Online Concurrent Transmissions at LoRa Gateway

Zhe Wang*, Linghe Kong*, Kangjie Xu*, Liang He†, Kaishun Wu‡, Guihai Chen*
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
†Department of Computer Science and Engineering, University of Colorado Denver, USA
‡College of Computer Science and Software Engineering, Shenzhen University, China
{wang-zhe,linghe.kong,xkjac04452017}@sjtu.edu.cn, liang.he@ucdenver.edu,wu@szu.edu.cn, gchen@cs.sjtu.edu.cn

***Abstract*—Long Range (LoRa) communication, thanks to its wide network coverage and low energy operation, has attracted extensive attentions from both academia and industry. However, existing LoRa-based Wide Area Network (LoRaWAN) suffers from severe inter-network interference, due to the following two reasons. First, the densely-deployed LoRa ends usually share the same network configurations, such as spreading factor (SF), bandwidth (BW) and carrier frequency (CF), causing interference when operating in the vicinity. Second, LoRa is tailored for low-power devices, which excludes LoRaWAN from using the listen-before-talk (LBT) mechanisms commonly used in wireless communication technologies, such as WiFi and ZigBee — LoRaWAN has to use the duty-cycled medium access policy and thus being incapable of channel sensing or collision avoidance. To mitigate the inter-network interference, we propose a novel solution achieving the online concurrent transmissions at LoRa gateway, called OCT, which recovers collided packets at the gateway and thus improves LoRaWAN's throughput. Moreover, OCT achieves the online concurrent transmission using only LoRa's (de)modulation information, thus can be easily deployed at LoRa gateway. We have implemented and evaluated OCT on USRP platform and commodity LoRa ends, showing OCT achieves: (i) >90% packet reception rate (PRR), (ii) 3×10^{-3} bit error rate (BER), (iii) 2x and 3x throughput in the scenarios of two- and three- packet collisions respectively, and (iv) reducing 67% latency compared with state-of-the-art.**

I. Introduction

The number of active Internet of Things (IoT) devices is expected to reach 10 billion by 2020 and 22 billion by 2025 [1], to support applications such as smart home [2], intelligent transportation [3], etc. Such large-scale and wide-range IoT deployments incur new demands for long-range and low-power communication technologies. Low-Power-Wide-Area-Network (LPWAN) is ideal for such long-range connections and offers excellent coverage with low-power operation, among which LoRa has attracted extensive attentions from both academia [4]–[6] and industry [7], [8] — the LoRa-based LPWAN, i.e., LoRaWAN [9], has been widely deployed in smart cities, smart environment monitoring and intelligent agriculture/industry [4], [8].

However, existing LoRaWANs suffer from severe inter-network interference. First, LoRa ends usually share the same network configuration, such as spreading factor (SF), bandwidth (BW) and carrier frequency (CF), causing interference when operating in the vicinity. Second, since LoRa is tailored for low-power or battery-powered devices, LoRaWAN exclusively uses the duty-cycle access policy rather than the listen-before-talk (LBT) mechanism, making LoRaWAN incapable of channel sensing or collisions avoidance. This, together with LoRaWAN's one-hop star topology, cause collisions at the LoRa gateway when multiple LoRa ends send packet simultaneously with identical network configuration.

To mitigate the inter-network interference and improve LoRaWAN's throughput, several works [10]–[12] have tried to enable LoRa's concurrent transmissions via (semi-)offline collision decoding, but have two inherent issues. First, these methods decode the collided packets in an (semi-)offline way, i.e., demodulating the collided packets after all their samples have been received, thus incurring significant latency to LoRa communication — a large latency easily causes timeout of reception window at LoRa ends, leading to miss of messages from the gateway [9], [13]. This is because achieving online concurrent transmissions requires to detect, synchronize and demodulate the collided packets in parallel, which are absent in existing methods — it is nontrivial to achieve online concurrent transmissions. Second, the large memory cost for storing the incoming signals becomes bottleneck of offline collision decoding when deploying at gateway because the available memory of gateway is limited due to its complex functionalities [14].

To fill this need, we propose a novel solution achieving the online concurrent transmissions at LoRa gateway, called OCT, enabling LoRa gateway to receive (and recover) multiple collided packets from different LoRa ends simultaneously. OCT achieves the online concurrent transmissions at LoRa gateway with three steps: preamble detection, start-of-frame-delimiter (SFD) detection, and packet decoding, all of which use only LoR's (de)modulation information and thus making OCT easily deployable at LoRa gateway. OCT works as follows: (i) detecting incoming packets by identifying the preamble from incoming signals; (ii) aligning collided packets by detecting their respective SFDs, after which the time offset (i.e., the time delay when compared with the first arrival packet) and power offset (i.e., the difference of signal power at the receiver) are determined; (iii) demodulating the overlapped symbols, and then decoding the collided packets, using the identified time/power offsets.

We have implemented and evaluated OCT using USRP N210-based platform and 3 commodity RN2483-based LoRa ends. The experimental results show OCT achieves: >90% PRR, 3×10^{-3} BER, 2x and 3x throughput in the scenarios of two- and three- packet collisions respectively, and reducing 67% latency compared with state-of-the-art mLoRa.

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.

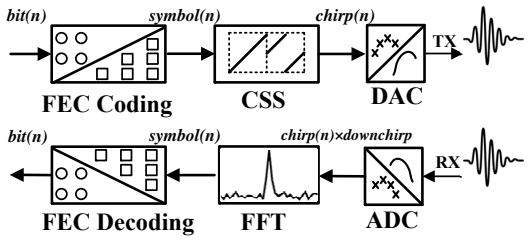


Fig. 1: Diagram of LoRa's PHY layer.

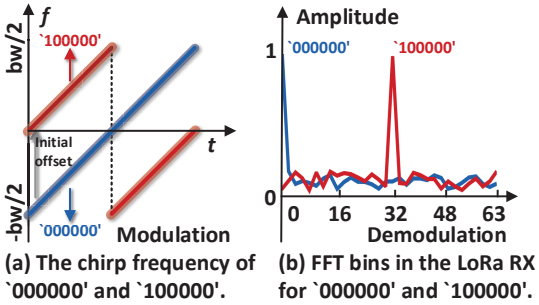


Fig. 2: (De)Modulation of LoRa.

The major contribution of this paper are summarized as follows:

- Analytically and empirically uncovering the feasibility of concurrent LoRa transmission.
- Designing a novel solution called OCT achieving the online concurrent transmissions at LoRa gateway, which can be easily deployed at LoRa gateway.
- Implementation and evaluation of OCT on USRP platform and commodity LoRa ends.

II. Preliminary

A. PHY Layer of LoRa

LoRa achieves long-range communication and enhances its anti-interference ability by integrating chirp-spreading-spectrum (CSS) modulation [15] with channel coding [16]. Fig. 1 illustrates the workflow of LoRa's TX/RX. At the TX, the incoming bits streams from upper layer are encoded first into redundant LoRa symbols by channel coding [17], [18]. The LoRa symbols are then modulated into chirps, converted into analog baseband signal by the DAC, and sent out at carrier frequency. To receive a LoRa packet, the RX down-converts the received signal into baseband signal and samples it with the ADC. After that, the LoRa chirps are demodulated with two steps: (i) multiplying with a predefined standard downchirp, and (ii) performing Fast Fourier Transform (FFT) on the resultant product. The LoRa symbol is identified as the index of the max FFT bin. At last, the decoder transforms the LoRa symbol into original bits by reverting the channel coding process.

LoRa's PHY layer is characterised by several parameters including BW, SF and coding rate (CR). According to LoRa$^{\text{TM}}$ modem specification [19], LoRa ends support 3 different BWs (125/250/500KHz) and 7 SFs ranging from 6–12. The hamming coding rate is controlled by CR which is typically 4/[5,8]. Specifically, LoRa spreads every $\mathbf{n}$-bit symbol into a

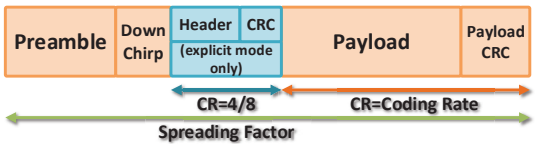


Fig. 3: Structure of LoRa packet.

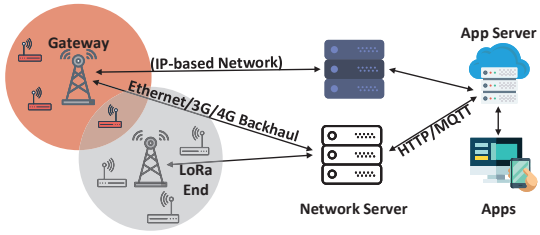


Fig. 4: LoRaWAN architecture.

chirp of $2^{\mathbf{n}}$ samples with a BW of $\mathbf{bw}$, where $\mathbf{n}$ is the SF, thus resulting in a theoretic data rate of $\frac{\mathbf{n}\cdot\mathbf{bw}}{2^{\mathbf{n}}}$.

Let us consider the LoRa symbol '000000'/'0' and '100000'/'32' in Fig. 2 as an example. Given LoRa symbol $\mathbf{s}$, the initial frequency $\mathbf{f}(0)$ has an offset of $\frac{\mathbf{bw}\cdot\mathbf{s}}{2^{\mathbf{n}}}$, when compared with the standard upchirp whose initial frequency is $-\frac{\mathbf{bw}}{2}$, as shown in Fig. 2a. With the frequency sweeping of CSS, the chirp frequency can be expressed as:

$$\mathbf{f}(t) = \begin{cases} -\frac{\mathbf{bw}}{2} + \frac{\mathbf{bw}\cdot\mathbf{s}}{2^{\mathbf{n}}} + \frac{\mathbf{bw}^2}{2^{\mathbf{n}}}t, & 0 \leqslant t < \frac{2^{\mathbf{n}}-\mathbf{s}}{\mathbf{bw}}, \\ -\frac{3\cdot\mathbf{bw}}{2} + \frac{\mathbf{bw}\cdot\mathbf{s}}{2^{\mathbf{n}}} + \frac{\mathbf{bw}^2}{2^{\mathbf{n}}}t, & \frac{2^{\mathbf{n}}-\mathbf{s}}{\mathbf{bw}} \leqslant t < \frac{2^{\mathbf{n}}}{\mathbf{bw}}, \end{cases} \quad (1)$$

where $\mathbf{f}(t)$ is the chirp's instantaneous frequency at time t. When demodulating at the RX, the received chirp signal is multiplied with a predefined downchirp which sweeps through the band from $\frac{\mathbf{bw}}{2}$ to $-\frac{\mathbf{bw}}{2}$, and thus the signal frequency will concentrate at $\frac{\mathbf{bw}\cdot\mathbf{s}}{2^{\mathbf{n}}}$. Applying FFT on the resultant product, we can observe a peak at the corresponding frequency bin (see Fig. 2b). The LoRa symbols are thus demodulated by detecting the index of the max FFT bin for every $2^{\mathbf{n}}$ samples.

Fig. 3 shows LoRa's packet structure. The LoRa frame begins with constant upchirp sequences (i.e., preamble), consisting of 8 standard upchirps followed by 2 chirps of sync word. The preamble is followed by 2.25 downchirps as SFD.

B. LoRaWAN

As the most widely used specification in MAC layer of LoRa [9], LoRaWAN defines a LoRa-based network architecture and regulates its frequency band, access policy, and transmitting power.

- **Network Architecture.** Fig. 4 shows the typical LoRaWAN architecture, consisting of 3 core components: LoRa end, LoRa gateway and network server. LoRaWAN uses a one-hop star topology, i.e., a LoRa gateway serves a group of LoRa ends within the range, and LoRa ends communicate with the gateway directly. As a result, collisions at the gateway will occur when multiple LoRa ends send packets simultaneously using the same SF, BW and CF, which is inevitable in dense LoRaWAN due to the limited combinations of network parameters. Also, the fact that most LoRa ends simply use the default

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.

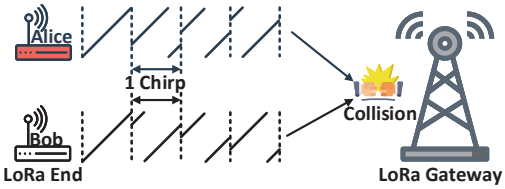


Fig. 5: Example of LoRa collision.

configuration (i.e., SF=7, BW=125KHz) [20] magnifies further the chance of collisions.

- **Frequency Regulation.** LoRaWANs are usually deployed in unlicensed Industrial Scientific Medical (ISM) band, e.g., 868/433MHz in Europe and 915/433MHz in USA. The use of unlicensed band facilitates LoRaWAN's deployment, but at the cost of intensified collisions — the limited channels of ISM band make collisions inevitable.
- **Access Policy.** The latest LoRaWAN specification [21] exclusively uses the duty-cycled access policy rather than the LBT mechanism to reduce the power consumption. As a result, LoRa ends can access the channel anytime according to their respective duty cycle, causing collisions in densely deployed LoRaWANs and frequent packet retransmissions thereof.
- **Transmitting Power.** LoRa radio usually uses a transmitting power of $[-4, 14]$dBm [19], [22]. The potentially different transmitting power, together with other factors such as communication distance and blocking, leading to the fact that collided packets usually have different signal power. We define *power offset* as the difference of signal powers between a pair of collided packets at the RX.

C. Basic Idea of OCT

Understanding the basics of LoRa's PHY layer and LoRaWAN, we next present the basic idea of OCT's collision decoding, using a typical LoRa collision example shown in Fig. 5: two LoRa TXs Alice (A) and Bob (B) send packets simultaneously using an identical network configuration, causing collisions at the gateway. Collided packets usually arrives at the RX with certain time offset. Such packet-level time offset consists, in general, several chirps cycles and the chirp-level time offset less than one chirp cycle (see Fig. 6a). Note sometimes the chirp-level time offset may be absent as shown in Fig. 6b, or there is no packet-level time offset at all, i.e., the collided packets are perfectly aligned in time.

For simplicity, let us first consider the case without chirp-level time offset. Denoting $\mathbf{s}_a$ and $\mathbf{s}_b$ as the symbols sent by A and B, the superposed chirps at the RX are aligned perfectly as shown in Fig. 6b. Given the initial phase of 0 and the linearly varying frequency in Eq. 1, the time domain signal $\mathbf{X}_a$ of $\mathbf{s}_a$ is

$$\mathbf{X}_a(t) = e^{j2\pi\boldsymbol{\theta}_a(t)}, \tag{2}$$

where $\boldsymbol{\theta}_a(t)$ is the integral of $\mathbf{f}_a$ over the interval $[0, t]$:

$$\boldsymbol{\theta}_a(t) = \begin{cases} (-\frac{\mathbf{bw}}{2} + \frac{\mathbf{bw}\cdot\mathbf{s}_a}{2^\mathbf{n}})t + \frac{\mathbf{bw}^2}{2^{\mathbf{n}+1}}t^2, & 0 \leqslant t < \frac{2^\mathbf{n}-\mathbf{s}_a}{\mathbf{bw}}, \\ (-\frac{3\cdot\mathbf{bw}}{2} + \frac{\mathbf{bw}\cdot\mathbf{s}_a}{2^\mathbf{n}})t + \frac{\mathbf{bw}^2}{2^{\mathbf{n}+1}}t^2, & \frac{2^\mathbf{n}-\mathbf{s}_a}{\mathbf{bw}} \leqslant t < \frac{2^\mathbf{n}}{\mathbf{bw}}. \end{cases} \tag{3}$$

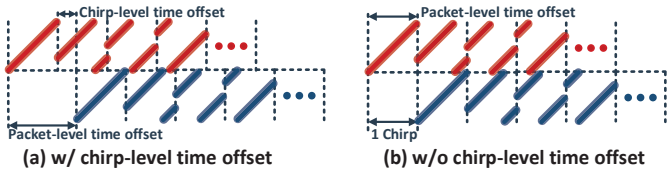


Fig. 6: Collision examples w/ and w/o chirp-level time offset.

Note the constant term $(2^n - s_a)$ is omitted in the second line for simplicity. The superposed chirp signal $\mathbf{Y}(t)$ at the RX is

$$\mathbf{Y}(t) = \mathbf{H}_a\mathbf{X}_a(t) + \mathbf{H}_b\mathbf{X}_b(t) + \mathbf{w}(t), 0 \leqslant t < \frac{2^\mathbf{n}}{\mathbf{bw}}, \tag{4}$$

where $\mathbf{H}_{a/b} = |\mathbf{H}_{a/b}|e^{j\Phi_{a/b}}$ is the channel parameter and $\mathbf{w}(t)$ is the noise.

Similarly, the time domain signal $\mathbf{X}_d$ of predefined downchirp is:

$$\mathbf{X}_d(t) = e^{j2\pi\boldsymbol{\theta}_d(t)}, \tag{5}$$

where $\boldsymbol{\theta}_d(t)$ equals to:

$$\boldsymbol{\theta}_d(t) = \frac{\mathbf{bw}}{2}t - \frac{\mathbf{bw}^2}{2^{\mathbf{n}+1}}t^2, 0 \leqslant t < \frac{2^\mathbf{n}}{\mathbf{bw}}. \tag{6}$$

At the RX, the received chirp signal is multiplied with the predefined downchirp. The resultant product $\mathbf{Y}_d(t)$ is

$$\begin{aligned} \mathbf{Y}_d(t) &= \mathbf{Y}(t)\mathbf{X}_d(t) \\ &= \mathbf{H}_a\mathbf{X}_a(t)\mathbf{X}_d(t) + \mathbf{H}_b\mathbf{X}_b(t)\mathbf{X}_d(t) \\ &= \mathbf{H}_a e^{j2\pi(\boldsymbol{\theta}_a(t)+\boldsymbol{\theta}_d(t))} + \mathbf{H}_b e^{j2\pi(\boldsymbol{\theta}_b(t)+\boldsymbol{\theta}_d(t))}, \end{aligned} \tag{7}$$

where the noise term is, again, omitted for simplicity.

Substituting Eq. 3 and Eq. 6 into Eq. 7, we get:

$$\mathbf{Y}_d(t) = \begin{cases} e^{j2\pi\frac{\mathbf{bw}\cdot\mathbf{s}_a}{2^\mathbf{n}}t} + e^{j2\pi\frac{\mathbf{bw}\cdot\mathbf{s}_b}{2^\mathbf{n}}t}, \\ \qquad 0 \leqslant t < \frac{2^\mathbf{n}-\mathbf{s}_a}{\mathbf{bw}}, \\ e^{j2\pi(\frac{\mathbf{bw}\cdot\mathbf{s}_a}{2^\mathbf{n}}-\mathbf{bw})t} + e^{j2\pi\frac{\mathbf{bw}\cdot\mathbf{s}_b}{2^\mathbf{n}}t}, \\ \qquad \frac{2^\mathbf{n}-\mathbf{s}_a}{\mathbf{bw}} \leqslant t < \frac{2^\mathbf{n}-\mathbf{s}_b}{\mathbf{bw}} \\ e^{j2\pi(\frac{\mathbf{bw}\cdot\mathbf{s}_a}{2^\mathbf{n}}-\mathbf{bw})t} + e^{j2\pi(\frac{\mathbf{bw}\cdot\mathbf{s}_b}{2^\mathbf{n}}-\mathbf{bw})t}, \\ \qquad \frac{2^\mathbf{n}-\mathbf{s}_b}{\mathbf{bw}} \leqslant t < \frac{2^\mathbf{n}}{\mathbf{bw}}, \end{cases} \tag{8}$$

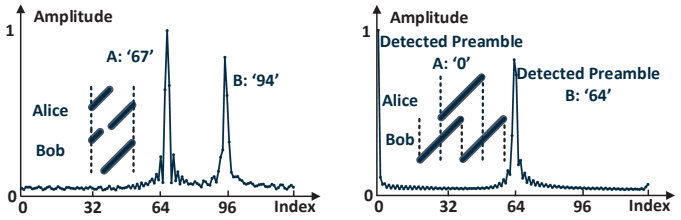
where $\mathbf{s}_a > \mathbf{s}_b$ and the channel parameter is omitted. Note that Eq. 8 focuses on the characteristics of collided signal itself and discards the negligible impact of noise and channel parameter, in view of the fact that LoRa's channeling coding and CSS (de)modulation make it robust to noise, Doppler and multi-path effect [4], [23].

Considering symbol $\mathbf{s}_a$ and $\mathbf{s}_b$ independently, the chirp signal frequency of A or B, after multiplied with the standard downchirp, only has two values with a difference of $\mathbf{bw}$. The instantaneous frequency thus becomes continuous over the whole chirp band, because the collided signal is also sampled with a rate of $\mathbf{bw}$ [24]. As a result, the collided signals, after being multiplied with the downchirp, is the superposition of two different frequency signals:

$$\mathbf{Y}_d(t) = e^{j2\pi\frac{\mathbf{bw}\cdot\mathbf{s}_a}{2^\mathbf{n}}t} + e^{j2\pi\frac{\mathbf{bw}\cdot\mathbf{s}_b}{2^\mathbf{n}}t}. \tag{9}$$

According to the linearity of Fourier Transform, we can detect two peaks after performing FFT on the product result

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.



(a) The FFT result of two collided symbols '67' and '94' w/o chirp-level time offset

(b) The FFT result of two collided preamble chirps w/ chirp-level time offset about half-chirp

Fig. 7: Two examples of FFT result on collided signals.

$\mathbf{Y}_d$, which correspond to the signal energy of $\mathbf{s}_a$ and $\mathbf{s}_b$, respectively — implying the possibility of demodulating the collided signal.

We have empirically validated the above analytically uncovered possibility of collision recovery, as shown in Fig. 7. In Fig. 7a, the chirp signals of two collided symbols '67' and '94' from A and B are aligned with almost no chirp-level time offset. Two peaks are observed in the normalized FFT result, whose maximum FFT bin's amplitudes differ by the power offset. These two peaks facilitate us to detect and distinguish collided symbols according to their power offset, even without the chirp-level time offset. Fig. 7b further shows an example when two packets collides with the chirp-level time offset. We align with the preamble upchirp of A and thus a preamble symbol of '0' is detected. The upchirp from B is shifted by the chirp-level time offset (around half-chirp cycle) and the original preamble symbol '0' becomes '64' due to the time offset. Thus, it's possible to detect preambles of multi-packet even when they are collided. Besides, we further explore chirp-level time offset to recover collided payload symbols.

D. Limitations of State-Of-The-Art

Several works tried to enable concurrent transmissions by offline collision decoding [10]–[12], usually with two steps: (i) the set of collided packets have to be completely received (and saved) due to the absence of online collision detection and synchronization; (ii) the collided packets are separated and demodulated respectively. As a result, these methods recover the collided packets in a (semi-)offline way, thus incurring large amount of memory overhead and significant latency to LoRa communication. For example, Class-A LoRa ends, as specified in LoRaWAN [9], [13], have only two fixed (and short) reception slots after each transmission, and a large latency will cause timeout of the reception window, i.e., LoRa ends will miss messages from the gateway. Also, timely acknowledgements are needed for some applications to ensure system reliability [8]. Different from these existing solutions, OCT achieves online concurrent transmissions at LoRa gateway without introducing the latency to completely receive collided packets, thus achieving concurrent transmissions in real-time.

III. Design of OCT

A. Overview

OCT enables LoRa gateway to decode collided packs with a comprehensive multi-packet reception mechanism, including preamble detection, SFD detection and packet decoding. Fig. 8 summarizes OCT's working flow.

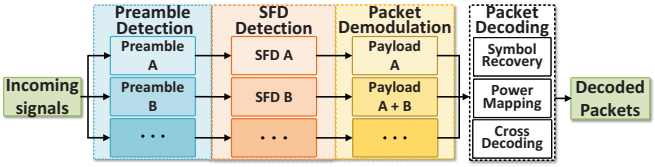


Fig. 8: Working flow of LoRa's collision decoding.

- **Preamble Detection.** LoRa identifies packets by detecting continuous upchirps in the preamble. OCT also detects preamble based on continuous upchirps, with the difference that OCT's preamble detection applies even when collision occurs — the OCT's targeting preamble could be superposed with other packets' preambles, SFDs, or payloads.
- **SFD detection.** Once the preamble is detected, SFD detection is performed based on the identified preamble symbols. OCT determines the start of payload which is at 2.25 downchirps after the SFD.
- **Packet Decoding.** The last step of OCT is packet decoding: first demodulating the incoming signal and then recovering the whole packet based on the demodulation results. Actually, OCT's packet demodulation runs in parallel with its preamble detection and SFD detection, i.e., OCT keeps detecting preambles upon receiving the incoming signal and aligning with the new packets while demodulating previously arrived packets. OCT decodes the collided chirps using the feedback information from preamble/SFD detection (i.e., the time and power offset) with three steps: symbol recovery, power mapping, and cross decoding.

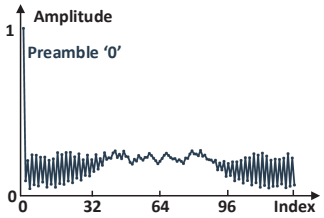
B. Preamble Detection

OCT's preamble detection concludes a LoRa packet is arriving by detecting a continuous upchirp sequence, i.e., the start of a LoRa frame. The preamble detection of standard LoRa only detects one packet or even being invalid when collisions occur, which OCT resolves. Depending on the collisions' time offsets, the preamble of current packet could be superposed with other packets' preamble, SFD or payload. Next we explain how OCT detects the superposed preamble.

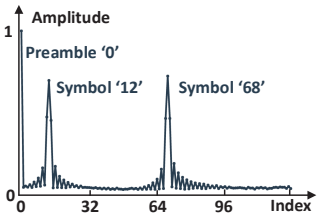
With Chirp-level Time Offset. When two packets collide at the RX with chirp-level time offset, the superposed situation of the preamble can be categorized into three cases according to the packet-level time offsets.

1) The preamble is superposed with the preamble of another packet, in which case the preamble chirps will produce two peaks. We can easily distinguish the preamble chirp of the two packets because they have different peak indexes due to the chirp-level time offset.
2) Second, the preamble may be superposed with the SFD of another packet. Since SFD only consists of downchirps, it has a negligible impact to the current preamble chirp.

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.



(a) The FFT result of superposed preamble and SFD chirps



(b) The FFT result of superposed preamble and two payload chirps

Fig. 9: The FFT results on collided preamble chirps.

3) Third, the preamble is superposed with the payload of another packet. As the preamble is a continuous upchirp sequence, we can still detect successive peaks at the same index no matter what the overlapped payload symbols are.

Fig. 7 and 9 illustrate the three cases when the preamble is superposed with another packet's preamble, SFD, and payload. In the first case, the overlapped preambles of two packets generate two peaks due to the chirp-level time offset, as shown in Fig. 7b. On the other hand, only one peak is generated in Fig. 9a for the second case, while three peaks exist in Fig. 9b for the third case because the current preamble chirp is superposed with two different payload symbols.

OCT uses a simple multi-peak detection mechanism to detect the peaks. Specifically, we define a *peak* as the local maximum whose amplitude is larger than its $\mathbf{k}$ adjacent FFT bins. By exploring the FFT result on current chirp, all peaks are detected and ranked by amplitude descendingly. For $\mathbf{m}$-packet collisions, there exist at most $(2\mathbf{m}-1)$ valid peaks after being aligned with the chirp of a specific packet.

OCT detects the preamble based on the index and energy/amplitude of the top $(2\mathbf{m}-1)$ peaks. OCT then tries to find continuous peak indexes, i.e., possible preambles, by traversing through all these peaks. OCT uses brute-force search to map possible preamble symbols with a specific packet, because the amplitudes of continuous preamble chirps may have different rankings when being overlapped with other packets' signal. We can identify two information, i.e., the current preamble chirp's symbol and amplitude, based on the detected preamble. First, we can coarsely estimate the beginning position of preamble's upchirp according to preamble symbol, and thus align current packet for the following SFD detection. Second, we can determine power offset at the RX for packets decoding based on the max FFT bin.

Without Chirp-level Time Offset. When two packets arrives at the RX without chirp-level time offset, OCT cannot distinguish superposed or non-superposed preamble chirps because both cases generate only one peak in the same FFT bin index after demodulation. However, we can detect two peaks when demodulating the superposed payload chirps as shown in Fig. 7a. OCT concludes a collision occurs if several continuous chirps have multiple peaks with power larger than a specific threshold. OCT then verifies whether we can form correct packet based on the detected peaks using packet decoding. Note when the packet-level time offset is equal or greater than one chirp cycle, we can still detect the two packets' preamble and SFD because they have at least one chirp mismatch.

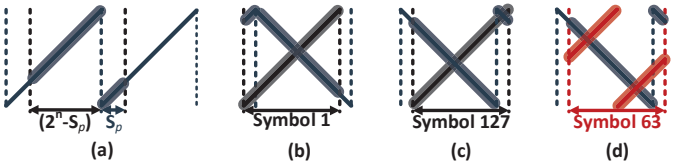


Fig. 10: SFD Detection.

C. SFD detection.

Whenever the preamble of a specific packet is detected, OCT aligns the current packet based on the preamble symbol, and determines the payload's beginning position via SFD detection in three steps.

1) In the first step, OCT aligns the current packet chirp-by-chirp according to the current preamble symbol. Denoting the detected preamble chirp symbol as $\mathbf{s}_p$, the beginning position of the current upchirp should be at the $\mathbf{s}_p$-th sampling point before the current position, and the beginning position of next chirp should be at the $(2^{\mathbf{n}}-\mathbf{s}_p)$-th sampling point after the current position, as illustrated in Fig. 10a.
2) In the second step, OCT determines the payload's position by detecting two consecutive downchirps in the packet's SFD. LoRa detects downchirp by multiplying with the upchirp and then performing FFT on the resultant product, which will detect symbol '0' if correct. However, if the alignment in the previous step is not accurate and there exists a mismatch between the downchirp and upchirp, symbol '1' or '127' may be detected, as shown in Fig. 10b and Fig. 10c. As a result, the packet's SFD may be missed when we try to detect two continuous downchirps whose symbol value is close to zero. As a mitigation, OCT uses a modified upchirp as shown in Fig. 10d, which always generates symbols of values around 64, even when the downchirp is not aligned perfectly. This way, OCT only needs to detect two sequential downchirps whose symbol value is close to 64 when detecting the packet's SFD.
3) In the last step, after detecting the packet's SFD, OCT uses the SFD symbols to further align the incoming chirps, with a similar approach as in the first step. Since the SFD has a fixed length of 2.25 downchirps, OCT can determine the beginning of packet payload, and hence identify the time offset.

D. Packet Decoding

OCT decodes collided symbols using the time/power offset, which allows to decode collided packets with either feasible chirp-level time offset or large power offset. Note we refer a time offset as feasible when it is not close to the duration of an integer times of chirp cycle. OCT's packet decoding includes three steps: the symbol recovery and cross decoding mainly exploit the time offset, while the power mapping is based on the power offset.

Symbol Recovery. When a packet arrives at the RX with a chirp-level time offset, OCT recovers the original symbol

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.

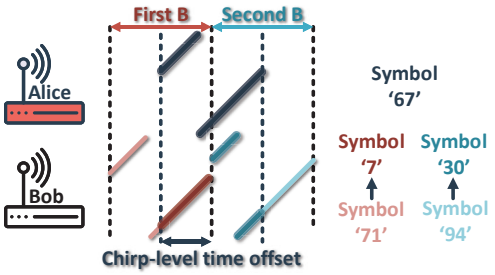


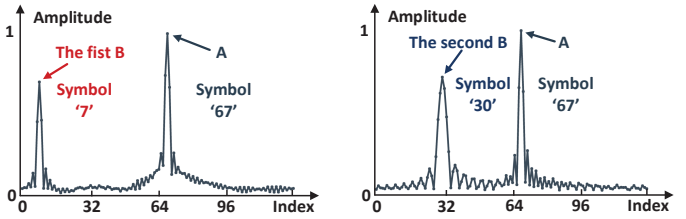
Fig. 11: Fundamentals of symbol recovery.

directly based on the chirp-level time offset. As the payload chirps of a specific packet may be superposed with other packets' preamble, SFD or payload, we use the most complicated case — i.e., the payload chirp is superposed with payload of another packet — to explain OCT's symbol recovery.

Considering the collided payload chirp from a specific TX (e.g., A), the original chirp signal of A is overlapped with two adjacent payload chirps of B, which usually represent different symbols because of the randomness introduced by channel coding, i.e., with a probability of $\frac{1}{2^{\mathbf{n}}}$. As shown in Fig. 11, A's chirp of symbol '67' is superposed with two B's chirps whose symbols are '71' and '94', respectively. Compared with A's chirp, B's chirp has a chirp-level time offset, i.e., half-chirp cycle. OCT will detect two different B's symbols when demodulating B's chirp, because the frequency of the superposed B's chirp is discontinuous.

OCT exploits such a difference to recover the collided payload, as illustrated in Fig. 11. We divide A's chirp into two segments according to the chirp-level time offset. While the first segment only involves the first B's chirp and current A's chirp, the second segment is the overlap of the second B's chirp and current A's chirp. We then complete each segment with 0s to form the whole chirp and demodulate the two segments respectively. Fig. 12 shows an example of the demodulation result. The first segment generates two peaks corresponding to the first B's chirp and current A's chirp, as observed in Fig. 12a. For the second segment, we can detect two symbols corresponding to the second B's chirp and current A's chirp as shown in Fig 12b. Because the two adjacent B's chirps usually represent different symbols, we can determine (i) the chirp symbol of A which is identical, and (ii) two adjacent chirp symbols of B which are different.

OCT can recover most payload symbols with feasible chirp-level time offset. However, the symbol recovery may be invalid in several cases. First, the payload symbols may be superposed with the preamble of another packet, in which case OCT cannot distinguish the adjacent preamble chirps because they are always continuous. Second, the adjacent payload chirps of B, which are superposed with A's chirp (see Fig. 11), may have the same symbol value and thus cannot be distinguished. Note that when the payload symbols are superposed with the SFD chirps of other packets, the SFD chirps have only negligible effects on payload symbols because SFD only contains downchirps. OCT further uses cross decoding to determine the collided symbols when symbol recovery is failed for specific chirps.



(a) The FFT result of the first segment after compensating 0s

(b) The FFT result of the second segment after compensating 0s

Fig. 12: The FFT result of two segments in symbol recovery.

Power Mapping. OCT's symbol recovery is invalid if packets arrive at the TX with infeasible time offset which is close to the integer times of chirp cycle. In these cases, OCT introduces power mapping — which is based on the stable but distinguishable power levels of collided packets, i.e., the power offset between packets — to map decoded symbols to the correct packets.

Power mapping works as follows. We first rank the detect symbols according to their amplitudes, which is identical with the rank of the powers of the collided packets. After this, we map the symbol with the largest power to the packet with the largest power, and the rest can be done in the same manner. We always choose the longest segment (see Fig. 11) as reference for demodulation which only involves one chirp from each collided packet. As a result, we can guarantee to map symbols to the correct packets.

Cross Decoding. OCT uses cross decoding as a complementary technique to symbol recovery and power mapping, grounding on the fact that the determined symbols can be used to recover other symbols in the collision.

Considering the case that several A's payload symbols are superposed with B's preamble, symbol recovery will be invalid because the preamble chirps are continuous. When aligning with A's payload chirps, B's original preamble symbol becomes:

$$s^p_{\mathbf{ba}} = (s^p_{\mathbf{b}} - s^p_{\mathbf{a}} + 2^{(\mathbf{n}-2)})\%2^{\mathbf{n}}, \tag{10}$$

where $s^p_{\mathbf{a}}$ and $s^p_{\mathbf{b}}$ are detected preamble symbols of A and B in preamble detection. We can determine A's payload symbols by excluding the peaks generated by B's preamble chirps.

For collided payload symbols, cross decoding can also be used to eliminate the peaks of known symbols in the FFT result, and thus determine the remaining unknown symbols. When aligning with A's or B's chirp, the original B's or A's symbols can be mapped into new symbols as follows:

$$\begin{aligned} s_{\mathbf{ba}} &= (s_{\mathbf{b}} - T_{\mathbf{b}})\%2^{\mathbf{n}}, \\ s_{\mathbf{ab}} &= (s_{\mathbf{a}} + T_{\mathbf{b}})\%2^{\mathbf{n}}, \end{aligned} \tag{11}$$

where $T_{\mathbf{b}}$ is the number of delayed samples of B when compared with A in one chirp cycle. For example, after aligning with A's chirp in demodulation, the original B's symbols become '7' and '30' due to the chirp-level time offset (see Fig. 11). When symbols '71' of B is decoded, we can determine the symbol '67' of A according to this mapping, and then the next B's symbol of '94'.

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.

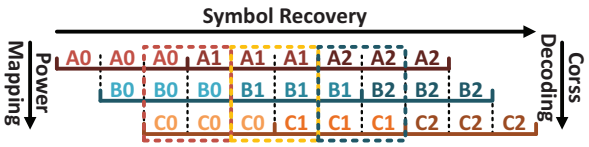


Fig. 13: Packet Decoding in Three-packet Collisions.

Workflow of Packet Decoding. OCT save the demodulated results of multiple packets according to their decoding state, as shown in Fig. 8. After all detected packets are demodulated and there is no newly detected preamble, symbol recovery and power mapping will be performed according to the time and power offsets. Then, cross decoding is applied to recover the undetermined symbols. At last, the recovered symbols are transformed into bits by reverting channel coding and the correctness of packets are checked by CRC. OCT keeps the correctly decoded packets and only retransmits packets that failed CRC.

Computation Overhead. While we need to demodulate two segments for each collided chirp as explained in symbol recovery, the chirp symbols of A and B are determined simultaneously according to the demodulation result, i.e., symbol recovery introduces no extra computation as if we demodulate two packets independently. As a result, the computational complexity of packet demodulation is $\mathbb{O}(\mathbf{mN}2^{\mathbf{n}}\log(2^{\mathbf{n}}))$, where $\mathbf{m}$ is the number of collided packets and $\mathbf{N}$ is the packet length in chirps. The packet decoding has a negligible overhead with a complexity of $\mathbb{O}(\mathbf{mN})$, because OCT only needs to compare several demodulated symbols to identify the correct mapping between symbols and packets.

E. Extending to Multi-packet Collisions.

OCT can be effortlessly extended to the case of multi-packet collisions, as illustrated in Fig. 8.

- **Preamble Detection.** OCT only needs to record more peaks in multi-packet collision detection, e.g., $2\times3-1=5$ peaks in case of 3-packet collisions.
- **SFD Detection.** OCT can detect the SFDs of multiple packets in the order they arrives at the RX. For example, OCT will first detect the preamble of the first arrival packet, then the preamble of second packets. Accordingly, OCT will detect the SFDs one by one according to the detected preamble symbols.
- **Packet Decoding.** OCT's packet decoding is also applicable to decode multi-packet collisions. As shown in Fig. 13, we divide each chirp into three segments according to the chirp-level time offsets. If any two segments are long enough to correctly restore the original chirp symbols, we can use symbol recovery to recover the corresponding packets. After that, we use cross decoding to eliminate known chirps and power mapping to match unknown chirps with small chirp-level time offset. For collisions with more than three packets, we can combine multiple segments together to avoid over-short segments.

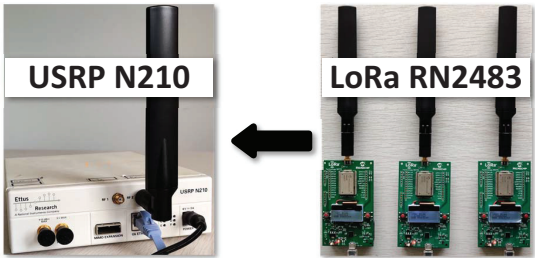


Fig. 14: Testbed of OCT.

F. Design Enhancement

Oversampling. LoRa's baseband chip supports sampling rate (SR) as high as 32MHz [25]. The USRP N210 device with SBX daughterboard also supports up to 40MHz SR in our experiment [26]. As a result, the original LoRa signals with low BW are usually oversampled at the RX. Denote the oversampling rate (OS) as $\mathbf{os}=\frac{\mathbf{sr}}{\mathbf{bw}}$, where $\mathbf{sr}$ is the value of SR. Considering LoRa chirps sent with BW = 125KHz and SF = 7, each chirp will have $1024 = 8\cdot2^7$ samples when sampled with 1MHz SR at the RX. Such an oversampling increases the probability that OCT can recover the collided packet using only the time offset.

Compensating Frequency Offset. Frequency offset shifts LoRa chirps in a packet to the same direction. OCT's preamble and SFD detection can eliminate this effect by aligning chirps with the correct position according to preamble and SFD symbols.

IV. Implementation

We have implemented a testbed using USRP N210 [27] and 3 RN2483 LoRa ends [22] to evaluate OCT (see Fig. 14).

- **USRP N210**, a universal software-defined radio with SBX daughterbord, is used as RX.
- **RN2483**, a fully certified commercial-off-the-shelf LoRa transceiver, is used as the TX.

At the TX side, up to 3 LoRa ends are controlled by the same host or laptop, facilitating controlling the transmitting power/delay. At the RX side, the USRP N210 is operated by another laptop and located in a fixed location, which has the same distance to the 3 LoRa ends. The time and power offsets at the RX are indirectly controlled by adjusting the transmitting power and delay at the TX [14], [28]–[30].

All the packet receiving functionalities are implemented at the USRP N210 with GNURadio [31]. We record the decoding state of each collided packet to realize the online concurrent transmissions, because the decoding process of each packet runs in parallel with each other. We demodulate the payload of the current packet at the same time of detecting preamble of incoming packet. When the SFDs of multiple packets are detected, we also demodulate the segments of overlapped payloads using the approach explained in symbol recovery. We preserve the demodulation results, i.e., the max FFT bin and index, of all packets. At last, the original chirps from different packets are recovered using packet decoding, which can be completed in real time.

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.

V. Experiment

A. Methodology

On/Off-line Experiments. Our experiments consist of the following two parts.

- First, we evaluate OCT in the case of two-packet collisions using online experiments, i.e., two LoRa ends send packets simultaneously while the USRP RX decodes the collided packets in real time. We use a large sending interval to avoid collisions involving more than two packets.
- Second, we conduct offline experiments for the case of three-packet collisions. Temporarily saving the collided packets' samples, we perform preambles and SFDs detection on them, and then decode the original chirps from the collided signal via packet decoding.

Metrics. We evaluate OCT using the following metrics.

- *Packet Reception Ratio:* a packet is successfully received if the header is decoded correctly.
- *Frame Reception Ratio:* FRR is the percentage of frames in the received packets that pass CRC.
- *Chirp Error Rate:* CER is calculated from the number of error chirps in the received packets.
- *Bit Error Rate:* BER is calculated from the number of error bits in the received packets.
- *Throughput:* average throughput of repeated experiments in bit per second (bps), is calculated under 18.11s and 20.18s total sending time for different payload length considering the sending delay and interval of 100 packets.

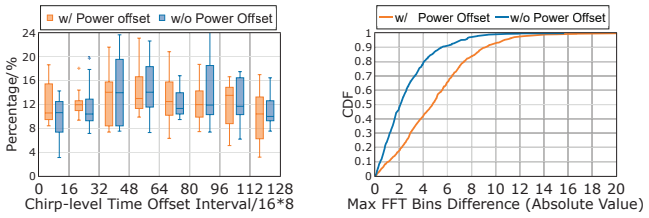
Configurations. Also, we use the following experiment configurations, each with 10 repeated experiments and 100 packets are sent from a single TX in each run. LoRa ends use the default configuration (i.e., SF=7, BW=125KHz).

- *Time Offset:* time offset is nearly uniformly distributed for all experiments. We use a multi-thread python script to make multiple LoRa ends send packets simultaneously with random delays.
- *Power Offset:* we use two settings of transmitting power, -3 and 0, corresponding to -4 and -1.7 dBm output power [22], leading to 2.3 dBm output power difference. The sending power is set to 0, which corresponds to a -1.7 dBm output power, when the power offset is not considered.
- *Payload Length:* we use two payload length, i.e., 8 and 16 bytes, leading to 32 and 48 chirps with LoRa's channel coding.

Baseline. To the best of our knowledge, OCT is the first solution to achieve the online concurrent transmissions at LoRa gateway. As a result, we compare OCT with (i) standard LoRa in the online experiments, and (ii) a state-of-the-art offline collision recovery solution for LoRa, called mLoRa [10], in the offline experiments. Note the comparison with mLoRa is based on the results reported in [10].

B. Existence of Time/Power Offsets.

OCT exploits the time and power offsets at the receiver to detect/synchronize/recover collided packets. So, we first present the experimental results on time/power offsets in different scenarios, to prove their common existence.



(a) Chirp-level time offset percentage. (b) Power offset CDF.

Fig. 15: Time/power offset statistic result.

Time offset. In Fig. 15a, we focus on the chirp-level time offset when SF=7. We map the chirp-level time offset into the number of delayed samples, which are divided into 8 intervals. It can be observed that the chirp-level time offsets are commonly observed in practice, with/without power offset.

Power offset. Fig. 15b plots the CDF of power offset. When power offset is enabled, more than 80% collided packets have a max FFT bin difference larger than 2, while 50% collided packet have such a difference in the case of without power offset, showing power offsets commonly exist in collisions even with the same transmitting power (likely due to hardware heterogeneity).

C. Experiment Result.

Fig. 16 summarizes our main results obtained in the cases of two/three-packet collisions.

PRR. As shown in Figs. 16a and 16e, OCT achieves high PRR (i.e., $>$80%) in all cases. Standard LoRa also achieve $>$75% PRR, because the random packet-level time offset and commonly-existed power offset.

FRR. As shown in Figs. 16b and 16f, OCT achieves $>$80% FRR. Standard LoRa achieves $>$70% FRR when the power offset is large, thanks to the capture effect [32]–[34]. However, standard LoRa's FRR degrades dramatically due to collisions when there is no power offset or the number of collided packets increases to 3.

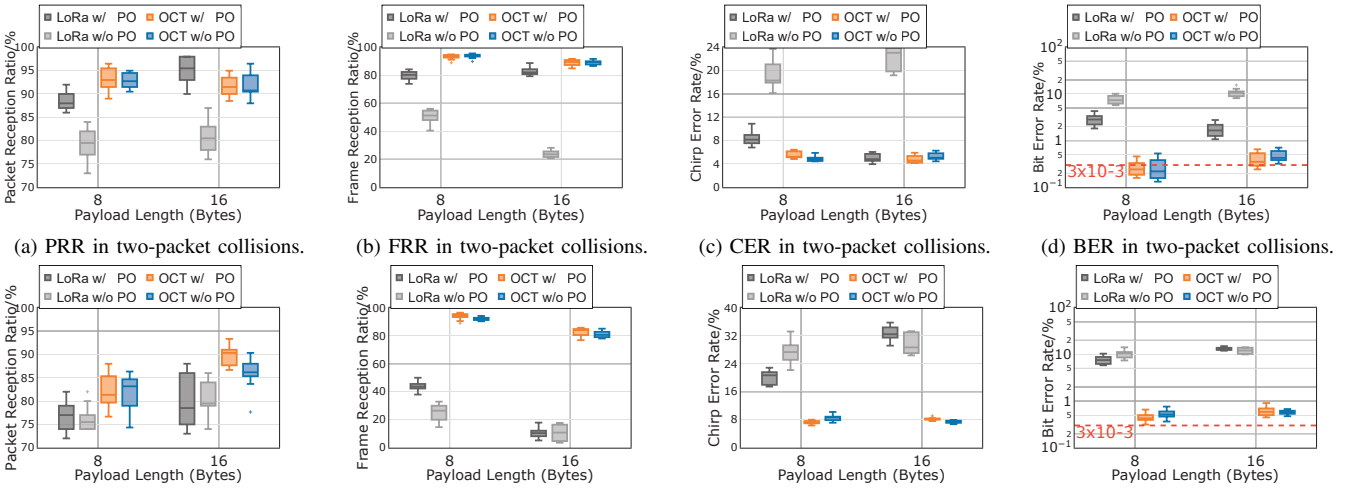
CER. The CER of OCT is around 5% in all experiments. While standard LoRa receiver can also achieve a stable CER of $<$10% when the power offset is large, the CER increases to 30% when there is no power offset.

BER. Thanks to LoRa's channel coding, OCT achieves around 3×10^{-3} BER. With standard LoRa, the BER is $>$1% even after correction, because the chirp errors usually occur consecutively and the overall CER is high.

Throughput. As shown in Fig. 17, OCT achieves 2x and 3x throughput in the scenarios of two- and three-packet collisions. While the throughput of standard LoRa degrades severely in three-packet collisions, OCT is still able to detect and decode most of the collisions.

OCT v.s. State-Of-The-Art. We also compare the latency of OCT with the standard LoRa and a state-of-the-art method, mLoRa [10], in Tab. I. Compared with standard LoRa (49.15ms) and mLoRa (152.79ms), OCT introduces negligible

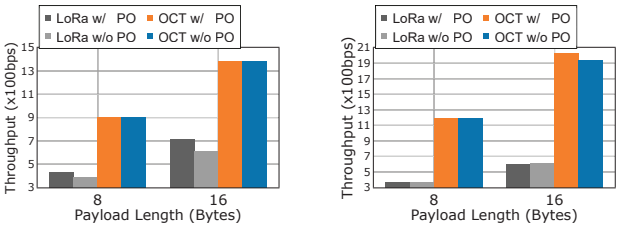
Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.



(a) PRR in two-packet collisions. (b) FRR in two-packet collisions. (c) CER in two-packet collisions. (d) BER in two-packet collisions.

(e) PRR in three-packet collisions. (f) FRR in three-packet collisions. (g) CER in three-packet collisions. (h) BER in three-packet collisions.

Fig. 16: PRR, FRR, BER and CER under w/ or w/o power offset (PO) in two-/three-packet collisions.



(a) Two-packet collisions. (b) Three-packet collisions.

Fig. 17: Average throughput in two-/three- packet collisions.

TABLE I: Comparison with state-of-the-art.

Technology	FRR	CER	BER	Latency
Standard LoRa	$10 \sim 50\%$	$> 10\%$	$> 10^{-2}$	49.15ms
mLoRa [10]	NA	$< 10\%$	$< 10^{-3}$	152.79ms
OCT	$> 90\%$	$< 10\%$	$< 3 \times 10^{-3}$	50.25ms

latency (50.25ms) and reduces 67% time consumption, respectively. mLoRa incurs large latency when decoding the collided packets, making it inapplicable for time-sensitive applications, e.g., Class-A LoRa ends require the gateway to response within two reception windows [13].

VI. Related Works

LoRaWAN Interferences. Plenty of works have been done to analyze or mitigate LoRaWAN interferences [12], [14], [23], [28], [35]–[42]. They analysis the scalability of LoRaWAN and reveal that LoRaWAN can support large amount of LoRa ends (hundreds to thousands) in a cell (several to tens Kms radius) due to the low duty cycle (less than 1%) [35]–[37]. However, the overall network performance is still attenuated significantly by the severe inter-network interference. [23] and [38] introduce directional antenna, dynamic network configuration and multi-gateway deployment to mitigate network interference.

Offline Collision Decoding. Existing works utilize offline collision decoding to realize concurrent transmissions. Capture effect and successive interference cancellation (SIC) are explored to receive the collided packets in LoRa and other protocols [28], [32]–[34], [39], [43], [44]. ZigZag [45], mZig [46] and mLoRa [10] explore collision-free samples to decode the collided parts via time-domain collision decomposition. While the collided symbols are decoded using the frequency offsets caused by hardware imperfections in [12], we reveal that it's possible to recover the collided symbols according to the time/power offsets commonly existent in collisions. [11] separate the collided symbols by tracking continuous frequency of chirps, which slides a window across each PHY sample and perform FFT. Both frequency track extraction and correlation based frame detection in [11] introduce intolerable computation overhead considering online implementation.

In general, all these methods have inherent limitations. First, they adopt a (semi-)offline way to decode the collided packets, which is not practical in real applications due to the latency and memory overhead. Second, existing methods cannot make fully use of features in LoRa collisions. mLoRa [10] will be invalid under small time offsets without using power offset. The computational complexity of FTrack [11] is very high because it slides across each PHY sample without using time offset. Compared with the existing methods, OCT is the first online concurrent transmission design exploring the frequency-domain (de)modulation information and thus can be easily deployed at LoRa gateway.

VII. Conclusion

We present a novel solution achieving the online concurrent transmissions at LoRa gateway, called OCT, which can be easily deployed at LoRa gateway. We have implemented OCT using USRP platform and commodity LoRa ends, and evaluated it with extensive experiments. The results show that OCT achieves more than 90% PPR and FRR in online concurrent transmissions.

Acknowledgment

This work was supported in part by NSFC grant 61972253, 61672349, U190820096, 61672353, 61672348, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] "State of the IoT 2018," https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/.

[2] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric internet of things," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 34–39, 2017.

[3] J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 122–128, 2015.

[4] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "Plora: A passive long-range data network from ambient lora transmissions," *ACM SIGCOMM*, 2018.

[5] A. Varshney, O. Harms, C. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt, "Lorea: A backscatter architecture that achieves a long communication range," *ACM SENSYS*, 2017.

[6] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "Lora backscatter: Enabling the vision of ubiquitous connectivity," *ACM IMWUT*, 2017.

[7] "LoRa report," https://www.gartner.com/en/documents/3715517.

[8] "LoRa application," https://en.four-faith.com/loraapplications/.

[9] "LoRaWAN specification," https://lora-alliance.org/resource-hub/lorawantm-specification-v11.

[10] X. Wang, L. Kong, L. He, and G. Chen, "mLoRa: A multi-packet reception protocol in lora networks," *IEEE ICNP*, 2019.

[11] X. Xia, Y. Zheng, and T. Gu, "FTrack: Parallel decoding for lora transmissions," *IEEE SENSYS*, 2019.

[12] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," *ACM SIGCOMM*, 2017.

[13] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM Transactions on Sensor Networks*, vol. 15, no. 2, p. 16, 2019.

[14] A. Rahmadhani and F. Kuipers, "When lorawan frames collide," *ACM WINTECH*, 2018.

[15] "LoRa Modulation Basics," https://www.semtech.com/uploads/documents/an1200.22.pdf.

[16] O. B. SELLER and N. Sornin, "Low power long range transmitter," Feb. 2 2016, US Patent 9,252,834.

[17] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, "A multi-channel software decoder for the lora modulation scheme," *IoTBDS*, 2018.

[18] M. Knight and B. Seeber, "Decoding lora: Realizing a modern lpwan with sdr," *GNURadio Conference*, 2016.

[19] "SX1272/73 - 860 mhz to 1020 mhz Low Power Long Range Transceiver Datasheet." https://www.semtech.com/uploads/documents/sx1272.pdf.

[20] N. Blenn and F. Kuipers, "Lorawan in the wild: Measurements from the things network," *arXiv preprint arXiv:1706.03086*, 2017.

[21] "Lorawan regional parameters v1.1rb," https://lora-alliance.org/resource-hub/lorawantm-regional-parameters-v11rb.

[22] "RN2483," http://ww1.microchip.com/downloads/en/DeviceDoc/50002346C.pdf.

[23] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" *ACM MSWIM*, 2016.

[24] C. Goursaud and J.-M. Gorce, "Dedicated networks for iot: Phy/mac state of the art and challenges," *EAI endorsed transactions on Internet of Things*, 2015.

[25] "SX1301," https://www.semtech.com/uploads/documents/sx1301.pdf.

[26] "SBX daughterboard," https://kb.ettus.com/SBX.

[27] "USRP N210," http://www.ettus.com/all-products/un210-kit/.

[28] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-hop lora networks enabled by concurrent transmission," *IEEE Access*, vol. 5, pp. 21 430–21 446, 2017.

[29] "python-loranode," https://github.com/rpp0/python-loranode.

[30] "RN2483 Command Reference," https://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf.

[31] "GNURadio," https://www.gnuradio.org/.

[32] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," *ACM SENSYS*, 2006.

[33] C. Gezer, C. Buratti, and R. Verdone, "Capture effect in ieee 802.15. 4 networks: Modelling and experimentation," *IEEE ISWPC*, 2010.

[34] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," *ACM SENSYS*, 2010.

[35] T. Elshabrawy and J. Robert, "Capacity planning of lora networks with joint noise-limited and interference-limited coverage considerations," *IEEE Sensors Journal*, 2019.

[36] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "Lora scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, p. 1193, 2017.

[37] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the lora low power wide area network technology," *European Wireless Conference*, 2016.

[38] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in lora networks," *ACM EWSN*, 2016.

[39] U. Noreen, L. Clavier, and A. Bounceur, "Lora-like css-based phy layer, capture effect and serial interference cancellation," *European Wireless Conference*, 2018.

[40] N. El Rachkidy, A. Guitton, and M. Kaneko, "Decoding superposed lora signals," *IEEE LCN*, 2018.

[41] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on lora networking: Research problems, current solutions and open issues," *IEEE Communications Surveys & Tutorials*, 2019.

[42] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in lora networks," in *IEEE ICDCS*, 2019.

[43] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless lans," *ACM MOBICOM*, 2008.

[44] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: Carving out mac layer opportunities," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 346–357, 2012.

[45] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," *ACM SIGCOMM*, 2008.

[46] L. Kong and X. Liu, "mzig: Enabling multi-packet reception in zigbee," *ACM MOBICOM*, 2015.

Authorized licensed use limited to: Southeast University. Downloaded on October 15,2021 at 02:08:23 UTC from IEEE Xplore. Restrictions apply.