particular Spreading Factor (SF) by selecting the bin with the strongest signal. The bottom part of Figure 1 illustrates the robustness of LoRa. The LoRa signal is integrated into coherent signal impulses located at a particular frequency offset, while the interference is more evenly spread across the bandwidth. Even so, strong interference or very low Signal-to-Noise Ratio (SNR) can jeopardize the proper mapping of symbols to their appropriate bins.

LoRa modulation also adds Forward Error Correction (FEC), which encodes 4-bit data into 5- to 8-bit redundancies that are spread across the packet. Higher Coding Rate (CR) values provide greater interference protection, at the cost of increased packet airtime. LoRa encoding is proprietary, but reverse engineering efforts have shown it to include gray indexing, whitening, and interleaving steps [19]. Thus it is hard to map corrupted symbols to the resulting bit patterns (although this can be used to improve future versions of OPR).

**LoRaWAN:** LoRa's Wide-Area Networking architecture (LoRaWAN) is composed of end devices that are connected with a single hop to one or more Gateway(s) which, in turn, forward packets to the Network Server (NS) through a backhaul network using a pub/sub bus over IP protocols. The NS performs end-device address checking, frame security checks (authenticity and nonces), acknowledgments, and MAC-layer requests and forwards application payloads to and from Application Servers (AS). The most recent version of LoRaWAN introduced a roaming architecture where a NS can have different roles, depending on the type of roaming involved. Join Servers (JS) handle requests to join the network from end devices which are forwarded by NSs, and finally, the AS handles the application layer payloads being forward to/from the NS.

This brief refresher of LoRaWAN's architecture is representative of common LP-WAN architectures and, importantly, highlights some features that we capitalize on: (i) transmissions from end devices are received by all gateways in range, and (ii) a significant amount of network operations are performed at the network backend, which is connected by high-performance network connections (wired Ethernet, wireless cellular) and uses, or has access to, Cloud computing resources (often, service providers deploy NS, JS and AP in the Cloud). Traditional wireless networks have many of these critical characteristics used by OPR.

**Frame Structure:** All LoRa messages start with a preamble, followed by a (Physical (PHY) layer) packet header, a Cyclic Redundancy Checks (CRC) of the header, and a payload, which includes

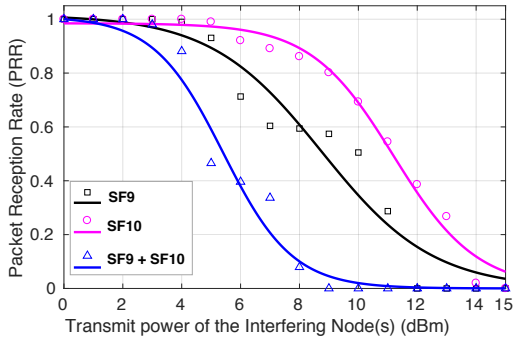


**Figure 3: Packet Reception Rate (PRR) of packets sent at SF8 when overpowered by another transmission(s) on the same channel. We can also observe the additive interference caused by two simultaneous transmissions (SF9 and SF10).**

a Medium Access Control (MAC) header, application payload, and a message integrity code as depicted in Figure 2. LoRa uplink messages also include a CRC of the entire PHY payload.

It is important to note that the security features of LoRaWAN impose restrictions on the design of OPR. LoRaWAN defines two device-specific root keys (*NtwKey* and *AppKey*) from which several session keys are derived. Importantly, the *AppKey* is used to derive the application session *AppSKey*, which is used to encrypt application payloads using Advanced Encryption Standard (AES). To not break the security of LoRaWAN, the access to these keys should be kept separate, and this means that OPR cannot exploit features in the data payload to correct packets.

**Motivating Experiments:** We designed three preliminary experiments to further motivate our work. First, we analyzed the effect of inter-technology interference which is expected to worsen as LP-WAN deployments increase. Next, we used a deployment on a city-scale test-bed to validate that a large number of received packets in LP-WAN networks are corrupt. Finally, we studied the impact of corruption on different regions within a packet. Across experiments, we observed that most corruptions are small comparatively to the total packet length (89% of the corrupted packets have less than 15 error bits as shown in Figure 9), which suggests that attempting to recover corrupt packets might be a viable path to noticeably improve the reliability of LP-WAN.

**Imperfect Orthogonality:** It has been shown that LoRa spreading factors are not perfectly orthogonal [32], and, particularly under near-far conditions, overlapping transmissions using different SFs can interfere with each other. Hence, even interference between LoRa transmitters is an important motivating factor as networks scale. Our first micro-benchmark shows the interaction between different spreading factors as described by an additive interference model. Packets with varying SF configurations were sent out synchronously to force packet collisions and interference.

Using attenuators, we calibrate a set of clients in order to simulate a real-world long-distance collision with the equivalent single strength. Figure 3 shows the packet reception rate of a node using spreading factor 8 (transmitting at 0 dBm) when sharing the same frequency with other node(s), using spreading factor 9, 10, or

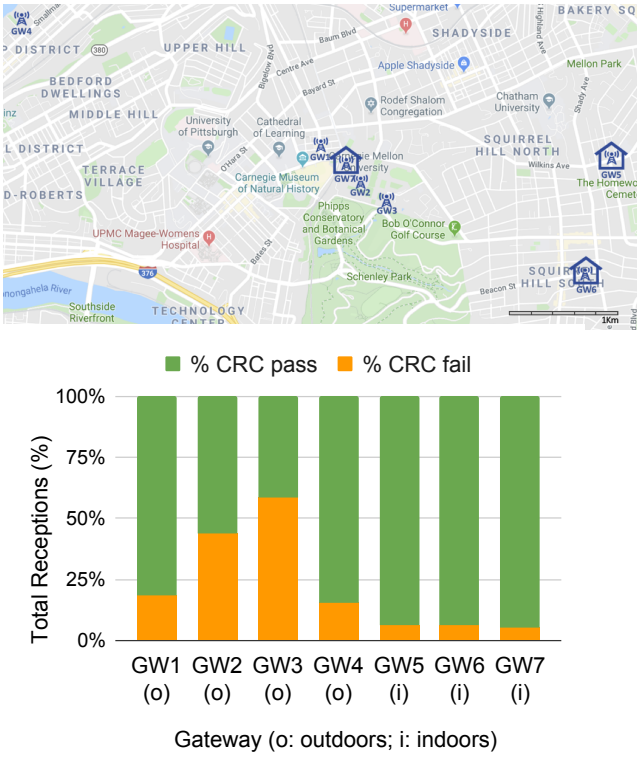


**Figure 4: Amount of failed CRC receptions across a city-scale 7 Gateways deployment. Gateways located outdoors experience a high number of receptions that fail CRC checks.**

both with increasing transmit power. Each data point was derived from 2000 samples that fit the inverse sigmoid function. We can observe the additive interference and capture effect caused by the two transmitters which confirm the values presented in [34].

**Corrupt Packets in LP-WAN Deployments:** In our next experiment, we used a real-world, 10 sq. km. test-bed in Pittsburgh, PA, to collect over 60M packets received at 7 LoRa gateways (4 of these gateways were outdoors and 3 indoors). Figure 4 shows the percentage of packets that pass or fail CRC checks across the 7 gateways. We can observe that packet corruptions vary greatly across gateways and, most importantly, outdoor gateways see a high number of corrupt packets. The gateways with the highest Line-of-Sight (LoS) coverage also show the most errors, with one gateway receiving over 58% corrupt packets. Multiple factors can affect packet loss, including packet duration, network load, the duration and distance of interference sources to the receiver, and the modulation used. To our understanding, the high number of corrupt packets in outdoor gateways is related to the coverage area of each gateway, since the outdoor test-bed is deployed on some of the tallest buildings around the city with optimal LoS coverage. We also note that these results are consistent with previous reports from other real deployments [26].

**Payload Vulnerability:** Figure 2 depicts the impact of errors associated with a specific location in packet. The preamble is particularly susceptible to interference, as enough symbols need to be correctly demodulated so that the packet is identified. Corrupt or

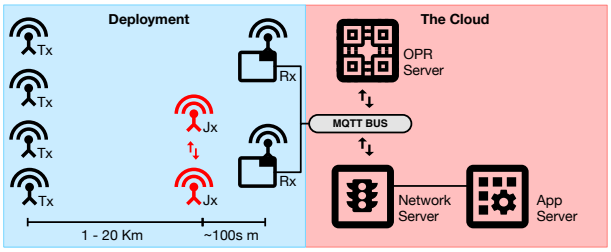


**Figure 5: OPR System Architecture: Valid and corrupted packets received at the Gateways (Rx) are tagged with metadata and sent over the MQTT BUS. OPR Server snoops on the bus and inserts the corrected framed.**

weak preambles likely account for the majority of dropped packets that do not trigger CRC failures. The LoRa PHY packet headers are always encoded with 1/2 FEC since it contains critical information about the rest of the packet, such as packet length and coding rate. An error in the header can not only make the payload undecodable but can also impact the resources of the receiver (such as battery consumption or decoding units on a GW). The PHY packet payload is usually encoded with a lower CR (the LoRa's Wide-Area Networking Protocol (LoRaWAN) standard defines the default $CR = 4/5$) to reduce packet airtime and demonstrates a fairly uniform bit error distribution as discussed in Section 7 and shown by Figure 10.

## 3 SYSTEM OVERVIEW

OPR is a software solution that recovers valid payloads that would normally be lost due to interference. It is most applicable to losses from strong interference or weak signal strength that are often the case in highly dense areas. Correcting these errors could allow LoRa clients to transmit at higher data-rates and/or at lower powers to preserve battery.

OPR can be easily integrated into common LP-WAN architectures. It employs a combination of packet reception redundancy and inter-packet signal strength metering, which are both unique characteristics of LP-WAN architectures. In this section, we provide an overview of OPR's system architecture, error detection, recovery techniques, and modes of operation.

**System Architecture:** Figure 5 depicts OPR's architecture, where user- and service provider-deployed gateways can be located indoors or outdoors. Instead of discarding corrupt packets (the default gateway configuration), OPR forwards invalid packets to the Cloud, which, for example, is supported on common LP-WAN gateway software by changing a configuration variable [7].

OPR also snoops on the traffic between the gateways and the LoRaWAN network server to collect metadata and sort the payloads, which are then post-processed along with the received byte-streams when possible. In other words, OPR opportunistically performs link-layer bit error correction at the Cloud, simultaneously improving battery life and range of low-power clients at the expense of speculative computation at the Cloud.

OPR's data collection transmits corrupt packets (typical LoRaWAN messages are 450 bytes long) along with the respective Received Signal Strength Indicator (RSSI) samples and metadata used to correlate receptions. Each of these packets are about 200 bytes after RSSI samples are down-sampled at the gateway. This is a small increase for the backend network, which is usually a wired Ethernet or high data-rate wireless cellular data link.

**Error Detection and Modes of Operation:** OPR has three primary modes of operation, depending on how many unique packets are received by surrounding gateways. (i) OPR needs to decide if post-processing is necessary at all or if a valid packet was detected. This is done by pairing corrupt receptions with timing information and metadata such as the gateway's geographical location. (ii) When OPR receives only one corrupt packet (single detection), it attempts to classify regions of the packet that might have been subject to interference using RSSI traces. (iii) When multiple corrupt copies of the same transmission are received at different gateways (multiple receptions), OPR compares the different receptions in order to generate a candidate error vector. In concept, RSSI traces could also be used in conjunction with the vectors to select the most probable error regions. Section 4 provides more details on these techniques.

**Error Recovery:** Error recovery (detailed in Section 5) is accomplished by: (i) exploiting the MAC frame structure, (ii) using the CRC and Message Integrity Check (MIC) fields as error correcting codes, (iii) majority voting across receptions from multiple gateways, and finally, (iv) cycle through possible combinations by expanding the search space, starting from the the bits deemed more likely to be corrupted.

## 4 ERROR DETECTION

Before recovery, corrupt packets need to be processed in order to detect which sections of the packet are corrupt. The goal is to output a Bit Corruption Likelihood (BCL) vector the same size as the number of bits in the packet. The vector is initialized with zeros (meaning correct reception of the bit) and can go up to 100 (meaning the bit was certainly corrupted during reception). This allows OPR to give varying degrees of confidence, depending on the method used. OPR performs different steps at each single corrupt packet, and when available, across multiple receptions.

### 4.1 Single Reception

We first consider the case of detecting erroneous bit locations from a single reception at one gateway. Figure 2 (presented earlier in Section 2) depicts the LoRaPHY and LoRaWAN frame structure. Observe that multiple unencrypted headers are kept mostly unchanged for all Uplink messages: the MAC and frame header (MHDR, FHDR) each contain an 8-bit field that is the same for almost all Uplink transmissions. OPR verifies these fields, obtaining clues about bits that might have been modified. It then uses the provided CRC to validate its hypotheses. As discussed in Section 6 (see Figure 9 and related discussion), the majority of the corrupted packets show only a small number of error bits. Thus, the probability that error bits occur in these common headers is very small and inversely proportional to the packet size.

**Metadata Collection and Gateway Synchronization:** OPR collects reception properties, such as the $SF$ and $CR$ used, gateway

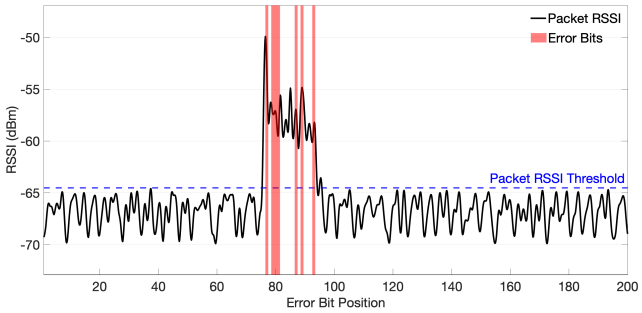


**Figure 6: RSSI sampling of a LoRa packet (SF7, 25 Bytes) reception provides clues to the location of Error Bits in the Packet**

location, and reception time. Using a combination of these data parameters, it is possible to uniquely identify packets and their geographical location. The precise start time of packets across gateways is valuable information for identifying which received signals across gateways correspond to the same packet, and for which common software time-synchronization protocols (e.g. Network Time Protocol (NTP)) are sufficient, as their accuracy is at a much finer granularity than the minimum airtime for a LoRaWAN packet (around 20 ms).

**Temporal Periodicity and Re-Transmission Combining:** Packets contain an unencrypted 4-byte device address and 2-byte frame counter. Since most nodes are statically deployed and have a fairly regular Uplink schedule, we can take advantage of the collected metadata to limit the scope of the search for possible devices and auto-correlate the addresses and counters with the fields of the damaged messages to identify errors in the LoRaWAN header. Due to highly noisy spectrum, it is possible that consequent re-transmissions from the same transmitter would be impacted. Knowing that re-transmissions occur after a fixed pre-established interval we can compare the both messages and identify the differences across the whole packet [11].

**RSSI Level Sampling:** Taking advantage of the frame structure is just an initial step. To detect errors inside the encrypted payload, OPR leverages a unique side-channel available in commercial LoRa hardware. Specifically, LoRa devices have an addressable register that returns real-time RSSI levels throughout the duration of a packet reception at a sample-granularity. This feature is used by monitoring the receive process using General Purpose Input/Output (GPIO) interrupt signals (*PreambleDetected*, *Synchronized*, *Receiving* and *RXDone*) which drive the collection of samples into a circular buffer up to hundreds times per bit. The samples are then down-sampled, compressed, and sent with the rest of the packet metadata. This contributes an additional 200 bytes for a 25-byte payload to the total message that is sent on the pub/sub bus, which only has to be transmitted with corrupt packets. While the data can be compressed even further, we leave this optimization for future work.

Figure 6 illustrates the process of using RSSI values to detect bit errors under interference. It shows a LoRa packet transmitted with SF7 that has suffered from interference from a similar LoRa packet using SF5, with 6dB higher power. This is a typical scenario where short bursts of interference are caused by coexisting wireless technology or networks operating in the same spectrum. If we align the received payload with the collected RSSI samples, we can observe that bit error locations are directly related to the additive interference caused by the collision. Particularly, a stronger interference has a higher chance of causing error bits. This is an important clue that OPR uses to hypothesize the location of the error bits, giving higher priority to the regions with the strongest interference.

**Maximizing Detections in Single Reception:** OPR can apply multiple bit error detection mechanisms at the same time. For example, if RSSI thresholds define a set of bits and consensus at the gateway defines another set of bits, we can XOR the two sets of hints together. In practice, one is typically most concerned about prioritizing the most likely bits that capture the true errors. In general, gateway consensus should take priority over RSSI-based hints. We use the notion of a BCL to capture the likeliness of a bit corruption for cases when the search mechanism must prioritize and cannot search the entire space. This could be because the server is heavily loaded or because network traffic delays consumed time that is normally spent on the search process. Ideally, the BCL can provide a sort of Just-In-Time (JIT) search strategy. Section 5 will demonstrate how OPR uses the BCL vector to recover the original payload.

## 4.2 Multiple Receptions

In an urban, densely populated area, it is likely that the same Uplink message will be detected by more than one gateway. While this naturally increases reliability, due to strong interference or low SNR, we still observe error bits that are often different across packets due to different interference sources and variations in multi-path propagation. Metadata filtering is used by OPR to identify damaged packets across multiple gateways. The key observation here is that transmitted signals propagate via distinct radio paths, and are affected by distinct reflections, fading, and interference before reaching each receiving gateway, translating into different error bits at each receiver.

**Independent Local Interference:** Consider an example of two receptions (at two different gateways) of the same packet, where one packet has errors in the header, and the other in the CRC. Both payloads are combined by OPR to determine the location of the errors by performing a bit-wise XOR across the packets. Having two copies of the same packet allows OPR to identify the impacted locations, and more copies allow for additional error correction, by using the majority of the affected bits [35]. The consistency of the CRC can help disambiguate ties where the majority rule fails.

We observe that even when the multiple receptions are subject to similar interference, packets are not affected in the same way. This observation is supported by an experimental setup described in Section 6.

**Common Source of Interference:** Table 1 shows the byte-stream representation of a real LoRa payload (TX) in hexadecimal format and receptions from three different receivers (RX1-3) that have been subject to the same interference source. Modified bits are highlighted. We can observe that most of the error bits are exclusive,

| Node | Payload Hex | Bit Errors |
|---|---|---|
| TX | 00000000000000000000000000000000 | - |
| RX1 | 00000000000020080000000000000000 | 2 |
| RX2 | 00000000020014060000000000000010 | 6 |
| RX3 | 00000000010020040000000000000000 | 3 |

**Table 1: Simplified byte-stream representation of the sent (TX) and received (RX) payload by three independent receivers that has suffered a collision**

**Algorithm 1** Bit Errors Detection

```
[pkts, payload_len] ← collect(data_and_meta)
pkt_errors ← zeros(payload_len)                ▷ init vector
for [a, b] in combnk(pkts, 2) do               ▷ all pair comb
    pkt_errors ← OR(pkt_errors, XOR(a, b))
end for
```

indicating that the same interference source will not cause the same behavior at each receiver. OPR uses this concept (Algorithm 1) to precisely identify the unique location of the error bits. One could also imagine expanding the search to include nearby bits in the BCL if additional compute resources are available (i.e. the BCL is below 30 bits).

The detection of the error bits only works if there is a difference in the byte-streams. For example, when looking at the third column of error bits (8, 6, 4) in binary representation (1000, 0110, 0100), it shows that RX3 bit error is a subset of RX2. Thus, if only RX2 and RX3 would have been received, the error bit would not be visible to OPR. Conveniently, the RX1 bit has not been affected and can be used to detect the error bit.

**Maximizing Detection with Spacial Diversity:** OPR can also take advantage of multiple bit error detection mechanisms from multiple receivers. Contrary to RSSI levels sampling, the spatial diversity of the gateways can precisely identify most of the affected bits. These are set in the maximum value in the BCL to give them priority during recovery. Likewise, obtained RSSI values are first averaged and added to the BCL.

## 5 ERROR RECOVERY

To recover errors, we cycle through possible combinations by expanding the search space, starting from the bits deemed more likely to be corrupted as given by the BCL. The performance of OPR is directly tied to the compute resources available, and, at the initialization phase, OPR profiles the compute resources available to define an upper bound on the number of error bits it will cycle through. It then searches though possible combinations that generate a valid CRC and further discards invalid patterns based on the MIC. The remainder of this section details OPR's initialization and aspects of this search.

**Search Mask Generation and Validation:** During OPR initialization, the error recovery module checks the hardware capabilities by using an internal benchmark. The benchmark is based on a CRC lookup table calculation to estimate how many bits can be

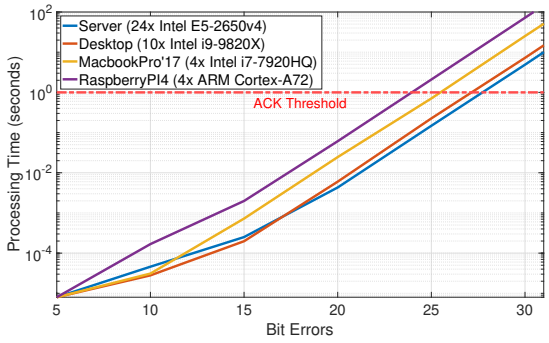


**Figure 7: Amount of time necessary to compute all hashes given the number of error bits and a 25 bytes payload**

searched given the LoRa ACK window time and estimates of network overhead. Figure 7 shows benchmarks of the capacity limits (CL) of four different device classes, ranging from low power to commercial server hardware. This establishes the upper bound of error bit combinations to be used on that particular platform.

When a bit error is detected by the error detection module, it forwards the BCL vector and the corresponding corrupted byte-stream (or the first received copy) in a multi-GW reception. The error recovery module only attempts the recovery of the transmission if the number of confirmed error bits (the ones marked as 100% Corruption Likelihood) is lower than the one supported by the hardware. A bitmask is generated by distributing the available CL bits across the hints provided by the BCL vector in decreasing order. Based on packets analyzed in our test-beds, we assume that interference has a bursty nature. Thus, we increasingly expand the search scope around these fields up to the maximum number of bits our system can check given LoRaWAN ACK time constrains.

In practice, CRC calculations should be completed within 0.8s to allow for network latencies so that packets are acknowledged within LoRaWAN's 1s acknowledge window. These constraints are only applied to packets that require acknowledgments; if idle, OPR will attempt to recover transmissions that are above the hardware limits up until a packet that requires processing arrives. In order to simplify the calculations, for the rest of the paper we assume an upper limit of 30 CL error bits unless mentioned otherwise.

**Hash Collisions:** LoRaWAN Frames contain a 16 bit CRC known as CRC-16/CCITT [17] and a 32 bit MIC. Assuming random input and an ideal CRC function, the hash will be one of $2^{16}$ different values. Given two random messages, the probability of their CRC being the same (probability of collision) is given by $1/2^{16}$ [38]. Thus, given a pool of $N$ unique messages, the probability of them having the same CRC is given by equation (1).

$$\frac{(N-1)}{2^{16}} \tag{1}$$

CRC-16/KERMIT is not a perfectly unbiased checksum algorithm, but it is a good approximation. Thus, by using the above equation we can estimate the probability of false positives generated by the hashing function. For example, assuming the 30 bit upper limit, we can demonstrate that the average number of generated messages

with the same CRC is given by $(2^{30} - 1)/2^{16} \approx 2^{14} = 16384$. In this setup, OPR will speculatively generate up to 16384 messages (based on the search generated bitmask) that match the same CRC and forward them to the LoRaWAN application server. When the speculative OPR packets are received by the LoRaWAN Application Servers, they are checked against their MIC and other MAC header fields such as the frame counter. The probability of OPR generating a message that checks the CRC and MIC is thus given by (2), where $M$ is the number of error bits in the message.

$$\frac{(2^M - 1)}{2^{16}} \times \frac{1}{2^{32}} \tag{2}$$

Considering the upper limit of 30 error bits, the approximate probability of a false positive is approximately $3.8 \times 10^{-6}$.

**Optimizing CRC Checks:** OPR makes heavy use of CRC validations in order to generate candidate packets based on the error patterns. The performance metrics shown in Figure 7 are the uppermost bounds of the time it takes to perform all the calculations of all possible error patterns. On average, it is expected that the true error pattern will be generated somewhere around half of this upper bound. Due to the linear nature of CRC codes and the bursty patterns of interference, recomputing the CRC over the entire packet is redundant [11]. Large speedups can be achieved by using dynamic programming and incremental CRC algorithms. The speedup depends on bit error location, and will perform best when the bit errors are clustered at the end of the packet.

**Traffic Amplification:** OPR has the potential to amplify the traffic in a LoRaWAN system that would normally be going between the main backend server and the LoRa applications. This is unfortunately required since candidate packets that satisfy the CRC can only be validated by end applications which have the required encryption keys to validate the MIC. A single packet reception with 30 bits flagged as corrupt could in the worst case generate up to $2^{14}$ messages that pass the CRC check. While extremely unlikely, a 25-byte transmission could be amplified to the point where it generates 410 kilobytes of traffic that the LoRaWAN application must filter. These are server-to-server interactions that are comparatively fast. Since the worst-case required bandwidth scales exponentially with the number of error bits, this quickly becomes an additional bottleneck (beyond just compute) for the number of bits that can be corrected. This amplification could be avoided in certain high-performance settings if LoRa applications were willing to provide the LoRaWAN server access to their encrypted data.

## 6 EXPERIMENTAL EVALUATION

We evaluate the effectiveness of OPR on a 10 sq. km. test-bed in a major U.S. city (shown in Figure 4) in addition to proof-of-concept experiments across our university campus network and inside a large office building as depicted in Figure 8.

Unless otherwise specified, to isolate OPR's performance we have disabled the standard 4/5 FEC codes enabled for LoRaWAN payloads. Forward error correction would increase the robustness of packets; however, they would still suffer similar bit errors that OPR will detect and correct. We present a more detailed discussion of the trade-off and potential synergy between OPR and FEC in Sec. 7. The

**Figure 8: Experimental LoRaWAN test-bed layout: there are 6 gateways in total, spread across an indoor office environment. Each red dot indicates a LoRaWAN receiver.**

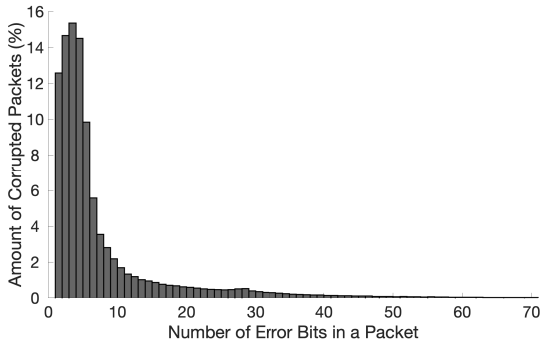


**Figure 9: Amount of bit errors from corrupted packets in an urban deployment experiment (1.2 million 40-byte payload LoRa transmissions)**

LoRaWAN does not take advantage of the spatial diversity arising from multiple receivers and it's requirement of 4/5 coding is a conservative approach that impacts the battery life of all devices, even the ones that might do not require it.

**Number of Bit Errors:** Using our campus test-bed, we characterize the number of bit errors experienced due to interference captured by logging over 1.2 million packets over 10 days. Figure 9 plots the histogram of bit error rates from our deployment. Our key observation is that 89% of the corrupted packets have less than 15 error bits – a mere 3.6% of the total packet length on average. In other words, packet bit corruptions are often relatively small compared to packet length. We observe that these data suffer from survivor bias, as these are the packets that were not subject to enough damage to make them completely undetectable. In hindsight, it is not unsurprising that most bit corruptions are small. Much of LoRa interference is caused by bursty higher-speed transmissions from indoor (and often cross-technology) gateways and devices whose transmissions are much shorter than LoRa. In practice, we see gateways that record as high as 57% packet failures across their clients (see Section 2). These losses are masked by costly retries or increased spreading factors that dramatically reduce client battery-life.

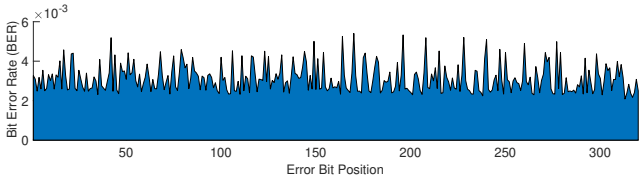


**Figure 10: Bit errors are uniformly distributed**

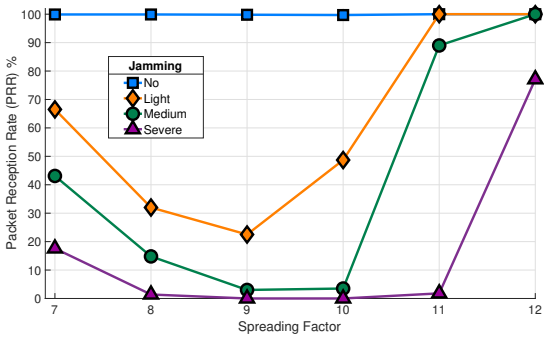


**Figure 11: Interference and Spreading Factor effect on Packet Integrity**

**Intra-Packet Bit Error Distribution:** Next, we look more closely at the positions of each error within the corrupt packets and see that they are primarily distributed uniformly as shown in Figure 10. This is important to note as it dramatically hampers the ability for standard Hamming Code error correction to perform well.

**Local Interference Sources:** We evaluate our system both indoors and outdoors on a live network with sensor traffic and interference sources. We observe that the outdoor tests were subject to significantly more interference as compared with the indoor installation (Figure 4). To our understanding, this is related to the large coverage area of each gateway, since the outdoor test-bed is deployed on some of the tallest buildings around the city.

Similar to the deployment scenario outlined in Figure 5, we performed a set of experiments that simulate the effect of interference from co-located networks on outdoor LoRaWAN deployments. We used one of our LoRaWAN gateways installed on the rooftop (Rx) of a four stories building and a transmitter LoRaWAN client (Tx) placed 400 meters in LoS from the gateway. We slowly increased the level of interference by using jammer clients (Jx) on a co-located network with close proximity to the receiver gateway (Rx): *Light*, *Medium* and *Severe* correspond to 1, 2, and 4 jamming clients (Jx). Each jamming client was transmitting at a 40-byte payload at either SF5 or SF6 with a uniform distribution of delay corresponding to 25-50% of their total air-time. The small spreading factors ensure that the transmissions are much shorter than regular LoRaWAN Transmissions (SF7-SF12), which is consistent with typical short range transmissions. Each data point in the plot represents 10000 LoRaWAN client transmissions (Tx). Figure 11 demonstrates the effect of the interference sources on the Packet Reception Rate (PRR) of the outdoor gateway (Rx) and the gain in robustness of LoRa CSS. The figure shows that shorter messages are subject to less

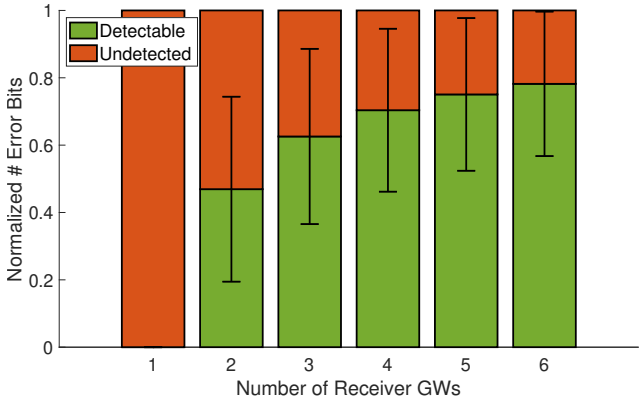


**Figure 12: Detecting corrupt bits in receptions using shared information across gateways only**

interference and reinforces the notion that minimizing packet duration can be effective in increasing network capacity. Note that as the SF increases, packets become more vulnerable to noise and we see a drop in PRR before the coding gain compensates to increase performance. As clients (Tx) switch to more robust transmission rates (higher SFs), they inevitably use twice the amount of air-time (per SF increase), thus making the transmission more susceptible to sporadic interference. Starting with SF10 the use of more robust modulations (even as packets get longer) helps limit the impact of interference, resulting in an increase in PRR.

**Error Detection Across Multiple Gateways:** To study how OPR detects errors in the presence of multiple receptions, we ran experiments on our indoor test-bed with multiple gateways as shown in Figure 8. We deployed transmitter nodes (Tx) across the building, sending 40-byte payloads at SF12 with hardware attenuators tuned to the receivers such that the received signal strength was just above LoRa's noise floor (-124dB for SF12). This simulates a dense urban scenario with high levels of multi-path fading. Transmitters usually operate near the receivers' sensitivity threshold, helping them use less power without overpowering concurrent transmissions (as discussed in the case of imperfect LoRa Orthogonality in Section 2). Interference sources (Jx) based on LoRa and 802.15.4g [15] were similarly placed across the building transmitting with a uniform distribution of delay corresponding to 5-40% of their total air-time. We log the exact value of each transmitted packet which is used as ground-truth for exactly how many and which bits are correctly detected by the system. Each time multiple gateways receive a packet, the value is XORed together to form the correction vector. Figure 12 shows the impact of the number of receivers on the accuracy of the final error vector, assuming a single jamming node. Note that this plot captures the total number of bit errors, including a mixture of packets that OPR was able to correct and some that it was not (due to interference in the header/preamble or exceeding the 30-bit correction limit). We see a logarithmic increase in the detection of corrupt bits as the number of receivers increases. The total number of packets that were recovered is discussed in the next section. Figure 13 generalizes the bit detection performance in the presence of multiple jamming nodes. We see that interference is additive and eventually saturates while our gain continues to increase with the

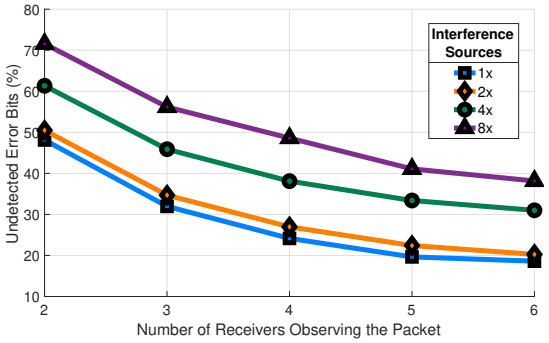


**Figure 13: Corrupt bits across gateways in the presence of different interference levels**

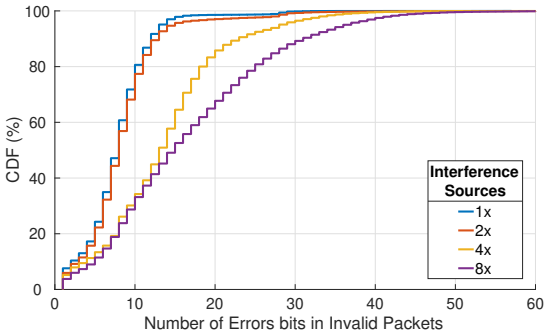


**Figure 14: CDF of number of bit errors**

| | | Predicted Bit Error | |
|---|---|---|---|
| | | Yes | No |
| Actual | Yes | 83% | 17% |
| | No | 0% | 100% |

**Table 2: Performance of bit error detector using XOR across multiple gateways**

number of gateways. Figure 14 shows the corresponding cumulative distribution function of bit errors as we increase the number of interference sources. This corroborates the data previously shown in Figure 9, and shows that the number of bit errors in a packet are within the number of bits OPR can recover.

**Bit Error Recovery Across Multiple Gateways:** Next, we evaluate how the number of gateways receiving a packet impacts our ability to recover data. The setup of this experiment was similar to the previous indoor experiment: a transmitter at SF12 sending 40-byte payloads and 6 gateways receiving the signal close to LoRa's noise floor. The added information at each receiver allows for detection and recovery of more bit errors. Table 2 presents the confusion matrix in this setup by exactly identifying error bits across receptions from multiple gateways, where we see that most of our predicted bit errors are in the correct location (87%) and OPR always correctly confirms non-erroneous bits (i.e. doesn't do any additional damage).

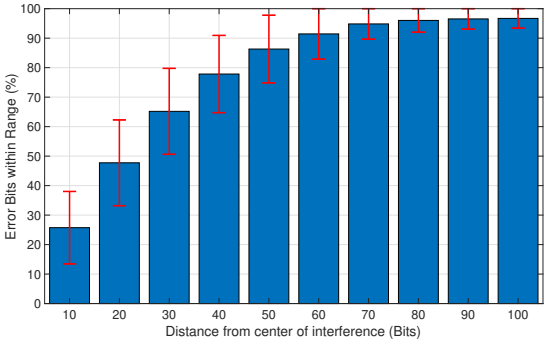


**Figure 15: Error search space using RSSI hints**

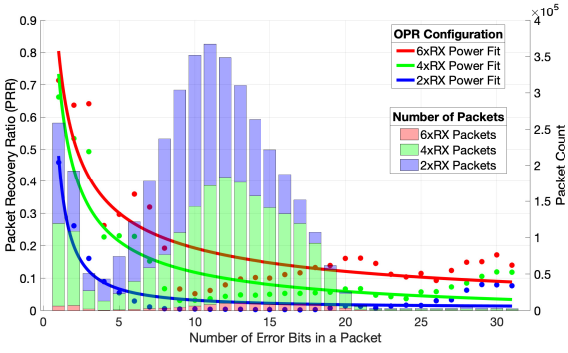


**Figure 16: OPR's Packet Correction Performance**

**RSSI Error Recovery Search Space:** When OPR only manages to receive a single invalid message from the gateways, it cannot use XOR or majority for recovery. Instead, OPR falls back to using RSSI hints to estimate the damaged location as described in Section 4.1. Our data demonstrate that due to the probabilistic nature of bit flips, the center of the errors does not always align with the perceived bit errors in the payload. OPR uses a running average filter in conjunction with a dynamic thresholding mechanism to detect the error centroids corresponding to the bit errors. The filtering function and thresholding parameters were derived from maximizing the bit-error detection and cross-validating over the data in order to avoid over-fitting. Figure 15 shows that a large percentage of bit errors (up to 80%) can be found within a distance of 30 bits. For short bursts of interference, this detector can be quite effective; otherwise, it is a bit of a Hail Mary.

**OPR Packet Correction Performance:** Using the data from our indoor data set, we computed the percentage of the total number of packets that OPR was able to correct that were received by 2, 4, and 6 gateways respectively and that normally would have failed their CRC check. Figure 16 shows the packet recovery rate along with the distribution of the total number of packets for each gateway set. The high-level takeaway is that OPR is able to recover between 20% and 72% of the packets with less than 10 bits of error (peak correction rate of 72% on 1-bit errors sensed by 6 gateways). We also note that it is more likely that a packet will be received by fewer gateways, but

| Code Rate | Coding Gain (dB) |
|---|---|
| 4/5 | 0.2 |
| 4/6 | 0.4 |
| 4/7 | 0.6 |
| 4/8 | 0.9 |

**Table 3: Code Rate vs. Coding Gain (vs. no codes) of LoRa's Hamming codes for a packet error rate of 20% as per SemTech's LoRa deployment guide [33].**

that a larger number of gateways improves the odds of being able to correct the packet. As one might expect, it gets significantly less likely that packets are found with many erroneous bits. We believe the shape of the histogram is a function of the delay and length of the jammer traffic used in this particular dataset. In practice, this would be largely a function of the particular environment. For certain clients at the edge of the network or subject to heavy interference, this would have a dramatic impact on their battery life (2x or more in terms of radio retries).

# 7 DISCUSSION

## 7.1 Rate Adaptation vs. OPR

In this section, we discuss how OPR inter-operates with LoRa's traditional rate adaptation schemes including forward error coding and change of spreading factor, which we loosely term coding strategies. We specifically seek to understand the regimes in which OPR will outperform standard coding strategies and can effectively provide identical resilience to coding, while significantly improving client battery life. To understand why such regimes should exist, recall that for a small number of bit flips, OPR can rely on signals observed across multiple gateways, as well as known header fields and CRCs, to provide error resilience without impacting client message length. In contrast, coding inevitably requires an expanded message length to improve resilience, as a result of trading off client battery life and network capacity for improved packet recovery rate.

To better understand this trade-off qualitatively and quantitatively, we analyze the coding schemes used by LoRa. LoRa uses a Hamming code with different code rates 4/5, 4/6, 4/7, or 4/8 as part of its modulation [2]. Unfortunately, Hamming codes, while simple to implement, are known to be significantly sub-optimal in performance, especially for short bursty errors or at low SNRs [22]. This is fundamentally because Hamming codes can lead to (detectable) error even if two bits within the same block are flipped and undetectable errors, if as little as three bits per block are in error. Indeed, SemTech's LoRa design guide has quantified the coding gain in dB of the different choices of coding rates [33] under Additive White Gaussian Noise reproduced in Table 3 for a packet error rate of 20%. The results show a relatively modest coding gain (lower than a dB), primarily due to choice of code. We surmise that this design decision was deliberate, to favor simplicity of devices and the decoding process, unfortunately at the expense of coding resilience.

In contrast, we argue that combining received signals across multiple base stations provides a significant improvement in SNR

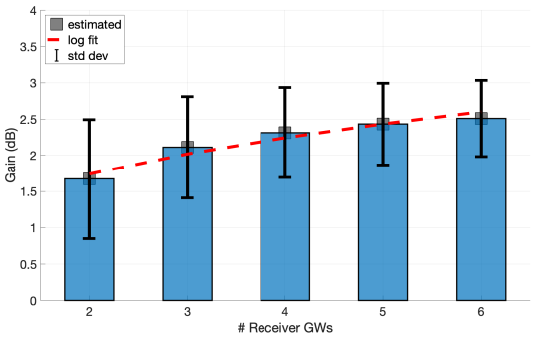


**Figure 17: OPR's gain in SNR vs. number of receiving gateways increases logarithmically.**

gain that scales in a logarithmic fashion with the number of gateways, allowing for more significant gain improvements without impacting client simplicity. Intuitively, this stems from the fact that sources of noise and interference across gateways, if independent, can significantly reduce probabilities of identical bits being in error. This gain is bounded by a $O(\log n)$ gain that stems from diversity coherent combining [9], where $n$ is the number of gateways. An analytical expression for this gain is challenging, without assumptions about the locations and nature of interfering sources.

We therefore present an experimental analysis of the SNR gains achieved with up to six gateways in Fig. 17. Our results provide two key takeaway messages: (1) First, the coding gains achieved over the one-gateway case can be as high as 2.5 dB in aggregate for six gateways, outperforming the gains achieved by LoRa's standard Hamming Forward Error Correction codes. (2) Second, the SNR gains scale further with a larger number of gateways. In other words, OPR can perform particularly favorably in comparison with codes with dense deployments of LoRa femtocells (for example, Comcast's pilot integration of LoRa gateways in set-top-boxes [31]).

Finally, we note that it may be possible to combine OPR with error correcting codes to achieve the best of both worlds. For instance, we could devise co-optimized decoding algorithms that both account for FEC and the locations of bursty bit errors uncovered by OPR. We note that our current implementation favors OPR to be used in isolation primarily because of the bursty error patterns that are unfavorable to LoRa's Hamming codes. However, we leave a detailed study of OPR combined with coding, and more broadly, devising energy-aware coding strategies for LP-WAN, to future work.

## 7.2 Security and Privacy Implications

OPR is designed to work alongside standard LoRaWAN infrastructure, intercepting traffic between the gateways and the Network Server and speculatively generating payloads that pass CRC validation. These payloads are then validated on the Network server which removes messages with invalid MIC (Message Integrity Check), a cryptographic function based on the encrypted payload. In this way, OPR remains compatible with LoRaWAN's security model. As noted before, OPR does not assume knowledge of the device-specific root keys (NtwKey and AppKey), and because of this, it cannot exploit features in the data payload to correct packets.

In a trusted environment, OPR could be part of the Network and Application Servers, which permit more efficient filtering and validation pipelines that exploit the payload structure itself. While OPR deals with encrypted packets, we note that these could still leak some information such as traffic patterns and message lengths. Network administrators should consider these privacy implications and protect OPR's endpoints and communication channels adequately.

### 7.3 Cloud Processing Latency

LP-WAN systems operate at a low data rate and have relaxed acknowledgment and reply windows (in the order of seconds). For OPR, this means that there is enough time to transfer packets to the cloud for processing. There is a trade-off between using local servers (introducing less transmission latency and thus allowing more time to be spent on processing each packet) and the scale of the compute available on the cloud. In Section 6, we have shown the trade-off between compute, latency, and number of recovered bits. One could even imagine a multi-tiered architecture where the location of OPR computation changes depending on available compute resources and network interconnect speeds.

## 8 RELATED WORK

**LP-WAN in the Industry:** LP-WAN's promise of inexpensive, low-power, and long-range connectivity for IoT devices attracted significant attention. Recent years have seen the appearance of many competing LP-WAN technologies, such as LoRa [2], SigFox [43], Weightless-N [39], LTE-M [14], NB-IoT [29], and others. We have also seen many commercial deployments from companies such as AT&T [3] and Comcast [8], and examples of real applications in precision agriculture [30], healthcare [20, 25], and smart cities [5], to name a few. Our observations on high packet loss in real-world LoRa deployments are consistent with previous reports from other deployments [26].

**Payload Recovery in LP-WAN:** The research community developed a considerable amount of work to analyze [1, 4, 21, 26] and improve LP-WANs [13, 27, 28, 36]. Focusing on payload recovery techniques, Charm [9] introduced a technique to coherently combine signals from multiple receivers and decode packets that would not be successfully received by any individual gateway. To recover collided packets, Choir [12] exploits frequency biases in LoRa, while FTrack [41] considers time domain and the frequency domain features in end devices. In contrast, OPR does not rely on custom radio front end or low-level radio signal processing, and thus can be implemented as a software-only solution that does not require any modification to the LoRa client or base station hardware. DaRe [23] is an application layer encoding technique for LoRaWAN which uses convolutional and fountain codes to add redundant data during a transmission session. This allows for a large payload to be split across multiple packets with increased redundancy at the cost of an increase in transmission time. Instead of adding redundant data at the application layer, OPR works with individual packets at the MAC layer. It takes advantage of already existing redundancy in received packets, due to the architecture of LP-WAN. Adding application-layer coding increases the time to decode a packet at the network infrastructure backend, and, in our work, we also trade off packet processing time to recover errors and allow the low-power end-devices to save more energy. We believe this is a reasonable trade-off, as the network backend has access to the scaling capabilities of the cloud, which can easily cope with LP-WAN latency and throughput requirements [1].

**Wireless Interference Management:** Researchers in the wireless networking community outside of LP-WAN have, similarly to OPR, proposed to recover bit errors across multiple receptions, but in WLAN access points [24, 40]. We argue that LP-WANs are more amenable to such techniques, and their characteristics allow us to take advantage of the significant compute available in the Cloud. Previous literature also studied how to recover corrupt packets by selectively re-transmitting parts of the packet that are deemed to be corrupted [18], and reconstructing packets from multiple corrupt ones [11]. Recently, researchers have also devised selective re-transmission schemes that exploit the fact that RSSI values in 802.15.4 are highly correlated with byte errors under interference [42]. Rateless coding [10] has also been employed to recover corrupted packets and effectively increase the communication distance, but requires a link-layer protocol to synchronize the sender and receiver. These previous techniques informed our work, but the design of LP-WANs differs significantly, invalidating some of these techniques: re-transmissions introduce significant power costs, and sender-receiver synchronization requires substantial downlink traffic, which is also inefficient in LP-WANS.

## 9 CONCLUSION AND FUTURE WORK

In conclusion, this paper presented a technique for improving client to gateway LP-WAN packet reception in the presence of interference. Two key insights are that (1) interference impacts a message received by multiple gateways in different bit locations, and (2) LoRaWAN radios have the ability to capture fine-grained RSSI values across the entire length of a packet. Given hints of where errors are located within a packet, LP-WAN systems have ample time to test all permutations on potential error bits until the packet passes its CRC check. The small subset of these candidate packets can then be passed to the application layer where, with a high probability, only the actual data packet passes the MIC used for LoRaWAN security. Unlike previous work that attempts to use receiver diversity to improve performance at the physical and MAC layers, our approach effectively pushes link-layer functionality into the cloud. This has the advantage of being immediately applicable through server-side updates to all existing LoRaWAN deployments.

In the future, we would be interested in studying the impact of how the underlying structure of chirp spread spectrum coding could be used to refine how our system searches through the error space. For example, if we know that particular symbols are much more likely to fail in a particular manner, we can structure the combinatorial options to decrease the search space. Likewise, it would be interesting to explore the trade-off of how much information is gained from each additional receiver that detects a partially corrupt packet. In the spirit of rate-adaptive codes, a system should expand or refine the number of bits it is able to error-correct based on the number of received packets. This could be beneficial in systems with extremely low signal-to-noise ratios like what one might expect with backscatter LoRa devices [37].

## ACKNOWLEDGMENTS

This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, award number: 2018-JU-2779, by national funds through the Portuguese Fundação para a Ciência e a Tecnologia under PhD grant PD/BD/114428/2016 and by US National Science Foundation under grant No. 1942902.

## REFERENCES

[1] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. 2017. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine* 55, 9 (Sep. 2017), 34–40. https://doi.org/10.1109/MCOM.2017.1600613

[2] LoRa Alliance. 2015. LoRaWAN What is it? - A Technical Overview of LoRa and LoRaWAN LoRa Alliance. https://www.tuv.com/media/corporate/products_1/electronic_components_and_lasers/TUeV_Rheinland_Overview_LoRa_and_LoRaWANtmp.pdf. [Online; accessed Dec-2019].

[3] AT&T. 2019. AT&T, Vodafone Business IoT Deployments. https://about.att.com/story/2019/att_vodafone_business.html. [Online; accessed Dec-2019].

[4] Norbert Blenn and Fernando A. Kuipers. 2017. LoRaWAN in the Wild: Measurements from The Things Network. *CoRR* abs/1706.03086 (2017). arXiv:1706.03086 http://arxiv.org/abs/1706.03086

[5] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. 2016. Long-range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios. *IEEE Wireless Communications* 23, 5 (2016), 60–67.

[6] Gonglong Chen and Wei Dong. 2019. Exploiting Rateless Codes and Cross-Layer Optimization for Low-Power Wide-Area Networks. (2019).

[7] ChirpStack. 2019. ChirpStack, open-source LoRaWAN® Network Server stack. https://www.chirpstack.io/. [Online; accessed Dec-2019].

[8] Comcast. 2019. MachineQ. https://machineq.com/. [Online; accessed Dec-2019].

[9] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe. 2018. Charm: Exploiting Geographical Diversity through Coherent Combining in Low-Power Wide-Area Networks. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 60–71. https://doi.org/10.1109/IPSN.2018.00013

[10] W. Du, Z. Li, J. C. Liando, and M. Li. 2016. From Rateless to Distanceless: Enabling Sparse Sensor Network Deployment in Large Areas. *IEEE/ACM Transactions on Networking* 24, 4 (Aug 2016), 2498–2511. https://doi.org/10.1109/TNET.2015.2476349

[11] Henri Dubois-Ferrière, Deborah Estrin, and Martin Vetterli. 2005. Packet Combining in Sensor Networks. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems* (San Diego, California, USA) *(SenSys '05)*. ACM, New York, NY, USA, 102–115. https://doi.org/10.1145/1098918.1098930

[12] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering Low-Power Wide Area Networks in Urban Settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. ACM, New York, NY, USA, 309–321. https://doi.org/10.1145/3098822.3098845

[13] Weifeng Gao, Wan Du, Zhiwei Zhao, Geyong Min, and Mukesh Singhal. 2019. Towards Energy-Fairness in LoRa Networks. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 788–798.

[14] Svetlana Grant. 2016. 3GPP Low Power Wide Area Technologies. *GSMA White Paper* (2016).

[15] Frederik Hermans, Hjalmar Wennerström, Liam Mcnamara, Christian Rohner, and Per Gunningberg. 2014. All Is Not Lost: Understanding and Exploiting Packet Corruption in Outdoor Sensor Networks. In *Proceedings of the 11th European Conference on Wireless Sensor Networks - Volume 8354* (Oxford, UK) *(EWSN 2014)*. Springer-Verlag New York, Inc., New York, NY, USA, 116–132. https://doi.org/10.1007/978-3-319-04651-8_8

[16] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. 2019. Netscatter: Enabling Large-scale Backscatter Networks. In *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation* (Boston, MA, USA) *(NSDI'19)*. USENIX Association, Berkeley, CA, USA, 271–283. http://dl.acm.org/citation.cfm?id=3323234.3323258

[17] ITU-T. [n.d.]. ITU-T Recommendation V.41. https://www.itu.int/rec/T-REC-V.41/en. accessed on Oct 09, 2019.

[18] Kyle Jamieson and Hari Balakrishnan. 2007. PPR: Partial Packet Recovery for Wireless Networks. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Kyoto, Japan) *(SIGCOMM '07)*. ACM, New York, NY, USA, 409–420. https://doi.org/10.1145/1282380.1282426

[19] Matthew Knight and Balint Seeber. 2016. Decoding LoRa: Realizing a Modern LPWAN with SDR. *Proceedings of the GNU Radio Conference* 1, 1 (2016). https://pubs.gnuradio.org/index.php/grcon/article/view/8

[20] Piers W Lawrence, Trisha M Phippard, Gowri Sankar Ramachandran, and Danny Hughes. 2017. Developing the IoT to support the health sector: A case study from kikwit, DR congo. In *International Conference on Emerging Technologies for Developing Countries*. Springer, 45–56.

[21] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. 2019. Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study. *ACM Trans. Sen. Netw.* 15, 2, Article 16 (Feb. 2019), 35 pages. https://doi.org/10.1145/3293534

[22] David JC MacKay and David JC Mac Kay. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.

[23] P. J. Marcelis, V. S. Rao, and R. V. Prasad. 2017. DaRe: Data Recovery through Application Layer Coding for LoRaWAN. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 97–108.

[24] Allen Miu, Hari Balakrishnan, and Can Emre Koksal. 2007. Multi-radio diversity in wireless networks. *Wireless Networks* 13, 6 (2007), 779–798.

[25] Juha Petäjäjärvi, Konstantin Mikhaylov, Matti Hämäläinen, and Jari Iinatti. 2016. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. In *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*. IEEE, 1–5.

[26] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov. 2016. Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 1–7. https://doi.org/10.1109/PIMRC.2016.7794569

[27] Z. Qin and J. A. McCann. 2017. Resource Efficiency in Low-Power Wide-Area Networks for IoT Applications. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–7. https://doi.org/10.1109/GLOCOM.2017.8254800

[28] M. Rahman and A. Saifullah. 2018. Integrating Low-Power Wide-Area Networks in White Spaces. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 255–260. https://doi.org/10.1109/IoTDI.2018.00033

[29] Rapeepat Ratasuk, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. 2016. NB-IoT system for M2M communication. In *2016 IEEE wireless communications and networking conference*. IEEE, 1–5.

[30] RCR Wireless News. 2015. Agricultural IoT Promises to Reshape Farming. https://www.rcrwireless.com/20151111/internet-of-things/agricultural-internet-of-things-promises-to-reshape-farming-tag15. [Online; accessed Dec-2019].

[31] Light Reading. 2017. Comcast Aims to Layer LoRa Into XB6 Gateway. https://www.lightreading.com/iot/iot-strategies/comcast-aims-to-layer-lora-into-xb6-gateway/d/d-id/736347. [Online; accessed Dec-2019].

[32] B. Reynders and S. Pollin. 2016. Chirp Spread Spectrum as a Modulation Technique for Long Range Communication. In *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. 1–5. https://doi.org/10.1109/SCVT.2016.7797659

[33] Semtech. [n.d.]. SX1272/3/6/7/8: LoRa Modem Designer's Guide.

[34] Semtech. October 2019. Corecell Reference Design for Gateway Applications Based on SX1302 and SX1250. https://www.semtech.com/products/wireless-rf/lora-gateways

[35] Lin Shu and Daniel J Costello. 2004. *Error control coding*. 704–712 pages.

[36] Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R. Smith, and Shyamnath Gollakota. 2017. LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 105 (Sept. 2017), 24 pages. https://doi.org/10.1145/3130970

[37] Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R. Smith, and Shyamnath Gollakota. 2017. LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 105 (Sept. 2017), 24 pages. https://doi.org/10.1145/3130970

[38] David Wagner. 2002. A generalized birthday problem. In *Annual International Cryptology Conference*. Springer, 288–304.

[39] Weightless SIG. 2015. Weightless Specification. http://www.weightless.org/about/weightless-specification. [Online; accessed Dec-2019].

[40] Grace R. Woo, Pouya Kheradpour, Dawei Shen, and Dina Katabi. 2007. Beyond the Bits: Cooperative Packet Recovery Using Physical Layer Information. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking* (Montréal, Québec, Canada) *(MobiCom '07)*. Association for Computing Machinery, New York, NY, USA, 147–158. https://doi.org/10.1145/1287853.1287871

[41] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: Parallel Decoding for LoRa Transmissions. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems* (New York, New York) *(SenSys '19)*. ACM, New York, NY, USA, 192–204. https://doi.org/10.1145/3356250.3360024

[42] Z. Zhao, W. Dong, G. Chen, G. Min, T. Gu, and J. Bu. 2017. Embracing Corruption Burstiness: Fast Error Recovery for ZigBee under Wi-Fi Interference. *IEEE Transactions on Mobile Computing* 16, 9 (Sep. 2017), 2518–2530. https://doi.org/10.1109/TMC.2016.2630696

[43] Juan Carlos Zuniga and Benoit Ponsard. 2016. Sigfox System Description. *LPWAN-IETF97, Nov. 14th* 25 (2016).