

S-MAC: Achieving High Scalability via Adaptive Scheduling in LPWAN

Zhuqing Xu¹, Junzhou Luo¹, Zhimeng Yin², Tian He^{1,2}, Fang Dong¹

¹*School of Computer Science and Engineering, Southeast University*

²*Department of Computer Science and Engineering, University of Minnesota*

Email: {zqxu, jluo, fdong}@seu.edu.cn, {yinxx283, tianhe}@umn.edu

Abstract—Low Power Wide Area Networks (LPWAN) are an emerging well-adopted platform to connect the Internet-of-Things. With the growing demands for LPWAN in IoT, the number of supported end-devices cannot meet the IoT deployment requirements. The core problem is the transmission collisions when large-scale end-devices transmit concurrently. The previous research mainly includes transmission scheduling strategies, collision detection and avoidance mechanism. The use of these existing approaches to address the above limitations in LPWAN may introduce excessive communication overhead, end-devices cost, power consumption, or hardware complexity. In this paper, we present S-MAC, an adaptive MAC-layer scheduler for LPWAN. The key innovation of S-MAC is to take advantage of the periodic transmission characteristics of LPWAN applications and also the collision behaviour features of LoRa PHY-layer to enhance the scalability. Technically, S-MAC is capable of adaptively perceiving clock drift of end-devices, adaptively identifying the join and exit of end-devices, and adaptively performing the scheduling strategy dynamically. Meanwhile, it is compatible with native LoRaWAN, and adaptable to existing Class A, B and C devices. Extensive implementations and evaluations on commodity devices show that S-MAC increases the number of connected end-devices by $4.06 \times$ and improves network throughput by $4.01 \times$ with PRR requirement of $> 95\%$.

I. INTRODUCTION

In recent years we have witnessed Low Power Wide Area Networks (LPWAN) emerged as an attractive communication platform for the Internet of Things (IoT). LoRa [1] is one of the most popular LPWAN technologies which has attracted much attention from both academia and industry. LoRa is a proprietary PHY-layer protocol, while LoRaWAN [2] is the corresponding MAC-layer protocol.

However, the number of connected end-devices is very limited in LoRaWAN networks, which cannot meet the requirements of future LPWAN applications [3]–[8]. The root cause of this problem lies in the transmission collisions, when large-scale end-devices transmit concurrently. With the increase of end-devices, the probability of transmission time overlaps increases, but the channel resources provisioned for transmission are limited, which results in large-scale transmission collisions.

Existing literature on this problem can be summarized into the following two categories: PHY-layer approaches and MAC-layer approaches. The researches on the former mainly include Choir [6], FTrack [7], etc., but they cannot work on commodity baseband chips. The researches on the latter are

mainly include transmission scheduling strategies [9]–[16], collision detection and avoidance mechanisms [17]–[20].

Although the above MAC-layer approaches has significant effects in certain scenarios, they still cannot adapt to the requirements of LPWAN scenarios, which may introduce excessive communication overhead, end-devices cost, power consumption, or hardware complexity. The problem of achieving scalability in LPWAN is still challenging for three reasons: (a) **expensive collision detection**. Low-power long-range communication causes serious path loss, which leads to carrier sensing failure. Furthermore, LoRa signals at receiver are often below the noise floor, which makes sensing LoRa signals time-consuming, resulting in low efficiency and high cost. (b) **multi-channel base station**. Although the parallel demodulation capability of the base station is improved by introducing the spreading factor (SF) in LoRa, end-device has to make a trade-off in terms of range, data rate, power consumption, collision probability. (c) **low communication frequency**. Low-power IoT end-devices usually have limited power budget, but they need to support a battery lifetime up to 10 years [2], so that they cannot interact with the base station frequently, which makes existing methods, such as [13]–[16], not really applicable in LPWAN.

In this paper, we propose the first scheduler based on the traffic characteristics of LPWAN applications and transmission features of LoRa PHY-layer to improve scalability in LPWAN. This is inspired by the observation that, in many application scenarios, such as metering, monitoring, and tracking, the transmission of end-device has periodic characteristics. By analyzing the transmission behaviour of a large number of end-devices in the real-live large-scale The Things Network (TTN) [21], the above idea is also confirmed. On the other hand, there is a key feature in LoRa PHY-layer that transmissions with different SF are collision-free. Thus, the key idea is as follows: the transmission time of end-devices during certain time periods can be predicted according to their periodic transmission characteristics. Meanwhile, by utilizing the SF information in the received packets, channels can be assigned efficiently for each transmission event. Therefore, the problem of improving scalability in LPWAN is modeled as a channel optimization scheduling problem based on end-devices transmission periodicity and LoRa PHY-layer collision behaviour.

Specifically, this work proposes S-MAC, a MAC-layer adaptive scheduler in LPWAN scenario. For the scenario with strict

periodicity requirement, *a strict scheduling model* is proposed to recommend the channel of each packet for each end-device. For the scenario with loose periodicity requirement, *a loose scheduling model* is presented to recommend the channel and the corresponding transmission offset of each packet for each end-device. Our major contributions can be summarized as follows:

- We design and implement S-MAC, a MAC-layer scheduler based on the periodic characteristics of LPWAN applications and transmission features of LoRa PHY-layer. This is mainly achieved by base stations' adaptive clock drift sensing, adaptive end-devices join and exit identification, and adaptive dynamic execution of scheduling strategy. More importantly, it is fully compatible with the native LoRaWAN protocol stack.
- A novel strict scheduling model is proposed for the scenario with strict periodicity requirement, while a loose scheduling model is presented for the scenario with loose periodicity requirement, which reduce transmission collision and maximize the packet reception rate.
- We implement S-MAC for extensive evaluations on its performance under various environment and parameter settings. Our experiment results show that S-MAC can increase the number of connected end-devices by $4.06 \times$ and improve network throughput by $4.01 \times$ with PRR requirement of $> 95\%$.

II. MOTIVATION

There are massive end-devices share limited communication medium in LPWAN. As the number of end-devices increases, the probability of transmission collision increases, such that the concurrency communication capability of end-devices is greatly limited. By analyzing applications in TTN network, it is found that the transmission of end-devices in most applications has periodic characteristics. On the other hand, by studying the transmission parameters of LoRa PHY-layer, we found that simultaneous transmissions only cause a collision when they both select the same SF and carrier frequency (CF), respectively. These two key findings inspired our original idea: by considering the periodic characteristics of LPWAN applications, and combining transmission features of LoRa PHY-layer, high scalability in LPWAN could be achieved. Our motivation as learned from the literature is detailed below:

Scalability Limitation of End-devices. Due to the huge transmission collisions, the current LPWAN technologies, especially LoRa, cannot meet the requirement of concurrent communication capability for large number of end-devices. Bor *et al.* [3] investigates the capacity limits in LoRa networks by developing models describing LoRa communication behaviour, and finds that a typical smart city deployment can only support 120 nodes per 3.8 ha. Haxhibeqiri *et al.* [22] creates a simulation model for evaluating the scalability of single gateway LoRaWAN network, and shows that as the number of end-devices increases to 1000, the loss is up to 32%. Liando *et al.* [4] utilizes several real LoRa end-devices to emulate large-scale end-devices concurrent transmission, and

concludes that when each end-device periodically transmits 120 packets within 15 minutes, the packet loss rate is 30%. As a result, the number of end-devices supported by each base station is not sufficient for IoT deployments.

Traffic Characteristics of End-devices. TTN is a real crowd-funded, crowd-sourced and crown-owned large-scale LoRaWAN network, which has been widely deployed around the world. Blenn *et al.* [23] captured data from TTN network for 8 months, and obtained 17,467,312 packets sent by 1,618 end-devices. By calculating the time difference between two consecutive unique packets with the above dataset, the periods of each end-devices was extracted, and a strong tendency to periodic behavior can be found. Approximately 93.7% of end-devices in TTN network is periodic [24], and this paper mainly focus on the scenarios of transmission with periodicity.

Transmission Features of LoRa PHY-layer. Spreading factor (SF) is the ratio between the symbol rate and chip rate [3], while carrier frequency (CF) is the centre frequency. They are two important transmission parameters in LoRa PHY-layer. Collisions only occur when the following conditions are met: end-devices use the same SF and CF respectively, and the transmission time overlaps. The change of SF will affect the communication distance, power consumption, etc., while the choice of CF is relatively free, which leaves us an opportunity to design scheduling algorithms to improve scalability.

In addition, with the improving scalability of LPWAN, it will trigger a lot of interesting applications for low-power, long-range and low-cost communications, such as smart cities, industrial assets monitoring, smart agriculture, smart grid and smart metering and wildlife monitoring and tracking, etc. Semtech [25] has standardized LoRa technology in 2015, and since has deployed it in 42 countries.

III. OVERVIEW OF S-MAC

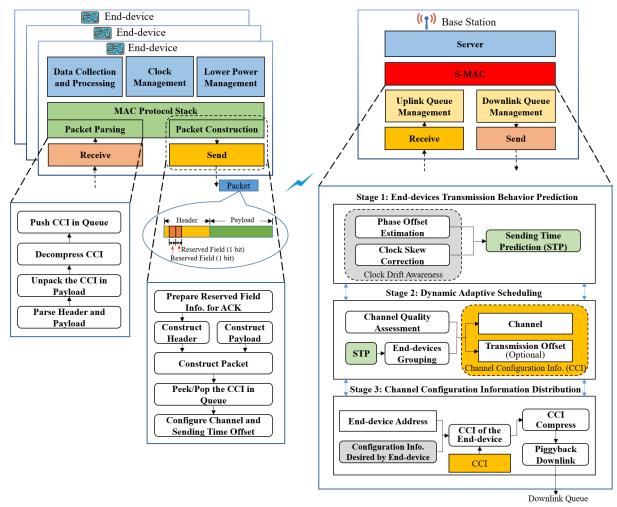


Fig. 1. The architecture of the S-MAC

S-MAC is a MAC-layer adaptive scheduler in LPWAN scenario. Fig. 1 illustrates how it works: First, end-devices periodically sends packets. Secondly, the base station predicts

the sending time of each end-device during certain time periods according to the transmission characteristics of each end-device, and performs dynamic adaptive scheduling. During a downlink window, the corresponding channel configuration information (CCI) is distributed to each end-device in a piggyback pattern. Finally, after receiving a downlink message, the end-device parses the CCI and stores it in a queue for applying when sending the subsequent packets.

For each end-device, the information of transmission period is carried in the stage of joining request. In the stage of data uplink, in order to be compatible with the native LoRaWAN protocol, we only use the reserved fields in packet header to identify and confirm the relevant status information. On the other hand, when receiving a downlink message which carries the CCI, the end-device will decompress it and push it to the queue. And when transmitting, it prefers the channel and transmission offset recommended in the CCI.

For base stations, in order to avoid transmission collisions of massive connected end-devices, the mechanisms taken by the base station mainly include: firstly, by utilizing status information of the received packets, the clock offset can be perceived and the transmission time of their subsequent packets can also be predicted. Secondly, End-devices Grouping algorithm is performed to recommend the channel and the corresponding transmission offset (optional) of each packet for each end-device. The core mechanism of the algorithm is that, in combination with the SF information provided by LoRa PHY-layer, each transmission event is effectively grouped and a unique channel (CF) is assigned to each group, such that there is no collision among the transmission events in the same group. Through the reference to SF, the number of end-devices that can be supported is further improved. Meanwhile, the quality of each channel may be different, so channel quality assessment is performed to evaluate the reliability of data transmission in each channel, so as to allocate a unique channel for different groups. Finally, according to the address of each end-device and the required CCI, the corresponding CCI is compressed and sent in a piggyback pattern.

IV. DESIGN

This section presents the detailed design of S-MAC.

A. Background

1) LoRaWAN Overview: LoRaWAN is a popular MAC-layer protocol stack, which is used by the vast majority of networks based on LoRa connectivity. The network architecture proposed by LoRaWAN specification [2] is shown in Fig. 2. It is a star topology over a single hop connection. The gateway is responsible for receiving messages from end-devices and forwarding them to the NetServer over an IP-based network. The NetServer is responsible for address checking, frame authentication and link configuration adjustment, etc.

LoRaWAN offers 3 types of end-devices, Class A, B and C, to meet the different requirements of various IoT applications. **The critical difference is how and when end-devices receive downlink messages.** For Class A end-devices, the downlink

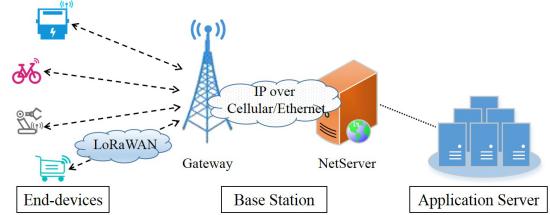


Fig. 2. LoRaWAN architecture

messages can be transmitted in one of the two available receive windows. For Class B end-devices, it is transmitted in one of the two receive windows or in time-synchronized windows. For Class C end-devices, it can be transmitted at any time.

In addition, according to LoRaWAN 1.1 Regional Parameters [26], LoRaWAN plans separate uplink and downlink channels for each regulatory region worldwide. For instance, 72 uplink channels from 902.3-914.2MHz, and 8 downlink channels from 923.3-927.5MHz can be used in the USA.

2) Key Transmission Options and Collision Behaviour: SF and CF are two key parameters in LoRa that affect whether transmission collisions will occur. When transmitting, the end-device will randomly select a CF from the list of available channels in the native LoRaWAN. SF is usually set by adaptive data rate (ADR) mechanism based on factors such as transmission distance and deployment environment, or set as given in advance. LoRa supports SF6 to SF12 to adjust the trade-off among data rate, range and power consumption, but only SF7 to SF12 are used in LoRaWAN.

Each base station is a multi-channel transceiver designed to support simultaneously receiving 8 LoRa packets using random SF on random CF. Since those SFs are orthogonal at the same CF, **simultaneous transmissions only cause a collision when they both select the same SF and CF, respectively.** In other words, when end-devices select different CFs or SFs for transmission, no radio collision occurs. This important PHY-layer feature makes it potential to support massive end-devices.

3) Random Channel Access Mechanism: Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a popular MAC protocol used in WLANs and other short-range wireless networks. However, as mentioned earlier, carrier sensing in low-power long-range communication may fail or be time-consuming, inefficient and expensive in reliably detecting on-going transmissions.

For these reasons, multiple LPWAN technologies, such as LoRaWAN, SigFox, adopt **ALOHA**, a random channel access MAC-layer protocol, in which end-devices do not perform carrier sensing when transmitting. This may lead to high collisions and low channel utilization among different transmissions. Existing works on LoRaWAN channel access mechanism evaluation [27]–[29] have shown that these collisions have a significant impact on communication services with the increase of communication end-devices.

B. The Core Design of S-MAC

The key to support massive and dense connected end-devices in MAC-layer is that the base station is capable of properly guiding the transmission of end-devices to reduce radio collision by utilizing the traffic characteristics of LPWAN applications and transmission features of LoRa PHY-layer. Therefore, the core of S-MAC is about how to design an adaptive scheduler for the base station to help end-devices select their transmission parameters, so as to effectively improve the overall scalability in LPWANs.

1) **Assumptions:** In order to establish the scheduling model accurately, the following reasonable assumptions are given.

- The transmission of end-devices is periodic. As mentioned in the previous section, end-devices have periodic transmission features in most applications, which provides a rational basis for this assumption.
- CF used by end-devices in transmission can be modified at will. In native LoRaWAN, end-devices will randomly select a CF from the list of available channels for each transmission. Thus, the base station can recommend to end-devices which CF to use for each transmission.
- The base station can be designed to have separate uplink and downlink physical channels. As is stated above, the uplink and downlink channels can use different frequency bands in the same region. On the other hand, the hardware design of such a full-duplex base station is also feasible, or it can be achieved by deploying multi-gateway [30].
- The clock precision of end-devices is limited, that is, the transmission periodicity is not very accurate. End-devices are constrained by low cost and low power in LPWAN, and they are vulnerable to ambient environmental factors, such as temperature, humidity.
- SF is not allowed to be modified at will. SF is closely related to transmission distance and power consumption of end-devices, and modifications to it may lead to other side effects. On the other hand, the SF used in each uplink transmission is actually known to the base station, which will help to further avoid transmission collisions.

2) **The Strict Scheduling Model:** First, we describe the strict scenario of end-devices transmission in Fig. 3. Each end-device has a transmission period. The word “*strict*” means that the end-device transmits a packet immediately at the beginning of a transmission period without any delay or advance. s_i^k, f_i^k in Fig. 3 are the start and finish time of the k -th transmission (we call it a transmission event) of end-device i , respectively.

To facilitate further explanation, some notation is defined in Table I. Then the problem can be defined as an optimization:

$$\max \sum_{x=1}^n (q_{k_x} \prod_{y=1, y \neq x}^n C(e_x^{j_x k_x}, e_y^{j_y k_y})) \quad (1)$$

Subject to:

$$C(e_x^{j_x k_x}, e_y^{j_y k_y}) = \begin{cases} 0, & \text{if } j_x = j_y \text{ and } k_x = k_y \text{ and} \\ & (s_x \leq f_y \text{ and } f_x \geq s_y) \\ 1, & \text{else} \end{cases} \quad (2)$$

$$1 \leq k_i \leq K, k_i \in \mathbb{Z}^+, \forall i, e_i \in E \quad (3)$$

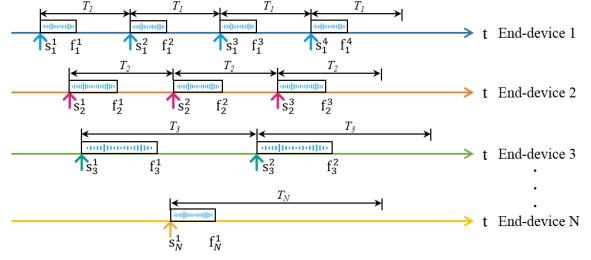


Fig. 3. The strict scenario of end-devices transmission

TABLE I
IMPORTANT NOTATION AND DESCRIPTION

Notation	Description
n	Number of transmission events
K	Number of available channels
e_i	Transmission event i
E	The set of transmission events, $E = \{e_1, e_2, \dots, e_n\}$
e_i^{jk}	The SF value of e_i is j ($7 \leq j \leq 12$, $j \in \mathbb{Z}^+$), and the CF (channel) of e_i is k ($k \in \{1, 2, \dots, K\}$)
s_i	The start time of e_i
f_i	The finish time of e_i , $0 < s_i < f_i$
q_i	The quality of channel i , $0 \leq q_i \leq 1$, $i \in \{1, 2, \dots, K\}$

k_i is the unique independent variable. $C(e_x^{j_x k_x}, e_y^{j_y k_y}) = 1$ in (2) denotes that e_x and e_y are collision-free, and vice versa. The objective in (1) is to assign an appropriate channel for each transmission event, such that the number of successfully delivered packets is maximized. Inequality (3) is used to constrain the range of channel selection for each end-device.

In view of this problem, we can break it into two parts. First, an End-devices Grouping Algorithm is designed to select a set of collision-free transmission events for each of the K groups such that the total number of collision-free events is maximized. Secondly, a Channel Assignment Algorithm is designed to assign a channel for each group such that the number of successfully delivered packets is maximized.

For **End-devices Grouping Algorithm**, suppose we have obtained a group of collision-free transmission events with the same SF, then any other group of collision-free transmission events with different SF can be merged to this group without radio collision. Inspired by this idea, we only need to solve the sub-problem in maximizing the total number of collision-free events in K groups which have the same SF. Then, groups with different SFs can be merged according to the number of elements in each group. Aiming at this sub-problem, there are two key observations that help to solve it optimally.

Theorem 1. Consider any set of non-empty transmission events E with the same SF, and let $e_m \in E$ and $f_m = \min\{f_i \mid e_i \in E\}$. $G = \{g_1, g_2, \dots, g_K\}$ is a set of available time for K groups, then (a) e_m is included in some optimal solution of collision-free events of E . (b) e_k ($e_k \in E$) can be arranged to the group g_n in some optimal solution, where $g_n = \max\{g \mid g \leq s_k, g \in G\}$.

The approach of solving the sub-problem by **Theorem 1**, has been proved to be the optimal solution in the Appendix. The core of the algorithm for the sub-problem is, for each

event, to find a group with the maximum available time from the groups where the event can be arranged. In some cases where events cannot be arranged to any group, channels will not be assigned for such events.

Consequently, an optimal solution is obtained for the End-devices Grouping problem. Suppose the *quicksort* algorithm is adopted (the same below) for sorting elements, then the time complexity of End-devices Grouping Algorithm is $O(n \log n + K \log K)$, and the corresponding space complexity is $O(n \log n)$.

For **Channel Assignment Algorithm**, a greedy approach can be used to assign the channel with higher reliability to the group with larger number of events.

Thus, an optimal solution can be obtained for the Channel Assignment problem, and its time and space complexity are both $O(K \log K)$.

3) The Loose Scheduling Model: In the strict scheduling model, if end-devices use the same SF and CF for transmission respectively, collisions may occur after experiencing the least common multiple (LCM) of their transmission periods. Therefore, in order to further reduce transmission collisions and increase the number and density of connected end-devices, in some application scenarios, the transmission start time of end-devices may be allowed to be appropriately offset. This is termed as “*loose*” accordingly. This section is devoted to have a detailed discussion for such a scenario.

As shown in Fig. 4, the start time of the k -th transmission event of end-device i can be selected in the time interval $[p_i^k, p_i^k + M]$, while in the strict scheduling model, the end-device transmits the packet at the time p_i^k . The start time of the k -th transmission event of end-device i is s_i^k .

Let's define some more notation. e_i^{jkl} indicates that the SF of the corresponding event e_i is j ($7 \leq j \leq 12$, $j \in Z^+$), the channel used is k ($k \in \{1, 2, \dots, K\}$) and transmission offset is l ($0 \leq l \leq M$). p_i is the earliest start time of e_i .

Thus, this problem can also be defined as an optimization:

$$\max \sum_{x=1}^n (q_{k_x} \prod_{y=1, y \neq x}^n C(e_x^{j_x k_x l_x}, e_y^{j_y k_y l_y})) \quad (4)$$

Subject to:

$$C(e_x^{j_x k_x l_x}, e_y^{j_y k_y l_y}) = \begin{cases} 0, & \text{if } j_x = j_y \text{ and } k_x = k_y \text{ and} \\ & (s_x \leq f_y \text{ and } f_x \geq s_y) \\ 1, & \text{else} \end{cases} \quad (5)$$

$$l_i = s_i - p_i, \forall i, e_i \in E \quad (6)$$

$$0 \leq l_i \leq M, \forall i, e_i \in E \quad (7)$$

$$1 \leq k_i \leq K, k_i \in Z^+, \forall i, e_i \in E \quad (8)$$

k_i and l_i are independent variables. $C(e_x^{j_x k_x l_x}, e_y^{j_y k_y l_y}) = 1$ in (5) denotes that events e_x and e_y are collision-free, and vice versa. The objective function in (4) is to arrange appropriate channel and transmission offset for each transmission event, such that the number of successfully delivered packets is maximized. Equation (6) is based on the fact that l_i is the offset between p_i and s_i . Inequalities (7) and (8) restrict the range of independent variables, respectively.

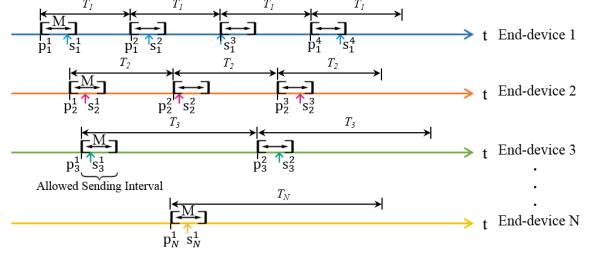


Fig. 4. The loose scenario of end-devices transmission

We break the problem into two parts again. First, an End-devices Grouping with Offset Algorithm is designed to select K groups of transmission events and set a transmission offset for each event, such that the total number of collision-free events is maximized. Secondly, the same Channel Assignment Algorithm as in the strict scheduling model can be performed to assign a channel for each group.

For **End-devices Grouping with Offset Algorithm**, similarly, we only need to solve the sub-problem in maximizing the total number of collision-free and offset-allowed events in K groups which have the same SF. To solve this problem, there is a key observation that determines our direction.

Theorem 2. *The decision problem corresponding to this sub-problem is NP-hard.*

Algorithm 1 End-devices Grouping with Offset

```

1: input:  $E, K$ , max offset value:  $M$ 
2: output: The set of  $K$  group events:  $S$ 
3: Sort the elements in  $E$ 
4: for  $i \leftarrow 1$  to  $K$  do
5:    $S_i \leftarrow \emptyset$ 
6:   while  $E \neq \emptyset$  do
7:      $event \leftarrow$  find the next best event for  $S_i$ 
8:     if  $event \neq \emptyset$  then
9:        $S_i \leftarrow S_i \cup \{event\}$ 
10:      pop  $event$  from  $E$ 
11:    else break
12:    end if
13:   end while
14: end for
15:  $S \leftarrow \{S_1, S_2, \dots, S_K\}$ 

```

The proof of **Theorem 2** can be found in the Appendix. As a result, we propose a greedy strategy based heuristic algorithm, as shown in Algorithm 1. The core ideas are described as follows: (a) Try to arrange all transmission events into one group first, and then arrange remaining transmission events into the second group, etc., so as to make full use of the function of transmission offset to arrange as many transmission events as possible for each group. (b) The transmission event with earlier finish time, earlier start time (if finish time is the same) and larger required offset (if finish and start time are the same, respectively) is prioritized to the group to allow as much time as possible for subsequent transmission events.

The time and space complexity of the above algorithm are $O(Kn^2)$ and $O(n \log n)$, respectively.

C. Dynamic Adaptive Scheduling

This section discusses how and when to perform the scheduling models proposed earlier. There are two key points: (a) Scheduling timing should be considered to adapt downlink opportunities for three types of end-devices. (b) The period of executing scheduling model should be dynamic and adaptive according to the currently accessed end-devices.

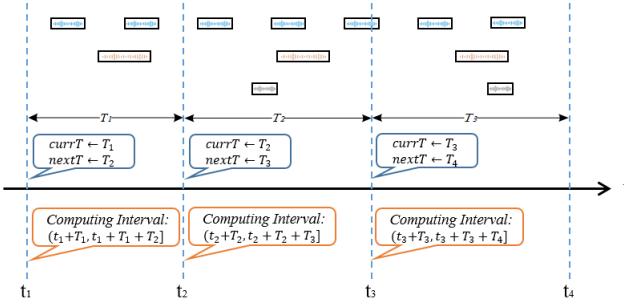


Fig. 5. The dynamic adaptive scheduling mechanism

Thus, the dynamic adaptive scheduling mechanism of base stations is proposed in Fig. 5. $currT$ denotes the current scheduling period, and $nextT$ indicates the next scheduling period. The update formula of $nextT$ is stated as:

$$nextT \leftarrow k \max\{T_1, T_2, \dots, T_n\} \quad (9)$$

where T_1, T_2, \dots, T_n denote the transmission period of currently accessed end-devices, respectively. k is a constant, which is related to factors such as the clock precision of end-devices and deployment environment.

Suppose $currT = T_1$, and $nextT = T_2$ at time t_1 , then the base station will perform the scheduling model at t_1 for transmission events whose start time is in $(t_1 + T_1, t_1 + T_1 + T_2]$. At time t_2 , the base station updates $nextT$ to T_3 for the reason that an end-device with a larger transmission period joins, and performs the scheduling model for events whose start time is in $(t_2 + T_2, t_2 + T_2 + T_3]$, etc. Thus, when an end-device successfully transmits a packet, the CCI of subsequent transmission events can be obtained in its downlink windows.

Therefore, it can be seen that the dynamic adaptive scheduling mechanism can dynamically adjust the scheduling period according to the transmission periodicity of all accessed end-devices, and is applicable not only to Class A devices, but also to Class B and C devices.

D. Clock Drift Awareness

As mentioned above, the base station needs to predict the start time of each transmission event in the next scheduling period. However, due to the clock of end-devices trend to drift, the prediction will introduce errors.

The existing studies show that clock drift can be summarized to two parts: phase offset and clock skew [31]. Thus, to estimate the relative phase offset between any end-device and the base station, the base station records the relevant information from the received packets to obtain the actual transmission

period of the corresponding end-device. To correct the clock skew, skew compensation with linear regression is used, which is the same as that mentioned in [31], [32].

In addition, to maintain the effectiveness of the scheduling model, the prediction precision should also be taken into account. Specifically, if a transmission event is predicted to occur in the interval $[s, f]$, and the error is e , then we update the transmission interval of it to $[s - e, f + e]$.

E. End-devices Join and Exit

In LoRaWAN, end-devices join a network via Over-The-Air Activation (OTAA) or Activation By Personalization (ABP) [26]. For end-devices with OTAA, they will periodically send join request message until receiving a join accept message, and then periodically transmit data message. For end-devices with ABP, they directly start periodic data transmission.

For the newly joined end-devices, according to Section IV-C, the base station will dynamically adjust its scheduling period; according to Section IV-D, the base station will also be aware of their clock drift, so that the newly joined end-devices can get their own CCI stably.

For the exit of end-devices, the base station monitors the uplink and downlink messages of each end-device to determines whether an end-device has exited. The base station will remove the exited end-devices from the list of the accessed end-devices. Meanwhile, as described in Section IV-C, the scheduling period of base station may also be adjusted adaptively, thus the dynamic exit of end-devices is implemented and channel resources are fully utilized.

V. CCI DISTRIBUTION AND RELIABILITY

The distribution of CCI is one of the important factors affecting the performance of the scheduler. On the other hand, reliability can also be achieved by the acknowledgment of messages in the downlink. To effectively distribute CCI, we have mainly taken the following measures:

Downlink ACK. A reserved field in uplink messages is used to confirm whether the last downlink message has been received. After obtaining the information, the base station can determine which part of the CCI needs to be sent to the end-device, instead of repeating the CCI that has been sent. It can not only improve the reliability of the system, but also reduce the downlink pressure. On the other hand, the CCI of the corresponding end-device is compressed before transmission to shorten the length of packet.

Separate Channels. In order to support massive connected end-devices, a full-duplex commodity base station is designed (see Section VI-A for details), which has separate uplink and downlink physical channels, thus ensuring that both uplink and downlink messages can be transmitted simultaneously.

Best-effort Service. For Class A devices, if the downlink message cannot be transmitted in the receive window 1 of the end-device, the base station will attempt to transmit in the receive window 2. For Class B and C devices, similar best-effort downlink services are also provided.

VI. EVALUATION

In this section, we evaluate the performance of S-MAC with other schedulers under various settings.

A. Experiment Setup

The system designs a full-duplex gateway with 8 parallel uplink channels and 1 separate downlink channel (Fig. 6(b)). The uplink transmission is based on a SX1301 digital baseband chip and two SX1255 front-end transceiver chips, while the downlink transmission is based on a SX1276 transceiver chip. The gateway is controlled by a Raspberry-Pi (Fig. 6(c)). The end-device is mainly composed of a transceiver chip SX1278 and a control chip STM32L152 (Fig. 6(a)). All the following experiments are performed in CN470-510MHz band. We deploy them in an industrial park (Fig. 6(d)).

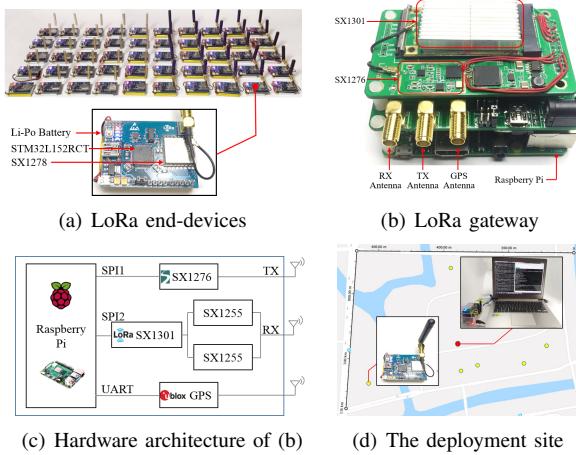


Fig. 6. Hardware utilized for the experimental system

It is not feasible to evaluate the scalability of large-scale LoRa networks in practice as the deployment of such networks will be prohibitively expensive. Furthermore, a real deployment would not allow us to test a larger number of configurations and topologies, which might be necessary for a general evaluation on scalability [3]. Therefore, the commonly used method in existing researches is to emulate a larger number of end-devices through multiple real end-devices [4], and we adopt the same method as the literature.

The experiment setup consisted of 50 end-devices and a single base station. The 50 end-devices were used to emulate a higher number of end-devices by changing their transmission periods. The emulation assumed a network of LoRa end-devices that were transmitted every 15 minutes (other values are also acceptable). Therefore, an end-device that transmitted 100 times periodically in 15 minutes (transmission period changed to 9s), emulated 100 LoRa end-devices, etc.

We evaluated the following 4 schedulers in our experiments: (a) ALG1-PA adopts the Pure ALOHA scheduler that comes with the native LoRaWAN. When a transmission period is reached, it randomly selects a channel for transmission. It provides a performance baseline for our experiments. (b)

ALG2-PARO denotes the Pure ALOHA with Random Offset scheduler. When a transmission period is reached, it can transmit after a random offset time using a random channel. (c) ALG3-CR is the scheduler described in *the strict scheduling model* (Section IV-B2), where the base station recommends channel for each end-device. When a transmission period is reached, it transmits immediately. (d) ALG4-COR is the scheduler described in *the loose scheduling model* (Section IV-B3), where the base station recommends channel and transmission offset for each end-device.

In addition, the maximum transmission offset is set to 2000ms for ALG2-PARO and ALG4-COR. End-devices transmit packets with frame size of 40 bytes.

B. Experiment Results

1) *PRR of SF7*: First, we performed a test to determine the impact on packet reception rate (PRR, the ratio of packets received by the base station to those sent by the corresponding end-device) with increasing number of end-devices in SF7.

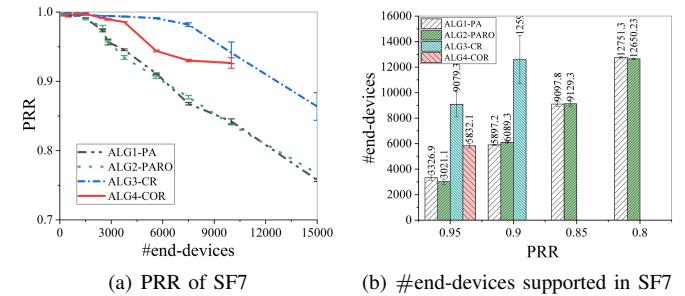


Fig. 7. The PRR of S-MAC in SF7 scenario

Fig. 7(a) depicts the result that in the case of 10,000 end-devices with SF7, S-MAC has an increase of 11.8% in PRR from 84.2% in ALG1-PA to 94.1% in ALG3-CR.

The results were then processed to determine the number of end-devices that could be supported under different PRRs ranging from 80%, 85%, 90% and 95%, as shown in Fig. 7(b). S-MAC can support up to 9,079 end-devices with PRR requirement of > 95%, increasing the number of end-devices by 1.73 \times compared with the native LoRaWAN.

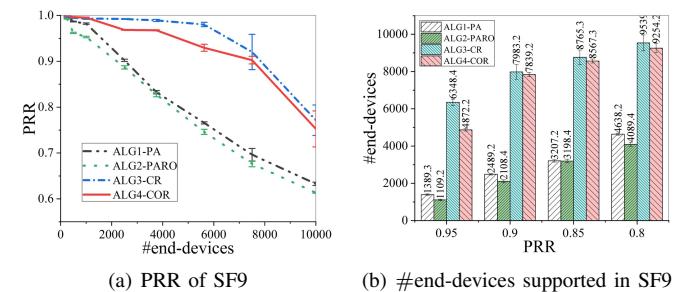


Fig. 8. The PRR of S-MAC in SF9 scenario

2) *PRR of SF9*: We then performed a test to determine the PRR of S-MAC when all end-devices used SF9 for transmission.

As can be seen from Fig. 8, in the case of 7,500 end-devices with SF9, PRR can be increased by up to 34.2% from 69.6% in ALG1-PA to 93.4% in ALG3-CR. S-MAC can support up to 6,348 end-devices with PRR requirement of $> 95\%$, increasing the number of end-devices by $3.57\times$.

3) *PRR of SF11*: We repeated the above experiment when all end-devices used SF11 for transmission.

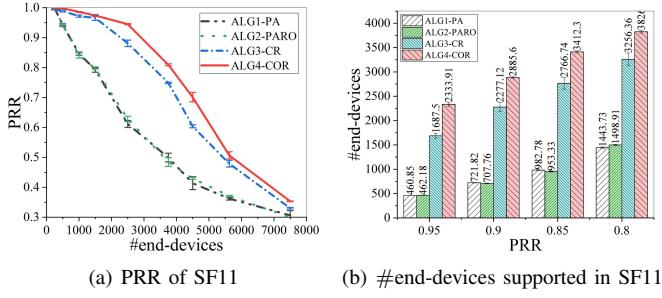


Fig. 9. The PRR of S-MAC in SF11 scenario

We can find that the PRR of S-MAC scheduler achieves a significant improvement. In the case of the same number of end-devices, PRR can be increased by up to 53.8% from 52.6% in ALG1-PA to 80.9% in ALG4-COR. It can support up to 2,334 end-devices with PRR requirement of $> 95\%$, increasing the number of end-devices by $4.06\times$.

Combining the above three experiments, it can be seen that:

With the increase of SF, the transmission time of packet will increase, and the probability of transmission collision will increase. When SF=7, the native LoRAWAN can also obtain a good PRR, and the advantage of S-MAC is not that obvious. While SF=11, the effectiveness of S-MAC is fully demonstrated as the probability of collision increases.

In addition, we can find that ALG4-COR is not always better than ALG3-CR. This is because, although ALG4-COR can make better use of each channel, its downlink efficiency is lower than that of ALG3-CR. In the case that the uplink transmission time is short, both ALG3-CR and ALG4-COR can allocate channels for transmission events well, but the downlink efficiency of ALG3-CR is better, which makes ALG3-CR more effective.

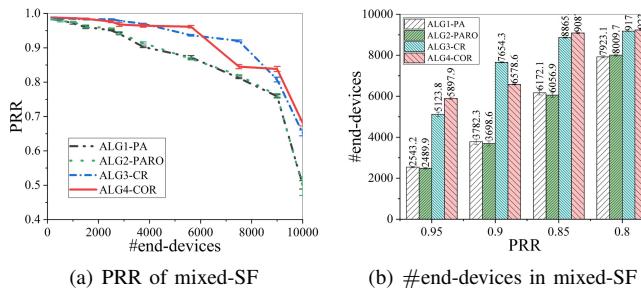


Fig. 10. The performance of S-MAC in mixed SF scenario

4) *PRR of mixed-SF*: We then performed a test to determine the performance of S-MAC when end-devices used different SFs. The setup for this experiment is as follows: End-devices

were deployed from near to far in Fig. 6(d), and they used the ADR mechanism to adjust their SF by their distance from the base station. According to statistics, the number of real end-devices using SF7 to SF12 for transmission was: 3, 8, 14, 14, 8 and 3, respectively.

Fig. 10(a) describes the impact on PRR with increasing number of end-devices in the above scenario. It can be found that PRR is increased by 13.3% from 81.2% in ALG1-PA to 92.0% in ALG3-CR, while increased by 35.6% from 51.4% in ALG1-PA to 70.2% in ALG4-COR. Fig. 10(b) illustrates the number of end-devices that can be supported under varying PRRs. The result shows that, in the case of PRR requirement of $> 95\%$, S-MAC can support up to 5897 end-devices in this scenario, increasing the number of end-devices by $1.32\times$ compared with the native LoRaWAN.

5) *Throughput of SF11*: We tested the impact on throughput with increasing number of end-devices in SF11 scenario in Fig. 11(a). The improvement of throughput under different PRRs is given in Fig. 11(b). When the number of end-device is 4,500, the network throughput reaches a peak of 1121.9 bps. The throughput has an increase of $4.01\times$ compared with the native LoRaWAN with PRR requirement of $> 95\%$.

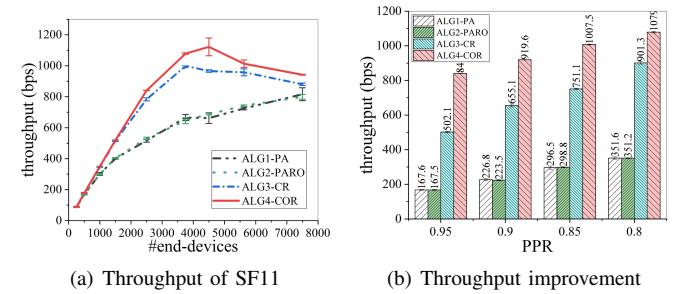


Fig. 11. The throughput of S-MAC in SF11 scenario

6) *Throughput of mixed-SF*: We then tested the throughput of S-MAC in mixed-SF scenario, which had the same deployment as Section VI-B4. Fig. 12 depicts the results of this experiment. When the number of end-device is 9,000, the network throughput reaches a peak of 2682.9 bps. The throughput is increased by $1.26\times$ compared with the native LoRaWAN with PRR requirement of $> 95\%$.

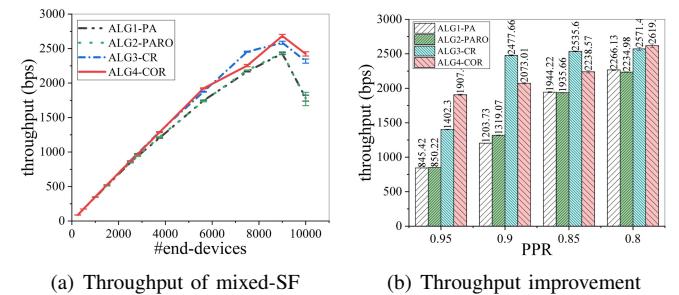


Fig. 12. The throughput of S-MAC in mixed-SF scenario

VII. RELATED WORK

Due to the huge transmission collisions caused by large-scale end-devices transmitting concurrently, the number of supported end-devices is very limited and cannot meet the requirements of LPWAN applications [3]–[6], [8], [33]–[35]. In order to solve this issue, existing literature can be summarized into the following two categories.

PHY-layer approaches. Choir [6] exploits the natural hardware offsets between low-power nodes to disentangle collisions from several LPWAN transmitters. FTrack [7] separates collided transmissions by considering both the time domain and the frequency domain features. In traditional wireless networks (such as WiFi, BLE and Zigbee), the work mainly involves parallel communication [36]–[40], and cross-technology interference in different scenarios [41], [42].

MAC-layer approaches. The main studies are as follows: (a) transmission scheduling strategies [9]–[16] are adopted in certain scenarios to reduce collisions, such as scheduling transmission powers, SFs, CFs or timeslots. (b) collision avoidance, detection or awareness mechanisms [17]–[20] are researched to reduce the probability of collision, such as Listen-Before-Talk, TDMA.

To sum up, the work in PHY-layer cannot work on commodity baseband chips, while the work in MAC-layer may introduce excessive communication overhead, end-devices cost, power consumption, or hardware complexity. On the other hand, the existing researches neither consider the traffic characteristics of LPWAN applications, nor combine its PHY-layer features, which will result in low scalability.

VIII. CONCLUSION

In summary, S-MAC is the first scheduler based on the periodic characteristics of LPWAN applications and transmission feature of LoRa PHY-layer to improve scalability in LPWAN. It is capable of adaptively perceiving clock drift of end-devices, adaptively identifying the join and exit of end-devices, and adaptively performing the scheduling strategy dynamically to reduce transmission collisions.

ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China 2017YFB1003000, NSF China under Grants No. 61632008, 61872079, 61702096, 61702097, 61902065, 61972085, 61906040, NSF of Jiangsu Province under grant BK20170689, BK20190345, BK20190335, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No.BM2003201, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No.93K-9, and supported by Collaborative Innovation Center of Novel Software Technology and Industrialization and Collaborative Innovation Center of Wireless Communications Technology. We sincerely thank Qianqian Lin for her help and support in making this work possible.

APPENDIX

Correctness and Optimality Proof Sketch of Theorem 1.

We assume that the events in E are sorted: $f_1 \leq f_2 \leq \dots \leq f_n$. Obviously, e_1 will be selected in some optimal solution. Ozws., we can replace the first event in any group with e_1 .

Suppose the optimal solution arranged events exactly as Theorem 1 does up to event e_i . Then, let's show that there is an optimal solution that is consistent with Theorem 1 up to event e_{i+1} . We distinguish three cases:

(1) $\forall g \in G, s_{i+1} \leq g$. Since the optimal solution contains and arranges event $e[1..i]$ exactly as Theorem 1 does, so both Theorem 1 and the optimal solution do not include e_{i+1} .

(2) $\exists g_a, g_b \in G, s_{i+1} \geq g_a, s_{i+1} \geq g_b$. We assume that the larger one is in group a (in group b is a symmetrical case), then Theorem 1 will select e_{i+1} . If not, we can replace next event in group a with e_{i+1} in the optimal solution. If the optimal solution arranges e_{i+1} in group b , we can swap those events starting from e_{i+1} between these two groups.

(3) $\exists g_a \in G, s_{i+1} \geq g_a. \forall g \in G - \{g_a\}, s_{i+1} \leq g$. e_{i+1} is arranged in group a . If the optimal solution doesn't select e_{i+1} , we can always replace the next event in group a with it.

Proof Sketch of Theorem 2. We polynomial reduce the set partition problem to the subset of the decision problem: can we select x ($x > 0$) events for K groups such that events in each group are collision-free before the deadline D . Given $V = \{v_1, v_2, \dots, v_n\}, \forall v_i \in Z^+$, the set partition problem is to decide if we can partition set V into K sets, A_1, A_2, \dots, A_K , such that $\sum_{v_i \in A_1} v_i = \sum_{v_j \in A_2} v_j = \dots = \sum_{v_k \in A_K} v_k$.

Our transformation is as follows: (a) compute $U = \sum_{i=1}^n v_i$. (b) for $v_i, 1 \leq i \leq n$, construct an event v_i such that $t_i = v_i$ and $p_i = 0$. (c) set $x = n, M = U/K$ and $D = U/K$.

Now, we prove that the n events can be scheduled into the K groups before the deadline D iff the set V can be partitioned.

(1) Suppose V can be partitioned into K sets, A_1, A_2, \dots, A_K , such that $\sum_{v_i \in A_1} v_i = \sum_{v_j \in A_2} v_j = \dots = \sum_{v_k \in A_K} v_k = U/K$. Then, we select those events to be scheduled in group 1 if their corresponding integers are in set A_1 : $G_1 = \{e_i \mid v_i \in A_1\}$, and select those events to be scheduled in group 2 if their corresponding integers are in set A_2 : $G_2 = \{e_i \mid v_i \in A_2\}$, and so on.

For $\forall x$, assume $G_x = \{v_1, v_2, \dots, v_m\}, 1 \leq m < n$, we have $\sum_{i=1}^m v_i = U/K$. We can set the transmission offset of the first event to 0, the offset of the second event to v_1 , etc., the offset of the m -th event is $\sum_{i=1}^m v_i - v_m \leq M$, and the finish time of this event is $\sum_{i=1}^m v_i \leq D$. Therefore, events in each group are collision-free before D .

(2) Suppose n events can be scheduled into the K groups. Let G_1 be the events scheduled in group 1 before D , G_2 be the events scheduled in group 2 before D , etc. Then, we partition the set V as follows: $A_1 = \{v_i \mid e_i \in G_1\}, A_2 = \{v_i \mid e_i \in G_2\}, \dots, A_K = \{v_i \mid e_i \in G_K\}$.

For $\forall x$, assume $G_x = \{e_1, e_2, \dots, e_m\}$, Then the finish time of the last event is $\sum_{i=1}^m t_i \leq D = U/K$. Because $\sum_{t_i \in G_1} t_i + \sum_{t_i \in G_2} t_i + \dots + \sum_{t_i \in G_K} t_i = U$, we must have $\sum_{t_i \in G_1} t_i = U/K, \sum_{t_i \in G_2} t_i = U/K, \dots, \sum_{t_i \in G_K} t_i = U/K$, so (A_1, A_2, \dots, A_K) is a partition of V .

REFERENCES

- [1] F. Sforza, "Communications system," Mar. 26 2013, uS Patent 8,406,275.
- [2] L. Alliance, "Lorawantm 1.0. 2 specification," *LoRa Alliance*, 2016.
- [3] 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.
- [4] 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.
- [5] D. Ismail, M. Rahman, and A. Saifullah, "Low-power wide-area networks: Opportunities, challenges, and directions," in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking (ICDCN)*. ACM, 2018, p. 8.
- [6] 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.
- [7] 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.
- [8] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] A. T. Hoang, Y.-C. Liang, and M. H. Islam, "Power control and channel allocation in cognitive radio networks with primary users' cooperation," *IEEE Transactions on Mobile Computing (TMC)*, vol. 9, no. 3, pp. 348–360, 2009.
- [15] L.-H. Yen, Y. W. Law, and M. Palaniswami, "Risk-aware distributed beacon scheduling for tree-based zigbee wireless networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 11, no. 4, pp. 692–703, 2011.
- [16] C. Cano, D. J. Leith, A. Garcia-Saavedra, and P. Serrano, "Fair coexistence of scheduled and random access wireless networks: Unlicensed lte/wifi," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 6, pp. 3267–3281, 2017.
- [17] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, and X. Vilajosana, "Lpdq: A self-scheduled tdma mac protocol for one-hop dynamic low-power wireless networks," *Pervasive and Mobile Computing*, vol. 20, pp. 84–99, 2015.
- [18] Y. Tang, Z. Wang, D. Makrakis, and H. T. Mouftah, "Interference aware adaptive clear channel assessment for improving zigbee packet transmission under wi-fi interference," in *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*. IEEE, 2013, pp. 336–343.
- [19] A. C.-C. Hsu, D. S. Wei, and C.-C. J. Kuo, "Coexistence wi-fi mac design for mitigating interference caused by collocated bluetooth," *IEEE Transactions on Computers (TC)*, vol. 64, no. 2, pp. 342–352, 2013.
- [20] L. Tytgat, O. Yaron, S. Pollin, I. Moerman, and P. Demeester, "Analysis and experimental verification of frequency-based interference avoidance mechanisms in ieee 802.15. 4," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 2, pp. 369–382, 2014.
- [21] "The things network," <https://www.thethingsnetwork.org/>, 2019, [Online; accessed 15-July-2019].
- [22] 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.
- [23] N. Blenn and F. Kuipers, "Lorawan in the wild: Measurements from the things network," *arXiv preprint arXiv:1706.03086*, 2017.
- [24] F. K. Andri Rahmadhani, "Understanding collisions in a lorawan," *SURF Wiki*, 2017.
- [25] "Semtech," <https://www.semtech.com/>, 2019, [Online; accessed 15-July-2019].
- [26] "Lorawan 1.1 regional parameters," <https://lora-alliance.org/resource-hub/lorawanr-regional-parameters-v11rb>, 2018, [Online; accessed 15-July-2019].
- [27] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A study of lora: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.
- [28] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.
- [29] P. Neumann, J. Montavont, and T. Noël, "Indoor deployment of low-power wide area networks (lpwan): A lorawan case study," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2016, pp. 1–8.
- [30] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in lorawan," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.
- [31] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *The Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, 2002, pp. 147–163.
- [32] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys)*. ACM, 2004, pp. 39–49.
- [33] K. Mikhaylov, J. Petaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the lora low power wide area network technology," in *European Wireless 2016; 22th European Wireless Conference*. VDE, 2016, pp. 1–6.
- [34] E. De Poorter, J. Hoebeke, M. Strobbe, I. Moerman, S. Latré, M. Weyn, B. Lannoo, and J. Famaey, "Sub-ghz lpwan network coexistence, management and virtualization: an overview and open research challenges," *Wireless Personal Communications*, vol. 95, no. 1, pp. 187–213, 2017.
- [35] 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.
- [36] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *ACM Special Interest Group on Data Communication (SIGCOMM)*, vol. 39, no. 4. ACM, 2009, pp. 159–170.
- [37] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking (MobiCom)*. ACM, 2014, pp. 617–628.
- [38] L. Kong and X. Liu, "mzig: Enabling multi-packet reception in zigbee," in *Proceedings of the 21st annual international conference on mobile computing and networking (MobiCom)*. ACM, 2015, pp. 552–565.
- [39] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu, "Snow: Sensor network over white spaces," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems (SenSys)*. ACM, 2016, pp. 272–285.
- [40] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra, "Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys)*. ACM, 2017, p. 9.
- [41] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the rf smog: making 802.11 n robust to cross-technology interference," in *ACM Special Interest Group on Data Communication (SIGCOMM)*, vol. 41, no. 4. ACM, 2011, pp. 170–181.
- [42] S. Yun and L. Qiu, "Supporting wifi and lte co-existence," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 810–818.