

Revolutionizing LoRa Gateway with XGate: Scalable Concurrent Transmission across Massive Logical Channels

Shiming Yu¹, Xianjin Xia¹, Ningning Hou², Yuanqing Zheng¹, Tao Gu²

¹ The Hong Kong Polytechnic University, Hong Kong SAR, China

² Macquarie University, Sydney, Australia

shiming.yu@connect.polyu.hk, xianjin.xia@polyu.edu.hk, ningning.hou@mq.edu.au,

yuanqing.zheng@polyu.edu.hk, tao.gu@mq.edu.au

ABSTRACT

LoRa is a promising technology that offers ubiquitous low-power IoT connectivity. With the features of multi-channel communication, orthogonal transmission, and spectrum sharing, LoRaWAN is poised to connect millions of IoT devices across thousands of logical channels. However, current LoRa gateways utilize hardwired Rx chains that cover only a small fraction (<1%) of the logical channels, limiting the potential for massive LoRa communications. This paper presents XGate, a novel gateway design that uses a single Rx chain to concurrently receive packets from all logical channels, fundamentally enabling scalable LoRa transmission and flexible network access. Unlike hardwired Rx chains in the current gateway design, XGate allocates resources including software-controlled Rx chains and demodulators based on the extracted meta information of incoming packets. XGate addresses a series of challenges to efficiently detect incoming packets without prior knowledge of their parameter configurations. Evaluations show that XGate boosts LoRa concurrent transmissions by 8.4× than state-of-the-art.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; • **Networks** → **Network protocol design**.

KEYWORDS

Internet of Things, LPWAN, LoRa, Logical Channel

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. *ACM MobiCom '24, September 30-October 4, 2024, Washington D.C., DC, USA* © 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0489-5/24/09...\$15.00

<https://doi.org/10.1145/3636534.3649375>

ACM Reference Format:

Shiming Yu¹, Xianjin Xia¹, Ningning Hou², Yuanqing Zheng¹, Tao Gu². 2024. Revolutionizing LoRa Gateway with XGate: Scalable Concurrent Transmission across Massive Logical Channels. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*, September 30-October 4, 2024, Washington D.C., DC, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3636534.3649375>

1 INTRODUCTION

Low-Power Wide-Area Networks (LPWANs) have emerged as a promising technology for connecting a vast number of physical things to the Internet. LoRa, as one of the LPWAN technology, features large coverage and ultra-low power communication, making it ideal for low data-rate IoT applications such as smart agriculture [29, 38] and smart metering [3, 17]. LoRa is poised to potentially become the de facto IoT technology, enabling ubiquitous low-cost IoT connectivity at city scales [32].

LoRa adopts a series of novel PHY- and MAC-layer designs to support massive IoT connections. *Multi-channel communication*: As a LoRa packet occupies a narrow bandwidth (e.g., 125 kHz by default), the entire LoRa spectrum can be divided into many sub-channels. Multiple LoRa nodes can concurrently communicate at different physical channels without mutual interference. *Orthogonal communication*: LoRa PHY modulates a packet with various packet parameters such as bandwidths (BW) and Spreading Factors (SF). Signals differing in SF or BW can be orthogonal and demodulated even in the same physical channel. In other words, orthogonal *logical channels* with different packet parameters can be built over the same physical channel. For example, the US902-928 MHz ISM band can be divided into 208 physical channels. With the above orthogonal configurations, thousands of logical channels can be created for concurrent transmissions.

Conventional commodity gateways rely on multiple hardwired Rx chains to receive LoRa packets from a few (e.g., nine channels [34]) out of the thousands of logical channels. Restricted by the existing gateway design, all connected nodes are crowded in a few channels, leading to serious packet

collisions and degraded network performance. Furthermore, since the operating channels of a gateway are typically pre-configured in advance, they may not be optimal for changing conditions. Existing gateways do not provide a sufficient number of channels for LoRa nodes to adapt to the ever-changing network conditions. Although one could increase the number of gateways or Rx chains per gateway to expand the operational channels, such a solution would necessitate a substantial number of gateways and Rx chains to effectively cover all logical channels in the entire LoRa spectrum.

Rather than assembling many Rx chains and locking each chain on a fixed channel as existing gateways do, we aim to fully support all possible LoRa channels using a single Rx chain. We achieve this by efficiently detecting incoming packets of any channels on-the-fly and judiciously dispatching software-controlled radio resources (e.g., Rx chains, packet demodulators) on demand for packet reception and demodulation. In addition to breaking the barrier of massive IoT connections and enabling scalable concurrent transmissions across all LoRa logical channels, we believe such a design holds great promise in supporting new communication and networking paradigms for LoRaWAN. For example, with the ability to receive packets without prior knowledge of their configurations in our design, LoRa nodes can seamlessly join deployed networks in a “come-and-be-served” manner, eliminating the need for cumbersome channel configurations for both nodes and gateways. Additionally, this new gateway design empowers LoRa nodes to freely select logical channels and parameter configurations for their next transmissions, facilitating fast adaptation to network dynamics.

To this end, this paper presents *XGate*, a novel gateway design that enables flexible packet reception over massive LoRa channels. At its core, *XGate* decouples the process of LoRa packet reception into two phases, *i.e.*, *packet detection* and *packet decoding*. During the packet detection phase, a packet detector scans the whole Rx spectrum to identify incoming packets and extract their detailed configurations. *XGate* employs a pool of software-based Rx chains and packet decoders, which can be dynamically configured based on the packet detection results, enabling flexible reception on specific LoRa channels.

However, implementing the idea in practice poses substantial challenges. Firstly, unlike traditional gateways that listen on fixed channels, *XGate* aims to receive packets from all logical channels within an Rx spectrum. As a packet may arrive in any logical channel unknown a priori, it is non-trivial to identify a packet in a large spectrum, let alone recover the parameters of a packet. The problem turns out to be even more challenging when receiving multiple packets concurrently from different channels. Unlike existing gateways that can use the preset channel information to remove interference from unwanted channels, the signals separated in different sub-channels are now superimposed in a common

spectrum, which can cause interference to the detection of concurrent packets (*i.e.*, termed *cross-channel interference*). Moreover, detecting packets with the raw signals of a large Rx spectrum, rather than the narrow-band signals of a single LoRa channel, may suffer SNR losses due to increased noise and interference. Therefore, it is crucial, yet challenging, for *XGate* to detect and receive packets without compromising sensitivity.

XGate introduces a novel design that enables the detection and reception of LoRa packets in a large spectrum while achieving high resilience to noise and interference. We find that it is feasible to use a chirp signal in one channel to dechirp signals from any other channels once the two signals have the same chirp slope (*i.e.*, rate of frequency change over time). The dechirped signals exhibit periodic patterns in the frequency and time domains, attributed to the repetitive signal structure of a LoRa preamble, which is leveraged to detect any incoming packets. We exploit the fact that the frequency and periodicity of dechirped signals essentially carry information about the channel and chirp profiles (e.g., BW and SF) of a packet. Concurrent packets differing in channels or chirp configurations are usually detected as different frequencies and distinctive periodicity patterns that can be separated in the frequency and time domains. Built on such observations, *XGate* employs a novel strategy that tracks the frequency changes of dechirped signals across consecutive detection windows to accurately recover communication channels and parameters for concurrent packets. *XGate* scans the signals in several rounds with diverse chirp slopes and window sizes, where short and long windows are used collaboratively to balance sensitivity and delays for packet detection. Lastly, *XGate* configures software Rx chains with the correct parameters to receive detected packets.

We implement and evaluate *XGate* using Software Defined Radio (SDR) and commercial off-the-shelf (COTS) LoRa nodes. Results demonstrate that *XGate* significantly outperforms existing solutions. Specifically, *XGate* supports up to 8.4× more concurrent transmissions in the same spectrum than state-of-the-art [36]. *XGate* can be seamlessly integrated with existing LoRa parallel decoding techniques [36, 47] to support 78 % more concurrent transmissions.

In summary, our work makes the following contributions: (1) *XGate* is the first-of-its-kind gateway design that enables the scaling of LoRa concurrent transmissions across logical channels, revolutionizing the potential for massive IoT communications; (2) We propose novel techniques to efficiently detect channel activities across all LoRa channels of a given spectrum and accurately extract the configurations of incoming packets; (3) We implement and evaluate *XGate* to validate its effectiveness. Our evaluations demonstrate the significant improvements achieved by our design. Beyond this, we believe the design of *XGate* will have broader impacts

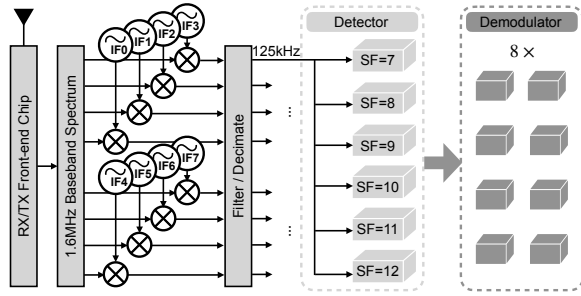


Figure 1: Block diagram of a COTS gateway [34].

on spectrum management, network planning, and channel adaptation in LoRaWAN operations and deployments.

2 BACKGROUND

LoRa PHY. LoRa is a physical layer technique for LPWAN communications [24]. It adopts a variant of Chirp Spread Spectrum (CSS) to modulate symbols, where the chirp frequency changes linearly with time. The duration of a chirp (*i.e.*, T_{chirp}) is controlled by Spreading Factor (SF) and bandwidth (BW) parameters of CSS, *i.e.*, $T_{chirp} = \frac{2^{SF}}{BW}$. A chirp with an increasing frequency from $-\frac{BW}{2}$ to $\frac{BW}{2}$ is called *base up-chirp* and represented as follows:

$$C(t) = e^{j2\pi(\frac{1}{2}kt - \frac{BW}{2}t)}, \quad (1)$$

where k indicates the frequency changing rate, termed *chirp slope*. Namely, $k = \frac{BW}{T_{chirp}} = \frac{BW^2}{2^{SF}}$.

LoRa changes the initial frequency of an up-chirp to modulate symbols. We can denote a modulated LoRa symbol $S(f_{sym}, t)$ as follows:

$$S(f_{sym}, t) = C(t) e^{j2\pi f_{sym}t}, \quad (2)$$

where f_{sym} is the initial frequency of a modulated symbol. To demodulate the symbol, a LoRa receiver first *dechirps* the modulated chirp by multiplying $S(f_{sym}, t)$ with a *down-chirp* denoted as $C^{-1}(t)$ which is the conjugate of a base up-chirp. The operation is represented as follows:

$$S(f_{sym}, t) C^{-1}(t) = e^{j2\pi f_{sym}t}. \quad (3)$$

Next, the receiver applies FFT to the dechirped signal to extract f_{sym} which indicates the symbol.

A LoRa packet starts with a preamble, followed by two up-chirps as synchronization words and 2.25 down-chirps as a Start Frame Delimiter (SFD), and ends with a packet payload. A preamble is composed of a variant number of base up-chirps with identical initial frequencies. A receiver radio leverages the signal structure of LoRa preamble to detect and demodulate a packet.

LoRa channels. LoRa operates in unlicensed ISM bands. Different from Wi-Fi and NB-IoT which have fixed channel plans, LoRaWAN does not specify the channel partition in the ISM bands. Instead, LoRaWAN allows end nodes to select channel frequency and bandwidth which uniquely define a *physical channel*. For example, a commodity LoRa radio

(*e.g.*, Semtech SX1276 [33]) supports up to ten bandwidth options ranging from 7.8 kHz to 500 kHz. Unlike conventional technologies that permit one node to communicate per channel, multiple LoRa nodes can simultaneously transmit at the same physical channel using orthogonal parameters (*e.g.*, different SFs and BWs) termed *logical channels*. As COTS LoRa radio supports SFs in 7~12, six logical channels can be created in the same physical channel. The total number of logical channels exceeds thousands in an ISM band.

LoRa gateway. Existing COTS LoRa gateways are typically equipped with Semtech chipsets [35]. Figure 1 shows a simplified block diagram of a COTS gateway consisting of eight Rx chains. The Rx chains can be programmed to receive on different physical channels. A gateway operates on the eight channels simultaneously to detect and receive incoming packets. Note that the operating channels of a gateway are usually configured beforehand and stay invariant after installation. It typically requires in-situ spectrum measurement and analysis to select and configure operating channels for deployed gateways, which can incur high measurement and maintenance overhead.

We note a large gap exists in using COTS gateways to support massive LoRa communications. As all connected nodes can only use the few operating channels of a COTS gateway, it can lead to poor spectrum utilization and sub-optimal network performance due to collisions. Whereas LoRaWAN defines many physical and logical channels with various frequencies and bandwidths. The channel settings available for LoRa nodes can be rather diverse. Ideally, if a gateway can cover all channels in a large spectrum, it can not only support more nodes with reduced collisions but also allow nodes to freely choose among diverse channel settings to better adapt to changing network conditions. However, because of the sheer amount of LoRa logical channels, it is challenging and even impractical to use COTS gateways to cover every channel with a dedicated Rx chain. It calls for a better gateway design that can detect and receive packets from all possible LoRa channels in a given spectrum.

3 RECEPTION FOR MASSIVE CHANNELS

Unlike COTS gateways receiving at a small fraction of fixed channels, we explore a new gateway design that can automatically detect channel activities, extract parameters of incoming packets, and configure Rx chains flexibly to receive in the whole spectrum. As LoRa nodes can adapt parameters (*e.g.*, channel, BW, SF) for every packet [1], if gateways are able to receive such packets configured on-the-fly, we can potentially scale LoRa concurrent transmissions to all logical channels. The reception over massive logical channels will also facilitate dynamic channel selection and data rate adaptation of nodes, and ease the pain of in-situ spectrum measurement and channel planning for network operators.

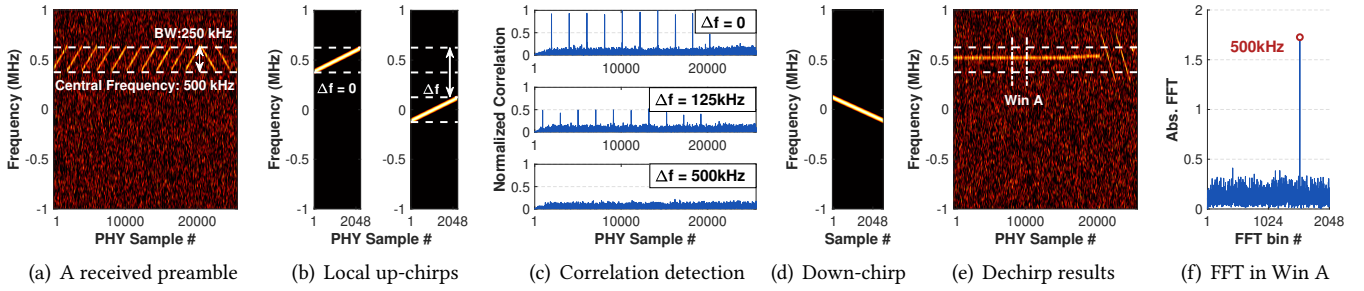


Figure 2: Packet detection in a large signal spectrum (2 MHz): (a) Raw signals of a LoRa preamble; (b,c) Correlation detection results with local up-chirps in different channel settings; (d-f) Results of dechirping preamble signals with a local down-chirp.

Without loss of generality, suppose N nodes communicate concurrently on different LoRa logical channels. Node i sends a packet in central frequency f_i with bandwidth BW_i and spreading factor SF_i . The signal of node i can be denoted as $S_i(f_{sym_i}, t) = C_i(t) e^{j2\pi f_{sym_i} t} = e^{j2\pi(\frac{1}{2}\frac{BW_i^2}{SF_i^2}t - \frac{BW_i}{2}t)} e^{j2\pi f_{sym_i} t}$. The received signals of a gateway covering N nodes over different channels can be represented as follows:

$$Y(t) = \sum_{i=1}^N h_i e^{j2\pi f_i t} S_i(f_{sym_i}, t) + n(t), \quad (4)$$

where h_i represents the wireless channel from node i to the gateway and $n(t)$ denotes the noise.

A conventional gateway needs the central frequency (f_i) and bandwidth (BW_i) information of an incoming packet to extract signals of the packet from $Y(t)$, and the spreading factor parameter (SF_i) to detect and demodulate the packet. For instance, COTS gateways need configurations of f_i and BW_i before packet reception. In contrast, our work explores a new gateway design that can receive packets without the knowledge of packet configurations or channel settings. We detect all meta-information of incoming packets (*i.e.*, f_i , BW_i and SF_i) from $Y(t)$ and flexibly configure Rx chains and packet demodulators to receive packets that can arrive at any channels with a variety of parameter settings.

4 XGATE DESIGN

4.1 Overview

XGate maintains a pool of ‘soft-wired’ Rx chains that can be dynamically configured on-the-fly to receive on any frequencies in diverse bandwidth settings. A key enabler of XGate is a powerful packet detector that can scan a large spectrum to detect the detailed parameters of incoming packets (*e.g.*, central frequency, bandwidth, spreading factor) and flexibly configure the software Rx chains to receive detected packets.

However, the implementation of XGate entails substantial challenges. First, as the number of LoRa logical channels can be rather large (*e.g.*, more than hundreds), it is non-trivial to detect packets on so many candidate channels without knowing incoming packets’ meta-information. Second, while

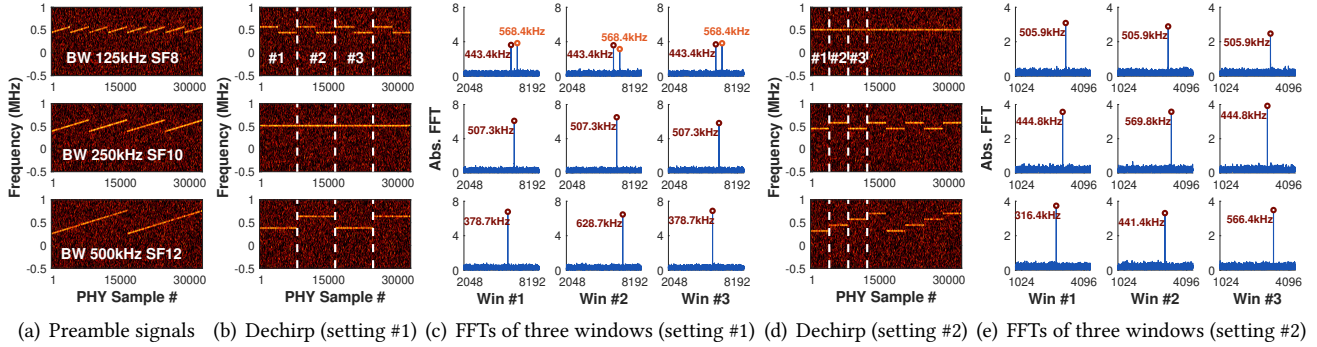
XGate monitors the whole Rx spectrum to detect LoRa packets, the signals originally separated in different sub-channels can potentially cause *cross-channel interference*. How to correctly detect the concurrent packets of different sub-channels and their detailed parameters is a major challenge faced by XGate. Moreover, as an Rx spectrum spans wider frequency bands than a single LoRa channel, it would include more noise and interference that may reduce signal quality. It is challenging to receive packets without compromising sensitivity. We present novel techniques to address these challenges in the following subsections.

4.2 Packet Detection over Logical Channels

4.2.1 Detecting Packet and Meta-information. As a packet may arrive from any logical channel of an Rx spectrum that is unknown to a gateway, XGate needs to detect not only the arrival of an incoming packet but also the channel and the parameter configuration of the packet.

Packet detection. XGate takes the raw signals of the whole Rx spectrum as inputs and detect incoming packets. A standard LoRa packet detection method uses correlation detection, which correlates incoming signals with a locally-generated LoRa preamble, to detect packets of a specific LoRa channel [8, 34, 52]. Can we extend the method to packet detection in the whole Rx spectrum? Figure 2(a) illustrates a LoRa packet received in a 2 MHz-wide spectrum. We correlate the signals with a local preamble chirp in the same SF and BW but different channel settings (see Figure 2(b)). As shown in Figure 2(c), we can only detect the highest correlation peaks when the central frequencies of the local chirp and the received signals are identical (*i.e.*, $\Delta f = 0$). It means that the channel (*e.g.*, central frequency) of a packet is required to bootstrap the correlation based detection in a particular channel. Obviously, such a method cannot be applied to XGate because the channel information of an incoming packet is not yet available.

How can we detect a packet without knowing the channel? Fortunately, as the Rx spectrum of a gateway covers many narrow LoRa channels, the signals of all LoRa logical



(a) Preamble signals (b) Dechirp (setting #1) (c) FFTs of three windows (setting #1) (d) Dechirp (setting #2) (e) FFTs of three windows (setting #2)

Figure 3: Detect BW-SF configurations: (a) Signals of three packets having the same chirp slope but different BW-SF settings; (b,c) Dechirped results with a down-chirp in (250 kHz, SF10); (d,e) Dechirped results with a down-chirp in (125 kHz, SF8).

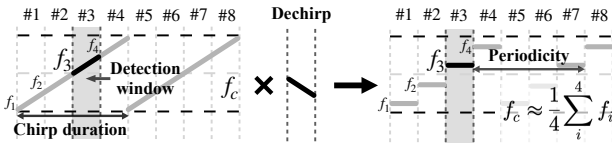


Figure 4: Dechirp with short windows.

channels are physically received. Thus, we can freely access signals of any LoRa logical channels in an Rx spectrum. For instance, we can use a down-chirp signal centered at frequency 0 as shown in Figure 2(d) to dechirp the signals displayed in Figure 2(a), which gives a result as shown in Figure 2(e). If a LoRa preamble is present, we will detect the same tone-frequency in successive windows. It allows us to detect a LoRa packet from the Rx spectrum even when the channel of the packet is unknown.

XGate extends the dechirp operation given by Eq.(3) from a single LoRa channel to an entire Rx spectrum to detect a packet, which can be represented as follows:

$$h_i e^{j2\pi f_i t} C_i(t) \cdot C_i^{-1}(t + \Delta t_i) = h_i e^{j2\pi f_i t} \cdot e^{j2\pi \Delta f_i t}, \quad (5)$$

where $h_i e^{j2\pi f_i t} C_i(t)$ denotes a received preamble chirp in a LoRa channel centered at frequency f_i , and $C_i^{-1}(t)$ is a local down-chirp centered at frequency 0 with the corresponding BW_i and SF_i . Here, Δt_i models the misaligned timing between symbol edges of incoming signals and the local down-chirp, which leads to a frequency component $\Delta f_i = (BW_i^2 / 2^{SF_i}) \Delta t_i$ in the dechirped results. Theoretically, the tone-frequency of dechirped signals (Figure 2(f)) is essentially $(f_i + \Delta f_i)$. Since edge misalignment Δt_i is within $\pm 0.5 \times$ of a chirp duration, the resulting Δf_i shall be within $\pm 0.5 \times$ of the chirp's bandwidth (i.e., BW_i). The tone-frequency $(f_i + \Delta f_i)$ deviates at most $\frac{BW_i}{2}$ from f_i (i.e., a channel's central frequency). Hereby, the tone-frequency of dechirped signals not only indicates the presence of an incoming packet, but also roughly estimates its central frequency.

Extracting packet parameters. After detecting an incoming packet, XGate further needs to extract the packet's BW and SF configurations for packet demodulation. Since LoRa chirps in different BWs or SFs are usually orthogonal,

if the BW/SF of down-chirp in Eq.(5) differs from that of a received packet, the packet signals would not be dechirped to tone frequency nor get detected. By exploiting chirp orthogonality, we may iterate through all BW-SF combinations to find the correct configurations of a packet.

However, such a method may suffer from ambiguous parameter detection since a packet can be dechirped by various chirps differing in BW or SF. For instance, the BW and SF of the three signals shown in Figure 3(a) are (125 kHz, SF8), (250 kHz, SF10) and (500 kHz, SF12), respectively. When we use a down-chirp in BW 250 kHz and SF10 to dechirp the three signals, we can detect strong tone frequencies not only from the signals of (250 kHz, SF10) but also from the other two signals. In fact, as the three signals have the same *chirp slope* defined as $k = BW^2 / 2^{SF}$ (Eq.(1)), they are not entirely orthogonal and can be dechirped by the same down-chirp with a slope of $-k$. In particular, Figure 3(b,c) plots the dechirped frequencies of those signals in three successive windows. We detect identical single-tone frequencies from successive windows for (250 kHz, SF10). While for (125 kHz, SF8), two frequencies are detected repetitively in every window. In this case, it is ambiguous to tell whether there is only one packet in (125 kHz, SF8) or two packets in (250 kHz, SF10).

XGate resolves such ambiguities by leveraging the fact that chirp signals differing in BW or SF would have different chirp lengths. In particular, if we use a shorter down-chirp (e.g., the shortest chirp length of the three signals) to dechirp the three signals shown in Figure 3(a), we will get a unique single-tone frequency in each short window, and the frequency varies across windows with distinctive periodic patterns for the three packets as shown in Figure 3(d,e). By detecting the periodicity of frequencies with short windows, XGate can disambiguate the three packets and uniquely recover BW and SF for each packet.

Putting it together. XGate uses down-chirps of different slopes, instead of iterating through BW-SF combinations, for packet detection. As multiple BW-SF settings share a

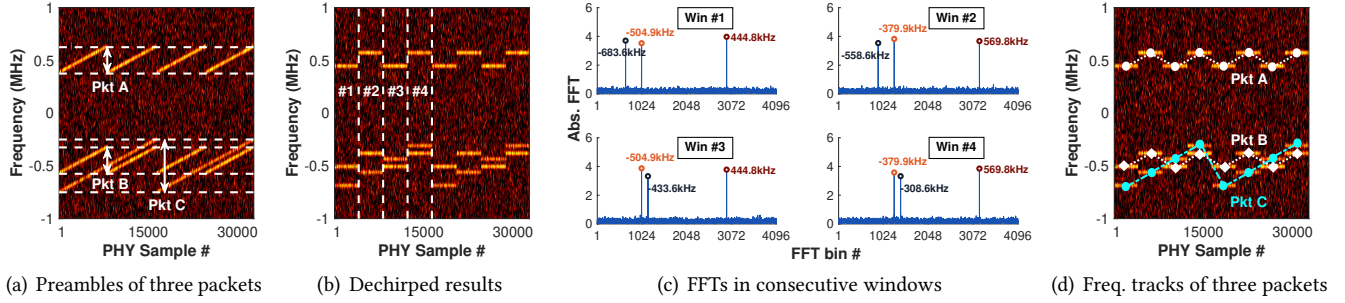


Figure 5: Detect concurrent packets: (a) Compound signals of packets A, B in (250 kHz, SF10) and packet C in (500 kHz, SF12); (b,c) Dechirped frequencies of signals from the same packet change at a fixed rate ($\Delta f=125$ kHz) over consecutive windows; (d) Track frequency changes to separate the three packets.

common chirp slope, we can efficiently check fewer slopes rather than all BW-SF settings to detect packets. To avoid ambiguities, we use a short window to dechirp and detect signal patterns, *e.g.*, typically choose a down-chirp with the shortest length of a given slope.

We rewrite Eq.(5) to represent the dechirp operation of a short window as below:

$$h_i e^{j2\pi f_i t} \cdot e^{j2\pi f_k t} C_w(k_i, t) \cdot C_w^{-1}(k_i, t) = h_i e^{j2\pi f_i t} \cdot e^{j2\pi f_k t}, \quad (6)$$

where $e^{j2\pi f_k t} C_w(k_i, t)$ is part of a long up-chirp located in the target window, f_k denotes the starting frequency of the chirp segment (Figure 4), k_i is the chirp slope, and $C_w^{-1}(k_i, t)$ is a corresponding down-chirp in slope k_i . According to Eq.(6), the tone frequency shown in any window of Figure 3(e) is essentially $(f_i + f_k)$, which signifies the starting frequency of the chirp segment in a short window. The starting frequency $(f_i + f_k)$ would change across windows and the periodicity of frequency changes equals the length of a complete chirp signal as Figure 4 illustrates.

XGate detects the periodicity of dechirped frequencies to infer the chirp length of received signals, from which the signals' BW and SF configurations can be restored unambiguously. Additionally, XGate calculates the average of the frequencies (*i.e.*, $(f_i + f_k)$'s) detected from different windows, which would be close to f_i , to estimate the channel frequency of a received packet. A refined estimation for precise channel frequency is presented in §4.4.

4.2.2 Detection of Multiple Packets. As multiple nodes transmit through different logical channels, XGate shall detect and receive all packets concurrently. Particularly, as concurrent packets have orthogonal parameters (*i.e.*, different chirp slopes), XGate could separate their signals accordingly by using down-chirps of different slopes to detect orthogonal packets in a divide-and-conquer way. However, if the signals of packets from different logical channels have a common chirp slope, they will interfere with each other (*i.e.*, *cross-channel interference*). For example, Figure 5(a) presents three packets in two chirp configurations, *i.e.*, (250 kHz, SF10) and (500 kHz, SF12), which have the same chirp slope. When we dechirp the compound signals of the whole spectrum, three

frequencies from different packets are detected in every window as shown in Figure 5(c), which adds interference to the packet detection and parameter extraction of the three packets. In this case, XGate needs to distinguish the frequencies of three packets and use the frequencies of each packet separately for correct packet detection and parameter extraction.

If two packets are apart in the frequency spectrum (*e.g.*, Pkt A in Figure 5(b)), we can easily separate concurrent packets by checking the ranges of detected frequencies. However, such a method may not function well if the frequencies of concurrent packets are close or overlapping each other (*e.g.*, Pkt B and C in Figure 5(b)). To achieve the highest communication concurrency, we expect XGate can correctly receive packets even if the packets have overlapping frequencies.

XGate leverages the signal structures of LoRa preambles to separate concurrent packets in the same chirp slope. The intuition is that the dechirped frequencies of preamble chirps from the same packet (*i.e.*, starting frequencies of preamble chirps) would change across windows at a fixed rate $\Delta f = k_i T_{win}$ that is determined by the chirp slope (k_i) and the length of a detection window (T_{win}). By contrast, the dechirped frequencies of preamble chirps from different packets will not change in Δf across windows (see Figure 5(c)). Based on this observation, XGate checks frequency changes across consecutive windows against Δf to track the preamble of a particular packet. XGate iteratively uses the method to disentangle preamble signals of concurrent packets. As shown in Figure 5(d), even though the frequencies of packets B and C are overlapping, we can separately track preamble signals for the two packets and recover packet configurations from their separated frequency patterns.

To summarize, XGate can detect all concurrent packets in an Rx spectrum in the following two steps: (1) Iterate through all chirp configurations to scan signals of the Rx spectrum for detecting LoRa preambles and packets' meta-information. The concurrent packets in different chirp slopes will be detected separately in different iterations without interfere to each other due to the orthogonality of their chirp signals. (2) For concurrent packets with the same chirp slope,

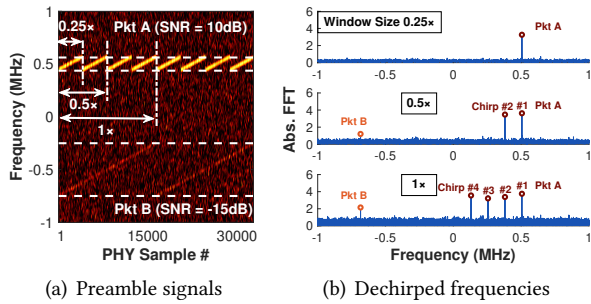


Figure 6: Use long windows to detect weak packets: (a) Received signals of two packets, i.e., Pkt A in (125 kHz, SF8) and Pkt B in (500 kHz, SF12); (b) Detected results with 0.25 \times , 0.5 \times , and 1 \times of the chirp length of Pkt B.

their signal frequencies are simultaneously detected. XGate iterates through all the detected frequencies and tracks the frequency changes across consecutive detection windows to disentangle their colliding preambles for concurrent packet detection and meta-information extraction.

4.3 Enhancing Packet Sensitivity

Although a short detection window is beneficial for avoiding ambiguities of packet parameters, it can suffer *SNR losses* when detecting packets in large SFs because short windows disperse the signal power of a long chirp in multiple windows. Figure 6 displays two packets received from two links differing in SNRs, where Pkt A (SF8) is from a strong link and Pkt B goes through a weak link using SF12. While XGate detects packets with a short window (e.g., 0.25 \times the chirp length of Pkt B), it detects Pkt A correctly but cannot detect Pkt B due to SNR losses caused by short windows as shown in Figure 6(b). We address this problem and enhance XGate to reliably detect both long-chirp and short-chirp packets.

A practical solution is to detect packets using longer windows, which can aggregate power from more samples of a chirp signal to improve detection sensitivity. As shown in Figure 6(b), a weak frequency of Pkt B is detected as window size enlarges from 0.25 \times to 0.5 \times of the chirp length of Pkt B, and the detected frequency becomes stronger with window size 1 \times . However, longer windows do not bring extra power gains for Pkt A. Instead, as window size exceeds the chirp length of Pkt A, multiple chirps of Pkt A are included in the same window and detected as distinctive frequencies (see Figure 6(b)), which can add more interference and cause ambiguity issues to the detection of packet parameters.

How can we deliver high sensitivity while avoiding the side effects of long detection windows (e.g., interference and ambiguity issues)? We notice that although different window sizes produce different frequency results, intrinsic relationships exist among those frequencies. Specifically, for a long-chirp packet (e.g., Pkt B), the detected frequencies remain invariant for different window sizes. Whereas for Pkt A, multiple short chirps are included in a long window.

The frequencies from different chirps are equally spaced by a fixed interval which equals the bandwidth of the chirps, as shown in Figure 6(b).

Based on such observations, XGate uses the diverse detection results of multiple window sizes for mutual verification, which can help resolve interference and ambiguity issues of long-window detection. For instance, after detecting Pkt A with window size 0.25 \times , when we next apply window size 1 \times to the same signals, we assure those four frequencies are indeed from Pkt A (see Figure 6). We use this extra information to manually remove the frequencies of Pkt A from the raw detection results of long windows (i.e., 1 \times) and focus on detecting the frequencies from Pkt B.

In practice, XGate scans an Rx spectrum multiple rounds to reliably detect packets in diverse SNRs and parameter configurations. XGate first detects high-SNR packets with short windows and gradually increases window sizes to search for any weak packets (i.e., usually in large SFs). To eliminate the implications of high-SNR packets on weak packet reception, XGate reuses the information of high-SNR packets detected in previous short-window rounds to facilitate interference cancellation for the detection results of long windows. As a side benefit, in case some packets cannot be reliably detected due to the impacts of interference and noises in a single round (e.g., Pkt B with window size 0.5 \times in Figure 6), XGate can aggregate the detection results of multiple rounds to further improve packet detection, which can effectively reduce false positive and false negative errors.

4.4 Massive Packet Reception

Figure 7 illustrates the general workflow of packet reception in XGate. The front end of RF radio receives signals of a large spectrum and down-converts them to baseband as digitized samples. XGate scans the whole Rx spectrum to detect LoRa packets and their parameters including channel frequencies, bandwidths, and spreading factors. XGate uses the detected parameters to configure software Rx chains to receive packets in parallel. The Rx chains are built and activated on-demand upon detecting new packets. A parameterized Rx chain extracts the signals of a packet from the Rx spectrum, which are finally passed to a standard LoRa decoder to demodulate and decode an incoming packet.

Note that a packet will not get received if it cannot be detected. We handle several practical issues that may impact the performance of packet detection in terms of sensitivity and accuracy, which is critical to the success of XGate.

Detecting large spectrum vs. small spectrum. While XGate scans the whole Rx spectrum to detect packets rather than a specific LoRa channel, a primary concern is whether the sensitivity of packet detection will be compromised since the Rx spectrum can include more noises and interference than a single LoRa channel. Figure 8 compares the signals of the same chirps from a packet received in a 2 MHz spectrum

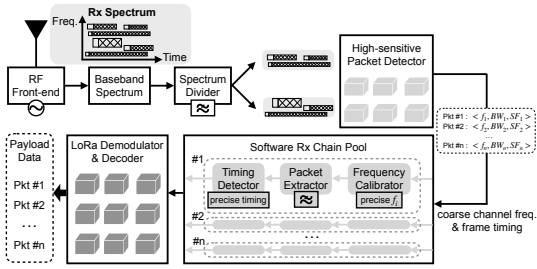


Figure 7: System architecture of XGate: XGate scans all available channels and automatically configures software Rx chains to receive massive LoRa packets.

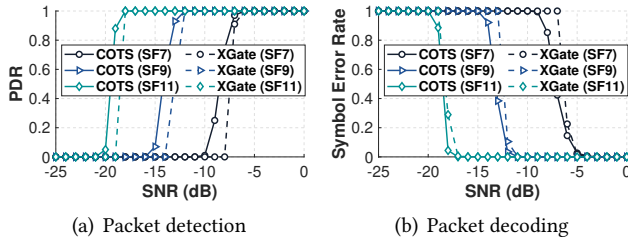


Figure 10: Performance of single packet reception

before and after being filtered by the target channel (250 kHz). As expected, the time-domain signal with 2 MHz spectrum has a higher noise floor than the signal with 250 kHz. Interestingly, we observe little changes of noise floors in the frequency domain after dechirping the two signals. This is because noises are uniformly distributed throughout the entire spectrum. A larger spectrum increases frequency ranges but not the noise strength per frequency bin. As XGate detects packets in the frequency domain, the detection sensitivity will not be affected much by the bandwidth of spectrum. However, we note that a large spectrum may suffer intensified cross-channel interference which can increase packet detection errors for XGate. To reduce interference, XGate partitions a larger Rx spectrum into multiple smaller chunks (*i.e.*, *Spectrum Divider* in Figure 7) and apply packet detectors to each small spectrum separately to detect packets.

Refining packet configuration. Recall that XGate detects the frequency patterns of dechirped signals to restore a packet’s parameter configurations. The SF and BW of a packet are inferred from the periodicity of frequency changes across windows, which has high resilience to errors. However, the central frequency of a packet is coarsely estimated as the average of dechirped frequencies, which is error-prone since the random time offsets between detection windows and chirp signals (*i.e.*, Δt in Eq.(5)) can alter the estimated frequencies. Our measurements show that the coarse estimation of central frequency often deviates from the actual frequency of a packet by 5%~35% of signal bandwidth (see Figure 9), which, if not corrected, can adversely affect the followed packet reception and decoding operations.

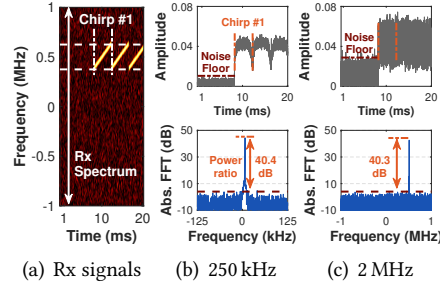


Figure 8: (a) Raw signals in a 2 MHz spectrum; (b,c) Time-domain and frequency-domain views of Chirp #1 under 250 kHz and 2 MHz bands.

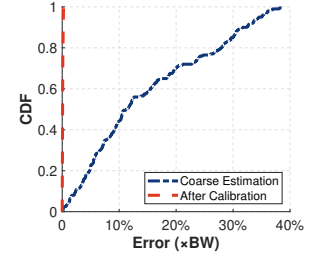


Figure 9: Estimation errors of packets’ central frequencies before and after calibration.

Our insight is that though the estimations of a packet’s channel frequency are not yet accurate enough, the detected meta-information enables us to roughly identify the frame structure of a LoRa packet (*e.g.*, preamble, SFD, *etc.*). Then, by using the signal structure of a LoRa packet, we can use existing time-offset correction methods [12, 46] to mitigate the impacts of misalignment of detection windows. The rationale is that different chirps of the same packet deviate from detection windows in an identical time offset. Yet, this time offset produces opposite frequency effects on preamble up-chirps and SFD down-chirps [12, 46]. We can use two coarsely located chirps from the preamble and SFD parts of a packet, denoted by $C_{pre}(t)$ and $C_{sfd}^{-1}(t)$, to cancel out the effects of time offset as below:

$$h_i e^{j2\pi f_i t} C_{pre}(t + \Delta t_i) \cdot h_i e^{j2\pi f_i t} C_{sfd}^{-1}(t + \Delta t_i) = h_i^2 e^{j2\pi(2f_i)t}, \quad (7)$$

where f_i denotes the central frequency of the i^{th} packet, h_i models the impact of wireless channel, and Δt_i stands for the time offset between detection windows and chirps. The frequency component from the resulting signals of Eq.(7) gives the accurate central frequency of the packet (*i.e.*, f_i).

XGate uses Eq.(7) to refine the detection of channel frequency. The frequency mismatch between up- and down-chirps can be eliminated. As plotted in Figure 9, the refined results achieve almost 100% accuracy. We note that Eq.(7) not only removes the effects of window time offset but also calibrates potential CFOs between sender and receiver radios. After frequency calibrations, a software Rx chain of XGate can correctly filter the interference of unwanted channels and detect precise frame timing from CFO-free signals for reliable packet demodulation and decoding (see Figure 7).

5 EVALUATION

5.1 Methodology

Implementation. We implement XGate on a software defined radio platform (USRP N210) based on the *gr-lora* project [9]. We use USRP to receive signals from LoRa nodes and forward received samples to a workstation running XGate for processing. We use COTS LoRa nodes with Semtech

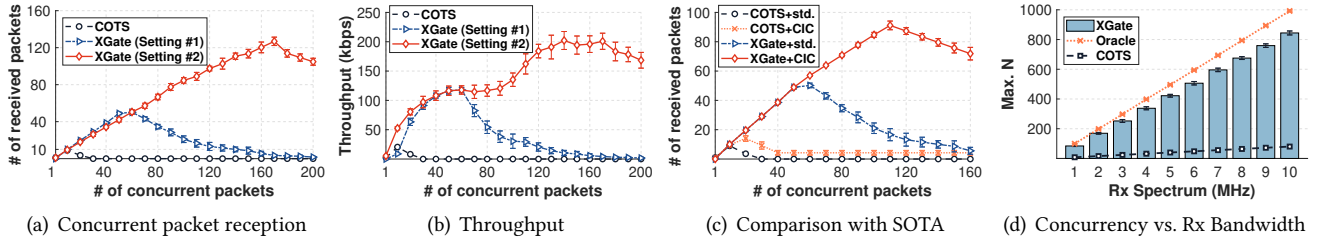


Figure 11: Performance of concurrent reception: (a,b) different numbers of concurrent transmissions in a 1.6 MHz spectrum; (c) comparison with state-of-the-art; (d) concurrency under different spectrum bandwidths.

SX1276 radio [33] as transmitters and use Arduino Uno to configure transmission parameters. We build a testbed consisting of 40 LoRa nodes and six gateways (*i.e.*, four COTS gateways and two USRP gateways). We conduct experiments on our campus spanning $1.08 \text{ km} \times 1.2 \text{ km}$ that represents a typical urban environment as illustrated in Figure 13(a). The gateways are mounted on the roof of a 20-storey building. All nodes operate in the 915 MHz ISM band.

Experiment Setup. We conduct extensive experiments to evaluate XGate, aiming to answer the following questions: (1) How many concurrent LoRa transmissions can be supported by XGate? (§5.2) and (2) How does XGate perform with practical network settings? (§5.3)

We collect over 10,000 packet traces from more than 200 links in our testbed. The collected data covers various channel conditions (*e.g.*, indoor and outdoor, low and high SNRs). All evaluations with <40 nodes are conducted via real-world experiments in the testbed. For performance evaluation of massive concurrent transmissions (*e.g.*, thousands of LoRa nodes), we synthesize the traffic based on the collected traces. Specifically, we add up multiple packet traces with random time offsets to emulate real-world network traffic. The synthesized traces are then replayed to evaluate the performance of XGate when more than 40 nodes are involved.

Baseline. We compare XGate with two baselines: (1) *COTS gateways* [34] receive LoRa packets with multiple Rx chains concurrently. The Rx chains of COTS gateways are pre-configured; all nodes communicate in fixed channels supported by the Rx chains of COTS gateway. (2) *CIC* [36] adopts collision resolving technique to receive concurrent LoRa transmissions in the same LoRa channel.

5.2 Basic Reception Performance

Packet detection & decoding. We first evaluate XGate in detecting and receiving packets from LoRa logical channels. We set up one gateway to receive packets from one LoRa node under different SNR conditions. The Rx spectrum of gateway is 1.6 MHz. The LoRa node randomly chooses a channel (*i.e.*, central frequency) for every packet with a default bandwidth of 125 kHz. We analyzed >100 packets for each SNR condition. XGate detects and receives a packet from the Rx spectrum without knowing the channel or parameter configuration. For comparison, we use a COTS LoRa

gateway, which knows the channels and parameter configurations of all packets in advance.

Figure 10 compares the packet detection ratios and decoding errors of XGate with a COTS gateway. We observe that even without the prior knowledge of channel or parameter configuration, XGate achieves comparable performance in packet detection and decoding to that of a COTS gateway which knows all the meta-information in advance. XGate can reliably detect a packet from unknown logical channels in ultra-low SNRs (*e.g.*, -12.4 dB for SF9 and -18.2 dB for SF11), and use detected parameters to decode incoming packets correctly. Compared to a COTS gateway, the absence of channel information in XGate only leads to 0.3 dB \sim 0.6 dB SNR losses on packet detection and decoding while achieving $>80\%$ packet detection ratio and $<20\%$ symbol error rate.

Gains on concurrency. We next evaluate the performance of XGate in supporting LoRa concurrent transmissions. We compare XGate against a COTS LoRa gateway. We configure both XGate and the COTS gateway to operate within the same Rx spectrum of 1.6 MHz. We control a varying number of LoRa nodes to transmit concurrently on different LoRa channels. The COTS gateway (SX1301) has nine Rx chains that receive at nine frequencies distributed equally in the Rx spectrum with a fixed 125 kHz bandwidth. We adopt two settings of channel partition for XGate. For Setting #1, XGate uses the same channel partition scheme as in the COTS gateway. By varying SF from 7 to 12, 54 LoRa logical channels are supported. Different from COTS gateway detecting and receiving packets on channels known in advance, XGate detects the channel and parameter configurations of packets on-the-fly. For Setting #2, XGate supports all LoRa logical channels in a 1.6 MHz spectrum. LoRa nodes randomly choose packet parameters from those supported by a COTS radio [33]. In particular, we test four bandwidths (*i.e.*, 62.5 kHz, 125 kHz, 250 kHz and 500 kHz) and seven SFs (*i.e.*, 6 \sim 12). The total number of LoRa logical channels is 168.

Figure 11(a) plots the number of received packets when different numbers of nodes transmit concurrently. The number of received packets first increases as more nodes transmit concurrently, then reaches the maximum and starts to drop when the number of concurrent nodes exceeds the number of available channels because of multiple packets colliding in

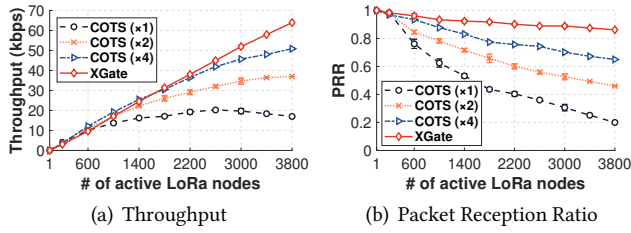


Figure 12: Performance of massive packet reception.

the same channel. As expected, a COTS gateway can receive nine concurrent packets at most, *i.e.*, one packet from each Rx chain. By contrast, XGate (Setting #1) can receive up to 51 concurrent packets, *i.e.*, one packet per logical channel. This number increases to 126 for XGate (Setting #2), *i.e.*, 14 \times higher than the COTS gateway. The performance gain in concurrency is mainly because XGate can support more LoRa logical channels.

Figure 11(b) compares the aggregate network throughput of XGate with a COTS gateway. The highest throughput of a COTS gateway is 20.4 kbps on average when nine packets are transmitted concurrently. XGate can achieve a maximum throughput of 118 kbps in Setting #1 and 200 kbps in Setting #2, which are 5.8 \times and 9.8 \times higher than a COTS gateway.

Comparison with the state-of-the-art. This experiment compares XGate with existing LoRa concurrent transmission strategies. We consider two state-of-the-art (SOTA) strategies (*i.e.*, COTS gateway [34] and CIC [36]) and evaluate four schemes that combine different gateway paradigms and packet decoders, *i.e.*, COTS gateway with a standard LoRa decoder (*COTS+std.*), COTS gateway with a CIC decoder (*COTS+CIC*), XGate (Setting #1) with a standard LoRa decoder (*XGate+std.*), and XGate (Setting #1) with a CIC decoder (*XGate+CIC*). We control various numbers of LoRa nodes to transmit concurrently on different channels using the same method as in the former experiment. When a COTS gateway is used, only nine concurrent channels (*i.e.*, one channel per Rx chain) are available in a 1.6 MHz spectrum, while XGate supports 54 logical channels in the same spectrum.

Figure 11(c) presents the number of received packets of the four schemes under different numbers of concurrent transmissions. As expected, a COTS gateway with the standard LoRa decoder receives nine concurrent packets at most, *i.e.*, one packet per Rx chain. When more than nine nodes transmit concurrently, collisions will occur on some channels, resulting in packet loss. CIC recovers two packets from collisions on average. It increases the maximum concurrency of a COTS gateway to 15 packets. This number is slightly less than that reported in OpenLoRa [26] because the bandwidth of channels in a COTS gateway is 125 kHz, while the bandwidth is 250 kHz in [26]. Unlike CIC, XGate increases concurrency in the dimension of LoRa logical channels. XGate with a standard decoder receives 51 concurrent packets in the same spectrum without collisions, which is 3.4 \times higher than

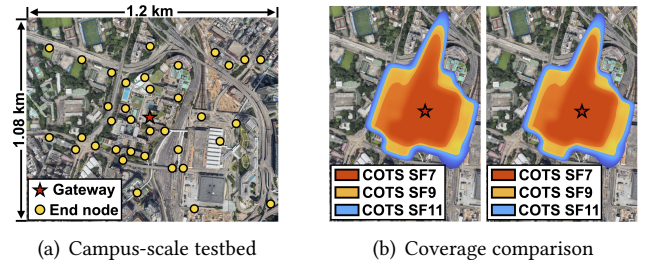


Figure 13: (a) Deployment map of testbed (each dot represents multiple surrounding sites); (b) Coverage comparison between XGate and a COTS gateway.

the concurrency of the compound SOTA scheme (*i.e.*, CIC's parallel decoding atop multiple Rx chains of a COTS gateway). Moreover, XGate can jointly work with CIC to support up to 91 concurrent transmissions (*i.e.*, 78% more than XGate with a standard LoRa decoder) as shown in Figure 11(c).

Scaling with Rx bandwidth. This experiment evaluates the maximum number of concurrent transmissions supported by XGate under different Rx spectrum settings. We increase the Rx spectrum of a gateway from 1 MHz to 10 MHz and set up various LoRa nodes to transmit concurrently using the same method as XGate (Setting #2) in the former experiment. We compare XGate with two benchmarks: (1) COTS gateway that uses dedicated Rx chains to receive pre-configured channels (*i.e.*, eight 125 kHz LoRa channels per 1 MHz); and (2) Oracle method that can divide a given spectrum into the maximum number of logical channels and know the parameters of all packets in advance.

We measure the maximum number of concurrent transmissions and present the results in Figure 11(d). The maximum concurrency increases with the bandwidth of Rx spectrum for all three strategies. The COTS method increases the slowest, and the Oracle increases the fastest. XGate matches closely with Oracle, *i.e.*, within 85%. Specifically, when spectrum bandwidth increases to 10 MHz, XGate receives 844 concurrent packets which is 10.5 \times higher than the COTS gateway method. The results indicate that XGate can support higher concurrency with an increased Rx spectrum.

Massive reception performance. This experiment evaluates XGate in supporting massive IoT communications in practical network settings. In particular, we deploy LoRa gateways in our testbed area (1.08 km \times 1.2 km) to cover up to 3,800 IoT sensors using a 4 MHz ISM spectrum. Each IoT sensor transmits a 20-Byte message every 30 min with a duty cycle $\leq 1\%$. We use a trace-driven method to investigate communications for thousands of sensors. Specifically, we measure LoRa packet profiles from more than 200 sites with SNR ranging from -20 dB to 10 dB and use the collected data to synthesize the signals of received packets with randomly selected link profiles. We consider using a single XGate gateway with a 4 MHz Rx spectrum or various COTS gateways with each covering 1 MHz only. Sensors

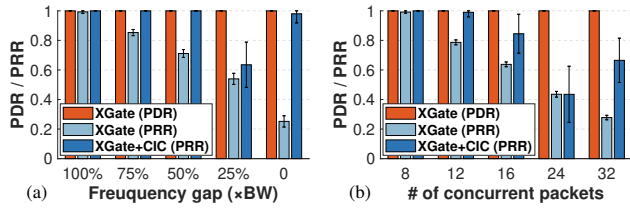


Figure 14: Impacts of cross-channel interference under (a) two packets with different frequency gaps, and (b) different numbers of concurrent packets.

can randomly choose channel frequency (in the 4 MHz spectrum), bandwidth (62.5 kHz, 125 kHz, 250 kHz and 500 kHz) and spreading factor (SF6~12) for every transmission when XGate is used. We use COTS gateways for benchmarking. LoRa nodes select only from the pre-configured channels (*i.e.*, nine channels per COTS gateway) to transmit messages with an ALOHA-based MAC.

Figure 12 compares the aggregated throughput and packet reception ratio (PRR) of XGate and COTS gateways under different numbers of active nodes. We see that the throughput of a COTS gateway first increases and then becomes saturated when the number of nodes exceeds 200. As more nodes transmit actively, it increases the chance of collisions and decreases PRRs (Figure 12(b)). A COTS gateway can only support 400 nodes with PRR>80%. When four COTS gateways are used, up to 1600 nodes can transmit actively with PRR>80%. By contrast, XGate can use fewer gateways to serve more active nodes with even higher PRRs because many LoRa logical channels are used to reduce potential collisions. As shown in Figure 12, a single XGate gateway can support 3,800 active nodes with PRR>85% in the same 4 MHz spectrum, achieving a throughput close to 64 kbps.

Gateway coverage. This experiment compares the coverage of XGate with that of a COTS gateway in an urban environment. To obtain the coverage maps, we divide our testbed area as shown in Figure 13(a) into 40x36 grids. We put LoRa nodes in each grid to send probe packets at 915 MHz with a default bandwidth of 125 kHz and fixed SFs (*i.e.*, 7, 9, and 11). We measure the SNRs of LoRa packets transmitted from each grid and plot a grid in colors if packets from the grid can be reliably received (*e.g.*, PRR>90%). Figure 13(b) visualizes the coverage area of a COTS gateway and XGate with different spreading factors. We see that XGate achieves almost the same coverage as the COTS gateway. Due to the severe signal blockage and power loss in our deployment area, the maximum communication ranges are 286 m, 340 m and 406 m for SF7, SF9 and SF11, respectively. The differences in communication ranges are smaller than 20 m between XGate and a COTS gateway.

5.3 Microbenchmarks

Impact of cross-channel interference. We conduct experiments to evaluate the impacts of cross-channel interference

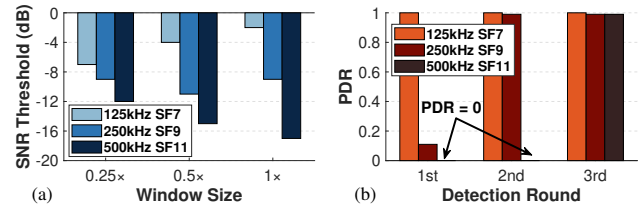


Figure 15: Impact of window size on (a) single packet and (b) concurrent packet detection.

on XGate’s performance. We first study the impact of frequency gaps between concurrent packets. We set up two LoRa nodes transmitting concurrently with the same parameters (BW 125 kHz, SF9). Both nodes have good SNRs (>0 dB). We change the central frequencies of two LoRa nodes and measure the PDR and PRR of XGate under various frequency gaps. As shown in Figure 14(a), XGate can reliably detect all packets in all frequency settings. However, some packets cannot be correctly decoded when concurrent transmissions are overlapped in frequency (*i.e.*, frequency gap < BW). In this case, the interference between overlapped packets can add errors to packet decoding, *i.e.*, similar to packet collisions. Such decoding errors of two concurrent packets can usually be recovered by a parallel LoRa decoder (*e.g.*, CIC), as marked by XGate+CIC in Figure 14(a).

Next, we evaluate XGate under different intensities of cross-channel interference. We change the number of LoRa concurrent transmissions in a fixed 1 MHz spectrum. All nodes have good SNRs and use the same packet configurations (BW 125 kHz, SF9). The central frequencies of LoRa nodes are set uniformly in the 1 MHz spectrum with equal frequency gaps between any two nodes. Namely, the frequencies of packets will overlap while node numbers exceed eight. Figure 14(b) reports the PDR and PRR of XGate with different numbers of concurrent transmissions. Similarly, XGate can always detect all packets correctly but fails to decode some packets when the number of concurrent packets > 8 (*i.e.*, frequency overlapping happens). More concurrent transmissions lead to intensified interference and higher symbol errors that cannot even be recovered by CIC (see Figure 14(b)).

In summary, cross-channel interference can primarily affect packet decoding, while the packet detection method of XGate has high resilience to interference. The reason is that symbols in a preamble have fixed signal patterns yet payload symbols have random signal patterns which are more vulnerable to interference than preamble.

Impact of detection window. This experiment evaluates how the setting of detection windows in XGate affects the performance of packet detection. We set up three LoRa nodes and a gateway in the testbed. The SNRs of links from the three nodes to the gateway are -2 dB, -10 dB, and -16 dB, respectively. The BW and SF configurations of the three nodes are (125 kHz, SF7), (250 kHz, SF9), and (500 kHz, SF11).

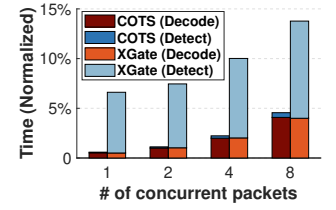
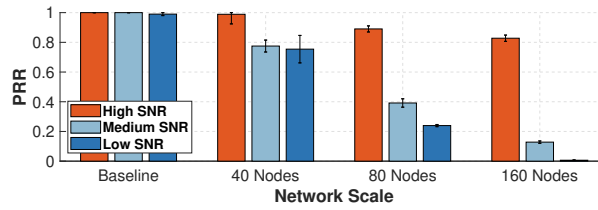
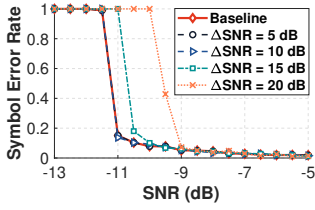


Figure 16: Near-far effect. **Figure 17: Near-far effect at different network scales.** **Figure 18: Time overhead.**

We employ three types of detection windows for XGate, *i.e.*, $0.25\times$, $0.5\times$, and $1\times$ of the chirp length of (500 kHz, SF11).

We first examine the packet detection capability of XGate with different detection windows. We control the three nodes to transmit individually with corresponding BW and SF, and change signal SNRs by varying the transmit power of LoRa nodes. We measure the minimum SNRs (*i.e.*, SNR threshold) of packets that XGate can detect reliably. Figure 15(a) plots the results. We see that XGate generally gives the highest sensitivity when the length of detection windows equals the chirp length of a packet. For instance, the chirp length of a packet in (125 kHz, SF7) is $0.25\times$ of the chirp length of (500 kHz, SF11). XGate, with a detection window of $0.25\times$, detects a packet of (125 kHz, SF7) at the highest sensitivity (*i.e.*, -7 dB). For the packets of (500 kHz, SF11), the SNR sensitivity increases (*e.g.*, from -12 dB to -17 dB) as detection window enlarges from $0.25\times$ to $1\times$ of the chirp length. Whereas for packets in (250 kHz, SF9), the SNR threshold first decreases and then increases. The optimal sensitivity (*i.e.*, -11 dB) is produced by window size $0.5\times$.

Next, we evaluate the full packet detection scheme of XGate, which detects packets in multiple rounds using different windows. We control the three nodes to transmit concurrently on three logical channels. XGate is configured to use three types of windows (*i.e.*, $0.25\times$, $0.5\times$, and $1\times$) to detect packets from all possible logical channels in three rounds. Figure 15(b) plots the packet detection ratios (PDRs) of the three nodes after each round of detection. As expected, the packets of (125 kHz, SF7) are detected with 100 % PDR in the first round, whereas packets from the other two nodes cannot be detected due to poor link qualities. The packets of (250 kHz, SF9), and (500 kHz, SF11) are gradually detected in the second and third rounds, as longer detection windows (*e.g.*, $0.5\times$ and $1\times$) are adopted.

Near-far effect. In practice, concurrent packets may differ in SNRs due to the different distances and heterogeneous link qualities of LoRa nodes (*i.e.*, near-far effects [11]). This experiment investigates the impact of SNR differences between concurrent packets on XGate performance. We set up two LoRa nodes (*i.e.*, one node in higher Tx power and the other in lower power) to transmit concurrently. Both nodes use a default BW 125 kHz. We configure SF7 for the strong node and SF9 for the weak node. In particular, we maintain a fixed SNR difference (*i.e.*, Δ SNR) between the two nodes. We change SNRs of the weak node from -13 dB to

-5 dB and measure Symbol Error Rates (SERs) of both nodes under different SNR conditions (SERs of strong packets are not displayed). We also measure SERs from the same packets of the weak node with the strong node absent and use the results for baseline comparison.

Figure 16 plots the SERs of weak packet reception in various Δ SNR settings. We see that the SER remains the same as the baseline when Δ SNR ≤ 10 dB. It means that XGate can correctly detect and receive weak packets as if no strong packets are present when the SNR difference between concurrent packets is smaller than 10 dB. As the SNR difference increases to 15 dB and 20 dB, though the concurrent packets use orthogonal SFs, a strong packet still increases the background noise power of a weak packet which in turn leads to higher SERs for weak packet reception (see Figure 16).

We next evaluate the near-far effect when more nodes are transmitting simultaneously. We test three network scales with 40, 80, and 120 nodes. All nodes are evenly distributed in the testbed area, connecting to the gateway with SNRs ranging from -5 dB to 5 dB. We set up gateways with the channel plan Setting #2. Nodes select channels and SF settings based on their SNRs, *i.e.*, large (low) SFs are used by nodes with low (high) SNRs. We group nodes into three SNR regimes, *i.e.*, high SNR (2~5 dB), medium SNR (-2 ~2 dB), and low SNR (-5 ~-2 dB). To investigate the near-far effects, we control all nodes to transmit concurrently and measure the packet reception performance of nodes in different SNR regimes. We also measure the packet reception performance of nodes in each SNR regime when they are communicating individually (*i.e.*, without interference from any concurrent nodes) and use the results for baseline comparisons.

Figure 17 illustrates the PRR of three SNR regimes. As expected, the PRRs of nodes in each SNR regime are close to 100 % when the nodes communicate individually. The PRRs of packets in high SNRs drop slightly as concurrent transmissions increase but remain higher than 80 % when 160 nodes transmit concurrently. In contrast, the PRRs of medium- and low-SNR packets drop dramatically as the number of concurrent nodes increases to 80 and 160. The reason is although different nodes transmit packets in different logical channels, the concurrent high-SNR packets increase background noises for the medium- and low-SNR packets coexisting in the same spectrum, which affects the reception performance of medium- and low-SNR packets. The more high-SNR packets, the lower PRRs of medium- and low-SNR packets.

System Name	Multi. logical ch.	# of logical ch.	Concur. per logical ch.	Overall Concur.	Method.
FTrack [47]	✗	1	<2	<2	Collision Recovery (CR) in ONE logical ch.
CoLoRa [42]	✗				
NScale [41]	✗				
CIC [36]	✗	1	2~3	2~3	
COTS [34]	✓	9/1.6MHz	1	9	H/w Rx chains
COTS+CIC	✓	9/1.6MHz	1.67	18~27([26]) 15(Our)	H/w Rx chains & CR
XGate	✓	168/1.6MHz	1	126	Auto-config & S/w Rx chains

Table 1: Comparisons between XGate and SoTA.

Real-time performance. This experiment evaluates the time overheads of XGate. We run XGate on a workstation to process different numbers of concurrent packets (BW 125 kHz, SF9) and measure the time spent on packet detection and decoding. We use the same number of demodulators and decoders for COTS gateway and XGate under each experiment setting. We normalize the processing time to percentages of the air-time of a packet. We also measure the time costs of corresponding operations in a COTS gateway for benchmark comparison. As presented in Figure 18, the costs of packet decoding are comparable for XGate and a COTS gateway. The major overhead of XGate comes from packet detection. In particular, the overall processing time of eight packets in XGate only accounts for <15 % of a packet duration. The result indicates that both packet detection and decoding operations of XGate can be completed in real-time.

6 RELATED WORK

Spectrum sensing. Spectrum sensing and channel occupancy detection [53] aim to identify idle channels for media access and interference avoidance. LoRa uses Channel Activity Detection (CAD) to detect packets [8, 40, 52]. LMAC [8] implements CAD-based carrier-sense multiple access (CSMA) in pre-fixed logical channels. LoRadar [52] leverages the CAD results in a narrow band channel to acquire channel occupancy states of a relatively wide band. Different from existing methods that operate on a pre-configured channel, XGate can sense all logical channels in the Rx spectrum.

Scalability enhancement. There have been significant efforts to improve the scalability of wireless networks (e.g., RFID [16, 28], Wi-Fi [5, 54] and others [21, 22, 43]). For LoRaWANs, PHY layer techniques [6, 26, 36, 41, 42, 44, 46, 47, 49] enable a gateway to resolve collisions. MAC layer methods [8, 11, 20, 31] apply scheduling and chirp design to avoid collisions and improve concurrency. LoRa backscatters [10, 11, 14, 15, 30, 37, 39] create additional LoRa links by shifting LoRa carrier frequencies. Unlike these works, XGate use all logical channels to support massive IoT communications.

Performance enhancement for LoRaWAN. Efforts [18, 24] have been made to improve LoRaWAN regarding throughput [2, 7, 27, 45], weak link reception [4, 13, 19, 25, 40, 50], energy efficiency [23, 48], security [12, 51], etc. Many

works (e.g., Chime [7] and DyLoRa [23]) achieve better performance by optimizing channel parameters. XGate supports flexible transmission parameters and auto-configuration, expanding the solution space for optimizing techniques to further enhance LoRaWAN performance.

7 DISCUSSION

XGate vs. SoTA. Table 1 compares XGate with the state-of-the-art. The concurrency data of FTrack, CoLoRa, NScale, and CIC are from [26]. Different from COTS gateways that use multiple Rx chains to receive concurrent channels, XGate uses a single Rx chain to detect and receive packets from all LoRa logical channels. Existing parallel decoding techniques resolve LoRa packet collisions to achieve concurrent transmissions in one logical channel. Though it is feasible to apply parallel decoding to each Rx channel of a COTS gateway (e.g., COTS+CIC), the overall concurrency remains low (e.g., <30), limited by parallel decoding capability. In contrast, XGate scales concurrency across hundreds of logical channels.

Overheads of XGate. The key advantage of XGate is that it fully covers all LoRa logical channels within an Rx spectrum. The packet reception of XGate can be configured on-the-fly via software, which has better flexibility and supports higher concurrency. XGate incurs higher computational and energy overheads associated with wider bandwidth reception (e.g., MHz-level), detection of concurrent packets and meta-information, etc. We notice XGate suffers slight SNR losses (e.g., near 0.6 dB) due to the lack of packet meta-information. We believe it is worth the trade-off considering the increased flexibility and scalability.

Logical channel selection. XGate unlocks a broad space of channel configurations for LoRa nodes, which allows a node to freely choose channels in an Rx spectrum without negotiations with a gateway (i.e., come-and-be-served). XGate automatically detects the parameters of an incoming packet and configures software Rx chains on-the-fly to receive the packet. In practice, a node can select channels properly based on the link states and workloads of logical channels.

8 CONCLUSION

This paper presents XGate, a novel gateway paradigm for LoRaWAN that revolutionizes the way packets are detected and received. Unlike conventional approaches, XGate utilizes a single Rx chain to efficiently scan the entire Rx spectrum, detect incoming packets across thousands of logical channels, and dynamically configure resources on-the-fly, such as software-controlled Rx chains and packet demodulators, to receive all detected packets. This breakthrough enables scalable LoRa concurrent transmissions across all available logical channels, fundamentally breaking the barrier of massive LoRa communications. Importantly, XGate opens up new possibilities for LoRaWAN communication and networking paradigms, facilitating dynamic channel configuration, data-rate adaptation, spectrum sharing, and more.

ACKNOWLEDGMENTS

We sincerely thank the anonymous shepherd and reviewers for their comments and feedback. This work is supported in part by HK GRF (Grant No. 15218022 and 15206123), NSFC (Grant No. 62102336), and the Innovation Capability Support Program of Shaanxi (No. 2023-CX-TD-08) Shaanxi Qinchuangyuan "scientists+engineers" team (No. 2023KXJ-040). Xianjin Xia and Yuanqing Zheng are the corresponding authors.

REFERENCES

- [1] LoRa Alliance. 2022. *Lorawan Specification*. "https://loro-alliance.org/about-lorawan".
- [2] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. 2020. A Cloud-Optimized Link Layer for Low-Power Wide-Area Networks. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services* (Toronto, Ontario, Canada) (*MobiSys '20*). Association for Computing Machinery, New York, NY, USA, 247–259. <https://doi.org/10.1145/3386901.3388915>
- [3] Yao Cheng, Hendra Saputra, Leng Meng Goh, and Yongdong Wu. 2018. Secure smart metering based on LoRa technology. In *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. 1–8. <https://doi.org/10.1109/ISBA.2018.8311466>
- [4] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony 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)*. IEEE, 60–71.
- [5] Manideep Dunna, Miao Meng, Po-Han Wang, Chi Zhang, Patrick P Mercier, and Dinesh Bharadia. 2021. SyncScatter: Enabling WiFi like synchronization and range for WiFi backscatter Communication.. In *NSDI*. 923–937.
- [6] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağın. 2017. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17)*. 309–321.
- [7] Akshay Gadre, Revathy Narayanan, Anh Luong, Anthony Rowe, Bob Iannucci, and Swarun Kumar. 2020. Frequency Configuration for Low-Power Wide-Area Networks in a Heartbeat.. In *NSDI*. 339–352.
- [8] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. 2020. LMAC: Efficient Carrier-Sense Multiple Access for LoRa. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, New York, NY, USA, Article 43, 13 pages.
- [9] Gr-LoRa GitHub community. 2021. *gr-lora projects*. "https://github.com/rpp0/gr-lora".
- [10] Xiuzhen Guo, Longfei Shangguan, Yuan He, Nan Jing, Jiacheng Zhang, Haotian Jiang, and Yunhao Liu. 2022. Saiyan: Design and implementation of a low-power demodulator for LoRa backscatter systems. In *Proc. USENIX NSDI*. 437–451.
- [11] Mehrdad Hesar, Ali Najafi, et al. 2019. Netscatter: Enabling large-scale backscatter networks. In *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI'19)*.
- [12] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. 2021. Jamming of lora phy and countermeasure. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.
- [13] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. 2023. Don't Miss Weak Packets: Boosting LoRa Reception with Antenna Diversities. *ACM Trans. Sen. Netw.* 19, 2, Article 41 (feb 2023), 25 pages. <https://doi.org/10.1145/3563698>
- [14] Jinyan Jiang, Zhenqiang Xu, Fan Dang, and Jiliang Wang. 2021. Long-range ambient LoRa backscatter with parallel decoding. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 684–696.
- [15] Mohamad Katanbaf, Anthony Weinand, and Vamsi Talla. 2021. Simplifying backscatter deployment: {Full-Duplex} {LoRa} backscatter. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. 955–972.
- [16] Linghe Kong, Liang He, Yu Gu, Min-You Wu, and Tian He. 2014. A parallel identification protocol for RFID systems. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 154–162.
- [17] Preti Kumari, Rahul Mishra, Hari Prabhat Gupta, Tanima Dutta, and Sajal K Das. 2021. An energy efficient smart metering system using edge computing in LoRa network. *IEEE Transactions on Sustainable Computing* 7, 4 (2021), 786–798.
- [18] Chenning Li and Zhichao Cao. 2022. Lora networking techniques for large-scale and long-term iot: A down-to-top survey. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–36.
- [19] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. 2021. NELoRa: Towards Ultra-Low SNR LoRa Communication with Neural-Enhanced Demodulation. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (Coimbra, Portugal) (SenSys '21)*. Association for Computing Machinery, New York, NY, USA, 56–68. <https://doi.org/10.1145/3485730.3485928>
- [20] Chenning Li, Xiuzhen Guo, Longfei Shangguan, Zhichao Cao, and Kyle Jamieson. 2022. CurvingLoRa to Boost LoRa Network Throughput via Concurrent Transmission. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. USENIX Association, Renton, WA, 879–895.
- [21] Songfan Li, Hui Zheng, Chong Zhang, Yihang Song, Shen Yang, Minghua Chen, Li Lu, and Mo Li. 2022. Passive {DSSS}: Empowering the Downlink Communication for Backscatter Systems. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. 913–928.
- [22] Yinyi Li, Chao Feng, Xiaojing Wang, Yangfan Zhang, Yaxiong Xie, and Xiaojiang Chen. 2023. {RF-Bouncer}: A Programmable Dual-band Metasurface for Sub-6 Wireless Networks. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 389–404.
- [23] Yinghui Li, Jing Yang, and Jiliang Wang. 2020. DyLoRa: Towards Energy Efficient Dynamic LoRa Transmission Control. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 2312–2320. <https://doi.org/10.1109/INFOCOM41043.2020.9155407>
- [24] 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 Transactions on Sensor Networks (TOSN)* 15, 2 (2019), 1–35.
- [25] Jun Liu, Weitao Xu, Sanjay Jha, and Wen Hu. 2020. Nepalai: towards LPWAN C-RAN with physical layer compression. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–12.
- [26] Manan Mishra, Daniel Koch, Muhammad Osama Shahid, Bhuvana Krishnaswamy, Krishna Chintalapudi, and Suman Banerjee. 2023. {OpenLoRa}: Validating {LoRa} Implementations through an Extensible and Open-sourced Framework. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1165–1183.
- [27] Di Mu, Yitian Chen, Junyang Shi, and Mo Sha. 2020. Runtime Control of LoRa Spreading Factor for Campus Shuttle Monitoring. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. 1–11. <https://doi.org/10.1109/ICNP49622.2020.9259383>
- [28] Jiajue Ou, Mo Li, and Yuanqing Zheng. 2015. Come and be served: Parallel decoding for COTS RFID tags. In *Proceedings of the 21st annual international conference on mobile computing and networking*. 500–511.

- [29] Antonino Pagano, Daniele Croce, Ilenia Tinnirello, and Gianpaolo Vitale. 2022. A Survey on LoRa for Smart Agriculture: Current Trends and Future Perspectives. *IEEE Internet of Things Journal* (2022).
- [30] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. 2018. PLoRa: A passive long-range data network from ambient LoRa transmissions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 147–160.
- [31] Yaman Sangar and Bhuvana Krishnaswamy. 2020. WiChronos: energy-efficient modulation for long-range, large-scale wireless networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.
- [32] Semtech. 2022. *New ABI Research White Paper*. "https://www.semtech.com/company/press/new-abi-research-white-paper-highlights-growth-of-lora-and-the-lorawan-open-protocol".
- [33] Semtech. 2022. *SX1276*. "https://www.semtech.com/products/wireless-rf/lora-connect/sx1276".
- [34] Semtech. 2022. *SX1301*. "https://www.semtech.com/products/wireless-rf/lora-core/sx1301".
- [35] Semtech. 2023. *LoRa gateway chips and reference designs*. "https://www.semtech.com/products/wireless-rf/lora-core".
- [36] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. 2021. Concurrent Interference Cancellation: Decoding Multi-Packet Collisions in LoRa (SIGCOMM '21). Association for Computing Machinery, New York, NY, USA.
- [37] Yihang Song, Li Lu, Jiliang Wang, Chong Zhang, Hui Zheng, Shen Yang, Jingsong Han, and Jian Li. 2023. μ Mote: Enabling Passive Chirp De-spreading and μ W-level Long-Range Downlink for Backscatter Devices. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association.
- [38] SJ Suji Prasad, M Thangatamilan, M Suresh, Hitesh Panchal, Christopher Asir Rajan, C Sagana, B Gunapriya, Aditi Sharma, Tusharkumar Panchal, and Kishor Kumar Sadasivuni. 2022. An efficient LoRa-based smart agriculture management and monitoring system using wireless sensor networks. *International Journal of Ambient Energy* 43, 1 (2022), 5447–5450.
- [39] Vamsi Talla, Mehrdad Hesar, Bryce Kellogg, Ali Najafi, Joshua R Smith, and Shyamnath Gollakota. 2017. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1, 3 (2017), 1–24.
- [40] Shuai Tong, Zilin Shen, Yunhao Liu, and Jiliang Wang. 2021. Combating Link Dynamics for Reliable Lora Connection in Urban Settings. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (New Orleans, Louisiana) (MobiCom '21)*. Association for Computing Machinery, New York, NY, USA, 642–655. <https://doi.org/10.1145/3447993.3483250>
- [41] Shuai Tong, Jiliang Wang, and Yunhao Liu. 2020. Combating packet collisions using non-stationary signal scaling in LPWANs. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. ACM, Toronto Ontario Canada, 234–246.
- [42] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. 2020. CoLoRa: Enabling Multi-Packet Reception in LoRa. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 2303–2311. ISSN: 2641-9874.
- [43] Anran Wang, Vikram Iyer, Vamsi Talla, Joshua R Smith, and Shyamnath Gollakota. 2017. FM Backscatter: Enabling Connected Cities and Smart Fabrics.. In *NSDI*, Vol. 17. 3154630–3154650.
- [44] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. 2019. mlora: A multi-packet reception protocol in lora networks. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 1–11.
- [45] Xianjin Xia, Qianwu Chen, Ningning Hou, and Yuanqing Zheng. 2023. HyLink: Towards High Throughput LPWANs with LoRa Compatible Communication. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (Boston, Massachusetts) (SenSys '22)*. Association for Computing Machinery, New York, NY, USA, 578–591. <https://doi.org/10.1145/3560905.3568516>
- [46] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. 2021. PCube: scaling LoRa concurrent transmissions with reception diversities. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 670–683.
- [47] 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*. ACM, New York New York, 192–204.
- [48] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2021. Litenap: Downclocking lora reception. *IEEE/ACM Transactions on Networking* 29, 6 (2021), 2632–2645.
- [49] Zhenqiang Xu, Pengjin Xie, and Jiliang Wang. 2021. Pyramid: Real-time lora collision decoding with peak tracking. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–9.
- [50] Zhenqiang Xu, Pengjin Xie, Jiliang Wang, and Yunhao Liu. 2022. Ostinato: Combating LoRa Weak Links in Real Deployments. In *2022 IEEE 30th International Conference on Network Protocols (ICNP)*. 1–11. <https://doi.org/10.1109/ICNP55882.2022.9940369>
- [51] Huanqi Yang, Hongbo Liu, Chengwen Luo, Yuezhong Wu, Wei Li, Albert Y Zomaya, Linqi Song, and Weitao Xu. 2022. Vehicle-key: A secret key establishment scheme for LoRa-enabled IoT communications. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 787–797.
- [52] Fu Yu, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2022. LoRadar: An Efficient LoRa Channel Occupancy Acquirer based on Cross-channel Scanning. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. 540–549. <https://doi.org/10.1109/INFOCOM48880.2022.9796845>
- [53] Tefik Yucek and Huseyin Arslan. 2009. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE communications surveys & tutorials* 11, 1 (2009), 116–130.
- [54] Renjie Zhao, Fengyuan Zhu, Yuda Feng, Siyuan Peng, Xiaohua Tian, Hui Yu, and Xinbing Wang. 2019. OFDMA-enabled Wi-Fi backscatter. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–15.