AdapLoRa: Resource Adaptation for Maximizing Network Lifetime in LoRa networks

Weifeng Gao†, Zhiwei Zhao†, Geyong Min‡
† School of Computer Science and Technology, University of Electronic Science and Technology of China, China
‡ Department of Computer Science, University of Exeter, Exeter, EX4 4QF, U.K.
weifeng@mobinets.org; zzw@uestc.edu.cn; G.Min@exeter.ac.uk

***Abstract*—LoRa has attracted much research attention due to its long communication range and low power consumption on end devices. In LoRa networks, the energy consumption on the end devices can be unfair, because some end devices have to use large spreading factors (leading to long transmission time) or large transmission power to reach a far-away gateway, and their energy consumption can be quite different. As a result, these end devices will run out of their batteries much faster, which may significantly reduce the network lifetime. The existing works have focused on the static resource allocation in LoRa networks to achieve energy fairness. However, due to the dynamic wireless environment, the static allocation can be inefficient in practice. In this paper, we develop AdapLoRa, a lifetime-aware dynamic network resource allocation system, to maximize the network lifetime of LoRa networks. AdapLoRa periodically adapts the resource allocation according to the link quality of end devices. A fine-grained network model is developed to capture the link quality variations and network interference. Finally, by considering the adaptation overhead (e.g., energy consumed by end devices to receive the configuration commands), we propose to gradually improve the network lifetime by periodically estimating network lifetime with different resource allocations. We implement AdapLoRa on a LoRa testbed, and the experimental results reveal that AdapLoRa improves the network lifetime by 23.7% compared with the state-of-the-art works.**

***Index Terms*—The Internet of Things, wireless networking, LoRa networks, energy fairness, resource allocation**

I. Introduction

The emerging LoRa technology has received much attention as Low-Power Wide-Area Networking (LPWAN) since its open standard allows us to build autonomous IoT networks without third-party infrastructure [1], [2].

Network lifetime is one of the most important design objects in low-power wireless networks , and is determined not only by the energy consumption on each end device, but also by the energy fairness among them. Existing works [3]–[7] mainly focused on reducing the energy consumption of end devices through more efficient and reliable transmissions. For example, unnecessary re-transmissions can be avoided in Charm [4] and Ftrack [7] by concurrently decoding multiple colliding LoRa signals. Unfortunately, the energy fairness problem of LoRa networks has not received sufficient attention. Basically, the network lifetime is defined as the time of the first end device running out of its battery.

978-1-7281-6992-7/20/$31.00 ©2020 IEEE

Energy unfairness in LoRa networks occurs mainly due to the following two reasons. First, some far-away end devices have to use large spreading factors (leading to low data rate) to reach a gateway. Compared to those end devices using high data rates, they spend longer time to send the same amount of data and consume more energy. As a result, they will drain their batteries earlier, shortening the network lifetime. Since the ratio between the highest and the lowest data rate can be as large as 22[1] [9], the problem of energy unfairness is severe in LoRa networks. Besides, spreading factor in LoRa performs multiplexing, i.e., two end devices can send packets simultaneously using different spreading factors even on the same frequency channel. Therefore, the selection of spreading factor results in different collision probability, and the retransmissions will take more energy from end devices, leading to short network lifetime.

Most of the existing works focus on the fairness of collision probability [10]–[12] (i.e., trying to balance the transmission success probability among the end devices), not the fairness of energy consumption. The energy unfairness problem in LoRa networks is lately explicitly studied in EF-LoRa [13]. EF-LoRa considers the impact of data rate and collisions to allocate the resources (i.e., spreading factor, channel and transmission power) in LoRa networks and improve energy fairness. However, EF-LoRa [13] statically allocates resources and only runs once when the network is first deployed. While in practice, LoRa wireless links are dynamic [14], i.e., the quality of a LoRa link fluctuates among time. It makes energy-fair resource allocation face the following challenges. First, static allocation cannot exploit the benefits of error correction scheme. LoRa uses Hamming code to achieve higher reliability, but its coding rate is limited and cannot adapt to the variation of link qulaity. Static allocation (e.g., [10], [13]) adopts a packet-level network model to reveal the relationship between resource allocation and network performance such as collision probability, which cannot explore the bit-level error correction. As a result, a fine-grained network model is required. Second, link dynamics requires end devices to change resource allocation adaptively (called ***resource adaptation*** in this paper), so gateways have to

[1]Since end devices sleep for most of the time, the energy consumption of sleeping can be larger than that of transmitting [8]. The energy consumption of the lowest data rate is about four times larger than that of the highest data rate, but it is large enough to cause energy unfairness.

frequently send resource adaptation decisions to end devices, inducing overhead on both latency and energy consumption (the impact of overhead will be described in Section III in detail). The additional downlink (i.e., transmissions from gateways to end devices) overhead makes the resource adaptation problem more complex than static resource allocation.

To tackle these challenges, we develop AdapLoRa, a dynamic network resource adaptation system for LoRa networks. We formulate the resource adaptation as an optimization problem to realize energy fairness and improve the network lifetime, considering the energy consumption overhead of resource adaptation. Keeping optimal resource allocation all the time is difficult, so AdapLoRa periodically estimates network lifetime at different resource allocations and decides whether to adapt a new resource allocation. However, the latency overhead of resource adaptation requires adaptations to finish as soon as possible, otherwise, adaptations may be out-of-date if it finishes after a long latency. AdapLoRa starts from adapting resources of end devices with the lowest resource adaptation latency, as long as an adaptation can improve network lifetime by more than a threshold, this adaptation will be performed in the network. Otherwise, AdapLoRa will adapt resources of end devices with the next lowest adaptation latency and will repeat the same procedure.

To get the improvement of resource adaptation, we have to compare network lifetime with different resource allocations. AdapLoRa first estimates network lifetime at next cycle that keeps the current resource allocation, using a linear regression method based on bit error rate measurement. This estimation is used as the baseline to evaluate other possible allocations. The network lifetime with a new resource allocation is estimated by periodically updating the network model.

AdapLoRa extends the link performance model presented in [15] to a fine-grained network model to relate resource allocation to network lifetime. Compared with [15], the proposed model takes into account the link interference in a network, and considers that packets are broadcast and can be received by multiple surrounding gateways. The model also analyzes the link performance with a powerful error correction scheme, RS code. By carefully adjusting RS coding rate, we can increase or reduce the redundancy in a packet to control the trade-off between error correction capability and energy consumption, so that the network lifetime can be improved adaptively.

We implemented AdapLoRa on a LoRa testbed of 4 gateways and 20 end devices on our campus. The gateways and end devices were based on LoRa chips. We evaluated the network lifetime on the testbed for AdapLoRa and EF-LoRa [13]. Our experimental results reveal that AdapLoRa can improve the network lifetime over EF-LoRa by 23.7%, respectively.

In summary, this paper makes the following contributions.

- We solve the complex resource adaptation problem by gradually improving network lifetime, considering adaptation overhead of both latency and energy consumption.
- We estimate network lifetime with different resource allocation with both measurement-based method and model-based method, and periodically update network model.
- A fine-grained network model is developed to estimate network lifetime. A powerful error correction scheme, i.e., RS code, is employed for higher reliability.

II. Related work

In this section, we study the existing works on energy fairness and resource allocation of LoRa networks and other networks, respectively.

Energy fairness in LoRa networks. EF-LoRa [13] proposes a network model that estimates the energy efficiency based on network layout. It allocates network resources to minimize the difference of the energy efficiency among end devices for fairness. However, this static algorithm only runs once before a network is deployed. EF-LoRa does not take dynamic wireless links into account. Differently, AdapLoRa adapts network resources with following novel designs. 1) AdapLoRa adopts a fine-grained network model that enables error correction to adapt to dynamic wireless links, while EF-LoRa is based on packet-level model and cannot consider the impact of error correction. 2) AdapLoRa considers the overhead of disseminating adaptation decision packets which can significantly affect the resource adaptation. 3) We build a LoRa network testbed on the campus and conduct a series of experiments to validate the performance of AdapLoRa, whereas EF-LoRa is only evaluated with simulations of NS-3.

Energy fairness in wireless sensor networks. Wireless sensor networks (WSNs) consist of energy-constrained sensors to monitor environment. Prior studies have tailored for prolonging the lifetime of WSNs. 1) Given that the communication range of sensors is very short, WSNs collect data in a multi-hop manner, and existing works mainly focus on energy-efficient routing or data rate control [16]–[23]. While in LoRa networks, end devices broadcast packets to gateways within a single hop, and we do not consider routing problem. 2) Since the network lifetime in WSNs is usually defined as the time that first node fails, researches also prolong the network lifetime by the max-min fairness optimization [24]–[26]. However, spreading factor shows properties of both data rate and multiplexing, making the energy fairness allocation more complex as we should jointly consider the impact of both of them (i.e., data rate determines the packet length thus energy consumption per transmission, and multiplexing impacts the transmission failure due to collisions, and these two properties show contradictory impact on the network lifetime). Besides, when adapting the resource allocation, the unique feature in LoRa networks is the overhead of the communication between end devices and gateways, which does not exist in traditional WSNs. The above reasons make it challenging to dynamically adapting the resources in LoRa networks.

Resource allocation in LoRa networks. In [10], [12], the authors realize the fairness of collision probability of end devices in a single gateway scenario. They do not consider the energy consumption gap of different spreading factors. LoRaWAN specification [9] provides Adaptive Data Rate to adjust spreading factor and transmission power, but it tries to use the smallest spreading factor and transmission power

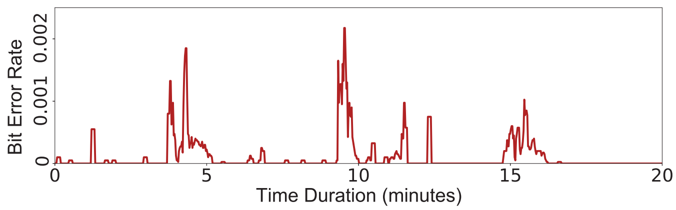

Fig. 1. LoRa link dynamics.

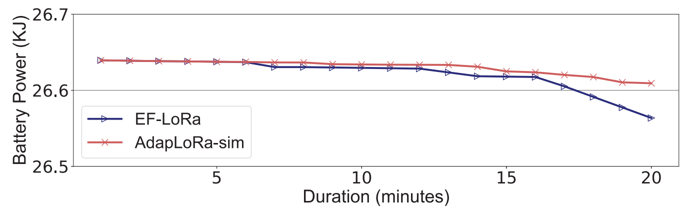

Fig. 2. Inefficiency of static allocation.

on individual end devices, which does not consider energy fairness for LoRa networks.

Resource allocation in cellular networks. Resource allocation problem in cellular networks such as spectrum assignment and power control has been widely studied [27], [28]. For example, techniques like partial frequency reuse (PFR) [29] or soft frequency reuse (SFR) [27] are used to mitigate inter-cell interference. The inter-cell interference can also be mitigated with efficient power control scheme [30]–[32]. Besides, to improve the throughput, data rate is adjusted by different modulation schemes and adapting to the dynamic wireless condition [33]–[36]. However, energy fairness problem in LoRa networks is different. 1) In LoRa networks, spreading factors not only perform orthogonality but also indicate different data rates. These two properties should be jointly considered in resource allocation. 2) Power control in cellular networks tries to reduce transmission power to limit communication range and reduce the interference. However, end devices in LoRa networks are not associated with a certain gateway, and packets can be received by all the surrounding gateways. if a LoRa end-device uses a small transmission power, although the interference can be mitigated, the number of gateways it can reach will also decrease, and reliability may be reduced.

III. Motivation

We conducted initial experiments on a LoRa network testbed deployed on our campus to show the dynamics of links in LoRa networks, and to demonstrate that a simple dynamic resource allocation can improve the performance over static resource allocation. We also analyzed the overhead to adapt network resource allocation in LoRa networks.

Dynamics of LoRa wireless links. We first measured link condition in a LoRa network, and used bit error rate (BER) to illustrate fine-grained link conditions.

In the experiment, an end device sent a packet to a LoRa gateway every 12 seconds. Basically, the size of a LoRa packet should be smaller than 256 bytes. In the initial experiments, we set the packet size to 100 bytes, including 7-byte header and 93-byte payload. Spreading factor 7 and transmission power 8 dBm were used. Figure 1 depicts BER of a link for a typical 20 minutes period. We observe that there are periods where BER shows dynamic behavior (i.e., fourth minutes, tenth minutes and fifteenth minutes in Figure 1). Although the dynamics of BER in Figure 1 is small (i.e., less than 0.002), it will lead to a high packet error rate for packet transmissions, with the length of a LoRa packet (e.g., several hundred bits). Therefore, LoRa link condition shows highly dynamic behavior and the energy consumption on the end devices changes dynamically, and the network lifetime also fluctuates. The experiments in [14] also show the dynamics of signal-noise-ratio (SNR) of LoRa links that verifies the above point. As a result, it is inefficient to use static resource allocation to always achieve long network lifetime. This dynamic can be dealt with by adaptive allocation of network resources so that the energy consumption on the end devices will not increase dramatically and impact the network lifetime.

Inefficiency of static resource allocation. In the initial experiment, we compared a simple dynamic resource allocation with the static resource allocation in [13] over a LoRa network with 2 gateways and 20 end devices. The static allocation is obtained according to the algorithm in [13], which does not change after we deploy the network. The simple dynamic allocation is implemented by modifying the existing adaptive data rate (ADR) in LoRa specification [9] (called AdapLoRa-sim in this paper). AdapLoRa-sim periodically adapts the resource allocation of end devices. It measures the average signal-noise-ratio (SNR) of 10 latest received packets on end devices and calculates the lifetime of the end devices based on the network model in [13]. At every cycle, AdapLoRa-sim determines whether to increase or decrease the SNR of end devices so that the lifetime of the end devices can be maximized (optimization problem as in [13]). The adjustment of SNR works based on ADR in LoRa specification.

According to [37], the fairness in a network can be achieved by optimize the performance of the worst node. In Figure 2, we use the remaining battery energy of the worst end device (i.e., minimum remaining energy) in a network for a 20 minutes duration to represent the energy fairness. Three spreading factors (8, 10, 12) and four frequency channels were used, and transmission power ranged from 2 dBm to 14 dBm. A fully charged battery on an end device had 26.64 kJ power. From Figure 2, we observe that during the first 20 minutes, the energy consumption of the static resource allocation is larger than that of the simple dynamic allocation for 76% thus implies significant potential improvement of network lifetime, which can be achieved by dynamic resource adaptation. However, dynamic resource adaptation in LoRa networks faces some challenges and we discuss them next.

Adaptation overhead. In LoRa networks, a server processes the data collected from end devices and makes decision periodically on whether to change the resource allocation. If the server decides to change the resource allocation at some moment, it needs to send resource adaptation commands to configure each involved end device. Thus, this resource adaptation will have overhead and will potentially affect the energy fairness and network lifetime.

Energy consumption. After sending an uplink packet to gateways, a LoRa end device opens two reception windows and waits for a packet downlinks from the gateways. Casals *et al.* [38] measured the energy consumption of different actions in LoRa end devices such as transmitting, sleeping and receiving. It is shown that receiving a packet in the first reception window can consume about 46% energy as transmitting a packet [38]. This is much larger than the energy consumption of sleeping for the duration of the window. Even worse, if no downlink packet is received during the first window, the end device has to wait for reception in the second reception window, consuming 74.7% energy as for transmitting a packet. Therefore, the reception of adaptation decision packets is likely to reduce the lifetime of these end devices, potentially affecting the network lifetime. Besides, some end devices do not need to receive adaptation decision packets and spend battery power, which will cause energy unfairness and affect network lifetime. As a result, a dynamic resource adaptation should be designed carefully so that overhead does not outweigh the improvement of network lifetime.

Transmission latency. LoRa networks usually consist of a large number of end devices, and the resource adaptation could finish after a long latency because the adaptation decision packets are sent to end devices in a unicast manner. For example, sending a 14-byte downlink packet with spreading factor 12 takes 991.8 ms, if 1000 end devices are required to change their resource allocations, it would take 1000×991.8ms=991.8s to complete the decision packets dissemination. In practice, the time is longer, because a gateway has to wait for an uplink packet before it can send packets to an end device. With such a considerable latency in disseminating adaptation decisions, realizing energy fairness in dynamic LoRa networks becomes challenging, because adaptation decision packets may arrive at end devices after a long latency when the link condition has already changed.

IV. Design of AdapLoRa

In this section, we present the design of AdapLoRa in detail. The objective of AdapLoRa is to optimize LoRa network lifetime with dynamic resource adaptation. We first formulate the problem of realizing the maximum network lifetime in Section IV-A. Energy consumption overhead of end devices receiving adaptation decision packets is considered in the formulation. To adapt to LoRa links dynamics with error correction, we propose a network model in Section IV-B to formulate network lifetime with resource allocation. However, this formulation can only get optimal network lifetime at a certain time, while achieving optimal network lifetime all the time requires resource adaptation whenever link condition changes. This makes the problem very difficult, and AdapLoRa overcomes this challenge by estimating network lifetime periodically and performing resource adaptation if the new resource allocation results in enough network lifetime improvement. The estimation of network lifetime for different resource allocations is described in Section IV-C. The notations used in this paper are summarized in Table I.

TABLE I
Notations used in this paper

Symbols	Notations
s_i	Spreading factor of end device i
p_i	Transmission power of end device i
f_i	Frequency channel of end device i
c_i	Coding rate of end device i
$\mathcal{N}$	Set of all the end devices
L	Network lifetime
r_i	Packet error rate of end device i
q	Symbol error rate
E_i	Remaining battery energy of end device i
E_t^i	Energy consumption for transmitting
E_s^i	Energy consumption for sleeping
E_r^i	Energy consumption for receiving a downlink
d_i	Distance between end device i to the gateway
β	Pass loss exponent
l_s^i	Number of symbols in the packet
T_i	Transmission time of end device i
T_g	Data generation interval of end device i

A. Problem formulation

In LoRa networks, the end devices broadcast packets to surrounding gateways, and the gateway relay the packets to a central server for processing. If the server decides to send packets to an end device (e.g., replying to an end device), it chooses one gateway with the highest received signal power to transmit the packets. We consider the scenario that all the end devices send a packet every T_g time, guaranteeing the duty cycle constraint in LoRaWANs.

We mainly consider four configurable LoRa network resources which can affect the performance of LoRa transmissions such as energy consumption and reliability.

1) Spreading factor, which impacts communication range and data rate.
2) Transmission power, which impacts energy consumption and communication range. It ranges from 2 dBm to 14dBm in steps of 2 dBm.
3) Frequency channel, which multiplexes the transmissions to reduce interference. LoRa networks operate in the 470 MHz frequency band in China.
4) Coding rate, which represents the error correction capability. LoRa adopts Hamming Code as Forward Error Correction (FEC), however, it can only correct at most one single error bit, inducing unnecessary transmission overhead and delays, and reducing the network lifetime [39]. AdapLoRa tries to mitigate this inefficiency by using RS code for error correction. It has been claimed that RS code is a powerful error correction scheme and can be implemented in LoRaWANs [40].

By choosing these parameters for end devices, different transmission reliability and energy consumption can be realized to achieve energy fairness and improve network lifetime.

We first define the resources in LoRa networks as: spreading factor s_i, transmission power p_i, frequency channel f_i and coding rate c_i. The problem of seeking the longest network lifetime can be formulated as an optimization problem in Eq. (1). In practice, this problem has to be solved every time the network experiences changes in the link dynamics.

$$\begin{aligned} &\max\ L(s,p,f,c), \\ s.t.\ &\forall i \in \mathcal{N},\ s_i \in [7,12] \\ &\forall i \in \mathcal{N},\ p_i \in [2,14] \\ &\forall i \in \mathcal{N},\ f_i \in [1,8] \\ &\forall i \in \mathcal{N},\ c_i \in \mathcal{C} \end{aligned} \tag{1}$$

where $L(s,p,f,c)$ denotes network lifetime that is determined by the resource allocation on each end device, and $\mathcal{N}$ is the set of the end devices. AdapLoRa uses RS code for error correction, and we use c_i to denote RS coding rate which is defined by $\mathcal{C} = \{5/6, 4/5, 3/4\}$ in this paper, where x/y means x data bits and $(y-x)$ redundant bits. Since the bottleneck of the network lifetime is those end devices with short device lifetime, we consider the network lifetime as the duration of the first end device running out of its battery:

$$L(s,p,f,c) = \mathrm{min L_i(s_i, p_i, f_i, c_i)}, \tag{2}$$

where L_i is the lifetime of end device i. Eq. (1) and Eq. (2) maximize the network lifetime by realizing the energy fairness in a LoRa network, which verifies the need of energy fairness in Section I. L_i is dependent on its energy consumption and residual battery energy E_i as follows:

$$L_i = \frac{E_i}{(\frac{E_t^i}{1-r_i} + E_s^i)} \cdot T_g, \tag{3}$$

where E_t and E_s denote the energy consumption for transmitting a packet and sleeping, respectively, r_i is the packet error rate (PER) of end device i, and T_g represents the data generation interval. The end device energy in a duty cycle is consumed by sleep state that is dependent on sleep duration and active state that is related to transmission power and time-on-air. According to [8], the relationship between energy consumption and transmission power (spreading factor) is computed by regression, based on experimental measurements.

Eq. (3) calculates the end device lifetime by considering how many transmissions the device battery can support. Since transmissions may fail due to the interference and noise, the energy consumption for transmitting one packet may involve several retransmissions. AdapLoRa uses PER r_i to measure the reliability and the number of transmissions.

We define the set of end devices that requires changing their resource allocations as $U \subseteq \mathcal{N}$, an resource adaptation decision is represented as a set $V = \{(i, s_i, p_i, f_i, c_i)\}$, where $i \in U$ and (s_i, p_i, f_i, c_i) is the new resource allocation of end device i. An end device i will have a new r_i with a different resource allocation, and the energy consumption on this end device will increase due to the reception of the adaptation decision packet. Both r_i and energy consumption affects network lifetime, so the lifetime of an end device L_i after adaptation changes to:

$$\hat{L}_i = \begin{cases} L_i, & \text{if } i \notin U \\ \frac{E_i}{(\frac{E_t^i}{1-r_i} + E_r^i + E_s^i)/T_g}, & \text{if } i \in U \end{cases} \tag{4}$$

where E_r^i denotes the energy consumption for receiving a packet. A packet reception will reduce sleeping time at an end device, so the energy consumption of sleeping E_s^i should be reduced in Eq. (4). However, this reduction is extremely small (about 0.2%) [8], and is ignored in Eq. (4).

B. Network model

To solve Problem (1), estimation of r_i in Eq. (3) is required. First, we need to find out the relationship between r_i and resource allocation (e.g., redundant bits for error correction) in a dynamic wireless environment. For example, if a link is experiencing high packet error rate, we can use more redundancy (i.e., more powerful error correction) to reduce packet error rate. Otherwise, we can reduce redundancy to save energy. Static resource allocation uses a packet-level network model to map resource allocation to energy efficiency (i.e., the number of delivered bits per energy consumption). However, the packet-level model cannot capture the property of error correction scheme in LoRa, because it requires bit-level performance analysis. As a result, a finer-grained estimation is required to model the error correction and capture the dynamic link quality in LoRa networks.

1) Symbol-level network model: The information bits in LoRa are carried in a chirp (symbol), and a packet consists of multiple symbols, so we estimate the link condition with symbol error rate (SER). SER is the probability that a symbol m from end device i is not correctly decoded [15]. However, the existing model does not consider the interference induced by the signals from other end devices. Interference comes from the other end devices that use the same channel and spreading factor as the target end device. It is formulated as follows,

$$\begin{aligned} I &= \sum_{\substack{j \in \mathcal{N}, \\ s_i, f_i, o_i}} p_j \cdot g_j \cdot a(d_j) & (A1) \\ &= \sum_{\substack{j \in \mathcal{N}, \\ s_i, f_i}} p_j \cdot g_j \cdot a(d_j) \cdot h_j, & (A2) \end{aligned} \tag{5}$$

where s_i and f_i mean the spreading factor and frequency channel of end device i, o_i indicates that the transmission between end device i and j overlaps, g_j is Rayleigh fading channel and can be modeled as a complex Gaussian random variable, $a(d_j)$ denotes path loss attenuation that follows from the Friis transmission equation and can be defined as:

$$a(d_j) = (\frac{v}{4\pi r d_j})^{\beta}, \tag{6}$$

where v is the velocity of electromagnetic wave, r is the carrier frequency and β is the path loss exponent.

The first equation $(A1)$ shows the cumulative interference from other end devices. They use the same spreading factor and channel as end device i, and their transmissions overlap with end device i. The second equation $(A2)$ simplifies Eq. $(A1)$ by considering the randomness of LoRaWANs MAC protocol (unslotted Aloha). It extracts o_i in $(A1)$ and replaces it with a probability Eq. (7):

$$\begin{aligned} h_j &= P\{SO, PO\} \\ &= P\{SO|PO\} \cdot P\{PO\} \\ &= \frac{t_i + t_j^s}{t_i + t_j} \cdot \frac{t_i + t_j}{T_g} \\ &= \frac{t_i + t_j^s}{T_g}, \end{aligned} \tag{7}$$

where SO and PO denote the events that there are symbol overlapping and packet overlapping between end devices i and j, t_i and t_j are transmission time of end devices i and j, t_j^s represents the period of a symbol on end device j, and T_g denotes the data generation interval. The calculation of transmission time t_i is mainly related to spreading factor, and is given in LoRa specifications [41].

With I_n, the cumulative interference of symbol n, SER q can be written as:

$$q = P\{\max_{n, n\neq m}(|N_n| + |I_n|) > |E_m| + |N_m|\}, \tag{8}$$

where N_n and N_m denote the noise envelopes, and E_m is the energy of symbol m. Symbol errors occur when the symbol is corrupted by noise and interference. Given $|N_m|$ as a Rayleigh distributed random variable [15], SER is written as:

$$\begin{aligned} q = P\{\max_{n,n\neq m} & \left(\left(1 + \sum_{\substack{j\in\mathcal{N}, \\ s_i, f_i}} p_j \cdot a(d_j) \cdot h_j\right) \cdot \phi_n\right) \\ & > \left(p_i \cdot a(d_i)/l_s^i + 1\right) \cdot \psi_m\}, \end{aligned} \tag{9}$$

where l_s^i is the number of symbols in a packet on end device i, and ϕ_n and ψ_m are both Rayleigh random variables following Gaussian distribution. If we let $\hat{\phi} = \max_{n,n\neq m}(\phi_n)$, the SER can be approximated by Eq. (10) [15].

$$\begin{aligned} q &\approx Q\left(\frac{-\mathbf{C}_i \mu_{\hat{\phi}}}{\sqrt{(\mathbf{C}_i \sigma_{\hat{\phi}})^2 + \sigma^2}}\right) \\ &\approx Q\left(\frac{-\mathbf{C}_i\left((H_M)^2 - \frac{\pi^2}{12}\right)^{\frac{1}{4}}}{\mathbf{C}_i^2\left(\sqrt{H_M} - \left((H_M)^2 - \frac{\pi^2}{12}\right)^{\frac{1}{4}}\right) + 0.5}\right), \end{aligned} \tag{10}$$

where $\mathbf{C}_i = (1 + \sum_{\substack{j\in\mathcal{N}, \\ s_j, c_j}} p_j \cdot a(d_j) \cdot h_j)/(p_i \cdot a(d_i)/l_s + 1)$, and $H_M = \sum_{i=1}^{M} \frac{1}{i}$ denotes the M^{th} harmonic number with $M = 2^{s_i} - 1$. $Q(w) = \frac{1}{\sqrt{2\pi}} \cdot \int_w^\infty exp(-\frac{u^2}{2})\, du$ is the Q-function, $\mu_{\hat{\phi}}$ and $(\sigma_{\hat{\phi}})^2$ are the mean and variance of $\hat{\phi}$, and σ^2 is the variance of the Rayleigh distributed random variable. Now we can get SER related to spreading factor s_i, transmission power p_i and frequency channel f_i.

2) Enhanced error correction scheme: Due to the inefficiency of Hamming code that LoRa specification uses, AdapLoRa improves the transmission reliability with RS code. Since Hamming code is integrated in the hardware, we implement RS code on LoRa packets which already have Hamming code implemented. we consider its impact on the network model. With RS (u,w) (i.e., coding rate is $\frac{w}{u}$), where u and w denote the number of the total symbols and information symbols, respectively, and there would be $u - w$ redundant symbols. A packet can be correctly decoded if the number of error symbols in this packet is less than $\frac{u-w}{2}$. In this case, we can directly calculate the packet error rate r based on the above symbol error rate q:

$$\dot{r} = 1 - \sum_{j=0}^{(u-w)/2} \binom{u}{\frac{u-w}{2}} q^j (1-q)^{u-j}, \tag{11}$$

where we have RS coding rate $c_i = \frac{w}{u}$ in Eq. (11). With a larger c_i, more redundant data will be added in a packet, so the transmission delay and energy consumption of a single transmission will increase. On the other hand, RS code can handle more errors, so the PER can be reduced, leading to less energy consumption for retransmissions. AdapLoRa properly allocates c_i to end devices to consider this trade-off and achieve long network lifetime.

3) Packet reception by multiple gateways: With the above PER, we can further extend the model to multiple-gateway scenarios which is more practical. Specifically, with D gateways deployed, the final PER can be reduced as a packet is possibly received by all the gateways [4]. As a result, PER with D gateways can be expressed as follows:

$$\hat{r} = 1 - \prod_{d=1}^{D} \dot{r}_d \tag{12}$$

With the above model, we can estimate network lifetime based on Eq. (3).

4) Model calibration: The above network model can be inaccurate in practice due to the changed wireless environment, and the adaptation decisions may be inefficient. Therefore, the network model should also be adaptive.

To estimate network lifetime of resource adaptation accurately, we calibrate the network model accordingly. An important parameter that controls model accuracy is the path loss exponent β in the received signal power model, $rp_i = p_i \cdot g_i \cdot a(d_i)$. With a large β, the received signal would be weak, and vice versa. In the static resource allocation, β is empirically set to 2.7 in open space and 4 when there are buildings or trees attenuating signals [42].

However, this coarse-grained β cannot reflect the real network environment, so we adjust the path loss exponent β to calibrate the network model and represent LoRa links more accurately. When receiving a packet, the gateways can read its received signal power (RSSI). By considering the theoretical received signal power from the model, we can transform the equation to: $g_i = \frac{rp_i}{p_i \cdot a(d_i)}$. Following the fact that $g_i \sim exp(1)$, the probability P_i that the received signal power equals to rp_i

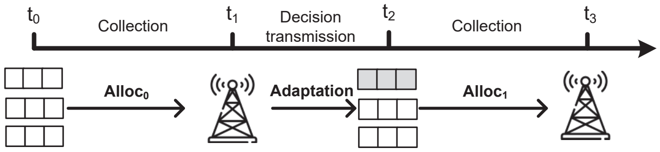

Fig. 3. Illustrating example of AdapLoRa.

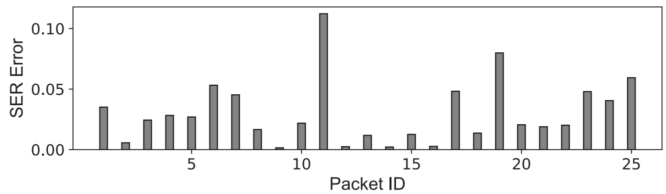

Fig. 4. Accuracy of model-based SER estimation.

can be estimated based on the probability density function of g_i. If P_i is larger than 50%, it means the received RSSI follows the theoretical model [43]: β for this link is acceptable and will not be changed; otherwise, we will adjust the value of β to guarantee the P_i is large enough.

Accuracy of network modeling. The network model is critical to estimate network lifetime. We conducted experiments to evaluate the accuracy of this model-based SER estimation. Figure 4 depicts the error of estimated SER, involving 20 packets with different resource allocation. We can see that more than half of the SER error is below 0.03. This is much larger than the BER error in Figure 5. The reason is that a symbol includes multiple bits, and as long as one bit error happens inside a symbol, this symbol is considered incorrect.

C. AdapLoRa resource adaptation

In the optimization problem in Eq. (1), we have to consider the adaptation of different number of end devices. For instance, if we change the allocation of two out of n end devices, there would be C_n^2 possible choices ($C_n^2 = \frac{n(n-1)}{2}$). Even worse, for each choice, the problem of finding the best allocation (which can be reduced to max-min fairness problem) is *NP-hard* [13]. So solving optimization problem in Eq. (1) is nearly impossible for large scale networks. Besides, to keep optimal network lifetime all the time with resource adaptation, we need the complete information on the condition of every link at every time. However, it is impossible to get this information because it would require frequent message exchanges, resulting in unacceptable overhead. To solve the above problem, AdapLoRa performs resource adaptations periodically to improve network lifetime with an acceptable overhead.

1) resource adaptation procedure: A typical adaptation cycle is shown in Figure 3. An adaptation cycle consists of two phase: in the first phase, gateways collect packets from all the end devices from t_0 to t_1. In the second phase, adaptation decisions are sent to end devices from t_1 to t_2; decisions are sent to end devices sequentially, so if a large of number of end devices receive decisions, the latency of this phase can be very long.

In the information collection phase (i.e., t_0 to t_1), end devices send packets to the server, carrying information such as the resource allocation $Alloc_0$ and bit error rate. From the collected information, the server estimates the network lifetime , say L_1, for the case that the resource allocation is not changed. This network lifetime L_1 will be used as the baseline to evaluate other possible resource allocations.

Now AdapLoRa estimates network lifetime for different resource allocations for comparison. Instead of seeking a resource adaptation with maximum network lifetime, which is likely to induce high latency for decision packets dissemination (i.e., t_1 to t_2), AdapLoRa improves network lifetime by changing the resource allocation with short dissemination latency which is enough to improve the network lifetime by a threshold τ. AdapLoRa first sort the end devices according to their decision packet dissemination latency, which is dependent on their spreading factors. Starting from an end device with the shortest decision packet dissemination latency, AdapLoRa estimates network lifetime L_1^i for different resource allocations $Alloc_i$ on this end device. If an L_1^i is larger than L_1 by a threshold τ, in the decision dissemination phase in Figure 3, the server will send a decision packet to this end device to change its resource allocation. Otherwise, AdapLoRa picks a subset of end devices with the next lowest dissemination latency and checks if a new resource allocation can improve network lifetime by at least τ. If not, AdapLoRa repeats the above procedure with another subset of end devices with the next lowest decision dissemination latency.

Since large scale networks involve a large number of possible subsets (2^n subsets with n end devices), in the worst case, we have to search for all these subsets. To reduce the searching space, AdapLoRa limits the maximum size of a subset based on the computation capability of the server (in the experiment, it is set to ten because we use a laptop as the server to run the algorithm). The above procedure is concluded in Algorithm 1.

Algorithm 1: AdapLoRa resource adaptation

$Initial()$; #Initial allocation
while $Receive()$ **do**
    $T_{old} = measure(BER)$; #Measuring network lifetime
    $\mathcal{S} = \mathcal{P}(\mathcal{N}))$; #All the subsets of end devices
    $Sort(\mathcal{S})$; #Sorting subsets by downlink latency
    $L_0 = Cal()$; #Lifetime without adaptation
    for $n \in [1, |\mathcal{S}|]$ **do**
        for $\mathcal{A} \in \mathcal{S}$ and $|\mathcal{A}| \leq 5$ **do**
            $(\mathcal{V}, L) = maxL(\mathcal{A})$;#Adaptation and lifetime
            if $L - L_0 \geq \tau$ **then**
                $Output(\mathcal{V})$;
                Break;
Function $maxL(\mathcal{A})\{$
    $L - L_0 = 99$; #Initial network lifetime
    while $L - L_0 > 0.05$ **do**
        for $i \in \mathcal{A}$ **do**
            for $s_i, p_i, f_i, c_i \in$ (**S,T,F,C**) **do**
                $L = max(L_i)$;# Allocation with max lifetime
                $Update(V(s_i, p_i, f_i, c_i))$;
                $L_0 = L$;
    Return (V, L) $\}$

The improvement of network lifetime is obtained by com-

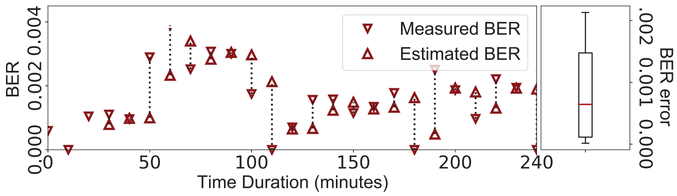

Fig. 5. Accuracy of BER estimation.

paring the network lifetime estimation that: 1) keeps the same resource allocation as last cycle (i.e., **without** resource adaptation) and 2) uses different resource allocations (i.e., **with** resource adaptation). The improvement is the difference between these two estimations. We describe the network lifetime estimation in Section IV-C2. The value of τ will be described in Section IV-C3.

2) Network lifetime without resource adaptation: Different from packet-level measurement, AdapLoRa does not wait for many received packets to estimate r_i. Instead, it is based on the bit error rate measurement from a few packets (the number of packets can be adjusted according to packet length), and even those collided packets which do not pass the CRC check are also considered to make the estimation more accurate.

In every adaptation cycle, network lifetime is updated based on the BER. If the current resource allocation does not change, the BER at next adaptation cycle is estimated based on the linear regression. BERs of multiple packets are utilized with different weights. Generally, a more recent packet should have a larger weight because it can reflect a more latest channel condition. The estimated BER can be written as:

$$\tilde{b} = \sum_{j=1}^{J} \alpha_j \cdot b_{-j}, \ where \ \sum_{j=1}^{J} \alpha_j = 1, \tag{13}$$

where b_{-j} denotes BER of the j^{th} latest received packet, α_j denotes weighting factors. In this paper, we use the last three packets to estimate BER without adaptation ($J = 3$), and their weights α_1 to α_3 are 70%, 20% and 10%, respectively.

With the estimated BER, PER can be calculated as follows and we can get the network lifetime based on Eq. (3).

$$r = 1 - (1 - \tilde{b})^N, \tag{14}$$

where N is the length of a packet in bits.

Accuracy of BER estimation. We conducted experiments to illustrate the accuracy of BER estimation. A LoRa end device sends a 100-byte packet to a gateway every 10 minutes using spreading factor 10 for four hours, and we compare the estimated BER with measured BER in Figure 5. Since estimation requires at least three previous packets, the first three packets do not show estimated BER. To make the estimation accuracy clear, the right figure shows the distribution of error of the estimated BER. It can be observed that more than 50% of the errors lie within 0.001, showing the accuracy of BER estimation for calculating network lifetime. Besides, we have carefully set the location of the end device and the gateway to make BERs a little high. Otherwise, the estimation accuracy cannot be clearly shown if most BERs are close to zero.

3) Network lifetime with resource adaptation: Intuitively, the network lifetime with different resource allocations can be obtained with the proposed symbol-level network model and the model calibration in Section IV-B.

Since the estimation of network lifetime with and without resource adaptation may be both inaccurate, an resource adaptation is possible to result in a shorter network lifetime if the threshold is too small. To avoid performance degradation, the network lifetime improvement threshold τ is empirically set based on the estimation inaccuracy. In this paper, τ is set to a proportion of the measured network lifetime without adaptation, so that the network lifetime improvement can adapt to the different network lifetime. In the evaluation, we conducted experiments on the selection of τ of 5%, 10% and 20%, and found that the threshold 10% shows the best network performance regarding to the network lifetime among them.

V. Evaluation

In this section, we validate the performance of AdapLoRa on a LoRa network testbed deployed on our campus, and compare the performance with the state-of-the-art work, EF-LoRa [13].

A. Experimental setting

Testbed setup. We deployed a LoRa testbed including 4 gateways and 20 end devices that covers our campus with around 1.5 square kilometers. Four gateways are deployed on the roof of different buildings, and we change the number of gateways by turning different gateways off. The end devices are deployed statically and have no mobility. We set up LoRa end devices by combining Arduino Uno boards with RFM95W modules (LoRa signal transceivers) operating at 470MHz frequency band. RAK2245 Pi HAT Board (equipped with SX1301) is used for LoRa gateways. Since LoRa end devices communicate with gateways within one hop, the packets from end devices can be received by all the surrounding gateways, so we do not need to design the network topology and determine the pairing of the end devices and gateways.

Since LoRa networks can involve hundreds of end devices with six available spreading factors and eight available channels, with a limited number of end devices (i.e, 20), we deploy our LoRa network and reduce the number of used channels and spreading factors to emulate a larger network with similar interference environment. Specifically, we use four channels and three spreading factors (i.e., 8, 10, 12) with 10% maximum duty cycle to emulate a network with around $20 \times 2 \times 2 \times 10 = 800$ end devices, using eight channels, six spreading factors and 1% duty cycle.

With the maximum duty cycle constraint, we let the end devices send a packet every 12 seconds (the time-on-air of a typical LoRa packet is less than one second) with randomly scattered starting time. Since the end devices may have different tasks of collecting sensing data from the environment and generate different amount of data, we make the end devices send packets with different payload size. Specifically, the payload size is randomly set from 60 bytes

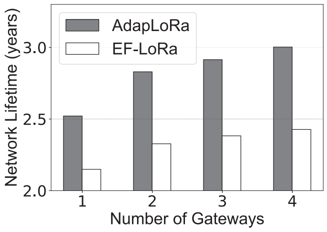

Fig. 6. Network lifetime.

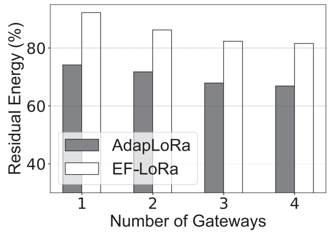

Fig. 7. Residual network energy.

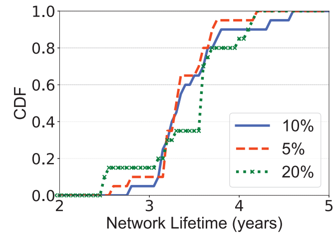

Fig. 8. Lifetime threshold τ.

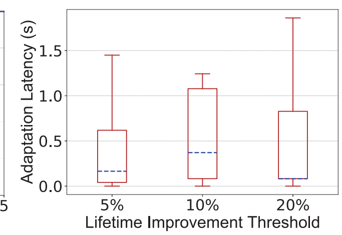

Fig. 9. Adaptation latency.

to 200 bytes, following the constraint of LoRa specification [9], and this size does not change during the experiments. Bandwidth and hamming coding rate are identical and set to 125kHz and 4/5. Transmission power ranges from 2 dBm to 14 dBm following the LoRaWANs specification [9]. Packets sent from end devices can be received by multiple gateways, and gateways will relay them to the server. When the server decides to adapt the resource allocation of the network, it sends adaptation decision packets to the corresponding end devices. The server will choose the gateway with the highest received RSSI to send the adaptation decision packet to an end device.

We implement RS code on LoRa end devices and gateways. Specifically, the encoder on the end devices is designed such that original packets are encoded based on RS coding rate. The encoded packets are passed to the InAir9B transceiver module for transmission. Similarly, a RS decoder on gateways will recover the received packets. In the experiments, we use three available RS coding rates: 5/6, 4/5 and 3/4. The testbed results will be attached for public access on: https://github.com/mobinets/AdapLoRa.

Metrics. We evaluated the performance using the following metrics: 1) Network lifetime: it is defined as the time that 10% end devices in a network have run out of their batteries. 2) Residual network energy: it is the average percentage of remaining battery power on end devices when a network ends, and it can reflect energy fairness of a network.

Benchmarks. We compared AdapLoRa with EF-LoRa [13] and implement them, the static resource allocation for energy fairness in LoRa networks. EF-LoRa allocates the resources by solving a max-min optimization problem with a greedy algorithm. The experiments with different network deployment are repeated for five times, and the average of the performance metrics such as network lifetime is used.

B. Network performance

Network lifetime. We first compared network lifetime of AdapLoRa with EF-LoRa on the testbed. AdapLoRa collected packets and made adaptation decisions every minute, while EF-LoRa employed a static resource allocation all the time. We changed the number of gateways from one to four, and networks with both methods ran for 20 minutes. We estimated their network lifetime by assuming that energy consumption drained on an end device repeats as this 20-minute period [8].

Experimental results in Figure 6 show that network lifetime can be improved with more gateways for both AdapLoRa and EF-LoRa, due to the reduced packet error rate. Besides, AdapLoRa performs better than EF-LoRa for every deployment, and the network lifetime difference between AdapLoRa and EF-LoRa increases with more gateways deployed (e.g., AdapLoRa outperforms EF-LoRa by about 17.4% with one gateway and this benefit rises to 23.7% with four gateways). This is because with more gateways, AdapLoRa can adjust the PER of the end devices adaptively, so there is more space for AdapLoRa to improve network lifetime, while EF-LoRa can only rely on the PER improvement by more gateways.

Residual network energy. Higher residual network energy means that when the first end device in the network dies, the other end devices still have much battery power and can work for a long time, indicating less energy fairness. We evaluated energy fairness through residual network energy, presented in Figure 7. Experimental results show that AdapLoRa has higher energy fairness than EF-LoRa, and this is because AdapLoRa periodically changes the allocation to improve the energy fairness thus network lifetime. Besides, by deploying more gateways, residual network energy of both methods is improved, thanks to the improved PER.

Network lifetime improvement threshold. The threshold of network lifetime improvement τ affects the frequency of resource adaptation. Compared with a small threshold, resource adaptation with a larger threshold will happen less frequently, because it is harder to find an adaptation to achieve large improvement in an adaptation cycle. Figure 8 illustrates network lifetime with different thresholds. Network lifetime can be calculated according to current battery power and PER. AdapLoRa performs better with larger thresholds (i.e., 20%) when the network lifetime is long (i.e., 3.6 years), but it performs worse when network lifetime is less than three years. The reason is that sometimes AdapLoRa with a large threshold cannot achieve a large improvement thus adaptation is not performed, and network lifetime is not improved.

Resource adaptation latency. The improvement threshold τ also has a significant impact on the latency of adaptation decision packets dissemination. This latency is expected to be short so that the estimated network lifetime can be accurate enough to make adaptation decisions. Figure 9 depicts the downlink latency to disseminate decision packets with different thresholds. Decision packets only carry adaptation results, so a decision packet is short as 14 bytes. Figure 9 shows that the dissemination latency is less than two seconds, because AdapLoRa tries to use the shortest latency to finish the decision packets dissemination, as described in Section IV-C1.

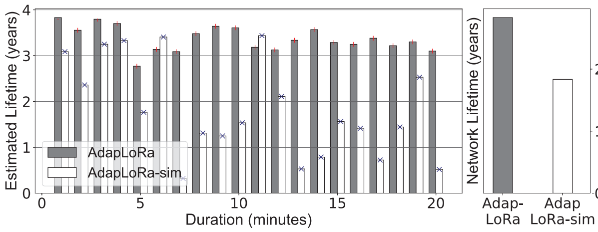

Fig. 10. Energy fairness ADR.

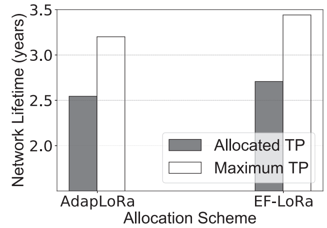

Fig. 11. Transmission Power.

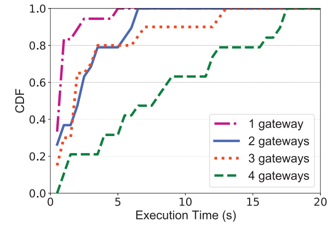

Fig. 12. Algorithm execution time.

C. Performance decomposition

To further analyze the performance gain of AdapLoRa, We first evaluate the gain of the proposed fine-grained network model and adaptation overhead consideration. We extended ADR provided by LoRa specification [9] to achieve energy fairness (AdapLoRa-sim), as described in Section III. Figure 10 depicts the estimated network lifetime at different time according to current PER, the calculation can be referred to Eq. (3). Results show that although the estimated lifetime fluctuates due to varying link quality, AdapLoRa outperforms AdapLoRa-sim most of the times. Finally, we obtain the real network lifetime based on the energy consumption during the 20 minutes (the right sub-figure), and we observe that AdapLoRa-sim that uses packet-level adaptation has shorter network lifetime than AdapLoRa, emphasizing the necessity of fine-grained network model and overhead consideration.

We decomposed AdapLoRa and evaluate AdapLoRa without the adaptation of transmission power. Figure 11 shows the network lifetime of AdapLoRa and EF-LoRa that keeps using the maximum transmission power (i.e., 14 dBm) with 2 gateways and 20 end devices. It can be observed that the network lifetime with maximum transmission power is always lower than that with configurable transmission power. The reason is that with the maximum transmission power, all the end devices have a long communication range. As a consequence, the interference increases significantly, and it is harder to achieve energy fairness than with the schemes that allocate different transmission power to end devices. Furthermore, even with the maximum transmission power, AdapLoRa can still outperform EF-LoRa by 7.5% with power allocation and 6.7% without power allocation, which indicates the benefits of dynamic energy fairness.

D. Algorithm execution time

Since downlinks latency not only includes the latency of adaptation decision packets dissemination but also includes the time for running the algorithm. We evaluated the execution time of the proposed algorithm with different number of gateways. Figure 12 shows that the algorithm execution time is longer with more gateways, because the topology (e.g., different distances from an end device to the gateways) makes it more complex as we have to calculate a PER of an end device on each gateway. Besides, 80% of the instances the execution time with 4 gateways is less than 1 minute, and for scenarios with one to three gateways, most of the execution times are less than 4 seconds (with one gateway) and 20 seconds (with two and three gateways), which is acceptable for low duty cycle LoRa networks. The small proportion of long execution time comes from the fact that sometimes AdapLoRa cannot find an adaptation that has enough lifetime improvement, so it has to seek through many possibilities.

VI. Limitation and future works

Although AdapLoRa can adapt the resources against the dynamic wireless environment, it is based on static network topology where the gateways are pre-deployed and the end devices do not move. As a result, 1) The network lifetime will be greatly affected by the deployment of the LoRa networks. Specifically, different distances between the gateways and the end devices lead to different received signal power, so the network lifetime can be further improved through proper network deployment. 2) The mobility of end devices will introduce much more dynamics in LoRa networks. We may analyze the pattern of the mobility of end devices and predict the transmission performance such as packet reception ratio to prolong the network lifetime under mobile LoRa networks.

VII. Conclusion

In this paper, we proposed AdapLoRa that maximizes network lifetime by dynamic resource adaptation in LoRa networks. Considering the dynamic wireless links and the overhead of resource adaptation, AdapLoRa periodically estimates the network lifetime with different resource allocations. A fine-grained network model is proposed to capture the dynamics of LoRa links and network interference, considering the impact of error correction scheme. As long as a new allocation can improve network lifetime by more than a threshold, this allocation will be performed in the networks. Extensive experiments on a testbed of 4 gateways and 20 end devices showed that AdapLoRa can achieve a longer network lifetime and better energy fairness, compared to other existing methods.

VIII. Acknowledgements

We thank anonymous shepherd and reviewers for helpful comments. This work is supported in part by the National Natural Science Foundation of China (No. 61972074 and No. 61972075), the China Postdoctoral Science Foundation (No. 2018M640909), the National Postdoctoral Foundation for Innovative Talents (No. BX201700046), and the National Key Research and Development Program of China (2017YFB1400102).

REFERENCES

[1] B. Ghena, J. Adkins, L. Shangguan, K. Jamieson, P. Levis, and P. Dutta, "Challenge: Unlicensed lpwans are not yet the path to ubiquitous connectivity," in *Proceedings of ACM Mobicom*, 2019, p. 43.

[2] O. Iova, A. Murphy, G. P. Picco, L. Ghiro, D. Molteni, F. Ossi, and F. Cagnacci, "Lora from the city to the mountains: Exploration of hardware and environmental factors," in *ACM EWSN*, 2017.

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

[4] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in *Proceedings of ACM/IEEE IPSN*, 2018.

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

[6] G. Chen and W. Dong, "Exploiting rateless codes and cross-layer optimization for low-power wide-area networks," in *Proceedings of IEEE ICNP*, 2019.

[7] X. Xia, Y. Zheng, and T. Gu, "Ftrack: parallel decoding for lora transmissions," in *Proceedings of ACM Sensys*, 2019, pp. 192–204.

[8] 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 TOSN*, vol. 15, no. 2, p. 16, 2019.

[9] L. Alliance, "Lorawan 1.1 specification," *Technical specification*, 2017.

[10] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *IEEE ICC*, 2017.

[11] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair adaptive data rate allocation and power control in lorawan," in *Proceedings of IEEE WoWMoM)*, 2018, pp. 14–15.

[12] 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.

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

[14] G. Akshay, N. Revathy, L. Anh, R. Anthony, I. Bob, and K. Swarun, "Frequency configuration for low-power wide-area networks in a heartbeat," in *Proceedings of USENIX NSDI*, 2020.

[15] T. Elshabrawy and J. Robert, "Closed-form approximation of lora modulation ber performance," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1778–1781, 2018.

[16] Z. Li, M. Li, and Y. Liu, "Towards energy-fairness in asynchronous duty-cycling sensor networks," in *Proceedings of IEEE INFOCOM*, 2012.

[17] W. Du, Z. Li, J. C. Liando, and M. Li, "From rateless to distanceless: Enabling sparse sensor network deployment in large areas," in *Proceedings of ACM Sensys*, 2014.

[18] T. Istomin, A. L. Murphy, G. P. Picco, and U. Raza, "Data prediction+ synchronous transmissions= ultra-low power wireless sensor networks," in *Proceedings of ACM Sensys*, 2016, pp. 83–95.

[19] S. Basagni, V. Di Valerio, G. Koutsandria, and C. Petrioli, "Wake-up radio-enabled routing for green wireless sensor networks," in *Proceedings of VTC*, 2017, pp. 1–6.

[20] C. Song, A. Wang, F. Lin, J. Xiao, X. Yao, and W. Xu, "Selective cs: An energy-efficient sensing architecture for wireless implantable neural decoding," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 2, pp. 201–210, 2018.

[21] F. Lai, M. Radi, O. Chipara, and W. G. Griswold, "Workload shaping energy optimizations with predictable performance for mobile sensing," in *Proceedings of IoTDI*, 2018, pp. 177–188.

[22] D. Gunatilaka and C. Lu, "Conservative channel reuse in real-time industrial wireless sensor-actuator networks," in *Proceedings of IEEE ICDCS*, 2018, pp. 344–353.

[23] N. S. Ramesan and F. Baccelli, "Powers maximizing proportional fairness among poisson bipoles," in *IEEE INFOCOM*, 2019.

[24] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Communications letters*, vol. 9, no. 11, pp. 976–978, 2005.

[25] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 1–39, 2009.

[26] S. Chen, Y. Fang, and Y. Xia, "Lexicographic maxmin fairness for data collection in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 7, pp. 762–776, 2007.

[27] G. Giambene, T. Bourgeau, and H. Chaouchi, "Iterative multi-level soft frequency reuse with load balancing for heterogeneous lte-a systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 924–938, 2017.

[28] W.-H. Lee, J.-S. Choi, Y.-H. Kim, J.-H. Lee, and S.-C. Kim, "Adaptive sector coloring game for geometric network information-based inter-cell interference coordination in wireless cellular networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 288–301, 2018.

[29] L. Su, C. Yang, and I. Chih-Lin, "Energy and spectral efficient frequency reuse of ultra dense networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5384–5398, 2016.

[30] Y. Fu, Y. Chen, and C. W. Sung, "Distributed power control for the downlink of multi-cell noma systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6207–6220, 2017.

[31] S. Li, Z. Shao, and J. Huang, "Arm: Anonymous rating mechanism for discrete power control," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 326–340, 2017.

[32] J. Zheng, Y. Wu, N. Zhang, H. Zhou, Y. Cai, and X. Shen, "Optimal power control in ultra-dense small cell networks: A game-theoretic approach," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4139–4150, 2017.

[33] A. Gudipati and S. Katti, "Strider: Automatic rate adaptation and collision handling," in *Proceedings of ACM SIGCOMM*, 2011.

[34] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "ptunes: Runtime parameter adaptation for low-power mac protocols," in *Proceedings of ACM/IEEE IPSN*, 2012, pp. 173–184.

[35] H. Lu and W. Gao, "Continuous wireless link rates for internet of things," in *Proceedings of ACM/IEEE IPSN*, 2018, pp. 48–59.

[36] H. Gupta, A. Eryilmaz, and R. Srikant, "Link rate selection using constrained thompson sampling," in *IEEE INFOCOM*, 2019.

[37] R. Sun, M. Hong, and Z.-Q. Luo, "Joint downlink base station association and power control for max-min fairness: Computation and complexity," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1040–1054, 2015.

[38] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of lorawan," *Sensors*, vol. 17, no. 10, p. 2364, 2017.

[39] M. Knight and B. Seeber, "Decoding lora: Realizing a modern lpwan with sdr," in *Proceedings of the GNU Radio Conference*, 2016.

[40] U. Coutaud and B. Tourancheau, "Channel coding for better qos in lora networks," in *Proceedings of IEEE WiMob*, 2018, pp. 1–9.

[41] Semtech, "Sx1272/3/6/7/8: Lora modem-designer's guide an1200.13," *Revision 1*, July 2013.

[42] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.

[43] R. D. Gupta and D. Kundu, "Generalized exponential distribution: different method of estimations," *Journal of Statistical Computation and Simulation*, vol. 69, no. 4, pp. 315–337, 2001.