

# X-MAC: Achieving High Scalability via Imperfect-Orthogonality Aware Scheduling in LPWAN

Zhuqing Xu<sup>†</sup>, Junzhou Luo<sup>†</sup>, Zhimeng Yin<sup>§</sup>, Shuai Wang<sup>†</sup>, Ciyan Chen<sup>†</sup>, Jingkai Lin<sup>†</sup>, Runqun Xiong<sup>†</sup>, Tian He<sup>†</sup>

<sup>†</sup>Southeast University, <sup>§</sup>City University of Hong Kong

{zqxu, jluo, shuaiwang, cychen, jklm, rxiong, tianhe}@seu.edu.cn; zhimeyin@cityu.edu.hk

**Abstract**—As an emerging Low-Power Wide Area Networks (LPWAN) technology, LoRa is dedicated to providing long-range connections for pervasive Internet-of-Things devices. As LoRa operates in the unlicensed spectrum, transmissions from multiple LoRa end-devices inevitably collide into each other, leading to packet losses and increased transmission delay. Targeting at collisions caused by interferences under the *same* spreading factor (SF) settings, researchers introduce multiple lines of techniques. Despite their efforts, these techniques commonly neglect the potential collisions caused by interferences under *different* SF settings, which are resulted by the imperfect orthogonality. Given the disparate transmission power configurations and diverse deployed locations, the collisions under different SFs commonly exist in practical networks, and significantly limit the LoRa reliability. In this paper, we present X-MAC, the first scheduler that is aware of imperfect orthogonality. Technically, X-MAC detects the collisions under different SFs via tracking historical transmissions, and further performs dynamic channel scheduling to avoid collisions caused by interferences both under the same and different SFs. Extensive evaluations on testbed devices show that, compared with the state-of-the-art methods, X-MAC boosts the network scalability (number of concurrent end-devices) by  $2.41 \times$  with packet reception rate (PRR) requirement of  $> 95\%$ .

## I. INTRODUCTION

In recent years we have witnessed the proliferation of emerging Low-Power Wide Area Networks (LPWAN). Among existing LPWAN technologies, LoRa is a widely adopted solution, capturing considerable attention from scientific and industrial communities. Since LoRa offers communication at a long distance, e.g., 10km, a LoRa network potentially covers a large area, calling for a high scalability. For this purpose, LoRa introduces chirp spread spectrum (CSS) and assigns different spreading factors (SF) to end-devices at different distances. When two end-devices transmit with different SF parameters, CSS is possible to offer additional resilience for alleviating transmission corruptions.

To further improve the scalability, researchers introduce various types of solutions. Based on the design principles, the existing literature is summarized into the following two categories: PHY-layer approaches and MAC-layer approaches. PHY-layer approaches (e.g., Choir [1], FTrack [2], and CoLoRa [3]) examine the signal characteristics from different LoRa devices, so that they could resolve collided transmis-

sions. However, these techniques require direct access to PHY samples and are not feasible on commodity chips. In contrast, MAC-layer approaches (e.g., S-MAC [4], LMAC [5], and others [6]–[11]) adjust the frequency channel, transmission timing, or power of LoRa end-devices that have the same SF. By doing so, these techniques effectively avoid transmission collisions under the same SF (termed as collisions caused by co-SF interferences) for improving the communication resilience and are directly applicable on commodity LoRa chips.

Although existing literature offers good network scalability, they focus on addressing collisions caused by co-SF interferences, since it was commonly believed that LoRa transmissions with different SFs are resilient to each other. However, recent studies [12]–[16] find that LoRa symbols with different SFs are quasi-orthogonal (imperfect orthogonal), thus could also lead to potential transmission collisions (termed as collisions caused by inter-SF interferences). This problem is especially critical when LoRa end-devices have different deployment locations and signal powers, which is general in practical LoRa networks. Since strong signal receptions from nearby transmitters could overwhelm weaker from faraway transmitters [1]–[3], and it is pretty common for end-devices to be spread over a wide area [16]–[21], such as in smart city, smart agriculture, and industrial monitoring, etc.

To improve the network scalability, this paper proposes X-MAC, the first scheduler that considers imperfect orthogonality in practical LoRa networks. Specifically, X-MAC collects the inter-SF interferences by tracking the successful and failed transmissions. Then it solves an optimization problem to assign a frequency channel (and a transmission timing) to each packet to be transmitted. If the inter-SF/co-SF interference between two LoRa packets causes a transmission collision, X-MAC will assign different frequency channels (or transmission timings) so that their transmissions will not collide with each other. X-MAC further proposes a greedy solution, which offers bounded performance at a moderate cost. It also offers additional designs, such as clock drift correction, closed-loop adaptive status update, and dynamic join/exit, to enable a practical MAC protocol in LoRa networks.

To summarize, X-MAC has the following contributions:

- We design X-MAC, a MAC-layer scheduler that is aware of the imperfect orthogonality for improving the network

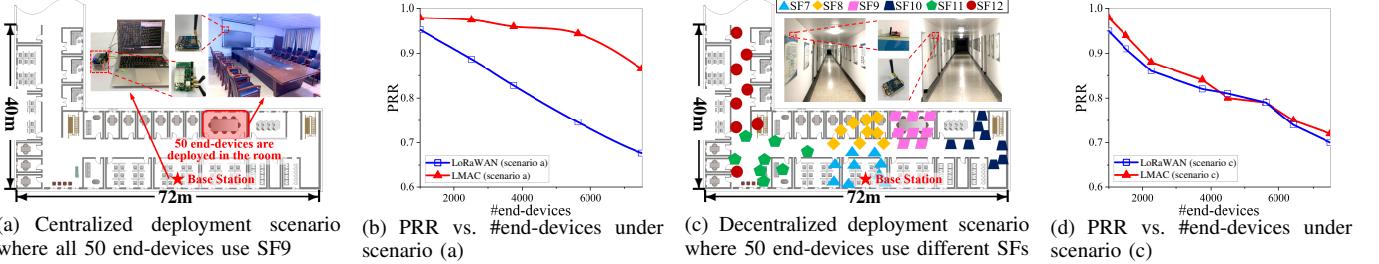


Fig. 1. The packet reception rate (PRR) of the native LoRaWAN and LMAC in different scenarios. Fig. 1(a) and 1(c) show the centralized and decentralized deployment scenarios, resp. Fig. 1(b) and 1(d) show the corresponding PRR results, resp. The bandwidth, coding rate, packet length, transmission power are set to 125kHz, 4/5, 40 bytes, and 14 dBm, resp. The detailed setup is in Section VI-A.

scalability. To the best of our knowledge, X-MAC is the first MAC protocol that considers imperfect orthogonality in practical LoRa deployment, in addition to the commonly considered collisions caused by co-SF interferences.

- We introduce a channel scheduling optimization that decides the frequency channel (and transmission timing) of each packet to be transmitted, followed by a greedy solution with performance bounds. Meanwhile, we design an imperfect-orthogonality aware scheme to detect collisions caused by inter-SF interferences.
- We implement X-MAC on a real-world testbed for extensive evaluations on its performance and cost under various scenarios. Experiment results show that, compared with the state-of-the-art methods, X-MAC boosts the network scalability (number of concurrent end-devices) by  $2.41 \times$  with packet reception rate requirement of  $> 95\%$ .

## II. BACKGROUND AND MOTIVATION

### A. LoRa Background

LoRaWAN is an ALOHA (i.e., random access) based MAC-layer protocol stack. The network architecture proposed by LoRaWAN specification [22] consists of end-device (ED), gateway (GW), network server (NS), and application server. EDs and a GW are connected in a single-hop star topology. The GW is responsible for receiving and forwarding packets to the NS over an IP-based network. The NS is accountable for address checking, authentication and link configuration, etc. For convenience, we refer to the combination of a gateway and network server as a base station (BS).

In LoRa networks, carrier frequency (CF) and spreading factor (SF) are two essential transmission settings. Specifically, CF is the central transmission frequency of a LoRa packet. SF decides the ratio between the modulation bandwidth  $BW$  and symbol rate  $R_s$  by  $SF = \log_2(BW/R_s)$ . A higher SF offers a longer transmission duration for a LoRa symbol, thus increasing its resilience against wireless interference and background noises. This increases the communication range but also leads to higher energy consumption.

LoRaWAN supports a SF choice from 7 to 12. If time-overlapped LoRa packets use the same CF and SF respectively, their transmissions will lead to **co-SF interference**. When the gateway demodulates these signals, the FFT results will produce several indiscernible peaks. Thus, only the packet with the strongest signal power is demodulated if its power

is at least  $\sim 4$  times (6dB) [21], [23] stronger than others. To increase the capacity of LoRa networks, recent studies, e.g., S-MAC [4] and LMAC [5], adjust the channels or transmission timing of packets via scheduling or channel activity detection, so that transmissions with the same SF are collision-free.

### B. Inter-SF Interference

In addition to co-SF interference, **inter-SF interference** occur between time-overlapped packets with the same CF but different SF parameters. This is because, with the CSS, LoRa symbols with different SFs are quasi-orthogonal (imperfectly orthogonal) [12]–[16], instead of perfectly orthogonal. As a result, ongoing LoRa transmissions will inevitably lead to wireless interference among each other, and the inter-SF capture condition must be met to demodulate the desired signal from the interfering signals [12]–[14], [16], which is given by:

$$SINR_{dsr} = \frac{P_{dsr}}{P_{int} + \sigma^2} \geq \theta_{SF_{dsr}} \quad (1)$$

where  $P_{dsr}$  refers to the received power of the desired signal,  $P_{int}$  is the received total power of the interfering signals using different SFs, and  $\sigma^2$  is the power of additive white Gaussian noise (AWGN) of the wireless channel. If the signal-to-interference-plus-noise ratio (SINR) of the desired packet exceeds the corresponding threshold  $\theta_{SF_{dsr}}$ , then this packet could be correctly demodulated.

Tracing the root, the CSS demodulator treats other symbols with different SFs as modulated noise when demodulating the desired signal. If the amplitudes of two signals differ significantly, correlation peaks may occur in positions different from the symbol time, and demodulation errors arise.

For simplicity, we refer to a transmission collision caused by co-SF interference as a **co-SF collision**, and a collision caused by inter-SF interference as an **inter-SF collision**.

### C. Limitations of State-of-the-art

Suppose that the distance of an end-device using SF12 to the gateway is 6 times that of an end-device using SF7 to the gateway. With log-distance path loss model [18], [19], [24], it gives the end-device using SF7 a 23 to 31 dB power margin over the end-device using SF12 (for path loss exponents of 3, 4, resp.). According to demodulation thresholds of inter-SF interference [13], [16], [25], it can be theoretically confirmed that any transmission with SF7 will at least destroy the concurrent transmissions with SF12 (and most probably SF11).

To examine these different types of interferences in actual LoRa networks and their impacts, we first conduct the following series of experiments in an indoor office shown in Fig. 1(a). In this scenario, all 50 end-devices are deployed in a room, and have the same SF parameter, e.g.,  $SF = 9$ . Due to the difficulty of deploying a large-scale LoRa network in practice [19], we adopt a common emulation solution, following [4], [20], which changes the transmission periods of end-devices to emulate a larger network. Fig. 1(b) shows the packet reception rate comparison between native LoRaWAN, a pure ALOHA-based solution, and the state-of-the-art co-SF MAC, LMAC [5], which utilizes channel activity detection (CAD) to avoid collisions among LoRa transmissions with the same SF. It can be found that the performance of LMAC significantly outperforms LoRaWAN under various network scales.

However, in typical LoRa networks, different LoRa end-devices are deployed in various locations, thus resulting in different received powers at the gateway and further leading to different choices of SF for ensuring communication resilience. Fig. 1(c) depicts the same indoor office as Fig. 1(a), but 50 end-devices are decentralized deployed and their SF parameters are decided by specific channel fading. Fig. 1(d) shows that the co-SF MAC (LMAC<sup>1</sup>) suffers from heavy packet losses and is only slightly better than native LoRaWAN. This is because the channel activity detection employed by LMAC is SF-selective, i.e., it can only detect transmissions in a specified SF, while remaining insensitive to other transmissions with different SFs. This experiment demonstrates the severe impacts of collisions due to imperfect orthogonality, which is commonly neglected in current solutions.

Overall, considering the high density and wide coverage of end-devices, in practical LoRa networks, it is common that end-devices with different SF settings are spread over a wide geographical area [1]–[3], [16]–[20]. As a result, the inter-SF interferences significantly affect the communication reliability and impose challenges for improving the LoRa network scalability, calling for specific solutions.

### III. OVERVIEW OF X-MAC

To the best of our knowledge, X-MAC is the first MAC-layer scheduler that considers the imperfect orthogonality (inter-SF collisions) for improving the scalability in practical LoRa networks. Fig. 2 illustrates its architecture.

End-devices report their transmission periods in their joining request in X-MAC.<sup>2</sup> To be compatible with LoRaWAN protocol, X-MAC preserves LoRaWAN packet structure, and fills reserved fields in the packet header for the additional control information, e.g., identify and confirm the relevant status. For each uplink packet to be transmitted, the end-device configures the specified channel configuration information (CCI), i.e.,

<sup>1</sup>The author has opened the source code in “<https://wands.sg/lmac/>”. We use the best configuration “Level 3” provided by the website for experiments.

<sup>2</sup>According to the 8-month data mining analysis of over 17 million packets sent by 1,618 end-devices in the world-wide LoRaWAN network platform The Things Network (TTN) [26], it is found that more than 93% LoRa end-devices have periodic transmission characteristics [27].

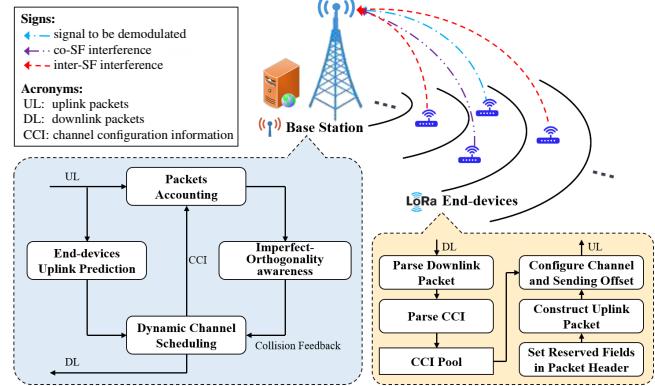


Fig. 2. The architecture of the X-MAC

frequency channel and transmission timing, which is pre-assigned by the base station operating our X-MAC scheduling. For each received downlink packet from its receive window, the end-device parses and saves the CCI (if any) carried in it.

The base station operates the following additional functions for improving the network scalability: First, the imperfect-orthogonality aware scheme collects inter-SF collisions by tracking the successful and failed transmissions. Second, by considering the detected inter-SF collisions, a dynamic channel scheduler configures a channel for each transmission to avoid co-SF collisions and currently detected inter-SF collisions, then constructs and distributes the required CCI to the receive window of each end-device in a piggyback way. In addition, by analyzing status information of the received packets, the base station detects the clock drift of each end-device, and predicts the transmission time of subsequent packets.

## IV. DESIGN

### A. System Model

In practical LoRa networks, a massive number of devices are deployed in different locations, requiring disparate SF settings depending on the specific channel status. Accordingly, the core of X-MAC is to capture the inter-SF collisions without much prior knowledge, and further schedule each transmission of end-devices to avoid packet corruption.

#### Assumptions.

- The transmissions of end-devices are periodic, while their periods are different depending on specific application settings. This follows the report in TTN, where 93% of the LPWAN applications are periodic [27], and this paper mainly studies scenarios with periodic transmissions.
- CF of a LoRa transmission is determined by the base station, while SF is disabled to adjust arbitrarily. Since in native LoRaWAN, end-devices can inherently randomly select a CF from the available channels for each transmission. On the other hand, SF is closely related to range, data rate and energy consumption of an end-device, and modifying it arbitrarily may break application constraints on delay, energy, etc. Further, X-MAC can achieve high scalability and reliability without modifying SF, after end-devices successfully communicate with the gateway.

- The gateway is full-duplex that supports simultaneous transmitting and receiving packets. Since LoRaWAN regional parameters specification [28] has offered separate uplink and downlink channels for most regulatory regions worldwide. In addition, it is feasible and low-cost to implement such a full-duplex gateway by using a separate RF chip for downlink transmission, and Semtech also introduces a full-duplex gateway reference design [29].

Fig. 3 describes a basic scenario of periodic transmissions from end-devices. Each end-device has a transmission period. The notations  $s_i, f_i$  denote the transmission start and finish time of a certain packet, resp. The end-device transmits a packet immediately at the beginning of a transmission period, which is termed as “*strict model*”. A packet transmission from  $s_i$  to  $f_i$  is hereinafter referred to as a *transmission event*, or simply an *event*. Table I summarizes the essential notations.

TABLE I  
IMPORTANT NOTATIONS AND DESCRIPTIONS

Notation	Description
$n$	Number of transmission events, where $n \in \mathbb{Z}_{++}$ .
$m$	Number of available frequency channels, where $m \in \mathbb{Z}_{++}$ .
$i, j$	Index of event, where $i, j \in \mathbb{Z}_n, \mathbb{Z}_n = \{1, \dots, n\}$ .
$k$	Index of CF, where $k \in \mathbb{Z}_m, \mathbb{Z}_m = \{1, \dots, m\}$ .
$E$	Set of total events, where $E = \{e_1, \dots, e_n\}$ .
$U, V$	Non-empty subset of events, where $U, V \neq \emptyset, V \subseteq U \subseteq E$ .
$e_i$	The $i$ -th event, where $e_i \in E$ .
$s_i$	Transmission start time of $e_i$ , where $s_i \in \mathbb{R}_+$ .
$f_i$	Transmission finish time of $e_i$ , where $f_i \in \mathbb{R}_+, s_i < f_i$ .
$\mathbf{C}$	$c_{ij}$ in $\mathbf{C} = [c_1 \ c_2 \ \dots \ c_n] \in \mathbb{S}^n$ denotes whether a co-SF/inter-SF collision will occur between $e_i$ and $e_j$ , where $c_{ij} \in \{0, 1\}$ .
$\mathbf{T}$	$t_{ij}$ in $\mathbf{T} = [t_1 \ t_2 \ \dots \ t_n] \in \mathbb{S}^n$ denotes whether $e_i, e_j$ overlap in time, where $t_{ij} \in \{0, 1\}$ . If $i \neq j, s_i \leq f_j, f_i \geq s_j$ , then let $t_{ij} = 1$ ; else $t_{ij} = 0$ .
$\mathbf{P}$	$p_{ij}$ in $\mathbf{P} = [p_1 \ p_2 \ \dots \ p_n] \in \mathbb{S}^n$ denotes whether $e_i, e_j$ use the same SF, where $p_{ij} \in \{0, 1\}$ .
$\mathbf{R}$	$r_{ij}$ in $\mathbf{R} = [r_1 \ r_2 \ \dots \ r_n] \in \mathbb{S}^n$ denotes whether a collision caused by the inter-SF interference will occur between $e_i$ and $e_j$ , when $t_{ij} = 1, x_i^T x_j = 1$ , and $p_{ij} = 0$ . $r_{ij} \in \{0, 1\}$ .
$\mathbf{X}$	$x_{ki}$ in $\mathbf{X} = [x_1 \ x_2 \ \dots \ x_n] \in \mathbb{R}^{m \times n}$ denotes whether CF $k$ is used by $e_i$ , where $x_{ki} \in \{0, 1\}$ .

To improve the scalability, we offer the following optimization problem by considering imperfect orthogonality.

$$\max_{\mathbf{x}_i} \sum_{i=1}^n \mathbf{1}^T \mathbf{x}_i (1 - \iota_{CL}^i(E \setminus \{e_i\})) \quad (2)$$

Subject to:

$$\iota_{CL}^i(U) := \begin{cases} 1, & \exists V \subseteq U, \iota_{OL}^i(V) \iota_{CF}^i(V) \iota_{SF}^i(V) \neq 0, \text{ or} \\ & \iota_{OL}^i(V) \iota_{CF}^i(V) (1 - \iota_{SF}^i(V)) \iota_R^i(V) \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$\mathbf{1}^T \mathbf{x}_i \leq 1, \forall i \in \mathbb{Z}_n \quad (4)$$

$$x_{ki} \in \{0, 1\}, \forall i \in \mathbb{Z}_n, \forall k \in \mathbb{Z}_m \quad (5)$$

where the function  $\iota_{OL}^i(V)$  is a 0-1 indicator:  $\iota_{OL}^i(V) = 1$  only if  $\forall e_j \in V, t_{ij} = 1$ . The function  $\iota_{CF}^i(V)$  is a 0-1 indicator:  $\iota_{CF}^i(V) = 1$  only if  $\forall e_j \in V, x_i^T x_j = 1$ . The function  $\iota_{SF}^i(V)$  is a 0-1 indicator:  $\iota_{SF}^i(V) = 1$  only if  $\forall e_j \in V, p_{ij} = 1$ . The function  $\iota_R^i(V)$  is a 0-1 indicator:  $\iota_R^i(V) = 1$  only if  $\iota_{OL}^i(V) = 1, \iota_{CF}^i(V) = 1, \iota_{SF}^i(V) = 0$ , and  $e_i$  is collided by events in  $V$ .

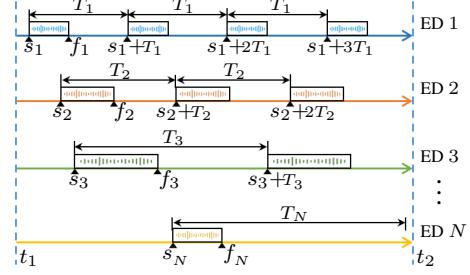


Fig. 3. The basic scenario of periodic transmissions from different end-devices

The objective (2) is to assign an appropriate frequency channel for each event, such that the number of successfully delivered packets is maximized. The 0-1 indicator function (3) defines whether  $e_i$  has a transmission collision (co-SF/inter-SF collision) with the events in  $U$ , where  $\iota_{CL}^i(U) = 0$  only if  $e_i$  does not collide with any subset events in  $U$ . The variable  $x_{ki}$  in (5) denotes whether  $e_i$  uses channel  $k$  for transmission. The Ineq. (4) denotes that for any  $e_i$ , it can only select at most one frequency channel for transmission. The vector  $\mathbf{x}_i = \mathbf{0}$  means  $e_i$  is not allowed to be transmitted.

However, it is challenging in practical LoRa networks to fully acquire inter-SF collision information for all subsets of events, i.e.,  $\iota_R^i(V)$ , for  $\forall V \subseteq (E \setminus \{e_i\}), i \in \mathbb{Z}_n$ , since their scale reaches  $O(n2^n)$ . Inspired by the following key insight, we can simplify the scale of the required inter-SF collision information to  $O(n^2)$ .

Suppose that an inter-SF collision occurs when  $n$  ( $n \geq 3$ ) events (e.g.,  $\{e_1, e_2, e_3\}$ ) are transmitted simultaneously. If we determine that the inter-SF collision occurs between 2 events (e.g.,  $e_1$  and  $e_2$ ) in  $n$ , then with the ensuing rational channel scheduling, the collision between these 2 events is avoided, i.e., for these  $n$  events, the inter-SF collision can only appear in  $(n-1)$  events (e.g.,  $\{e_1, e_3\}$ , or  $\{e_2, e_3\}$ ). The rest may be deduced by analogy. To this end, by well sensing the inter-SF collision between each pair of events and channel scheduling, the inter-SF collision among multiple events can be effectively handled. Accordingly, the problem is updated as follows.

$$\max_{\mathbf{x}_i} \sum_{i=1}^n \mathbf{1}^T \mathbf{x}_i \prod_{j=1, j \neq i}^n (1 - c_{ij}) \quad (6)$$

Subject to:

$$\mathbf{c}_i = \mathbf{t}_i \circ (\mathbf{X}^T \mathbf{x}_i) \circ \mathbf{p}_i + \mathbf{t}_i \circ (\mathbf{X}^T \mathbf{x}_i) \circ (\mathbf{1} - \mathbf{p}_i) \circ \mathbf{r}_i \quad (7)$$

$$\mathbf{1}^T \mathbf{x}_i \leq 1, \forall i \in \mathbb{Z}_n \quad (8)$$

$$x_{ki} \in \{0, 1\}, \forall i \in \mathbb{Z}_n, \forall k \in \mathbb{Z}_m \quad (9)$$

where the symbol  $\circ$  denotes the *Hadamard product*. The vector  $\mathbf{c}_i$  in Eq. (7) is the  $i$ -th column in matrix  $\mathbf{C}$ , and each component  $c_{ij}$  in  $\mathbf{c}_i$  indicates whether a collision (co-SF or inter-SF collision) occurs between  $e_i$  and  $e_j$ .

To tackle this problem, the matrix  $\mathbf{R}$  (explained in Table I) used in Eq. (7) requires to be determined primarily. Thus, we first design an imperfect-orthogonality aware scheme to detect inter-SF collisions. Then, a performance-bounded channel scheduling solution for the above combinatorial optimization is proposed to maximize packet delivery. Besides, additional designs are offered to make it practical in LoRa networks.

## B. Imperfect-Orthogonality Aware Scheme

To detect inter-SF collisions, the first intuition is to model the propagation channel by theoretical analysis [12], [13], [30] or learning-based approaches [23], [24]. However, inter-SF collisions are closely related to the deployment sites of networks, ambient environment, and signal power from other end-devices, etc., making it challenging for them to adapt to various environments and avoid such collisions effectively. On the other hand, these methods require rich prior knowledge, such as distances between end-devices and the gateway, transmission parameters, or multi-spectral images, etc., which greatly increases the practical deployment complexity.

To this end, inspired by the observation that inter-SF collision information can be identified by tracking the expected and actual packet reception results of channel scheduling, the key idea of proposed practical imperfect-orthogonality aware scheme is to check the reception of packets that meet the precondition of inter-SF collisions, e.g., for the case where two end-devices transmit overlapped packets with the same CF and different SFs, if packets are lost frequently, it is considered that there is an inter-SF collision with high probability. Otherwise, it is inferred that they are collision-free.

---

### Algorithm 1 Imperfect-Orthogonality Aware Scheme

---

```

1: input:  $E_{ps}$ ,  $E_{cs}$ ,  $\mathbf{Q}^d$ 
2: output:  $\mathbf{Q}^{d+1}$ ,  $\mathbf{R}$ 
3:  $E_{dl} \leftarrow \{e_j \mid e_j \in E_{ps}, e_j \text{ was sent according to the CCI}\}$ 
4:  $E_{rx} \leftarrow \{e_j \mid e_j \in E_{ps}, e_j \text{ was received by the BS}\}$ 
5: for each  $e_i \in (E_{dl} \setminus E_{rx})$  do
6:    $E_{cn} \leftarrow \{e_j \mid e_j \in E_{ps}, t_{ij} = 1, \mathbf{x}_i^T \mathbf{x}_j = 1, p_{ij} = 0\}$ 
7:    $e_j \leftarrow \text{event with max RSSI in } (E_{cn} \cap E_{rx})$ 
8:   if  $e_j = \emptyset$  then
9:      $e_j \leftarrow \text{event with max history avg. RSSI in } (E_{cn} \cap E_{dl})$ 
10:  end if
11:  Get indexes  $x, y$  of EDs to which  $e_i, e_j$  belong, resp.
12:   $q_{xy} \leftarrow q_{xy} + 1; q_{yx} \leftarrow q_{yx} + 1$ 
13: end for
14: for each  $e_i \in E_{rx}$  do
15:    $E_{cn} \leftarrow \{e_j \mid e_j \in E_{ps}, t_{ij} = 1, \mathbf{x}_i^T \mathbf{x}_j = 1, p_{ij} = 0\}$ 
16:    $e_j \leftarrow \text{event with min RSSI in } (E_{cn} \cap E_{rx})$ 
17:   Get indexes  $x, y$  of EDs to which  $e_i, e_j$  belong, resp.
18:   if  $q_{xy} > 0$  then  $q_{xy} \leftarrow q_{xy} - 1; q_{yx} \leftarrow q_{yx} - 1$ 
19: end for
20: for any  $e_i, e_j \in E_{cs}$  do
21:   Get indexes  $x, y$  of EDs to which  $e_i, e_j$  belong, resp.
22:   if  $q_{xy} \geq \gamma$  then  $r_{ij} \leftarrow 1$  else  $r_{ij} \leftarrow 0$ 
23: end for

```

---

In light of this, we design the imperfect-orthogonality aware scheme (Algorithm 1), which utilizes the expected received packets of channel scheduling, and the corresponding actually received packets to detect inter-SF collisions (i.e., acquire  $\mathbf{R}$ ), then the updated  $\mathbf{R}$  is further employed as an input of the next round of channel scheduling (Section IV-C), and the entire process is performed alternately iteratively (Section

V-B). As shown in Algorithm 1, the notations  $E_{ps}$ ,  $E_{cs}$  denote the event sets in the past and future time intervals, resp.  $\mathbf{Q} \in \mathbb{S}^N$  indicates the number of inter-SF collisions between end-devices, where  $N$  is the number of joined end-devices. It is initially a zero matrix and is updated in each round  $d$ . The constant  $\gamma$  denotes the confidence threshold of an inter-SF collision. In the implementation of X-MAC, we set  $\gamma$  to 3 to balance collision detection efficiency, performance, and stability. It is evaluated and elaborated in Section VI-E, VI-F.

Specifically, for each unexpected lost packet  $e_i$  that was sent according to the CCI but not received by the BS, find the interfering packet  $e_j$  most probably to cause its loss (line 6-10), which provides the most interfering power (refer to the demodulation condition in Eq. (1)), and increase the number of inter-SF collisions between the transmitters of  $e_i, e_j$  by 1 (line 11-12). On the other hand, for each received packet, find the interfering packet that has the least impact on its reception, and decrease the number of corresponding inter-SF collisions by 1 (line 15-18). With those,  $\mathbf{R}$  is updated by  $\mathbf{Q}$  (line 20-23).

## C. Performance-bounded Grouping for Channel Scheduling

This section first analyzes the complexity of the proposed channel scheduling problem, describes the shortcomings of the classic solutions, and followed by a greedy solution with performance bounds.

**Theorem 1.** *The decision problem corresponding to the combinatorial optimization defined in (6)-(9) is NP-hard.*

**Proof.** The problem of selecting the largest number of collision-free events (e.g.,  $n_x$ ) from  $n$  events is a case of optimization (6)-(9) under  $m = 1$ . This problem is reduced by maximum independent set problem of graph  $G$  in poly-time.

The transformation is as follows: (a) For each vertex  $v_i \in G$ ,  $\forall i \in \mathbb{Z}_n$ , construct an event  $e_i$  such that  $s_i = 1$ ,  $f_i = 2$ . (b) For any  $v_i, v_j \in G$ , if there exists an edge  $(v_i, v_j)$ , then  $c_{ij} = 1$ , else  $c_{ij} = 0$ . Then, it holds: there exists  $n_x$  collision-free events iff  $G$  has an independent set of  $n_x$  vertices.  $\square$

It is challenging to find exact poly-time solutions for the optimization (6)-(9) in light of Theorem 1. The intuitive idea is to employ search-based methods, such as branch-and-bound [31], binary-search [32], to find a (near) optimal solution. In addition, we also leverage semidefinite programming based methods, such as [33], [34], to achieve a decent approximate solution. However, the extensive experiments suggest that these methods introduce huge search space and large-scale computation overhead, and their perform efficiency affects the network scalability, since time-consuming solutions reduce CCI downlink opportunities (detailed in Section V-B).

Given these points, we propose a greedy based algorithm, which selects as many collision-free transmission events as possible for each of  $m$  groups, and assigns a unique frequency channel for each group. It can efficiently obtain a satisfactory solution with a bounded guarantee in most cases, and is suitable for base stations with various computing powers.

Specifically, we first construct a  $G$  by the following rules: (a) For each event  $e_i$ , construct a vertex  $v_i$ . (b) Suppose that for any  $e_i, e_j$ ,  $\mathbf{x}_i^T \mathbf{x}_j = 1$  is satisfied. If  $c_{ij} = 0$ , insert an edge  $(v_i,$

$v_j$ ) in  $G$ . Thus, the problem of finding the maximum collision-free events set for a group is transformed into the problem of finding the maximum clique in  $G$ . As is proved, no poly-time algorithm can approximate the maximum clique size within a factor of  $n^{1-\epsilon}$  ( $\epsilon > 0$ ), unless  $P = NP$  [34]. To this end, we further design a random graphs theory [35] based lightweight scheme in Algorithm 2 to trade off between efficiency and approximation ratio of the solution. The core idea is to start with the largest degree vertex (line 5-7), and repeatedly choose a vertex with as large degree as possible that is connected to all previously selected vertices (line 8-12).

#### Algorithm 2 Performance-bounded Events Grouping

```

1: input:  $E, m, R$ 
2: output: The set of  $m$  group events:  $S$ 
3: With  $R$  as an input, construct a graph  $G$  from the set  $E$ 
   according to the above rules
4: for  $k \leftarrow 1$  to  $m$  do
5:    $v \leftarrow$  the vertex with the largest degree in  $G$ 
6:    $S_k \leftarrow S_k \cup \{v\}$ 
7:    $\Gamma \leftarrow \{\text{adjacent vertices of } v\}$ 
8:   while  $\Gamma \neq \emptyset$  do
9:      $v \leftarrow$  the vertex with the largest degree in  $\Gamma$ 
10:     $S_k \leftarrow S_k \cup \{v\}$ 
11:     $\Gamma \leftarrow (\Gamma \setminus \{v\}) \cap \{\text{adjacent vertices of } v\}$ 
12:   end while
13:   Remove vertex  $v$  in  $G$ , for any  $v \in S_k$ 
14: end for
15:  $S \leftarrow \{S_1, S_2, \dots, S_m\}$ 
```

**Definition 1.**  $G(n, p)$  is a random graph [35] on  $n$  vertices, where each edge appears independently with probability  $p$ .

**Theorem 2.** Let  $p \in (0, 1)$  be a constant. With high probability, the above greedy scheme produces a maximal clique of size  $(1 - o(1))\log_{1/p}n$  in  $G(n, p)$ .

**Proof.** The greedy scheme only ends with a vertex set which is a maximal clique, i.e., no more vertices can be added to it. Then, the expected number of maximal cliques of size  $z$ :

$$\mathbb{E}_z = \mathbb{E}[\text{number of maximal cliques of size } z]$$

$$\begin{aligned} &= \binom{n}{z} p^{\binom{z}{2}} (1 - p^z)^{n-z} \\ &\leq \binom{ne}{z} p^{\frac{z(z-1)}{2}} (1 - p^z)^{n-z} \quad (\text{for } 1 \leq z \leq n, \binom{n}{z} \leq \binom{ne}{z}) \\ &\leq \left(\frac{ne}{z}\right)^z p^{\frac{z(z-1)}{2}} e^{-(n-z)p^z} \quad (\text{for any } x, 1 + x \leq e^x) \\ &\leq (p^{-\frac{1}{2}} ne^2)^z e^{-np^z} \quad (p^z \leq 1) \end{aligned}$$

When  $z = \log_{1/p}n - C\log_{1/p}\log_{1/p}n$ , we get

$$\begin{aligned} \mathbb{E}_z &\leq (p^{-\frac{1}{2}} ne^2)^{\log_{1/p}n} e^{-(\log_{1/p}n)^C} \\ &= \exp(\ln[(p^{-\frac{1}{2}} e^2 p^{-u})^u e^{-u^C}]) \quad (\text{let } u = \log_{1/p}n) \\ &= \exp(\frac{u}{2} \ln 1/p + 2u + u^2 \ln 1/p - u^C) \end{aligned}$$

When  $C > 2$ ,  $\sum_{z \leq \log_{1/p}n - C\log_{1/p}\log_{1/p}n} \mathbb{E}_z = o(1)$  holds. This shows that it is almost impossible to find a maximal clique with less than  $\log_{1/p}n - C\log_{1/p}\log_{1/p}n$  vertices. In other words, with high probability all maximal cliques have size at least  $\log_{1/p}n - C\log_{1/p}\log_{1/p}n$  (for any  $C > 2$ ).  $\square$

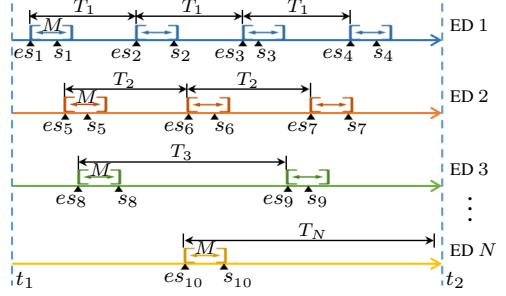


Fig. 4. The loose scenario of transmissions from different end-devices

Next, let's analyze the performance and efficiency. Theorem 2 shows that the above greedy scheme can find a maximal clique of size roughly  $\log_{1/p}n$  in  $G(n, p)$  with high probability. By contrast, it is implied in [35] that the size of the largest clique in  $G(n, p)$  is roughly  $2\log_{1/p}n$  with high probability. Therefore, it is roughly a 2-approximation greedy scheme in most cases. On the other hand, the time and space complexity of the above algorithm are  $O(mn^2)$  and  $O(n^2)$ , respectively.

## V. ADVANCED DESIGN

### A. Loose Channel Scheduling

In some applications, the transmission start time is allowed to be offset. As shown in Fig. 4, given the maximum offset  $M$ , the start time  $s_i$  of the event can be selected in the time interval  $[es_i, es_i + M]$ , where  $es_i$  is the earliest transmittable start time. This is termed as “*loose model*” accordingly.

It can also be defined as an optimization problem:

$$\max_{\mathbf{x}_{i,s_i}} \sum_{i=1}^n \mathbf{1}^T \mathbf{x}_i \prod_{j=1, j \neq i}^n (1 - c_{ij}) \quad (10)$$

Subject to:

$$0 \leq s_i - es_i \leq M, \forall i \in \mathbb{Z}_n \quad (11)$$

$$\mathbf{1}^T \mathbf{x}_i \leq 1, \forall i \in \mathbb{Z}_n \quad (12)$$

$$x_{ki} \in \{0, 1\}, \forall i \in \mathbb{Z}_n, \forall k \in \mathbb{Z}_m \quad (13)$$

where the definition of  $c_{ij}$  is the same as (7). The objective (10) is to arrange appropriate channel and transmission offset (i.e.,  $s_i - es_i$ , we use notation  $l_i$  for simplicity) for each event  $e_i$  to maximize the number of successfully delivered packets.

**Theorem 3.** The decision problem corresponding to the combinatorial optimization defined in (10)-(13) is NP-hard.

**Proof.** Optimization (6)-(9) is a special case of optimization (10)-(13) under  $M = 0$ . By Theorem 1, this result holds.  $\square$

Theorem 3 implies the fact that it is highly challenging to obtain an optimal solution with poly-time complexity. To this end, we propose an effective heuristic algorithm, which sets a transmission offset for each event, selects as many collision-free events as possible for each of  $m$  groups, and assigns a unique frequency channel for each group.

As shown in Algorithm 3, first, for each event in current  $E$ , compute the number of collisions between it and another event in  $E$ , and arrange the event with the least number of collisions to the current group (line 6-9). Then, for each event in the remaining event set, adjust the minimum required transmission offset to make it collision-free with all events in this group.

If the required offset exceeds the maximum offset  $M$ , it is no longer considered in this group (line 10-12). Repeat these steps until no more events can be added to this group.

The time and space complexity of the above algorithm are  $O(mn^3)$  and  $O(n)$ , respectively.

### Algorithm 3 Transmission Events Grouping with Offset

```

1: input:  $E, m, \mathbf{R}, M$ 
2: output: The set of  $m$  group events:  $S$ 
3: for  $k \leftarrow 1$  to  $m$  do
4:    $U \leftarrow \emptyset$ 
5:   while  $E \neq \emptyset$  do
6:     Taking  $\mathbf{R}$  as an input, for  $\forall e_i \in E$ , compute its
       #collisions by  $\sum_{j=1}^{|E|} c_{ij}$  under  $\mathbf{x}_i^T \mathbf{x}_j = 1$ 
7:     Find the event  $e_i$  in  $E$  that has the least collision
8:      $S_k \leftarrow S_k \cup \{e_i\}$ 
9:      $E \leftarrow E \setminus \{e_i\}$ 
10:    Taking  $\mathbf{R}$  as an input, for  $\forall e_i \in E$ , adjust  $\min l_i$ 
        such that for  $\forall e_j \in S_k, c_{ij} = 0$  under  $\mathbf{x}_i^T \mathbf{x}_j = 1$ 
11:     $U \leftarrow U \cup \{e_i \mid e_i \in E, l_i > M\}$ 
12:     $E \leftarrow E \setminus U$ 
13:   end while
14:    $E \leftarrow U$ 
15: end for
16:  $S \leftarrow \{S_1, S_2, \dots, S_m\}$ 
```

### B. Closed-loop Adaptive Status Update

To make the design practical in LoRa networks, the following critical challenges remain to be addressed. First, the imperfect-orthogonality awareness timing should coordinate with the channel scheduling timing. Second, the channel scheduling timing should adapt to the downlink opportunities of all 3 classes (i.e., A, B, C) of end-devices. Third, the design should support the dynamic join/exit of end-devices, etc.

Thus, we propose a closed-loop adaptive status update mechanism, as shown in Algorithm 4. The notations  $TS_{ref}$ ,  $TS_{last}$  depict the current and last reference timestamps, resp.  $T_{curr}$ ,  $T_{next}$  denote the current and next channel scheduling time interval, resp.  $\{T_1, T_2, \dots\}$  denote the transmission period set of currently joined end-devices.  $\eta$  is a constant that can impact downlink CCI packet length, CCI downlink requests success ratio (the ratio of the successful downlink CCI requests to the total downlink CCI requests), etc. To decide the value of  $\eta$ , we adopt the Monte Carlo method to simulate the downlink CCI distribution process for Class A end-devices. The highest CCI downlink requests success ratio is achieved when  $\eta = 4$ .

The designed mechanism fully considers the downlink communication limitation of Class A end-devices, i.e., the downlink packet is received only in one of the two available windows after an uplink transmission. Moreover, it can adaptively adjust the length of channel scheduling time interval according to the dynamic join/exit of end-devices, without affecting the operation of imperfect-orthogonality awareness.

### C. Clock Drift Correction

The base station requires to predict the start time of each transmission event during the next channel scheduling time

---

### Algorithm 4 Basic Closed-loop Adaptive Status Update

---

```

1:  $TS_{last} \leftarrow TS_{ref} \leftarrow$  current reference timestamp
2:  $T_{next} \leftarrow T_{curr} \leftarrow$  non-zero initial time interval
3: while true do
4:    $T_{next} \leftarrow \eta \max\{T_1, T_2, \dots\}$ 
5:    $E_{cs} \leftarrow$  predict events in  $[TS_{ref} + T_{curr}, TS_{ref} + T_{curr} + T_{next}]$ 
6:   Update  $\mathbf{R}$  by performing imperfect-orthogonality
      awareness for scheduled events  $E_{ps}$  in  $[TS_{last}, TS_{ref}]$ 
7:   Perform channel scheduling solution for  $E_{cs}$  with  $\mathbf{R}$ 
8:   Sleep until time  $TS_{ref} + T_{curr}$ 
9:    $TS_{last} \leftarrow TS_{ref}$ 
10:   $TS_{ref} \leftarrow TS_{ref} + T_{curr}$ 
11:   $T_{curr} \leftarrow T_{next}$ 
12: end while
```

---

interval. However, due to the clock of each end-device trends to drift, the prediction will introduce error.

To this end, the same scheme as S-MAC [4] is adopted to sense the clock drift of each end-device, including phase offset estimation and clock skew compensation. On the other hand, considering the prediction precision, if a transmission event  $e_i$  is predicted to occur in the interval  $[s_i, f_i]$ , and the error is  $\varepsilon$ , we update its transmission interval to  $[s_i - \varepsilon, f_i + \varepsilon]$ .

### D. Dynamic Join, Exit, and Others

For network accesses of end-devices, X-MAC supports both modes proposed by LoRaWAN, i.e., OTAA and ABP [28].

For newly joined end-devices, the base station dynamically adjusts its channel scheduling time interval, and corrects their clock drifts, so that they can get valid CCI steadily.

For the exit of end-devices, the base station traces each end-device's uplink and downlink packets to determine whether any end-device has exited. Once detected, the base station will remove the exited end-device from the joined list, and the channel scheduling time interval is also adaptively adjusted.

For downlink, once receiving an uplink packet from a Class A end-device, the base station, according to its reserved field, transmits the downlink packet with the required CCI of subsequent transmission events in one of the receive windows. It improves the reliability while reducing the downlink pressure.

## VI. EVALUATION

### A. Experiment Setup

**Experimental platform:** The testbed consists of EDs, a GW and a NS. An ED is mainly composed of a microcontroller unit (MCU) STM32L152, a transceiver chip SX1278 and a Li-Po battery. The full-duplex GW supports 8 parallel uplink channels and 1 downlink channel in CN470-510MHz band, whose hardware architecture is the same as that of S-MAC [4]. Specifically, the uplink employs a SX1301 digital baseband chip and two SX1255 RF front-end chips, the downlink utilizes a SX1278 transceiver chip, and they are controlled by a Raspberry Pi 3 via 2 serial peripheral interfaces (SPI). The NS adopts a laptop with an Intel Core i5 CPU and 8GB RAM.

**Scenarios:** We evaluate the performance and overhead of X-MAC in the indoor office in Fig. 5(a), outdoor university

campus in Fig. 5(b), 5(d), and mixed indoor-outdoor in Fig. 5(c). In these scenarios, 50 EDs and a BS are deployed, resp. All EDs operate as Class A devices to evaluate the lower-bound performance of X-MAC since they have the fewest downlink opportunities. All EDs periodically transmit unconfirmed packets. We set the default bandwidth (BW), coding rate (CR), packet length, and transmission power (TP) as 125kHz, 4/5, 40 bytes, and 14 dBm, resp. Each experiment under each configuration lasts at least 120 minutes and is conducted 5 times in different times of the day over 5 months.

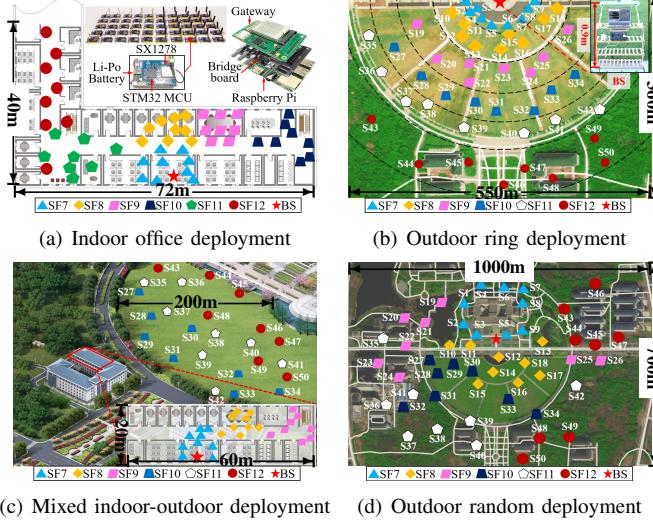


Fig. 5. Multiple experimental fields

**Method:** Evaluating the scalability of large-scale LoRa networks in practice is challenging as the manufacture, configuration, deployment, testing, and maintenance will be prohibitively expensive [19]. In view of this, the method widely accepted by existing research is to emulate a larger number of EDs by altering the transmission periods of real-world EDs [4], [5], [20], and we also adopt this method.

Specifically, we assume a real-world ED transmits every 15 minutes as a standard transmission period (relative reference value only, other values are acceptable). Thus, a real-world ED that transmits  $\ell$  ( $\ell > 0$ ) times periodically in 15 minutes, emulates  $\ell$  LoRa EDs. For instance, if the transmission period of 50 real-world EDs is set to 6s, it means the relative number of concurrent EDs (#end-devices) is 7500 (15minutes\*60s/6s\*50EDs). Thus, #end-devices is used to denote the relative number of concurrent EDs, which corresponds to a transmission period of 50 real-world EDs in our experiments.

**Baseline:** X-MAC is compared with the following state-of-the-art methods:

- LoRaWAN [22]: an unslotted ALOHA based method, where an end-device randomly selects a frequency channel from a list of available channels for each transmission.
- S-MAC [4]: a pure channel scheduling based method. Similarly to X-MAC, it also contains both strict and loose models for different application scenarios. Limited by the transmission periods of end-devices, we set  $M$  to 4s.
- LMAC [5]: a channel activity detection based method. The best configuration “Level 3” provided by the authors

of LMAC is employed. Each packet is only allowed to be transmitted in its own transmission period.

## B. PRR Performance of X-MAC

1) *PRR in the indoor office scenario:* We conduct an indoor experiment shown in Fig. 5(a) to evaluate the packet reception rate (PRR, the ratio of received packets to the total transmitted packets) of X-MAC with the increase of end-devices.

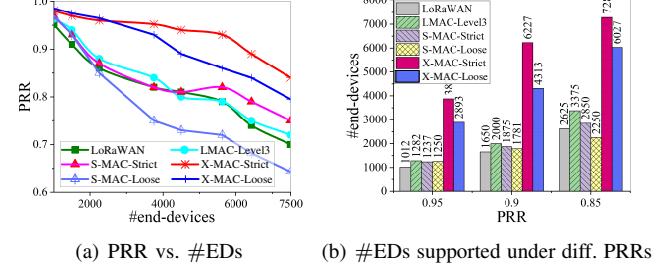


Fig. 6. PRR of X-MAC in the indoor scenario

As shown in Fig. 6(a), X-MAC achieves a significant PRR performance boost compared with the state-of-the-art methods. Furthermore, Fig. 6(b) indicates the relative number of end-devices supported under different PRRs of 95%, 90%, and 85%. With PRR requirement of > 95%, X-MAC brings up the number of concurrent end-devices by 2.02× compared with the state-of-the-art methods.

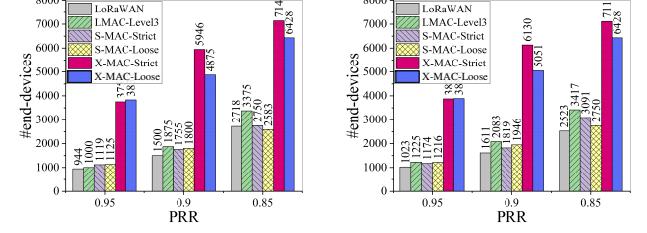


Fig. 7. PRR of X-MAC in the outdoor scenarios

2) *PRR in the outdoor scenarios:* We conduct outdoor experiments with the scenarios in Fig. 5(b) and 5(d), respectively, to evaluate the PRR of X-MAC. As shown in Fig. 7, with PRR requirement of > 95%, X-MAC increases the number of concurrent end-devices by 2.41× compared with the state-of-the-art methods, and 2.97× compared with LoRaWAN in the ring deployment scenario. X-MAC boosts the number of concurrent end-devices by 2.17× compared with the state-of-the-art methods in the random deployment scenario.

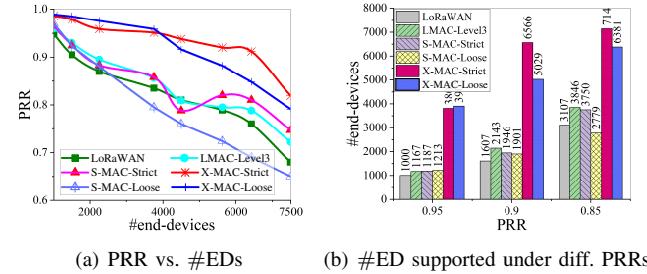


Fig. 8. PRR of X-MAC in the mixed indoor-outdoor scenario

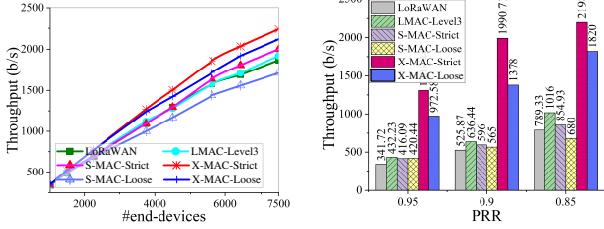
3) *PRR in the mixed indoor-outdoor scenario:* We also conduct an experiment shown in Fig. 5(c) to evaluate the PRR of X-MAC. As illustrated in Fig. 8, with PRR requirement of

> 95%, X-MAC brings up the number of concurrent end-devices by  $2.22\times$  compared with the state-of-the-art methods.

To sum up, the PRR results in Fig. 6 - 8 highlight the following observations. First, the PRR of S-MAC and LMAC drastically drops to similar capability as LoRaWAN in these multi-SFs decentralized deployed scenarios. Second, X-MAC achieves similar superior PRR performance in the above experimental scenarios where SFs are distributed from near to far. Third, the PRR of X-MAC using the loose model is not always better than that using the strict model. It is confirmed that the performance of the loose scheduling algorithm always beats the strict scheduling algorithm (Section VI-D). The root is that the downlink CCI distribution efficiency is weaker than that in the strict scheduling scenario. It is expected to be further improved by reducing the resolution of transmission offset, employing multi-gateway for downlink, etc.

### C. Throughput Performance of X-MAC

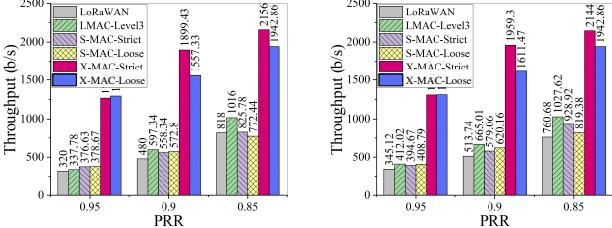
1) *Throughput in the indoor office scenario:* We evaluate the throughput of X-MAC with the scenario in Fig. 5(a). As shown in Fig. 9(a), X-MAC achieves the best throughput in the indoor environment. Furthermore, the throughput under different PRRs of 95%, 90%, and 85% is given in Fig. 9(b). With PRR requirement of > 95%, the network throughput supported by X-MAC reaches 1308.44 bps, bringing up to 2.03× improvement compared with the state-of-the-art methods.



(a) Throughput vs. #end-devices (b) Throughput under diff. PRRs

Fig. 9. Throughput of X-MAC in the indoor scenario

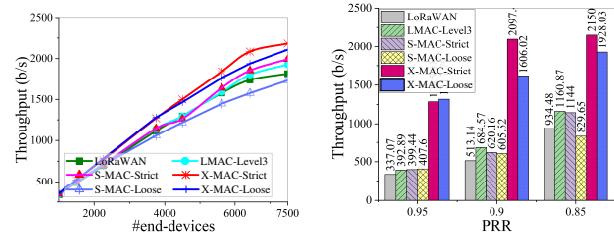
2) *Throughput in the outdoor scenarios:* We evaluate the throughput of X-MAC with the scenarios in Fig. 5(b) and 5(d), resp. As shown in Fig. 10, X-MAC also wins the best throughput. With PRR requirement of > 95%, X-MAC achieves  $\sim 1.3$  Kbps network throughput, which boosts  $2.18\times$  -  $2.42\times$  improvement compared with the state-of-the-art methods.



(a) The ring deployment scenario (b) The random deployment scenario

Fig. 10. Throughput of X-MAC in the outdoor scenarios

3) *Throughput in the mixed indoor-outdoor scenario:* We also evaluate the throughput of X-MAC with the scenario in Fig. 5(c). The results are illustrated in Fig. 11. With PRR requirement of > 95%, X-MAC brings up to  $2.23\times$  throughput improvement compared with the state-of-the-art methods.



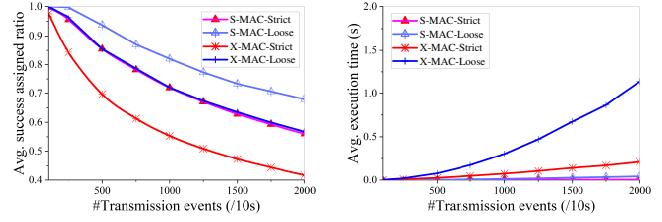
(a) Throughput vs. #end-devices

(b) Throughput under diff. PRRs

Fig. 11. Throughput of X-MAC in the mixed indoor-outdoor scenario

### D. Performance and Efficiency of Channel Scheduling

To evaluate the channel scheduling algorithms, we randomly generate different numbers of events in 10s interval, perform each algorithm in turn, and count their avg. success assigned ratio and the corresponding avg. execution time. As shown in Fig. 12(a), the algorithms' performance of X-MAC is worse than that of S-MAC, but the PRR and throughput of X-MAC evaluated earlier are much better than that of S-MAC. Thus, it can be seen that the imperfect-orthogonality aware scheme of X-MAC makes a great contribution to improving network scalability. Fig. 6(b) indicates the efficiency of X-MAC-Strict is very close to that of S-MAC, and the efficiency of X-MAC-Loose is also acceptable in most single-GW scenarios.



(a) Performance of algorithms

(b) Efficiency of algorithms

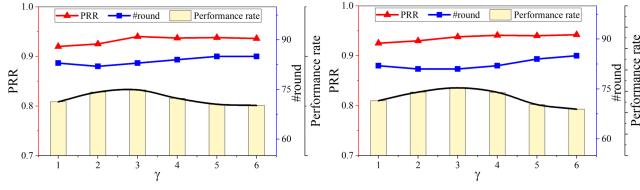
Fig. 12. Performance and efficiency of channel scheduling algorithms

### E. Inter-SF Collision Threshold Setup of X-MAC

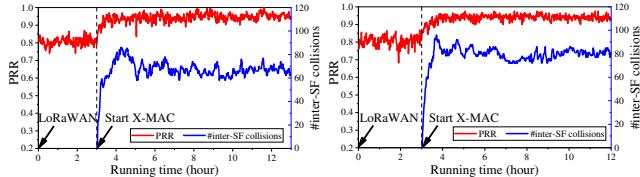
We perform indoor and mixed indoor-outdoor experiments shown in Fig. 5(a) and 5(c) respectively to determine the inter-SF collision threshold  $\gamma$  in X-MAC. The transmission period for all 50 end-devices is set to 10 seconds. The  $\gamma$  can affect PRR and inter-SF collision detection efficiency, and Fig. 13 shows the results of these metrics under various  $\gamma$ . The collision detection efficiency is defined by “#round”, which denotes the scheduling rounds required to receive a total of  $N_p$  packets ( $N_p = 15,000$  here, other large values are acceptable). The “performance rate” is the ratio of PRR to the required #round. We observe that PRR is stable when  $\gamma \geq 3$ , and the best performance ratio is achieved with  $\gamma = 3$ .

### F. Stability and Reliability of X-MAC

We then further conduct indoor and mixed indoor-outdoor experiments shown in Fig. 5(a) and 5(c) respectively to evaluate the stability and reliability of X-MAC under  $\gamma = 3$ . The transmission period for all 50 end-devices is also set to 10s. The network server runs the native LoRaWAN for 3 hours, and then starts X-MAC. As shown in Fig. 14, the system rapidly detects the inter-SF collisions after X-MAC is started, and the PRR boosts from  $\sim 80\%$  to  $94\%$ . In addition, with the change of time and environment, the detected inter-SF collisions fluctuate slightly, while the PRR remains stable.



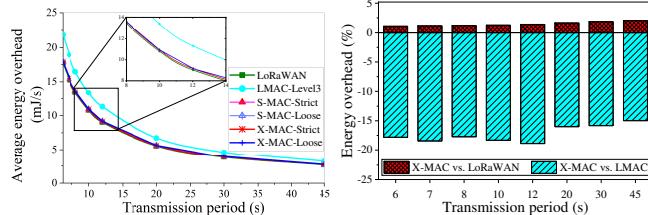
(a) The indoor office scenario (b) The mixed indoor-outdoor scenario  
Fig. 13. Performance vs. diff.  $\gamma$  in two scenarios



(a) The indoor office scenario (b) The mixed indoor-outdoor scenario  
Fig. 14. The stability and reliability of X-MAC

#### G. Energy Consumption Overhead of X-MAC

We conduct experiments with scenarios in Fig. 5(a) and 5(c) respectively to evaluate the energy consumption of transceivers in end-devices under various transmission periods. A portable power monitor with  $\mu$ A-resolution is employed to estimate the average energy. Fig. 15 shows the energy consumption results of mixed indoor-outdoor scenario (due to similar results and limited space, the figure of another scenario is omitted). We observe that X-MAC only increases the energy consumption by  $\sim 1\%$  to  $2\%$  compared with LoRaWAN, while it decreases by  $\sim 15\%$  to  $19\%$  compared with LMAC. In addition, an end-device can employ a ring buffer with tens of bytes to access the CCI, which can also be met by almost all MCUs.



(a) Energy vs. transmission period (b) X-MAC vs. LoRaWAN, LMAC  
Fig. 15. Energy consumption of transceivers in mixed indoor-outdoor scenario

#### VII. RELATED WORK

To achieve high scalability for LoRa networks, researchers introduce various approaches to resolve or avoid collisions.

**PHY-layer approaches.** Eletreby *et al.* [1] exploited nodes' natural hardware offsets to resolve collisions. Xia *et al.* [2] considered the time and frequency domain features to separate collided packets. Xia *et al.* [17] employed multiple antennas of a gateway to scale the concurrent transmissions. Tong *et al.* [3], [36] and Xu *et al.* [37] utilized subtle packet time offset to decompose multiple packets. Zhang *et al.* [38] proposed an interference cancellation method based on the segmentation of received signal. These works can resolve collisions and support more concurrent transmissions, but they require direct access to PHY samples and are not feasible on commodity chips.

**MAC-layer approaches.** Gamage *et al.* [5] avoided transmission collision by detecting whether the specified CF/SF

channel was idle. It mitigated collisions and balanced loads of channels, but it failed to detect transmissions with different SFs, which still met severe packet loss in practical LoRa networks. Xu *et al.* [4] utilized pure channel scheduling to avoid collisions. It improved the scalability in many scenarios, but it did not consider potential collisions among different SFs, which still encountered large collisions in decentralized deployments. Other methods, such as tuning TPs, SFs, CFs, or timeslots [6]–[11], [30], performed well in some scenarios, but they may introduce excessive communication overhead, end-devices cost, energy consumption, or hardware complexity.

#### VIII. DISCUSSION

**Transmission power control scheme.** An alternative to solve inter-SF collisions is to tune the TP of EDs close to (or away from) the GW, such that both transmissions are received by the GW. To achieve this, one way is to employ the channel propagation model and collision condition to decide a TP for each ED. However, it relies heavily on rich prior knowledge and empirical analysis, and is easily affected by the ambient environment. The other way is to schedule the TP for each ED. However, according to our experiments in real environment, it can cause long-term sharp fluctuations in PRR, especially there are large-scale EDs in the network or EDs join dynamically.

**Fast fading resistance.** The severe burst drop in signal strength, or a rapid change in channel conditions, may cause packet loss. However, X-MAC can cope with this problem partly. It blames these packet losses on inter-SF collisions and avoids these collisions in the channel scheduling of subsequent events. Furthermore, once the channel conditions are stabilized, this mistake will be corrected by the imperfect-orthogonality aware scheme. Therefore, X-MAC achieves a good resistance to fast fading. The endurance experiments conducted in different scenarios (Section VI-F) achieve stable PRR and #inter-SF collision, which also partly confirms this.

#### IX. CONCLUSION

This paper presents X-MAC, the first scheduler that considers imperfect orthogonality in practical LoRa networks. We propose an imperfect-orthogonality aware scheme to detect inter-SF collisions without much prior knowledge. Then, we design strict and loose channel scheduling algorithms to adapt to different scenarios. Further, we offer additional designs to make it practical. The evaluation shows X-MAC improves the network scalability by  $2.41\times$  with PRR requirement of  $> 95\%$ .

In the future work, we plan to extend the work to multi-GW scenarios. Further, we will explore achieving high scalability under traffic with properties such as burstiness, self-similarity.

#### ACKNOWLEDGMENT

We thank the shepherd and anonymous reviewers for insightful comments. This work was supported by National Key R&D Program of China 2018AAA0100500, Fundamental Research Funds for the Central Universities, NSFC under Grants No. 62172091, 61602112, 61632008, 61972085, and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CityU 21216822).

## REFERENCES

- [1] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*. ACM, 2017, pp. 309–321.
- [2] X. Xia, Y. Zheng, and T. Gu, "FTrack: parallel decoding for LoRa transmissions," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems (SenSys)*, 2019, pp. 192–204.
- [3] S. Tong, Z. Xu, and J. Wang, "CoLoRa: Enabling Multi-Packet Reception in LoRa," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2020, pp. 2303–2311.
- [4] Z. Xu, J. Luo, Z. Yin, T. He, and F. Dong, "S-MAC: Achieving high scalability via adaptive scheduling in LPWAN," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2020, pp. 506–515.
- [5] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for lora," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2020, pp. 1–13.
- [6] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.
- [7] J. Lee, W.-C. Jeong, and B.-C. Choi, "A scheduling algorithm for improving scalability of LoRaWAN," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 1383–1388.
- [8] M. Centenaro and L. Vangelista, "Boosting network capacity in LoRaWAN through time-power multiplexing," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 1–6.
- [9] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3097–3109, 2018.
- [10] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of lorawans through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, 2018.
- [11] M. Centenaro and L. Vangelista, "Time-power multiplexing for LoRa-based IoT networks: an effective way to boost LoRaWAN network capacity," *International Journal of Wireless Information Networks*, vol. 26, no. 4, pp. 308–318, 2019.
- [12] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *2016 Symposium on Communications and Vehicular Technologies*. IEEE, 2016, pp. 1–5.
- [13] D. Croce, M. Guicciardo, S. Mangione, G. Santaromita, and I. Timmirello, "Impact of LoRa imperfect orthogonality: Analysis of link-level performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, 2018.
- [14] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in lora networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 788–798.
- [15] A. Balanuta, N. Pereira, S. Kumar, and A. Rowe, "A cloud-optimized link layer for low-power wide-area networks," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2020, pp. 247–259.
- [16] L. Amichi, M. Kaneko, E. H. Fukuda, N. El Rachkidy, and A. Guitton, "Joint allocation strategies of power and spreading factors with imperfect orthogonality in LoRa networks," *IEEE Transactions on Communications (TCOM)*, vol. 68, no. 6, pp. 3750–3765, 2020.
- [17] X. Xia, N. Hou, Y. Zheng, and T. Gu, "PCube: scaling LoRa concurrent transmissions with reception diversities," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2021, pp. 670–683.
- [18] V. Toro-Betancur, G. Prem sankar, M. Slabicki, and M. Di Francesco, "Modeling communication reliability in LoRa networks with device-level accuracy," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2021, pp. 1–10.
- [19] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*. ACM, 2016, pp. 59–67.
- [20] 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 (TOSN)*, vol. 15, no. 2, p. 16, 2019.
- [21] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, "Throughput, Coverage and Scalability of LoRa LPWAN for Internet of Things," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–10.
- [22] L. Alliance, "LoRaWAN 1.0.2 Specification," *LoRa Alliance*, 2016.
- [23] S. Demetri, M. Zúñiga, G. P. Picco, F. Kuipers, L. Bruzzone, and T. Telkamp, "Automated estimation of link quality for LoRa: A remote sensing approach," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2019, pp. 145–156.
- [24] L. Liu, Y. Yao, Z. Cao, and M. Zhang, "Deeplora: Learning accurate path loss model for long distance links in lpwan," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications (INFOCOM)*, 2021.
- [25] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, "LoRa throughput analysis with imperfect spreading factor orthogonality," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 408–411, 2018.
- [26] "The Things Network," <https://www.thethingsnetwork.org/>, 2021, [Online; accessed 20-May-2022].
- [27] N. Blenn and F. Kuipers, "LoRaWAN in the wild: Measurements from the things network," *arXiv preprint arXiv:1706.03086*, 2017.
- [28] "LoRaWAN 1.1 Regional Parameters," <https://lora-alliance.org/resource-hub/lorawanr-regional-parameters-v11rb>, 2018, [Online; accessed 20-May-2022].
- [29] "Semtech SX1302CFDGW1," <https://www.semtech.com/products/wireless-rf/lora-core/sx1302cfdgw1>, 2021, [Online; accessed 20-May-2022].
- [30] M. Hamnache, R. Kacimi, and A.-L. Belyot, "L3fsa: Load shifting strategy for spreading factor allocation in lorawan systems," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 216–224.
- [31] Y. Wang, S. Cai, and M. Yin, "Two efficient local search algorithms for maximum weight clique problem," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 30, no. 1, 2016.
- [32] C. Lu, J. X. Yu, H. Wei, and Y. Zhang, "Finding the maximum clique in massive graphs," *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1538–1549, 2017.
- [33] N. Bansal, "Approximating independent sets in sparse graphs," in *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms (SODA)*. SIAM, 2014, pp. 1–8.
- [34] N. Bansal, A. Gupta, and G. Guruganesh, "On the Lovász theta function for independent sets in sparse graphs," *SIAM Journal on Computing (SICOMP)*, vol. 47, no. 3, pp. 1039–1055, 2018.
- [35] A. Frieze and M. Karoński, *Introduction to random graphs*. Cambridge University Press, 2016.
- [36] S. Tong, J. Wang, and Y. Liu, "Combating packet collisions using non-stationary signal scaling in LPWANs," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2020, pp. 234–246.
- [37] Z. Xu, P. Xie, and J. Wang, "Pyramid: Real-time lora collision decoding with peak tracking," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2021, pp. 1–9.
- [38] Q. Zhang, I. Bizon, A. Kumar, A. B. Martinez, M. Chafii, and G. Fettweis, "A Novel Approach for Cancellation of Nonaligned Inter Spreading Factor Interference in LoRa Systems," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 718–728, 2022.