LMLoRa: Enhancing Link Performance for Mobile LoRa Networks

Ciyuan Chen[†‡*], Zhuqing Xu[§*], Xiaohua Jia[‡], Jingkai Lin[£], Runqun Xiong[†], Dian Shen[†], Xirui Dong[†], Junzhou Luo[†]
[†]*School of Computer Science and Engineering, Southeast University*
[‡]*Department of Computer Science, City University of Hong Kong*
[§]*Nanjing University of Aeronautics and Astronautics;* [£]*Michigan State University*
{cychen, rxiong, dshen, sirid, jluo}@seu.edu.cn; zhuqingxu@nuaa.edu.cn; csjia@cityu.edu.hk; linjingk@msu.edu

***Abstract*—LoRa, as a typical representative of Low Power Wide Area Networks (LPWAN), has been widely used to connect massive IoT devices. However, in mobile applications, there is massive packet loss in LoRa transmission due to link performance degradation, especially when LoRa end-devices move far from the gateway or into obstructed areas. Existing studies take little account of end-device movement, particularly when the movement pattern is unknown. We propose LMLoRa to enhance the Link Performance for Mobile LoRa networks in general scenarios. The key observation is that repeating the original packet content enhances link performance and allows smaller and more energy-efficient transmission parameter selections. Technically, LMLoRa proposes a link performance estimation model for mobile LoRa networks based on packet content repetition. Second, we exploit key hardware features of LoRa to obtain much continuous RSSI information for link quality prediction. Additionally, LMLoRa develops a channel frequency allocation policy to mitigate transmission collisions. Finally, LMLoRa designs a communication mechanism to assist the estimation model and work the whole system with low communication overhead. We design and implement LMLoRa in complex real-world environments, results show that LMLoRa enhances packet reception rate by 33.4% and energy efficiency by 14.4% on average compared with the state-of-the-art.**

***Index Terms*—LoRa, Mobile Transmission, Link Performance, Communication Mechanisms, MAC Layer**

I. Introduction

In recent years, Low Power Wide Area Networks (LPWAN) have become a key technology in the Internet of Things (IoT) due to their low power consumption, long-distance transmission, wide-area coverage [1], [2]. Among many LPWAN techniques, LoRa has received attention from both academia and industry [3] because of its linear frequency Chirp Spread Spectrum (CSS) modulation and LoRaWAN protocol supported by many vendors. LoRa has been employed in various mobile scenarios, including smart farming, asset tracking, smart city, etc.

However, in these mobile scenarios, LoRa end-device transmission is susceptible to link performance degradation. Since the reception signal level varies with time, frequency, and location in diverse mobile environments, the signal strength may become insufficient for reception on the receiver. It is difficult to select the appropriate transmission parameters to align with the channel condition changes caused by the end-devices movement, consequently leading to a decrease in packet reception rate (PRR) and an increase in energy consumption. Existing studies mainly focus on fixed LoRa networks, typically such as Adaptive Data Rate (ADR) scheme proposed by LoRaWAN [4], and other efforts [5]–[11] to enhance link performance. The above methods commonly assume that LoRa end-devices have fixed locations, i.e., without location changes. As a result, they will suffer from low PRR and slow or difficult convergence in mobile scenarios (e.g., asset tracking). Due to the above limitations, LoRaWAN also proposes blind ADR [12] for mobile applications. It attempts to address the trade-off between coverage and battery life by pre-configuring multiple fixed data rates and intervals for end-devices instead of a single data rate. However, blind ADR still has a high packet loss rate and energy consumption in harsh environments due to the inability to adjust appropriate transmission parameters according to the channel condition changes. There are few studies [13] that focus on mobile LoRa end-devices, but they assume the mobile trajectories of end-devices are known. Thus, there is still a lack of research for general mobile LoRa networks.

The end-device mobility presents challenges to guarantee link performance in LoRa networks mainly due to the following factors: (a) Difficulty in establishing communication efficiently. Due to the very low duty cycle mode and data rate of LoRa, establishing a stable communication link in a fixed LoRa network may take several days [14], [15], and it will be even more difficult in a mobile network due to link condition changes. Besides, due to the movement of end-devices, there is less useful reference information than in the fixed LoRa networks. (b) Difficulty in determining the right transmission parameters. LoRa employs multiple parameters for transmission, it is necessary to choose the right parameters to establish a stable connection in the mobile scenarios without costing too much energy, as well as to consider the scalability of the network. (c) Persistently fluctuating and even disconnected links. When the transmission duration increases, the signal-to-noise ratio (SNR) fluctuates significantly during a single packet transmission [6]–[8], which will be more serious in mobile situations. Especially when end-devices move far

* Both authors contributed equally to this work.

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

away from the gateway or encounter obstructed areas such as buildings, forests, etc., the transmission will suffer from severe link performance degradation.

In this paper, we propose LMLoRa to enhance Link Performance for Mobile LoRa networks. This is inspired by the observation that repeated transmission of original packet content can increase its successful reception probability under unstable link conditions. Meanwhile, the repeated transmission allows for the selection of smaller transmission parameters, thereby avoiding energy consumption increases. We call this observation *Repetition Effect* (see Section II-B for details). It effectively addresses the issue of PRR degradation caused by end-device mobility. Thus, the key idea of LMLoRa is to select the optimal parameters based on *Repetition Effect* for mobile LoRa networks, ensuring successful transmission without excessively increasing end-device energy consumption.

Specifically, we establish a link performance estimation model for mobile LoRa networks to estimate the PRR under different transmission parameters. This estimation model is based on the *Repetition Effect* and will select the optimal parameters for maximum PRR with the least energy consumption. Then, to make the link estimation model more accurate, we employ a key hardware feature of LoRa that the end-device can receive continuous RSSI information from a single LoRa packet. It enables us to dig out more information from a single packet in mobile environments with variable link conditions. The major contributions of this paper are as follows:

- We observe and thoroughly analyze *Repetition Effect* in various environments under different transmission parameters. We propose LMLoRa to enhance the link performance for general mobile LoRa networks.
- We propose a novel link performance estimation model for mobile LoRa network based on our observed *Repetition Effect*. Second, we propose a prediction model based on continuous RSSI to predict the future link quality. Then, we develop a channel frequency allocation policy to mitigate transmission collisions. Finally, we design a communication mechanism to assist the estimation model and work the whole system with little overhead.
- We implement LMLoRa on a real-world testbed and evaluate it under various mobile scenarios. The experimental results show that, compared with the state-of-the-art method blind ADR, LMLoRa improves PRR by an average of 33.4% and energy efficiency by 14.4%.

II. Background and Motivation

We first introduce LoRa transmission characteristics and then illustrate preliminary experiments that motivate this work.

A. Background

The LoRaWAN protocol operates at the media access control (MAC) layer and employs a star topology comprising end-devices, a gateway, and a network server. The end-device facilitates data transmission, while the gateway serves as a relay, transmitting packets from end-devices to the network server. LoRa utilizes the linear frequency CSS modulation at the physical layer. These robust signal processing capabilities increase the LoRa transmission range, enabling urban coverage up to 5 km and rural coverage up to 15 km areas [16].

Transmission Parameters. Carrier Frequency (CF), Spreading Factor (SF), and Transmission Power (TP) are three key parameters that affect the performance of LoRa transmissions. SF indicates the number of chirps used to encode a single symbol, typically ranging from SF7 to SF12. An increase in SF expands the transmission range, with higher energy consumption and a lower data rate. Because SF values are discrete, they create tiered transmission distances, with each SF corresponding to a transmission range. For instance, utilizing SF7 yields a communication range of 5 km, whereas SF12 achieves a range of 10 km, with a PRR of 70% [17].

According to LoRaWAN specifications [18], the available TP values are $\{2, 5, 7, 10, 12, 14, 16, 17, 20\}$ dBm, where a higher TP expands the communication distance but incurs greater energy consumption.

CF denotes the central frequency at which LoRa signals are transmitted. Regions and countries across the globe allocate distinct frequency bands for LoRa communications. A gateway can only concurrently process signals on up to 8 CFs.

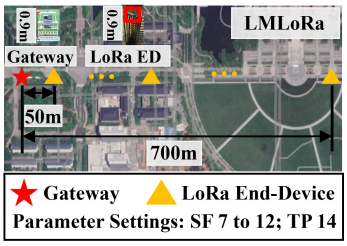
When packets transmit with the same SF and CF and overlap in time, transmission collision will inevitably occur. Therefore, we need to allocate appropriate CFs to end-devices based on network congestion and SF adoption to mitigate transmission collisions.

Adaptive Data Rate (ADR). The Adaptive Data Rate (ADR) mechanism in LoRaWAN continuously monitors the SNR and RSSI of uplink transmissions to calculate an optimal SF and TP for each end-device, consequently optimizing network capacity and energy efficiency. ADR is effective for end-devices that have fixed locations but may struggle to maintain performance when in mobile scenarios [12]. This is because ADR has difficulty adapting to end-devices' mobility and finding suitable parameters.

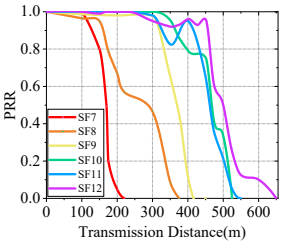
Blind ADR [12] is a mechanism proposed by LoRaWAN for mobile applications like GPS-plus-LoRaWAN pet tracking. Blind ADR pre-configures multiple SFs and transmission intervals without considering real-time channel condition feedback, such as transmitting at SF12 once an hour, at SF10 twice an hour, and at SF7 three times an hour, to adapt to end-device movements at different distances. Although blind ADR offers a solution for mobile applications over ADR, it is easy to lose packets in a blind way.

B. Motivation

End-devices transmission may suffer from link performance degradation during movement, such as when moving away from the gateway, due to a sudden drop in PRR occurring at the communication range edge of each SF (see *Edge Effect* below). This motivates us to enhance the link performance for mobile LoRa networks to address any potential performance degradation. Moreover, we observe *Repetition Effect* to solve the problem in mobile scenarios by repeating packet content.



(a) Complex Campus Deployment



(b) PRR of different SFs

Fig. 1. The Edge Effect of SFs. The experimental results in Fig. 1(b) indicate that within each SF communication range, the PRR is relatively stable, while at the communication edge, the PRR will rapidly decrease. Once end-devices move to each SF communication edge, they will suffer from severe link performance degradation and should adjust transmission parameters in time.

Edge Effect. SF is a critical parameter in LoRa transmissions. Different SFs correspond to distinct data rates and transmission ranges. As depicted in Fig. 1, we observe a sudden drop in the PRR occurring at the communication edge of each SF, which we call the *"Edge Effect."* The experimental setup in Fig. 1(a) is outlined below: The TP is 14dBm, and the packet length is 40 bytes. The end-device transmits packets to the gateway at different points using different SFs across a 700-meter distance to measure the PRR within each SF's transmission range. At each measurement point, the end-device transmits 100 packets. Fig. 1(b) illustrates that within the communication range of each SF, the PRR remains high, but with a sharp decline to 0 at the edge of the communication range. For instance, SF 7 exhibits a PRR exceeding 80% within the 150-meter range; however, beyond this, the PRR drops to near 0 from 150 to 200 m, a mere 50-meter span. Similarly, SF 10 maintains a PRR above 80% within 400 m but experiences substantial fluctuations between 400 and 500 m.

It means that once the end-devices are across different SF ranges without a timely SF switch, the PRR may drop sharply, indicating a significant and rapid decline in link performance, which easily occurs during the movement.

The *Edge Effect* serves as a representation of the unstable link conditions that end-devices will encounter during movement. This motivates us to enhance the link performance of mobile LoRa networks so that the end-devices can still have stable link performance when the environment changes.

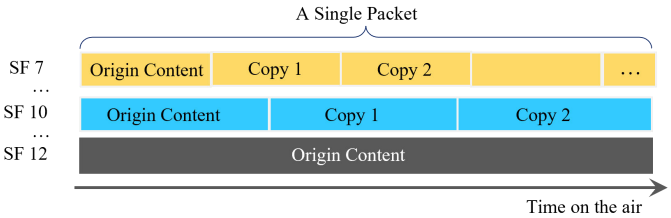
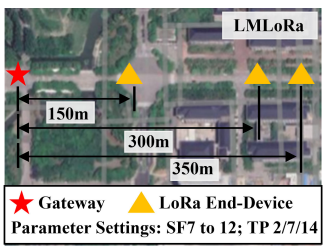
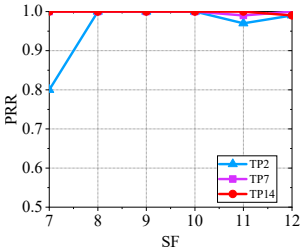


Fig. 2. The packet structure under *Repetition Effect*.

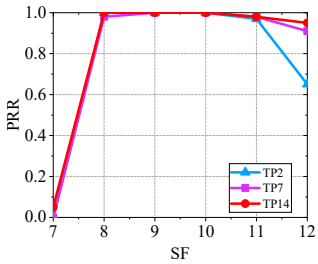
Repetition Effect. *Repetition Effect* means we can improve the PRR and select smaller and more energy-efficient parameters by transmitting original packet content repeatedly. Specifically, as depicted in Fig. 2, we synchronize the transmission duration of all SFs by repeating the Origin Content with many



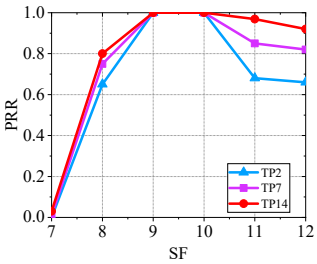
(a) Complex Campus deployment



(b) PRR at 150m with repetition



(c) PRR at 300m with repetition



(d) PRR at 350m with repetition

Fig. 3. The *Repetition Effect* for LoRa transmission. By repeating the packet's original content, the PRR is increased, and smaller transmission parameter selections are enabled. As shown in Fig. 3(d), due to content repetition, SF 9 achieves a higher PRR than SF 11 and SF 12.

copies. For example, the time takes for SF 12 to transmit a packet is equivalent to SF 7 transmitting more than 20 times with the same original content. We conduct experiments at different distances on campus, including 150 m, 300 m, and 350 m, to verify *Repetition Effect* with each SF. Each packet is transmitted for the same time on air with different SF, TP, and repetition numbers.

Fig. 3 illustrates the PRR achieved by different SFs at different distances through the *Repetition Effect*. Take Fig 3(d) as an example, which has the most unstable link quality and is more representative of complex mobile environments. Firstly, observe the curve of the same TP; take TP=7 as an example; the PRR rises with an increase in the SF but eventually declines because a larger SF increases the interference resistance but simultaneously reduces repetition numbers. It indicates that an individual copy of content loss has a larger impact on PRR when using large SF. And the *Repetition Effect* allows smaller SFs to achieve a higher PRR. Secondly, we observe the combination of SF and TP to reach a certain PRR. Taking about 95% PRR as an example, SF 11 and SF 12 require TP 14 to reach above 95% PRR, while SF 9 reaches 95% PRR even with TP 2. This is because, during the same time on air, the packet transmitted by SF 9 is repeated about 6 times, while SF11 is only repeated twice, and SF 12 once, resulting in SF 9 with a higher PRR and smaller TP.

The advantages of the *Repetition Effect* are mainly as follows: Firstly, it alleviates the sudden drop in PRR when link performance degrades, such as the *Edge Effect*. Since *Repetition Effect* reduces packet loss caused by link fluctuations through repeated transmissions, thereby enhancing the link performance of the original packet content. Secondly, it allows the use of smaller transmission parameters at long distances, thereby reducing energy consumption. It can be

observed that when in the transmission range of SF 7 and 8, their PRR is not inferior to the larger SFs, due to the fact that smaller SFs benefit more from repeated transmissions in their transmission range, whereas larger SFs have almost no repetitions. SF9 and 10 have the highest PRR due to their long transmission distances and moderate content repetition. Larger transmission parameters, such as SF11 and SF12, are only necessary when transmission distances are significantly longer or when environments are particularly harsh, as they offer greater resistance to interference. *Repetition Effect* increases the transmission rate while reducing energy consumption by enabling the use of smaller SF values.

Repetition Effect can significantly improve the PRR of each SF at different distances and enhance link performance for mobile LoRa networks to cope with changing link conditions. Within the transmission range of small SFs like SF 7 to 10, it is advisable to use a more energy-efficient SF as much as possible, which can improve PRR, expand transmission distance, and consume less energy through *Repetition Effect*.

III. Problem Definition

We primarily use PRR to indicate the link performance, with the objective of maximizing PRR. The input mainly includes the continuous RSSI information, SNR, and current network status contained in the latest received packet. The output consists of the transmission parameters, including the SF, TP, and the number of repetitions. In addition, under the premise of successfully transmitting the original packet content, we need to ensure as small transmission parameters and as few communication exchanges as possible to reduce end-devices energy consumption. The goal is formulated as follows:

$$max \sum_{i}^{n} PRR_i(Rep, SNR, SF, TP) \qquad (1)$$
$$min\ E_i \qquad i \in \{1, 2, ..., n\}$$

where PRR_i represents the PRR of each packet i, indicating the link performance, n is the total amount of packets. Rep means the repetition numbers of the original packet content. E_i is transmission energy consumption of each packet i. This multi-objective problem includes two main objectives: first, we aim to maximize the PRR. When the PRR is equal, we select the parameter that consumes the least amount of energy.

We design the following system to address the problem of enhancing link performance in mobile LoRa networks.

IV. Design

A. Overview of LMLoRa

To the best of our knowledge, LMLoRa is the first system to enhance the link performance for general mobile LoRa networks. Fig. 4 illustrates its architecture.

We first define a link performance estimation model that qualitatively describes the effect of transmission parameters such as SF, TP, and repetition numbers (Rep) on PRR. This model takes the SNR predicted by the prediction model as input, leverages the transmission setting selector to search the optimal transmission parameters, and then outputs them.

Second, we observe the correlation between the continuous RSSI information of the latest received packets and the movement of end-device. Combined with the relationship between transmission parameters and SNR, we design a prediction model that serves as an input to the estimation model.

Third, we develop a channel frequency (CF) allocation policy to mitigate transmission collisions. The policy mainly leverages the network congestion information contained in the latest downlink packet to select the appropriate CF. The CF is then combined with the parameters output by the estimation model to form the final transmission settings to uplink packets.

Finally, the above parts work together through a communication mechanism. The communication mechanism enables the gateway and end-devices to communicate with minimal overhead. End-devices first transmit uplink probes to notify the transmission requirements. Then the gateway transmits downlink packets indicating the current network status. End-devices then select the optimal transmission parameters.

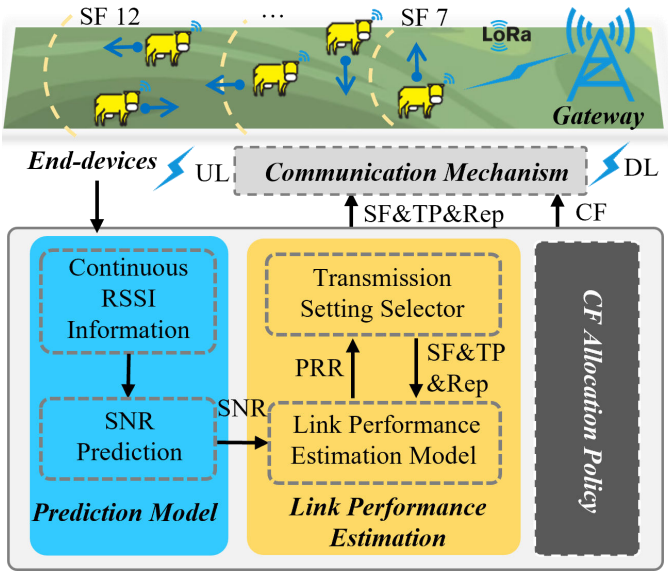


Fig. 4. The architecture of the LMLoRa.

B. Link Performance Estimation for Mobile LoRa Networks

The estimation model establishes the relationship between transmission parameters and PRR. The input is the predicted SNR at the uplink moment, and the output is the transmission parameters that theoretically achieve the highest PRR while consuming the least energy.

The PRR of a single LoRa packet is determined by the successful reception probability of the different structures of this packet. Fig. 5 illustrates the structure of a LoRa packet, which comprises three parts: the preamble, header, and payload. According to the LoRaWAN, we mainly repeat the preamble and payload of the original packet and then form the new uplink packet. Therefore, we need to reformulate the PRR calculation based on the *Repetition Effect*.

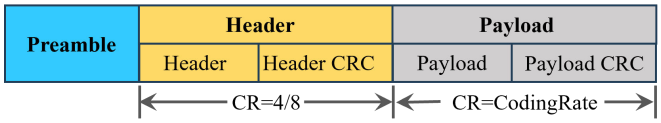


Fig. 5. LoRa packet structure.

Each component consists of a sequence of chirp symbols; thereby, the PRR can be estimated from the symbol error rate. In an Additive White Gaussian Noise (AWGN) channel, the symbol error rate P_s is determined as [5], [19]:

$$P_s(SNR, SF) = 0.5 \times Q(\sqrt{10^{\frac{SNR}{10}} \cdot 2^{SF+1}} - \sqrt{1.386 \cdot SF + 1.154}), \quad (2)$$

where $Q(\cdot)$ denotes the tail function of the standard normal distribution. SNR is the Signal Noise Ratio of the packet.

The preamble comprises three elements: the variable preamble, the frame synchronization word (Sync Word), and the frame start delimiter (SFD). The variable preamble consists of $(4+n)$ up-chirp symbols, with n varying from 6 to 65535. Sync Word spans 2 symbols in length. SFD is a standard down-chirp signal with a width of 2.25 symbols. Let Rep be the repetition times of the origin preamble. The probability of preamble detection with *Repetition Effect*, denoted as P_{pre} is:

$$P_{pre}(Rep, SNR, SF) = 1 - P_s(SNR, SF + log_2(8.25 + Rep \cdot n)). \quad (3)$$

The header typically encompasses metadata concerning the data packet, including payload length, encoding rate, and CRC options. LoRa employs the Forward Error Correction (FEC) encoding mechanism. The header is uniformly encoded at a coding rate (CR) of 4/8, signifying that an equal number of redundant bits are added for each quartet of data bits. Given a header length L_h, the header detection probability P_{header} can be calculated as follows:

$$P_{header} = [(1-P_s)^4 + 3(1-P_s)^7 P_s]^{\lceil L_h/4SF \rceil}. \quad (4)$$

The payload length is variable. The CR of the payload is mainly influenced by transmission distance, link quality, and the payload data volume. Typically, the CR is denoted as a ratio, for instance, $4/5$, $4/6$, $4/7$, or $4/8$, wherein the numerator indicates the number of actual data bits, and the denominator accounts for both the data bits and the additional error correction bits. For example, a coding rate of $4/5$ implies that for every 4 data bits, 1 redundant bit is integrated. 4/5 and 4/6 are limited to error detection only, while 4/7 or 4/8 offer error correction. Consequently, different CRs yield a distinct probability of successful payload decoding. Let L_p represent the payload length, we determine the payload decoding probability of the original content OP_{pl} as follows:

$$OP_{pl} = \begin{cases} (1-P_s)^{\lceil L_p/SF \rceil} & \text{CR=1,2} \\ [(1-P_s)^4 + 3(1-P_s)^{(CR+3)} P_s]^{\lceil L_p/4SF \rceil} & \text{CR=3,4.} \end{cases} \quad (5)$$

Then, the payload decoding probability P_{pl} based on *Repetition Effect* is calculated as:

$$P_{pl}(Rep, SF) = 1 - (1 - OP_{pl})^{Rep} \quad (6)$$

The successful decoding probability for the whole packet with *Repetition Effect* is:

$$P(Rep, SNR, SF) = P_{pre} \times P_{header} \times P_{pl} \quad (7)$$

The estimation model runs the transmission setting selector to obtain the optimal transmission settings, including SF, TP, and repetition numbers. The selector traverses the parameters until it identifies the most energy-efficient transmission settings that meet the PRR requirements.

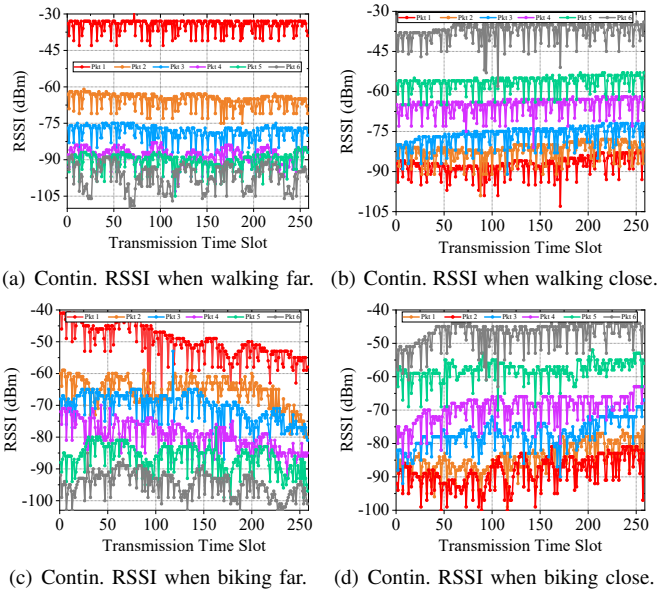
Algorithm 1 Parameter Selection for Link Performance

1: **Input:** The downlink packet from the gateway (GW)
2: **Output:** Optimal SF, TP and repetition number Rep for each transmission
3: Get continuous RSSI info. from the downlink packet.
4: $RSSI_{ED_i}^{SF_i} \leftarrow$ Using the prediction model in Section IV-C to predict the received RSSI by the GW at the moment that the end-device (ED) transmits the uplink packet;
5: $SNR_{ED_i}^{SF_i} \leftarrow RSSI_{ED_i}^{SF_i} + \sigma$, where σ is the environment noise obtained from the received packet;
6: $PRR \leftarrow$ -∞;
7: $Energy \leftarrow +\infty$;
8: **for** $SFj \leftarrow \{SF7, SF8, ..., SF12\}$ **do**
9: **for** $TPp \leftarrow \{TP2, TP4, ..., SF20\}$ **do**
10: **for** $Repr \leftarrow \{1, 2, ..., \left\lfloor \frac{\text{packet length restriction}}{\text{single packet lenth}} \right\rfloor\}$ **do**
11: $prr \leftarrow$ Calculation of $PRR(SFj, TPp, Repr)$ using Equation (7);
12: $energy \leftarrow$ Calculate the energy of transmitting this packet with the parameters in Line 12;
13: **if** $(prr > PRR)$ or $(PRR = prr$ and $Energy > energy)$ **then**
14: $PRR \leftarrow prr$, get the opt. SF_j, TP_p, Rep_r;
15: $Energy \leftarrow energy$
16: **end if**
17: **end for**
18: **end for**
19: **end for**
20: Each ED uses the CF allocation policy in Section IV-D to obtain CF_i.

Algorithm 1 represents the operation of the whole system. Based on the communication mechanism in Section IV-E, each end-device (ED) with an uplink requirement transmits an uplink probe packet, then the end-device(s) can receive the downlink packet from the gateway, which is the input of Algorithm 1. The key part is in Lines 6 to 19, executing the link performance estimation model. By traversing all the appropriate values, the parameter with the least energy consumption that satisfies the successful reception of the packet is selected. Lines 3 to 5 execute the prediction model in Section IV-C as input to the estimation model. Each ED gets

continuous RSSI information based on the downlink packet from the gateway. The RSSI at the actual uplink can be predicted from the received continuous RSSI, and thereby, the corresponding actual SNR can be calculated. Line 20 allocates CF to mitigate transmission collisions.

C. SNR Prediction with Continuous RSSI



(a) Contin. RSSI when walking far. (b) Contin. RSSI when walking close.

(c) Contin. RSSI when biking far. (d) Contin. RSSI when biking close.

Fig. 6. Continous RSSI under different mobile scenarios. Continuous RSSI shows a decreasing/increasing trend as the end-device moves far/close to the gateway, and the faster it moves, i.e., the longer the distance, the more obvious the trend becomes.

In mobile scenarios, the link quality experiences fluctuation; the SNR of the received downlink packet may differ from the SNR of the actual uplink packet. Therefore, it is crucial to estimate the SNR at the moment the packet is really sent. Current studies [5], [13] mainly use the average SNR of historical packets. However, these methods are not applicable in scenarios where end-devices move unpredictably, such as when the duty cycle is large, making historical data of limited reference value. Since the SNR is determined by RSSI and noise level, we can derive SNR from RSSI.

By the function of LoRa packet reception that RSSI can be sampled during the reception of the payload and used to extract a more high-signal RSSI measurement [20], we obtain hundreds of RSSI values from a single packet to predict future RSSI, thereby obtain the future SNR. As shown in Fig. 6, we conduct experiments in various mobile environments. It can be seen that the change of RSSI during the packet reception is affected by the moving speed (walk and bike) and direction (go close and far). Specifically, when the end-devices move faster, such as biking, the continuous RSSI shows a more obvious trend of increasing (close) or decreasing (far).

This inspires us to leverage the continuous RSSI of the latest received packets to predict the future RSSI and the corresponding SNR at the moment of uplink. Specifically, since the newer the data, the closer it is to the true value, we employ an exponentially weighted moving average (EWMA) method [21], considering the computational and storage limitations of end-devices. Using EWMA, a more updated RSSI has a larger weight, reflecting a more accurate RSSI change than a simple average value. According to the relationship between SF, TP, and SNR [5], the PRR_i in the uplink moment can be expressed as follows:

$$PRR_i(Rep, SNR, SF, TP) = PRR_i(Rep, SNR(RSSI_{pre}; RSSI_c, SNR_c) + Gap(SF_c, TP_c; SF, TP), SF, TP), \tag{8}$$

where $RSSI_c$, SNR_c, SF_c, TP_c are the current RSSI, SNR, SF, and TP. $SNR(\cdot)$ is used to calculate the future SNR based on the current SNR, RSSI, and the predicted RSSI. Considering that packets sent with different transmission parameters exhibit varying SNRs [5]. We use $Gap(\cdot)$ to calculate the SNR gap between using the current transmission parameters and the estimated transmission parameters.

D. CF Allocation Policy under Limited Downlink

The above estimation model allowed us to determine transmission parameters, including SF, TP, and repetition numbers. In addition, we need to identify CF, which will influence the transmission collisions. The CF allocation policy can be divided into two parts. First, considering the high concurrency of LoRa applications, we provide different lengths of CF instructions for different network states. Second, the policy indicates the busy state of each CF and provides indication information for end-devices to select CFs.

Specifically, we first determine the instructions' length. The gateway transmits downlink instructions of different lengths depending on different network congestion. Because there are more uplinks in a congested network, more downlink packets and faster downlink speeds are needed to serve more end-devices. The congestion is defined as the number of probs received per unit time. Users customize the congestion threshold according to the network scale. In scenarios of high network congestion, the downlink instruction is condensed into a single byte, where each bit signifies the occupancy status of a CF, i.e., a '0' denotes idle, while a '1' indicates busy. In scenarios of a relatively idle network, the downlink instruction extends to 4 bytes to accommodate a more granular indication of channel activity, distinguishing among 16 levels of busyness. Second, to indicate the CF status, the gateway listens to the packet received on each CF and determines the busy status of each CF based on the number of packets received per unit of time. Downlink packets including CF busy status are transmitted at Phase 1 at different granularities.

Once end-devices receive the downlink packet, they first normalize the busyness levels of various CFs and then select an appropriate CF for transmitting data packets based on those normalized probabilities.

E. Communication Mechanism with Low Overhead

We propose an energy-efficient and practical mechanism to realize the communication between the gateway and end-

devices. Since LoRa typically operates in a low-duty cycle mode and has sparse traffic, frequent interactions with the gateway will lead to substantial communication overhead. Second, the gateway has limited parallel reception capacity, and an excess of downlink traffic will constrain the concurrent transmission of the system. Thus, it is important to minimize communication overhead while still allowing the end-devices to get the necessary information for parameter selection.

Our gateway operates in full-duplex mode [22], enabling it to transmit and receive packets simultaneously. The LoRaWAN regional parameters specification [18] assigns distinct uplink and downlink channels for most regulatory regions globally. In addition, we can utilize independent RF chips for downlink transmission, which makes a full-duplex gateway both practical and economical. Thus, the following interaction process exclusively utilizes the independent downlink channel of the full-duplex gateway, which means there is no conflict between uplink and downlink.

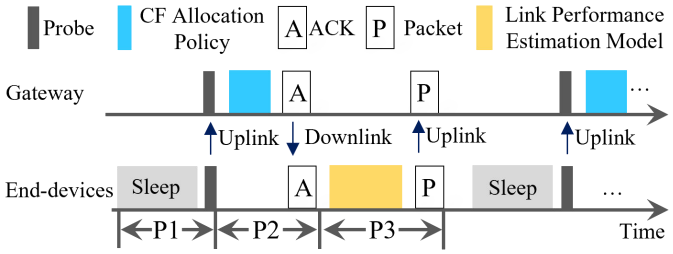


Fig. 7. The communication mechanism of LMLoRa.

Specifically, as depicted in Fig. 7, the communication mechanism is structured into three phases. (1) *P1: The Probing Phase.* The end-device remains in SLEEP mode during periods without communication needs and wakes up into TX mode upon requiring communication. The end-device employs an extra uplink channel to send a 1-byte probe detection packet to the gateway by the maximum SF and TP to maximize the probability of successful reception of the prob. The extra channel is employed specifically to ensure the probing phase will not occupy the standard uplink channel or impinge on the gateway's downlink capacity. Then the end-device switches to Channel Activity Detection (CAD) mode to await the downlink instructions from the gateway. Notably, energy consumption in CAD mode is only half of that in receiving (RX) mode [20], [23]. (2) *P2: Adaptive Downlink Phase at the Gateway.* After receiving the end-devices' probe, the gateway adaptively transmits downlink packets including the current network status (see details in Section. IV-D), which is informed by probes. (3) *P3: End-device Uplink Phase.* Once the end-device has received the gateway's downlink instructions, the end-device can obtain the RSSI of the downlink packet and the CF busyness levels. By inputting the RSSI into the prediction model and running the link performance estimation, end-device can determine the optimal SF, TP, CF, and repetition numbers to uplink packets.

Utilizing this lightweight and practical communication mechanism, end-devices obtain the current network and environment status by only a single communication with the gateway, ensuring the transmission efficiency of the system.

V. Evaluation

A. Experiment Setup

Experimental platform: We develop a real-world testbed consisting of end-devices, a gateway, and a network server to evaluate LMLoRa comprehensively. Each end-device is equipped with an STM32L152 microcontroller unit (MCU), an SX1278 transceiver chip, and a lithium-polymer battery. The gateway operates in full-duplex mode, handling eight concurrent uplink channels and a single downlink channel within the CN470-510MHz frequency band. For uplink communication, the gateway employs an SX1301 digital baseband processor and two SX1255 RF front-end chips, while downlink communication is managed by a single SX1278 transceiver. A Raspberry Pi 3 controls these components via two serial peripheral interface connections. The network server runs on a laptop with an Intel Core i7 processor and 16 GB of RAM.

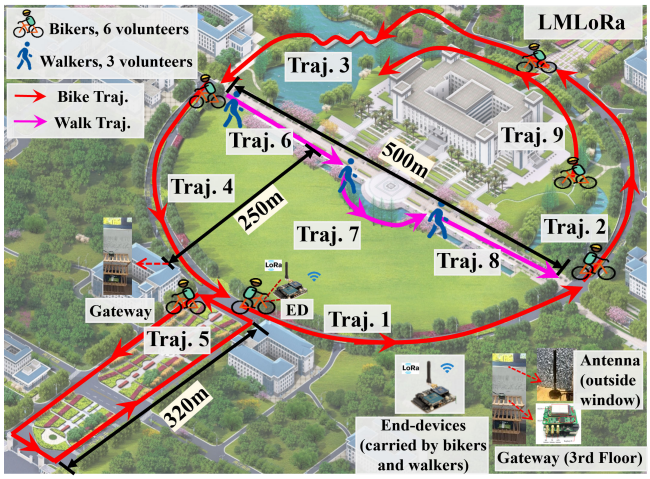


Fig. 8. The complex campus experimental scenario with 9 trajectories. 6 volunteers bike (Traj. 1 to 5 and 9), and 3 volunteers walk (Traj. 6 to 8), each carrying an end-device to move around their own trajectory. The gateway is placed on the third floor with the antenna outside the window.

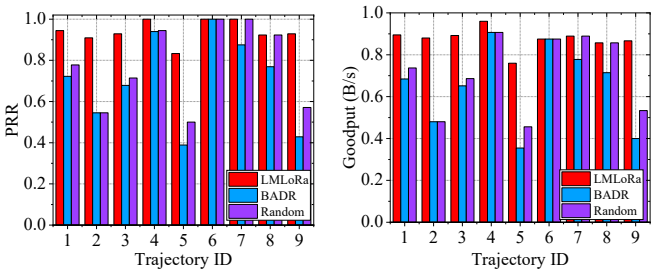
Scenarios: As shown in Fig. 8, we deploy a real-world testbed in a complex campus. The gateway is deployed on the third floor of the building, and the antenna is placed outside on the balcony. The network server is placed next to the gateway. The end-devices are carried by 9 volunteers, each carrying an end-device and moving along their respective trajectories (the movement pattern is unknown by the end-devices), with trajectories (abbr. Traj.) 1-9. These trajectories contain different environments, specifically including tree shading (trajectories 1 to 4), building shading (trajectories 5 and 9), and open lawn (trajectories 6 to 8). We arrange for the volunteers on trajectories 1 to 5 and 9 to move by bike and volunteers on trajectories 6 to 8 to move by walking. The bikers and walkers start simultaneously and move at a stable speed on their own trajectory, with average speeds of around 1.5 m/s and 0.5 m/s, respectively. The settings of transmission from

end-devices are as follows: bandwidth (BW), coding rate (CR), and single packet length are configured to 125 kHz, 4/5, and 20 bytes, respectively. The packet length under *Repetition Effect* should not exceed the longest length of each SF specified by LoRaWAN specification [18]. The experiment is run five times, and the results are averaged.

Baseline: Since LoRaWAN has proposed the default method for mobile applications: blind ADR (abbr. BADR) [12], we mainly compare LMLoRa with blind ADR. We also implement a *Random* comparison method, where the end-devices select one SF at random for each transmission. In the LoRaWAN Technical Paper [12] for blind ADR, end-devices transmit at different SFs every ten minutes; we modify it to suit our scenario. Specifically, end-devices use SF 12 once, SF 10 twice, and SF 7 three times every six transmissions [12], [24], [25]. As we mentioned in related work (see Section VI for details), existing researches [4]–[11] are mainly based on fixed LoRa networks or applications with known trajectories [13], and are not suitable for mobile LoRa networks with unknown movement patterns. Thus, we do not compare with them.

B. Link Performance of LMLoRa

We evaluate PRR and goodput under 9 trajectories to verify the link performance improvement; goodput refers to the amount of origin content transmitted per unit time. We use goodput rather than throughput because the former is more responsive to effective transmission under *Repetition Effect*.



(a) PRR of LMLoRa. (b) Goodput of LMLoRa.

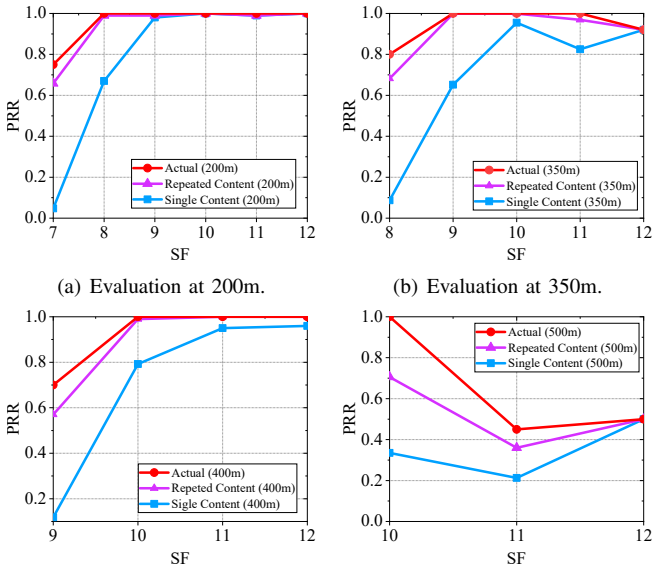
Fig. 9. Evaluation of Link Performance.

As illustrated in Fig. 9(a), the PRR of LMLoRa is consistently higher than that of blind ADR and *Random*. Specifically, trajectories 1 to 4 reflect PRR increases as the distance decreases, and at different distances, the PRR of LMLoRa is consistently higher than that of blind ADR and *Random*. Trajectories 5 and 9 indicate a lower PRR with the shading of buildings. According to the above 6 trajectories, compared to blind ADR and *Random*, LMLoRa can improve PRR more in harsh communication environments, such as long-range transmission and building obstruction. This is because LMLoRa can select more appropriate transmission parameters in time to ensure good link performance in poor environments. Trajectories 6 to 8 have the highest PRR due to the unobstructed environment and the slower moving speed. LMLoRa has a higher PRR and lower performance fluctuation than blind ADR and and *Random* due to its environment perception and packet content repetitions. *Random* outperforms blind ADR due to its more uniform SF selection.The results of the above experiments show that LMLoRa can improve the PRR by 33.4% on average compared to blind ADR, and by 21.4% on average compared to *Random*, with a maximum improvement of 1.2× over blind ADR under building obstructed.

From Fig. 9(b), goodput follows a similar trend to PRR due to its definition. The good performance of LMLoRa in trajectories 2, 3, 5, and 9 shows that the increase in transmission duration due to *Repetition Effect* is negligible compared to the duration of the whole experiments, and it is not very harmful to the goodput of the system.

C. Ablation Experiment of LMLoRa

We evaluate the performance of the main parts of LMLoRa. The ablation experiment mainly includes the evaluation of the performance of the *Repetition Effect* and prediction model.



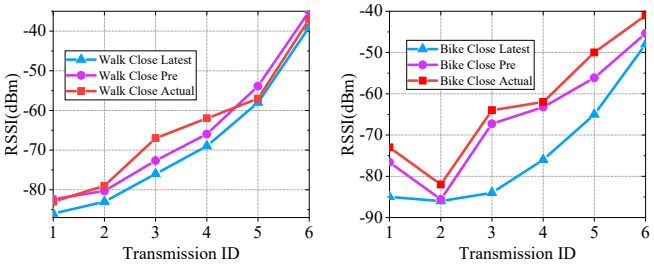
(a) Evaluation at 200m. (b) Evaluation at 350m.

(c) Evaluation at 400m. (d) Evaluation at 500m.

Fig. 10. Evaluation of Repetition Effect.

1) Verification of Repetition Effect: We verify Eq. (7) at different distances, as shown in Fig. 10. The term Single Content refers to the original content without repetition. Repetition content denotes the estimated PRR of the original packet with repetition, which is calculated using Eq. (7) based on the PRR of single content. Actual means the actual results obtained from experiments using *Repeated Effect*, the experimental setup is the same as Section II-B. We can see in Fig. 10(a) that in the short range (200 m), the PRR of single content rises as the SF increases. Repetition can significantly boost the PRR for smaller SFs, for example, the PRR of the single packet in SF 8 is 67%, but the PRR of Repetition reaches 99%, and its true PRR is 100%. It can be observed that Repetition is often lower than the actual PRR, suggesting that meeting the estimated PRR will ensure the actual PRR. In Fig. 10(b), 10(c) and 10(d), the trends are roughly the same, but due to the long transmission duration of SF 11 and 12, leading

to fewer repetitions and less PRR improvement. The PRR decrease of SF 11 may be due to the link fluctuations or the similar performance between SF 11 and SF 10, resulting in a lower PRR for single packets with SF 11.



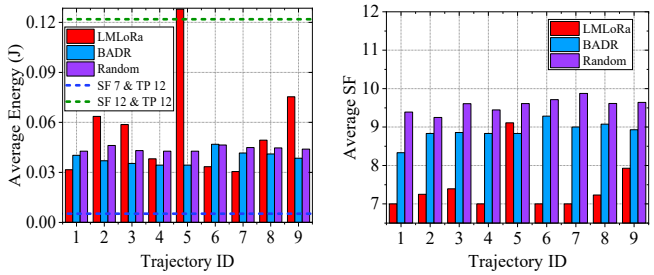
(a) Prediction when walking close. (b) Prediction when biking close.

Fig. 11. Evaluation of Prediction Model.

2) Verification of Prediction Model: We verify the prediction model under different speeds (walking and biking) and directions (going close and far). Fig. 11 shows the results of two of these four experiments due to space constraints. We compare the prediction results (Pre) with the real results (Actual), as well as with the existing methods (Latest). We use the RSSI of the latest packet (Latest) for the comparison, considering the average of the last few packets has a large error in mobile scenarios. It can be observed that our prediction is closer to the Actual compared to the Latest.

D. Evaluation of End-device Energy Consumption

We evaluate the average energy consumption of the LMLoRa, blind ADR and *Random* and analyze the effect of SF selection on energy consumption.



(a) Energy under Multiple Traj. (b) SF under Multiple Traj.

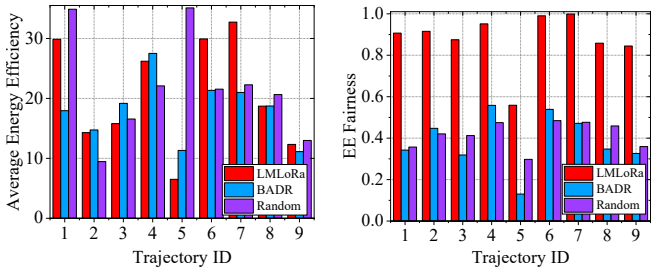
Fig. 12. Evaluation of End-device Energy Consumption.

Fig. 12(a) illustrates the average total energy of each transmission round; the energy consumption of LMLoRa includes detection energy, CAD energy, transmission energy, etc. We can see that when the communication condition is better, such as trajectories 1, 4, 6 to 8, the energy consumption of LMLoRa is lower than or similar to that of blind ADR and *Random*, which is due to the fact that LMLoRa selects the appropriate transmission parameters to adjust the channel condition changes. LMLoRa consumes a lot of energy when the environment is more complex (trajectories 2, 3, 5, and 9), especially when link performance is degraded due to building occlusion. This is because LMLoRa selects larger and more energy-consuming transmission parameters and more repetition times when it detects worse link conditions to ensure the PRR. The results show that LMLoRa reduces 28.7% energy consumption at best than blind ADR. SF 12 and SF 7 refer to the energy consumption when transmission using SF 12 and SF 7 with TP 12. LMLoRa is close to the minimum value in better environments. Thus, LMLoRa can achieve a higher PRR without excessively consuming cost in most cases

Fig. 12(b) depicts the average SF used in each transmission. It can be seen that LMLoRa tends to choose a smaller SF compared to blind ADR and *Random*. This is because the repetition effect of LMLoRa makes it preferable to achieve a better PRR with a smaller SF to avoid unnecessary energy losses. Blind ADR, on the other hand, periodically selects different levels of SFs, resulting in a larger average of values. Only when the environment is consistently severe (like trajectory 5) does LMLoRa use larger SFs for most transmissions, resulting in worse results than blind ADR. LMLoRa reduces the SF by 1.49 SF on average compared to blind ADR.

E. Evaluation of End-device Energy Efficiency

We evaluate energy efficiency (EE) and the corresponding energy efficiency fairness under the 9 trajectories of each transmission round. The EE fairness is measured by Jain's fairness index with the formula $(\sum_1^P EE_p)^2/(P\sum_1^P EE_p^2)$, where EE_p denotes the energy efficiency of p^{th} transmission round of each trajectory, P is the total transmission round amount. Fairness demonstrates the performance fluctuation of the three methods in different environments.



(a) EE under Multiple Traj. (b) EE Fairness under Multiple Traj.

Fig. 13. Evaluation of End-device Energy Efficiency and its Fairness.

As depicted in Fig. 13(a), the energy efficiency of LMLoRa is similar to or even higher than the other two methods. It is only in harsh environments that LMLoRa becomes less energy efficient. This is because the primary goal of LMLoRa is to ensure the PRR, which requires additional energy to guarantee successful transmission. Results show that LMLoRa is on average 14.4% more energy-efficient than blind ADR.

As indicated in Fig. 13(b), compared to blind ADR and *Random*, LMLoRa has higher energy efficiency fairness, i.e., the smaller the performance fluctuation in different environments, which is because LMLoRa chooses different transmission parameters and the number of repetitions according to different environments in order to ensure PRR.

VI. Related Work

Our work is related to studies on link performance improvement in LoRa networks and studies on mobile LoRa networks.

A. Enhancing Link Performance in LoRa Networks

Existing methods mainly assume that LoRa end-devices have fixed locations. Based on the design principles, the existing literature can be divided into two categories: PHY layer methods [4], [6]–[11] and MAC layer methods [4], [5].

PHY layer methods. The PHY layer approaches analyze link signal strength to improve or adjust the link quality, which has not been implemented in the current commercial LoRa chips. PolarTracker [6] is an attitude-aware channel access method for floating LPWAN, which improves PRR by orchestrating transmissions to coincide with moments when the LoRa end-device achieves optimal antenna alignment. PolarScheduler [7], [8] is also a dynamic transmission control method for floating LoRa networks. PolarScheduler fine-tunes the transmission duration and configuration to align with periods when antennas are optimally positioned, thereby improving the PRR and throughput for aquatic IoT applications. XCopy [9] is a system achieving reliable LoRa communication by coherently combining the signal power from multiple retransmissions of weak links, thereby improving the SNR and PRR of LoRa packets in challenging environments. CurvingLoRa [10] is a PHY-layer design that enhances LoRa network throughput by employing non-linear chirp modulation to enable more reliable concurrent transmissions, even in challenging signal conditions with high interference. Ostinato [11] improves the communication reliability of LoRa networks in real-world deployments by transforming standard packets into pseudo packets with repeated symbols, thereby boosting signal SNR.

MAC layer methods. The MAC layer methods mainly select suitable transmission parameters to improve communication reliability. LoRaWAN ADR [4] is a mechanism designed to optimize network performance and prolong end-device battery life by dynamically adjusting the transmission parameters based on current radio conditions. DyLoRa [5] is a dynamic LoRa transmission control system optimizing energy efficiency in LoRa networks by dynamically adjusting transmission parameters based on the physical link properties.

However, the above methods mainly consider fixed end-device geographic location and are not suitable when LoRa end-devices move across a wide area.

B. Mobile LoRa Networks

There are a few approaches for mobile LoRa networks; some [12], [13] consider the mobility of end-devices, while others focus on mobile gateways.

Blind ADR [12] proposed by LoRaWAN enables dynamic adjustment of data rates without relying on network feedback, optimizing for both coverage and battery life by selecting different data rates based on the changing conditions. ShuttleNet [13] is a low-cost LoRa-based networking solution for monitoring campus shuttles, which optimizes the trade-off between network reliability and data throughput.

There also some work focuses on mobile LoRa gateways, but they assume end-devices fixation. FlyingLoRa [26] is an energy-efficient data collection scheme in UAV-assisted LoRa networks, which optimizes the UAV's trajectory, scheduling, and transmission parameters to reduce the energy consumption of end devices. LoRaDrone [27] and MLoRaDrone [28] integrating UAVs equipped with LoRa gateways to reduce the energy consumption of LoRa end-devices by optimizing transmission parameters and UAV flight paths. Delafontaine [29] et al. present a drone-aided localization system for LoRa networks through the integration of UAVs as mobile gateways within existing LoRaWAN infrastructure.

In summary, the existing studies have not improved the link performance specifically for general mobile LoRa networks with unknown end-device mobility patterns. The methods for fixed LoRa end-device locations are not suitable in mobile environments where the link conditions are changing. The existing studies for mobile networks mainly use blind methods or are based on fixed mobile patterns.

VII. Conclusion and Future Work

In this paper, we present LMLoRa, to enhance the Link Performance in Mobile LoRa networks under unknown end-device mobility patterns. In mobile scenarios, link condition fluctuations caused by end-device movement will result in link performance degradation, such as massive packet loss at the edge of each SF communication range when LoRa end-devices move far from the gateway. Inspired by the observation that repeating the original packet content can improve the PRR and allow the selection of smaller transmission parameters with less energy consumption, thereby enhancing the link performance. We develop a link performance estimation model based on the *Repetition Effect*. Then, we propose a continuous RSSI prediction model to provide inputs to this estimation model. Third, we design the CF allocation policy to mitigate transmission collisions. Finally, we present a communication mechanism to work the whole system. We conduct experiments in complex campus environments, and the results show that LMLoRa improves the PRR by 33.4% and energy efficiency by 14.4% on average compared with blind ADR. In future work, we will evaluate the performance of LMLoRa in a larger scenario with more mobile end-devices.

Acknowledgement

We thank the shepherd and anonymous reviewers for insightful comments. This work is supported by the National Key R&D Program of China with Grant number 2021YFB2900100; the National Natural Science Foundation of China under grants 62232004, 62172091, 61632008, 62272101, and 62402221; Natural Science Foundation of Jiangsu Province under grant BK20230083; the Jiangsu Provincial Key R&D Program under grant BE2022065-4; the Jiangsu Provincial Key Laboratory of Network and Information Security under grant BM2003201; and the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under grant 93K-9.

REFERENCES

[1] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, "A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[2] C. Li and Z. Cao, "LoRa networking techniques for large-scale and long-term IoT: A down-to-top survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–36, 2022.

[3] A. W.-L. Wong, S. L. Goh, M. K. Hasan, and S. Fattah, "Multi-hop and mesh for LoRa networks: Recent advancements, issues, and recommended applications," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–43, 2024.

[4] "LoRa Alliance Inc." https://lora-alliance.org/resource-hub/ lorawanr-specification-v11, 2017.

[5] Y. Li, J. Yang, and J. Wang, "DyLoRa: Towards energy efficient dynamic LoRa transmission control," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2312–2320.

[6] Y. Wang, X. Zheng, L. Liu, and H. Ma, "Polartracker: Attitude-aware channel access for floating low power wide area networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1807–1821, 2022.

[7] R. Li, X. Zheng, Y. Wang, L. Liu, and H. Ma, "Polarscheduler: Dynamic transmission control for floating LoRa networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 550–559.

[8] X. Zheng, R. Li, Y. Wang, L. Liu, and H. Ma, "PolarScheduler: Dynamic Transmission Control for Floating LoRa Networks," *ACM Transactions on Sensor Networks*, 2024.

[9] X. Xia, Q. Chen, N. Hou, Y. Zheng, and M. Li, "XCopy: Boosting Weak Links for Reliable LoRa Communication," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[10] C. Li, X. Guo, L. Shangguan, Z. Cao, and K. Jamieson, "CurvingLoRa to Boost LoRa Network Throughput via Concurrent Transmission," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 879–895.

[11] Z. Xu, P. Xie, J. Wang, and Y. Liu, "Ostinato: Combating lora weak links in real deployments," in *2022 IEEE 30th International Conference on Network Protocols (ICNP)*. IEEE, 2022, pp. 1–11.

[12] "Blind ADR," https://lora-developers.semtech.com/documentation/tech-papers-and-guides/blind-adr.

[13] D. Mu, Y. Chen, J. Shi, and M. Sha, "Runtime control of LoRa spreading factor for campus shuttle monitoring," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 2020, pp. 1–11.

[14] S. Li, U. Raza, and A. Khan, "How agile is the adaptive data rate mechanism of LoRaWAN?" in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 206–212.

[15] J. Finnegan, R. Farrell, and S. Brown, "Analysis and enhancement of the LoRaWAN adaptive data rate scheme," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7171–7180, 2020.

[16] Z. Sun, H. Yang, K. Liu, Z. Yin, Z. Li, and W. Xu, "Recent advances in LoRa: A comprehensive survey," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–44, 2022.

[17] 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, pp. 1–35, 2019.

[18] "LoRa Alliance, LoRaWAN Specification v1.1," [Online]. Available: https://lora-alliance.org/resource-hub/lorawanspecification-v1-1, 2021.

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

[20] "SX1276 datasheet," https://www.mouser.com/datasheet/2/761/sx1276-1278113.pdf, 2015.

[21] Z. Xu, J. Luo, Z. Yin, S. Wang, C. Chen, J. Lin, R. Xiong, and T. He, "Leveraging Imperfect-Orthogonality Aware Scheduling for High Scalability in LPWAN," *IEEE Transactions on Mobile Computing*, 2024.

[22] "Semtech SX1302CFDGW1 Introduction," [Online]. Available: https://www.semtech.com/products/wirelessrf/lora-core/sx1302cfdxxxgw1, 2022.

[23] A. Gamage, J. Liando, C. Gu, R. Tan, M. Li, and O. Seller, "LMAC: Efficient carrier-sense multiple access for LoRa," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–27, 2023.

[24] A. Farhad, D.-H. Kim, and J.-Y. Pyun, "R-arm: Retransmission-assisted resource management in LoRaWAN for the internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7347–7361, 2021.

[25] M. Chen, L. Mokdad, J. Ben-Othman, and J.-M. Fourneau, "Dynamic parameter allocation with reinforcement learning for LoRaWAN," *IEEE Internet of Things Journal*, 2023.

[26] R. Xiong, C. Liang, H. Zhang, X. Xu, and J. Luo, "FlyingLoRa: Towards energy efficient data collection in UAV-assisted LoRa networks," *Computer Networks*, vol. 220, p. 109511, 2023.

[27] C. Chen, J. Luo, Z. Xu, R. Xiong, Z. Yin, J. Lin, and D. Shen, "LoRaDrone: Enabling Low-Power LoRa Data Transmission via a Mobile Approach," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2022, pp. 239–246.

[28] C. Chen, J. Luo, Z. Xu, R. Xiong, D. Shen, and Z. Yin, "Enabling large-scale low-power LoRa data transmission via multiple mobile LoRa gateways," *Computer Networks*, vol. 237, p. 110083, 2023.

[29] V. Delafontaine, F. Schiano, G. Cocco, A. Rusu, and D. Floreano, "Drone-aided localization in LoRa IoT networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 286–292.