DFH: Improving the Reliability of LR-FHSS via Dynamic Frequency Hopping

Fanhao Zhang, Fu Yu, Xiaolong Zheng*, Liang Liu, Huadong Ma
Beijing University of Posts and Telecommunications, Beijing, P. R. China
{zhangfanhao, yufu, zhengxiaolong, liangliu, mhd}@bupt.edu.cn

***Abstract*—Long Range-Frequency Hopping Spread Spectrum (LR-FHSS) is a novel wireless communication technology to improve the coverage of Low-Power Wide-Area Network (LPWAN). But our measurement finds that given the same set of sub-channels, different Frequency Hopping Sequence (FHS) can result in a reliability difference of up to 52.6% in terms of Packet Reception Rate (PRR). The key observation indicates that the reliability of LR-FHSS is significantly influenced by the FHS besides the link quality. Hence, in this paper, we propose DFH that takes both link quality and FHS into consideration to improve the reliability of LR-FHSS. We first propose using the hop Signal-to-noise Ratio (SNR), a new indicator to reflect the quality of the sub-channels and establish the PRR prediction model according to hop SNR and FHS. Based on the model, we design an interleaving-based search algorithm to decide the optimal FHS. We implement and evaluate DFH on the commercial transceivers and SDR-based gateway. The results of experiments in real environments show that DFH can improve the PRR by up to** $2.76\times$**, compared to the standard LR-FHSS.**

***Index Terms*—LR-FHSS, frequency hopping sequence.**

I. Introduction

In the past decade, various wireless communication techniques have been widely used in both indoor and outdoor environments [1]–[3]. In the complex indoor environments where links are severely blocked by the wall, traditional short-range techniques such as WiFi and Bluetooth have limited communication distance and reliability. To well support indoor applications such as environmental monitoring [4], [5] and localization service [6]–[8], recent studies try to deploy LoRa to support communications in large buildings with complex indoor environments. As shown in Fig. 1(a), LoRa utilizes the Chirp Spread Spectrum (CSS) modulation that maps the data symbol to the initial frequencies of a liner chirp. However, existing studies show that the performance of LoRa is also unsatisfied in complex indoor environments [9]–[12].

To cope with the performance degradation of LoRa, Semtech proposes Long Range-Frequency Hopping Spread Spectrum (LR-FHSS), a novel wireless communication technology to further increase wireless coverage and support reliable communication networks. Different from LoRa which uses CSS, the key feature in LR-FHSS is using frequency hopping spread spectrum which has been proven to be able to achieve reliable communication under interference [12]. Specifically, LR-FHSS divides the whole data symbols into multiple segments and transmits them in different frequencies, as shown in Fig. 1(b). Each packet contains headers and payloads in different hops. Each hop is in a sub-channel with the bandwidth as narrow as $488Hz$, which is much less than LoRa whose bandwidth is 7.81-$500kHz$. Given the same transmission power, the narrower bandwidth can well concentrate the energy and achieve better ability of penetrable communication. Hence, LR-FHSS can maintain reliable long-range communication in indoor scenarios. The Operating Channel Width (OCW) in Fig. 1(b) defines the bandwidth of whole sub-channels. The order of sub-channel used for all hops is denoted as Frequency Hopping Sequence (FHS). In each hop, LR-FHSS implements Gaussian Minimum Shift Keying (GMSK) to modulate each symbol.

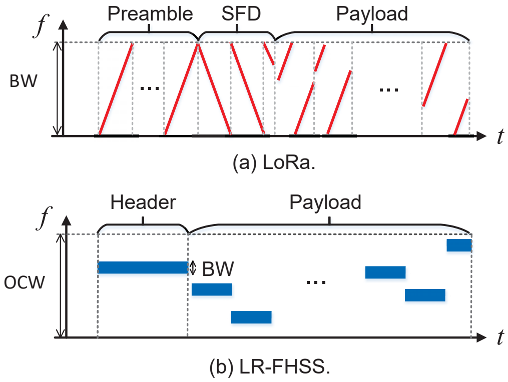


Fig. 1: Time-frequency signal of LoRa and LR-FHSS.

Though promising, our real measurement study finds that LR-FHSS is still seriously influenced by dynamic interference in sub-channels. But directly using existing methods to resolve interference is not suitable for LR-FHSS. The frequency hopping in existing wireless techniques such as Bluetooth and Time Slot Channel Hopping (TSCH) is packet-level hopping, which means that the completed packet is transmitted in each channel. The decoding result of each packet only depends on the quality of the transmitted channel. But for LR-FHSS whose frequency hopping occurs within each packet as shown in Fig. 1(b), only a part of the packet is transmitted in each hop. The transmission of each hop affects the final decoding result of the whole packet. From our experiments, we observe that nodes transmitting in the same group of sub-channels but using different FHS have significantly different reliability performances. The key reason is that different FHS will cause different error patterns of symbols after the deinterleaving of

*Xiaolong Zheng is the corresponding author.

LR-FHSS. If the error symbols are bursty, the Forward Error Correction (FEC) will fail and the packet will be dropped. Hence, existing adaptive hopping studies that only consider the channel quality are not enough for LR-FHSS, and selecting the optimal FHS under dynamic interference is also necessary.

However, to adaptively decide FHS in LR-FHSS, we still face three key challenges. First of all, we need a metric that can reflect the quality of each sub-channel while existing link quality is measured in packet-level granularity. How to identify the suitable metric to achieve fine-grained estimations in each sub-channel is unknown. Furthermore, due to the ignorance of FHS in existing methods, the relationship between FHS and reliability is still unknown. A model that describes the relationship is necessary for following optimization but missing in the literature. Finally, the selection of optimal FHS is an NP-hard problem because the relationship between different FHS and Packet Reception Rate (PRR) is non-linear caused by interleaving and convolution encoding process in LR-FHSS. How to online obtain the optimal FHS is also challenging under highly dynamic channels.

To solve these challenges, we propose DFH, a dynamic frequency hopping method for LR-FHSS that takes both link quality and FHS to improve the transmission reliability in dynamic environments. Firstly, we adopt hop Signal-to-noise Ratio (SNR) instead of the packet's SNR used in existing methods to indicate fine-grained conditions in sub-channels. Secondly, to establish the model between hop SNR, FHS, and PRR, we first establish the model between hop SNR and Symbol Error Rate (SER) and then propose a Monte Carlo based simulation to predict PRR based on the hop SER and FHS. Furthermore, we carefully select the parameters to achieve a good balance between prediction accuracy and overhead. Finally, to online select optimal FHS, we propose an interleaving-based search algorithm to reduce the candidate FHS and speed up the process.

The contributions of this work are summarized as follows:

- We propose DFH, a novel method that dynamically adjusts the FHS, instead of simply blacklisting the poor channels, to improve the reliability of LR-FHSS.
- We propose a model to predict LR-FHSS PRR based on hop SNR and FHS. We also design an interleaving-based search algorithm to quickly decide optimal FHS.
- We implement DFH on the commercial equipment SX1262 node and SDR-based gateway and evaluate DFH in the real deployed environment. The experimental results show that DFH improves the PRR by up to $2.76\times$, compared to the standard LR-FHSS.

II. Related Work

In this section, we first discuss the existing adaptive frequency hopping methods in other wireless systems and then the dynamic control methods in LoRa networks.

A. Adaptive Transmission in Frequency Hopping System

For frequency hopping systems, there are many studies focusing on adaptively selecting parameters for transmission. They adopt the blacklist-based methods which identify poor channels by different indicators and block poor channels in use. In Bluetooth, the standard Adaptive Frequency Hopping (AFH) [13] and methods in [14], [15] rely on Packet Delivery Ratio (PDR) or the Received Signal Strength Indicator (RSSI) to estimate channels and block channels with higher interference. In system using TSCH, RSSI [16]–[19] and PDR [20]–[22] are used to indicate channel quality and disable poor channels in use. There are also some studies using machine learning to estimate channels. Authors in [23] adopt reinforcement learning to self-supervised deep learning to predicate future link quality, respectively.

These methods are not suitable for LR-FHSS to enhance reliability. On the one hand, the frequency hopping in Bluetooth and TSCH is packet-level hopping. A complete packet is transmitted in each hopping channel. The decoding result of a packet only depends on the quality of the specific used channel. But for LR-FHSS, the data payload is divided into multiple segments and transmitted in different channels. Besides the channel quality, the FHS of hops also affects the error pattern in the received packet. Hence, the above methods that don't consider the FHS are inefficient for LR-FHSS. On the other hand, LoRaWAN supports a large number of nodes connecting one gateway, blacklist-based methods will cause multiple nodes to select the same group of channels for transmission. Then serious collisions of packets cause the degradation of overall channel capacity.

B. Adaptive Transmission in LoRaWAN

There are many studies focusing on improving LoRaWAN's performance. Authors in [24]–[26] resolve the dynamic link caused by the attitude changing of nodes. Authors in [27]–[31] resolve the packet collisions to improve network capacity in LoRaWAN. Standard LoRaWAN adopts Adaptive Data Rate (ADR) [32] to dynamically select the data rate and Transmission Power (TP) according to the link quality for a device. Recent works focus on the dynamic control of LoRa transmission parameters to improve energy efficiency and reliability. DyLoRa [33] establishes an energy efficiency model to adjust transmission parameters such as TP and Spreading Factor (SF). To gauge the quality of the link, DyLoRa utilizes the average SNR from a pre-determined number of recent data packets. In [34], the authors put forward an ADR algorithm centered around SNR. They estimate the SNR threshold for each SF, following which the algorithm assigns the minimum SF as per the given threshold. AdapLoRa [35] dynamically allocates SF and TP parameters based on the ever-changing link conditions. EARN [36] leverages an adaptive SNR margin to cope with dynamic link changes. It also formulates link performance as an Energy Per Packet (EPP) model to strike a balance between the delivery ratio and energy consumption.

However, none of the existing methods take the FHS into consideration. By our measurement, the LR-FHSS packets with similar SNR but different FHS have a significantly different bit error rate. Using packet's SNR only cannot accurately reflect the transmission reliability. Additionally, for LR-FHSS,

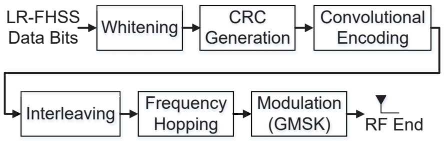
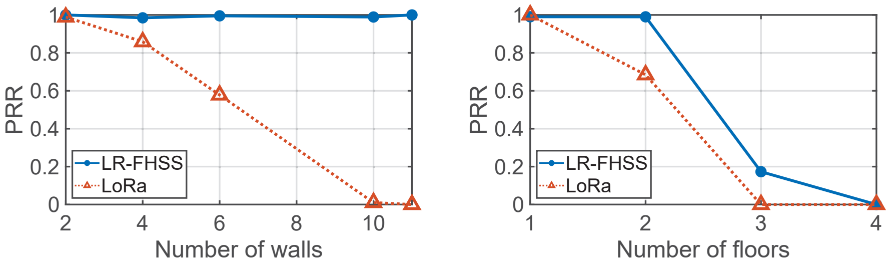


Fig. 2: Workflow of LR-FHSS transceivers.



(a) Different numbers of walls.

(b) Different numbers of floors.

Fig. 3: Indoor LR-FHSS and LoRa performance comparison.

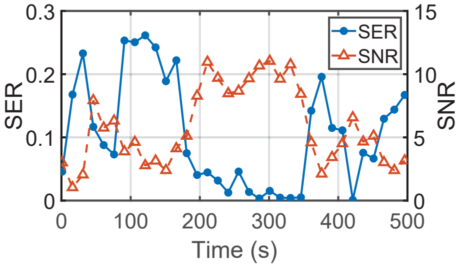
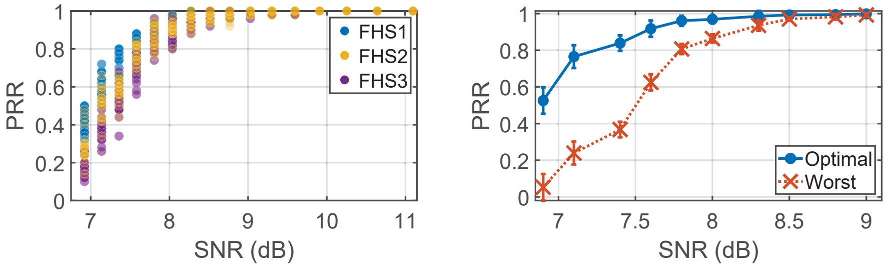


Fig. 4: SER under dynamic interference.



(a) Different FHS.

(b) Optimal and worst FHS.

Fig. 5: PRR under different FHS.

blindly adjusting the transmission parameters such as TP is inefficient due to the ignorance of FHS.

III. Background and Motivation

In this section, we first introduce the background knowledge of LR-FHSS and then show the influence of FHS on LR-FHSS to motivate our work.

A. Background of LR-FHSS

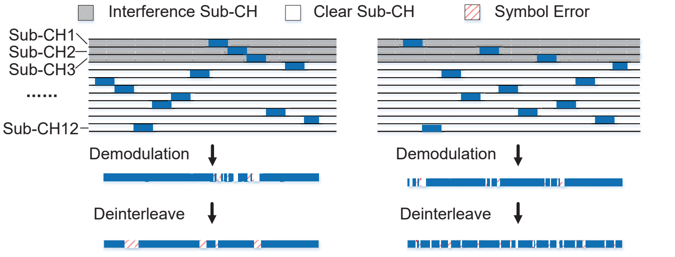
Fig. 2 shows how an LR-FHSS transmitter works. Given the 0/1 bit stream, LR-FHSS first performs data whitening to eliminate data correlation. Then, it adds additional bits to achieve Cyclic Redundancy Check (CRC). Next, convolutional code is implemented to provide FEC, with available Coding Rate (CR) of 1/3 or 2/3. After encoding, the data stream will be interleaved, which scatters adjacent data blocks to cope with burst symbol errors. Then, LR-FHSS divides the whole symbol stream into multiple 50-bit segments, each segment is transmitted in one hop. After that, LR-FHSS decides the sub-channel to be used by each hop. The occupied bandwidth of each sub-channel is denoted as OBW which equals $488Hz$. The number of sub-channels depends on OCW which can be configured to $137kHz$, $336kHz$, and $1.5MHz$. The order of sub-channels used by hops is represented by FHS. Standard LR-FHSS randomly selects the FHS for each transmission. Then, LR-FHSS performs GMSK modulation and sends the signals by the front end.

We investigate the performance of LR-FHSS and LoRa in the indoor environment to show LR-FHSS has a better ability for penetrable communication. For LR-FHSS, the OCW and CR are configured to $137kHz$ and 1/3. The bandwidth and SF of LoRa are $125kHz$ and 12 which provides the longest communication distance in LoRaWAN. The CR of LoRa is configured to 4/5. The transmission power and central frequency of the two techniques are $14dBm$ and $868MHz$. For fairness, we implement the SDR-based gateways of LR-FHSS and LoRa to decode packets.

Fig. 3 presents the results. As shown in Fig. 3(a), the PRR of LoRa suffers from serious degradation when penetrating 4 walls. But LR-FHSS can achieve a high PRR of 0.99 even through 10 walls. We also demonstrate LoRa and LR-FHSS when passing different numbers of floors. In Fig. 3(b), when being blocked by 2 floors, LoRa only achieves PRR of 0.68, which is 30.6% lower than LR-FHSS. The PRR of LoRa is close to 0 when passing 3 floors while LR-FHSS can still achieve PRR of 0.17. The results show that LR-FHSS has better communication reliability in indoor scenarios.

B. Motivation

Our measurement of the LR-FHSS network deployed in a teaching building on campus finds that the dynamic interference in the channel seriously degrades the reliability of transmission. The LR-FHSS node works in the EU $868MHz$ frequency band with CR=1/3 and OCW=$137kHz$. A node sends packets of length 20-byte every 15 seconds. We implement an SDR-based gateway to decode packets. The node and the gateway are separated by 6 walls with a distance of $60m$. We measure the SER and the SNR and plot the results in Fig. 4. We can find that in $[100s, 200s]$ the average SER is 0.21, which is $7.14\times$ higher than the average SER



(a) Worst FHS results in burst errors. (b) Optimal FHS results in dispersed errors.

Fig. 6: Symbol error distribution when using different FHS.

in $[200s, 300s]$. This is because there are people moving around during $[100s, 200s]$, blocking the link and resulting in a decrease in overall SNR, while during $[200s, 300s]$, there are no people moving, resulting in a higher SNR for the packet. The result shows though LR-FHSS has better reliability, it is still influenced by link dynamics.

But existing studies are inefficient in LR-FHSS because they only focus on selecting channels with good quality but do not adjust the FHS which also affects the reliability. We control channel interference to ensure that the packet's SNR varies between $6.9dB$ and $11dB$ while other configurations are the same as the experiment in Fig. 4. Each packet has 12 hops transmitted on 12 sub-channels, as shown in Fig. 6. We add interference in the first three sub-channels. Then FHS can be expressed as $FHS = [hop_1, ..., hop_{12}]$, where hop_i indicates that the i-th hop is transmitted on the hop_i sub-channel. We randomly select three FHS and direct the node to transmit using the same group of sub-channels. FHS1=[4 10 **1** 7 11 **2** 6 **3** 9 12 5 8], FHS2 = [**1** 9 5 6 11 7 **2** **3** 8 10 12 4], and FHS3 = [10 5 **2** **3** 6 12 **1** 11 9 7 4 8], where the bold hop id is the sub-channels under interference. In Fig. 5(a), we demonstrate the PRR corresponding to each FHS under different packet SNR. The result shows that the FHS will significantly affect the reliability of LR-FHSS.

Furthermore, we traverse all possible FHS and identify the optimal and worst FHS that can bring the highest and lowest PRR are shown in Fig. 5(b). The optimal FHS is [**1** 12 5 8 **2** 7 10 **3** 9 6 11 4] and the worst FHS is [6 7 12 9 8 5 **1** **2** **3** 10 4 11]. The average PRR of optimal FHS is 0.89, which is 30.9% higher than the PRR of worst FHS. The reason is that using different FHS for the same sub-channels generates different distributions of symbol errors, affecting the error correction performance of Viterbi decoding in LR-FHSS. Fig. 6 shows an example, after receiving the signal, the receiver first performs GMSK demodulation and obtains the symbols. The SER is similar because packets are transmitted by the same sub-channels. However, after deinterleaving, symbol errors of the two symbol streams that use different FHS have different distributions. The symbol errors in Fig. 6(a) are more concentrated than symbol errors in Fig. 6(b). The burst errors will be beyond the ability of error correction in convolutional codes and cause more packet loss. The above analysis motivates us to enhance the reliability of LR-FHSS by focusing on the selection of FHS.

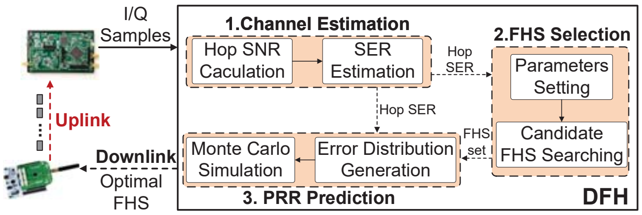


Fig. 7: Overview of DFH.

IV. Design

In this section, we first present an overview of DFH and then introduce the design details of each component.

A. Overview

Fig. 7 shows the framework of DFH which consists of three main components. When one LR-FHSS packet is received, DFH first calculates the SNR of each hop from the I/Q samples and estimates the corresponding SER of each hop. Then DFH decides the parameters of generating FHS based on the hop SER, including the number of bad sub-channels, interleaving width, and step size. After obtaining the set of candidate FHS by the interleaving-based algorithm, DFH predicts the PRR based on the estimated SER and all the candidate FHS. For each candidate FHS, the PRR prediction component first generates the error distribution according to the hop SER, then employs the Monte Carlo based method to predicate PRR for each FHS. Finally, DFH regards the FHS which has the highest predicated PRR as the optimal FHS, and sends it to the end device via downlink. Then the following transmission can use the new FHS to improve reliability.

B. Channel Estimation

To indicate the quality for each sub-channel, we use the fine-grained hop SNR as the indicator. Using hop SNR brings two benefits: firstly, we can obtain the channel condition in sub-channels by only receiving one packet, which can quickly adapt to highly dynamic interference. Secondly, compared with the packet's SNR, using hop SNR can help DFH achieve fine-grained estimation for each sub-channel. When the gateway receives the LR-FHSS packet, DFH decodes the packets and then calculates the SNR of each hop, which we denote as SNR_n for the n-th hop. The total number of hops in a packet is represented by N.

After obtaining the indicator, we need to estimate the SER of hop in each sub-channel. The SER is used to select FHS and predict the PRR in the following module.

LR-FHSS employs 2-GMSK modulation with a Bandwidth-Time product of 1. According to existing studies, the theoretical model [37] depicting the relationship between SNR_n and SER_n can be represented by

$$SER_n = \frac{1}{2}e^{-\alpha \cdot 10^{SNR_n/10}} \tag{1}$$

where α depends on the Bandwidth-Time product of GMSK, for LR-FHSS, α is set to 0.8.

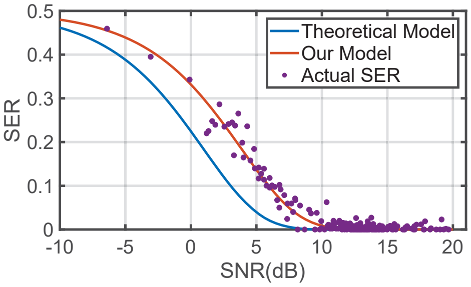


Fig. 8: SER offset between model predictions and measurement results.

To demonstrate the efficiency of the theoretical model, we gather data via our implemented gateway. We collect LR-FHSS packets in an indoor environment. We change the distance between the gateway and the node from 10m to 100m. The node sends 20-byte packets at a center frequency of $868MHz$, with OCW, CR, and TP of $137kHz$, 1/3, and $14dBm$, respectively. We calculated the actual SER of the packet and compared it with the theoretical model. As shown in Fig. 8, the distance exists between the theoretical model in the blue line and the actual SER in purple points. So, we introduce an offset δ and our model in red line can be represented by:

$$SER_n^{'} = \frac{1}{2}e^{-\alpha \cdot 10^{(SNR_n+\delta)/10}} \tag{2}$$

The value of δ depends on the environment. To obtain δ, we first receive packets in different SNR and calculate the corresponding SER. Then we obtain the optimal δ which can minimize the estimation error between estimated SER and actual SER by following the equation,

$$\arg\min_{\delta} |SER_r - SER_n^{'}| \tag{3}$$

where SER_r is the actual SER calculated by received packets in the real environment. δ in our environment is $2.9dB$.

C. FHS Selection

To select an optimal FHS under dynamic channel conditions, we first model the selection of optimal FHS as bipartite graph matching. Selecting FHS can be seen as establishing a mapping relationship between frequency hops and sub-channels. One FHS represents one matching within the complete bipartite graph. The essence of this approach is to interpret frequency hopping as a mapping or matching process between two distinct but interconnected entities (frequency hops and sub-channels). Next, we define what is the ”optimal” FHS. The optimal FHS is a bipartite matching that successfully maximizes a specific defined objective function. The objective function represents the expected result or objective in a mathematical model, which is the predicated PRR in this case.

Specifically, we consider a complete bipartite graph $G = (H, C, E)$, where $H = \{h_1, \cdots, h_N\}$ is the set of available hops, $C = \{c_1, \cdots, c_N\}$ is the set of available sub-channels and E is the edge of graph. If $\pi = \{\pi(1), \cdots, \pi(N)\}$ is a permutation of $\{1, \cdots, N\}$, then a matching of G is

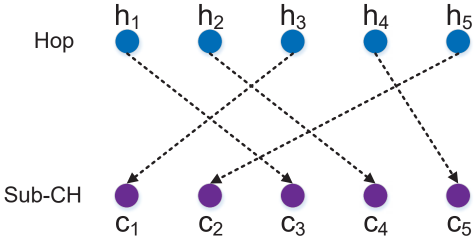


Fig. 9: A matching of the complete bipartite graph.

$$\begin{aligned} m = \{(h_1, c_{\pi(1)}), \cdots, (h_N, c_{\pi(N)})\} \\ (h_i, c_{\pi(i)}) \in E, 1 \le i \le N \end{aligned} \tag{4}$$

where $c_{\pi(i)}$ represents the sub-channel matched by i-th hop. Fig. 9 shows a matching of the complete bipartite graph G.

According to the previous description, the selection of the optimal FHS can be modeled as a complete bipartite graph maximum matching problem, with the PRR as the objective function. However, it’s important to note that the weight of this objective function is not linearly accumulated but rather varies with the matching, resulting in an NP-hard problem [38]–[40]. The traditional solution to the NP-hard problem is Simulated Annealing Algorithms (SAA) [41]. However, existing studies [42], [43] show that the SAA has a slow convergence speed, so it is not suitable for selecting optimal FHS online.

In Section III-B, different FHS result in varying sequence error patterns input into the Viterbi decoder after demodulation and interleaving, leading to different PRR. By analyzing the optimal and worst FHS, we find that the worst FHS causes the hops in poor sub-channels to approach each other after deinterleaving, and then more concentrated symbol errors make the FEC fail. Hence, the optimal FHS means that the symbol errors can be sparsely distributed after deinterleaving. By analyzing the detailed process of interleaving and deinterleaving, we find that the distribution of symbol errors mainly depends on the distance of hops in poor sub-channels. The distance between the a-th hop and b-th hop can be calculated by $min(|b - a|, |a + N - b|)$.

More specifically, if we put neighboring hops in poor channels, then the symbol errors will be concentrated. Hence, to maintain symbol errors sparse, we should increase the distance between hops in poor sub-channels as much as possible.

To select the optimal FHS, we first identify the bad sub-channels that cause high symbol errors. We determine the threshold th to decide bad sub-channels dynamically.

$$th = average(SER) \tag{5}$$

Then, based on th, we regard the sub-channels which $SER > th$ as the bad sub-channels, then we can obtain the number of bad sub-channels N_{bad}.

Our goal is to make the distance between the frequency hops allocated to these bad sub-channels sufficiently large. So we propose the interleaving-based methods where the number

```
Algorithm 1: Interleaving-based search algorithm
Input:
$H$: the set of frequency hops.
$SER$: hop SER.
$width$: width of interleaving.
$step$: step of interleaving.
Output:
$\mathbb{S}_{FHS}$ : candidate set of FHS.
for $st$ in $1 : length(H)$ do
    $row = \lceil st/width \rceil$
    $col = st \bmod width$
    $index = st$
    $S = []$
    for $i$ in $1 : length(H)$ do
        $S_i = H_{index}$
        $row++$
        $index = (row - 1) \cdot width + col$
        if $index > length(H)$ then
            $col = col + step$
            $row = 1$
            if $col > width$ then
                $col = (col \bmod width) + 1$
            end
            $index = (row - 1) \cdot width + col$
        end
    end
    $FHS = Convert(SER, S)$
    $\mathbb{S}_{FHS}$.add($FHS$)
end
```

of columns (or the width) $width$ during interleaving equals N_{bad}, and the interleaving step size $step$ is set to $\lfloor width/2 \rfloor$. Through the interleaving algorithm, we can obtain a sequence $S = [S_1, ..., S_N]$, where S_i represents the S_i-th hop being transmitted on the sub-channel corresponding to the i-th SER.

Algorithm 1 shows the workflow of the interleaving-based algorithm to select a candidate set of FHS. The algorithm takes as inputs the frequency hops H, the interleaving width $width$, and the interleaving step $step$. The interleaving process begins, with the starting point of interleaving ranging from 1 to $length(H)$. Points in a sequence can be mapped to row and column indices and this relationship can be reversed as well. We begin by initializing the row row, column numbers col, and the index of sequence $index$, followed by outputting the content indicated by $index$ into the S. Subsequently, we shift the row one level down and infer $index$ to continue outputting. If the position indicated by $index$ exceeds $length(H)$, col increases by $step$. If col surpasses the matrix $width$, the interleaving process resumes from the leftmost column that has not been outputted. Finally, we convert S into FHS through hop SER SER. Through this procedure, we are able to obtain a candidate set $\mathbb{S}_{FHS}$ of FHS.

```
Algorithm 2: Monte Carlo based simulation
Input:
$SER$: hop SER.
$FHS$: frequency hopping sequence.
$Y$: reference sequence.
Output:
$PRR$ : predicated PRR.
$R_{pkt} = 0$
for $i$ in $1 : K$ do
    $Y_{tmp} = Y$
    $err\_index$ = Random_error (SER, FHS)
    $Y_{tmp}$ = Sym_flip ($Y_{tmp}$, $index$)
    $D$ = Deinterleave ($Y_{tmp}$)
    if Viterbi (D) is correct then
        $R_{pkt}$++
    end
end
$PRR$ = $R_{pkt}/K$
```

D. Prediction Model

After channel estimation and FHS selection, we obtain the hop SER and a candidate FHS set. Then, for each FHS in $\mathbb{S}_{FHS}$, we predicate its corresponding PRR and regard the FHS that has the highest PRR as the optimal FHS in the next transmission. By the SNR-to-SER model proposed in Section IV-B, we can obtain the error probability of each symbol after demodulation and deinterleaving. However, the final PRR after FEC cannot be directly calculated because the process of Viterbi decoding used in LR-FHSS relies on maximum likelihood which is non-linear. Several works [44]–[47] have proposed methods to calculate the burst error probability after Viterbi decoding on a binary symmetric channel or an AWGN channel. However, they can only obtain the lower bound of PRR instead of the accurate PRR.

To accurately predict the PRR of each FHS in $\mathbb{S}_{FHS}$, we propose the Monte Carlo based simulation [48]. Given the SNR and FHS, we generate multiple possible symbol sequences that contain different symbol error patterns after demodulation. Then, we deinterleave them and input them into the Viterbi decoder. Finally, we calculate the PRR. The whole process is shown in Algorithm. 2, the inputs contain the SNR, FHS, and reference sequence Y. Y is used to generate symbol sequences with different symbol error patterns. We generate Y by inputting a random payload into the following processes including whitening, CRC generation, convolutional encoding, and interleaving.

For each FHS, we generate K symbol sequences to calculate PRR. In function $Random\_error()$, we first calculate the number of error symbols in each hop according to SER, and then we randomly select the positions of each error symbol within the hop. Next, we flip the 0/1 symbol in the selected position to generate symbol sequences by the function $Sym\_flip()$ and input it to deinterleaving and Viterbi decoder. Finally, we calculate the PRR by the ratio of successfully

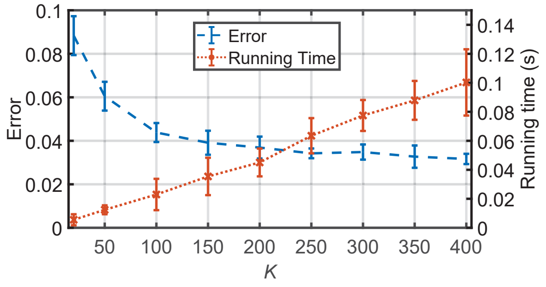
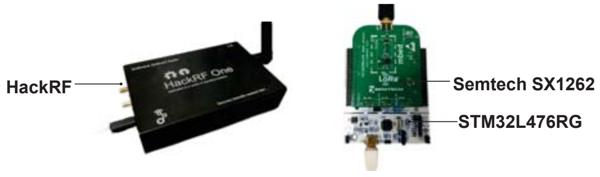


Fig. 10: Trade-off between accuracy and cost.



(a) Gateway.

(b) LR-FHSS node.

Fig. 11: Experiment equipments.

decoding sequence and whole K sequences.

Then we should select the suitable value of K, too large K brings higher time consumption and affects online selection. But if K is too small, the accuracy of PRR prediction will be low. Hence we need to achieve a good balance between accuracy and overhead. We first collect some LR-FHSS packets under different SNR in the real environment and decode them to obtain the actual PRR as the baseline. The corresponding setting is the same as the experiments in Section IV-B. Then, we use the Monte Carlo based simulation with different K. We record the difference between the predicted PRR and the real PRR as the error and the method's running time. The results are shown in Fig 10. It shows that the running time increases linearly as the increasing number of K. When $K > 200$, the error is stable, so K is set to 200 in DFH.

V. Evaluation

A. Experiment Setting

To demonstrate the performance of DFH, we implement DFH in an SDR-based gateway and conduct experiments in a real indoor network consisting of commercial nodes. As shown in Fig. 11, the gateway uses Hack-RF One as the front end. We implement the decoding of LR-FHSS and a prototype of DFH by Matlab. The node consists of SX1262 transceiver and STM32L476RG microprocessor. The node works in LoRaWAN Class B. After transmitting an LR-FHSS packet, the node switches to the receiving mode and listens to the LoRa packet sent by the gateway. The LoRa packet contains the optimal FHS calculated by the algorithm in the gateway. Then the node will use the received FHS for the next LR-FHSS transmission. By default, the OCW, CR, number of available sub-channels, and payload length of uplink LR-FHSS transmission are $137kHz$, 1/3, 35 and 20-byte, respectively. Each packet contains 12 hops. The BW, SF, and CR of downlink LoRa transmission are $125kHz$, 9 and 4/8. To avoid interference between uplink and downlink, the central frequency of LR-FHSS and LoRa are configured to $868MHz$ and $915MHz$.

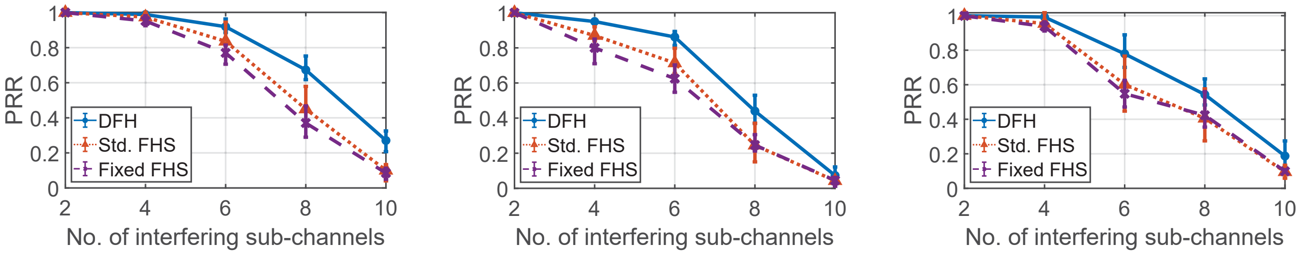
For comparison, we also implement four representative methods as baselines. The first one is the standard LR-FHSS (denoted as Std. FHS) which randomly selects sub-channels and DFH in each transmission. We also implement LR-FHSS which uses randomly selected fixed sub-channels and FHS (denoted as Fixed FHS) for comparison. Third, we implement ETSCH [18] as the representative of blacklist-based methods. ETSCH uses the Non-intrusive Channel-quality Estimation (NICE) technique to decide the blacklist which blocks sub-channels with poor quality. To cope with the dynamic interference in unfixed channels, ETSCH will adaptively update the blacklist before selecting channels used by nodes. We also implement a method that adaptively adjusts the transmission power to improve reliability (denoted as TP-method). It relies on the SNR of the whole packets to select suitable transmission power. After obtaining the average SNR of the whole packet, TP-method calculates the expected power which can make PRR higher than 0.9 according to the model of the packet's SNR to PRR which is measured from our real-world network.

In the following, we first demonstrate the performance of DFH under various types of interference with different intensities. Then we evaluate the performance of the main components in DFH. To accurately control the intensity of the interference, we perform trace-driven simulation. We first collect the trace which contains real signals of LR-FHSS packets and interference traces to simulate signals under different interference conditions. Then we synchronize the signal trace with interference trace and combine them to generate signals under interference with different intensities. Besides the trace-driven simulation, we also conduct real-time experiments to evaluate DFH in real deployed indoor network that contains multiple nodes. Finally, we compare the performance of DFH with ETSCH in large-scale network. We also conduct trace-driven simulations to emulate the concurrent transmissions by a large number of nodes.

B. Performance of DFH under Different Interference

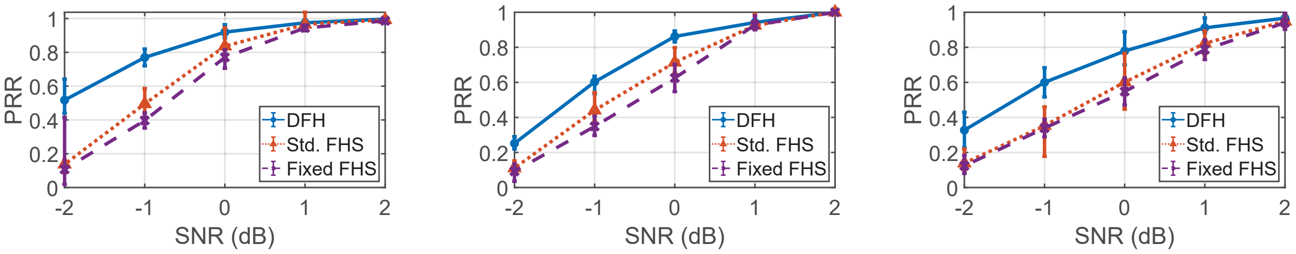
The purpose of DFH is to improve the reliability of LR-FHSS, so we demonstrate the performance of DFH under three types of interference with different intensities.

1) Different Numbers of Interfering Sub-channels: We evaluate the performance of DFH under different numbers and types of interfering sub-channels. The SNR of each sub-channel is set to $[-1, 1]dB$. We change the number of interfering sub-channels from 2 to 10. The interference contains white noise, homogeneous interference caused by other nodes that also send LR-FHSS, and hybrid interference which combines the above two interference with a mixture ratio of 1:1.



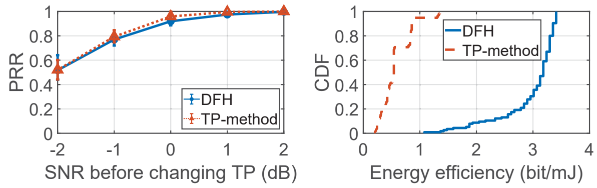
(a) White noise interference. (b) Homogeneous interference. (c) Hybrid interference.

Fig. 12: Performance of DFH under different numbers of interfering sub-channels.



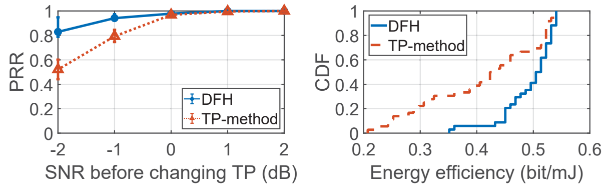
(a) White noise interference. (b) Homogeneous interference. (c) Hybrid interference.

Fig. 13: Performance of DFH under different SNR.



(a) PRR under white noise interference. (b) CDF of energy efficiency.

Fig. 14: Performance under different transmission power.



(a) PRR under white noise interference. (b) CDF of energy efficiency.

Fig. 15: Performance under same transmission power.

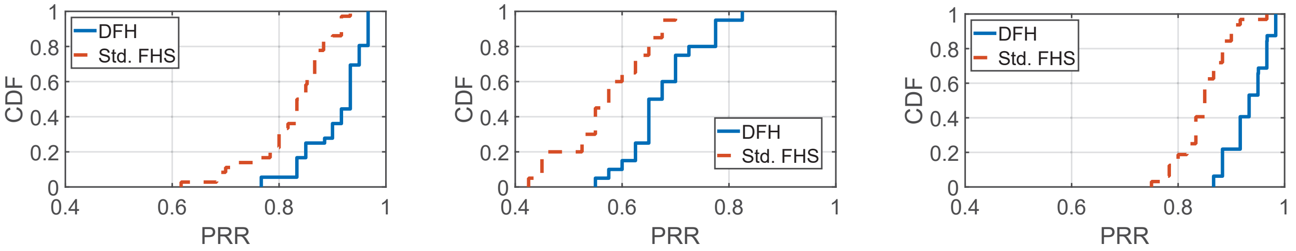
In Fig. 12, we can find that the PRR of DFH is higher than the other two methods under three different interference. Under white noise, when the number of interfering sub-channels changes from 2 to 10, the PRR of DFH decreases from 1.00 to 0.27 because of the increase of SER caused by noise. When the number of interfering nodes is 10, the PRR of DFH is 0.27, which is 1.69× and 2.18× higher than Std. FHS and Fixed FHS. Under homogeneous interference, compared with Std. FHS and Fixed FHS, DFH can improve the PRR by up to 79.3% and 78.5% when the number of interfering sub-channels is 8. Then, under hybrid noise, the average PRR of DFH is 0.70, which is 14.5% and 16.5% higher than Std. FHS and Fixed FHS.

2) Different SNR: Then, we demonstrate the performance of DFH under different SNR. We randomly select 6 sub-channels and add interference to them.

The results are shown in Fig. 13. Under white noise, with the change of SNR from $-2dB$ to $2dB$, the PRR of all three methods increases. When the SNR is $-2dB$, DFH improves the PRR by $2.76\times$ and $3.81\times$ compared to Std. FHS and Fixed FHS, respectively. Under homogeneous interference, the PRR of DFH is also higher than the two methods. DFH can achieve the average PRR of 0.57 when SNR is lower than 0, which is 35.6% and 61.0% higher than Std. FHS and Fixed FHS.

C. Energy Efficiency of DFH

We also conduct experiments to show the energy efficiency of DFH. The number of interfering sub-channels is set to 6. We demonstrate the PRR under white noise interference with different SNR levels and calculate the energy efficiency. As shown in Fig. 14, DFH has a similar performance of PRR with TP-method, the average PRR of DFH and TP-method method is 0.84 and 0.85, respectively. However, TP-method brings more energy consumption due to the increasing power of transmission. The medium energy efficiency of DFH and TP-method is $3.18bit/mJ$ and $0.54bit/mJ$. DFH can achieve average efficiency of $2.98bit/mJ$, which is $5.20\times$ larger than

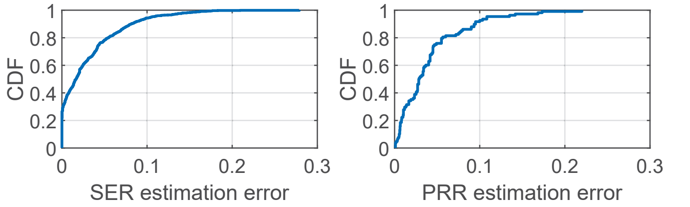


(a) CR=1/3 OCW=137kHz.

(b) CR=2/3 OCW=137kHz.

(c) CR=1/3 OCW=336kHz.

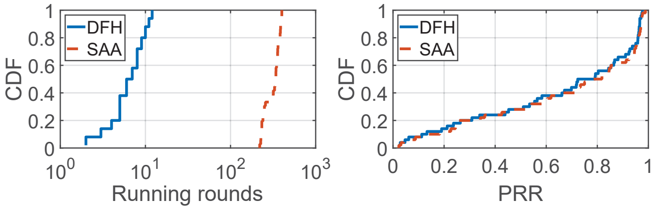
Fig. 16: Performance of DFH under different CR and OCW.



(a) Channel estimation.

(b) PRR inference.

Fig. 17: Accuracy of channel estimation module and PRR inference module.



(a) Time consumption.

(b) Accuracy.

Fig. 18: Performance of FHS selection module.

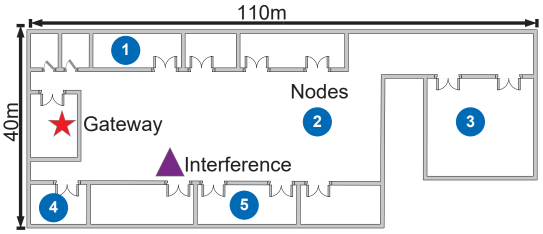


Fig. 19: Deployment of nodes and the gateway.

TP-method. It shows that our method can significantly increase the battery life of nodes.

Furthermore, we also compare DFH and TP-methods under the highest transmission powers. In this case, TP-method cannot further increase transmission power. As shown in Fig. 15(a), when the SNR is $-2dB$, the PRR of DFH is 0.83, which is 59.2% higher than TP-method. Meanwhile, in Fig. 15(b), the medium energy efficiency of DFH and TP-method are $0.51bit/mJ$ and $0.44bit/mJ$ because DFH can achieve high PRR under same energy consumption. The experimental results show that DFH can significantly improve reliability and reduce energy consumption.

D. Performance under Transmission Parameters

We compare the performance of DFH and Std. FHS under different CR and OCW.

1) Different CR: We evaluate the impact of CR on performance with the OCW of $137kHz$ under white noise interference. We add white interference in the half number of occupied sub-channels. Fig. 16(a) and (b) illustrate the CDF of the PRR for the two methods under $CR = 1/3$ and $CR = 2/3$. For DFH, the PRR under $CR = 1/3$ and $CR = 2/3$ is 0.91 and 0.68 because lower CR brings more redundant bits for error correction. For Std. FHS, the medium PRR under $CR = 1/3$ and $CR = 2/3$ are 0.85 and 0.58. Both in $CR = 1/3$ and $CR = 2/3$, the PRR of DFH is higher than Std. FHS.

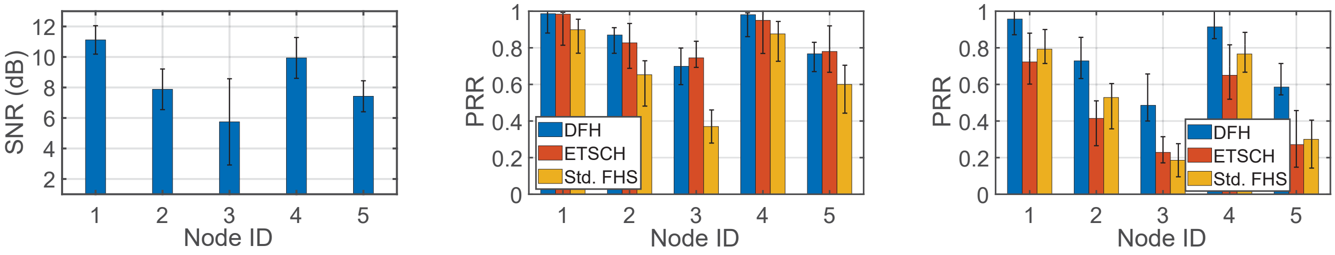
2) Different OCW: Fig. 16(a) and (c) present the CDF of the PRR for the two methods under different OCW. When the OCW is $336kHz$, the average PRR of DFH and Std. FHS is 0.93 and 0.85, which is 2.7% and 2.6% higher than it under $137kHz$. This is because a wider bandwidth provides more available channels, reducing the likelihood of interference from adjacent sub-channels.

E. Performance of DFH 's Components

Then, we conduct experiments to evaluate the efficiency of the main components of DFH. We evaluate DFH under three types of interference, the detail setting is same with experiments in SectionV-B.

1) Channel Estimation Module: To demonstrate the accuracy of hop SER estimation, we calculate the CDF of estimation error which is defined by the $|SER_e - SER_r|$ where SER_e represents the estimated hop SER by our module and SER_r represents the real measured hop SER. As shown in Fig. 17 (a), the errors mainly locate in $[0, 0.15]$, the medium error is 0.02 and the average error is 0.03, which shows that our method can achieve relatively high accuracy.

2) PRR Inference Module: Then, we evaluate the accuracy of PRR inference. We calculate the CDF of estimation error which is defined by the $|PRR_e - PRR_r|$ where PRR_e represents the predicted PRR by our module and PRR_r represents the real measured PRR. As shown in Fig. 17 (b),



(a) SNR of each location.

(b) Duty cycle=10%.

(c) Duty cycle=70%.

Fig. 20: Performance in the real deployed network using different duty cycle.

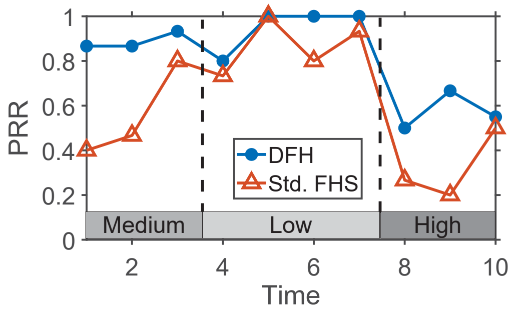


Fig. 21: PRR under dynamic interference.

the errors are primarily concentrated in the range of $[0, 0.2]$, the medium error is 0.03 and the average error is 0.04, which can help us select the optimal FHS.

3) FHS Selection Module: In order to evaluate the effectiveness of the FHS selection algorithm, we compared our method with SAA. According to existing methods [49], [50], the initial temperature T_b, the final temperature T_e, the cooling rate β, and the number of iterations for each temperature t of the SAA are set to 100, 20, 0.8, and 50. The running rounds required for the two methods to achieve convergence are shown in Fig 18 (a), DFH can achieve average running rounds of 7.72, which is 97.6% lower than SAA. The medium running rounds of DFH and SAA are 6 and 341 respectively. It shows that our interleaving-based method can significantly reduce time consumption and cope with highly dynamic channels.

Fig. 18(b) shows the CDF of PRR corresponding to the selected optimal FHS for DFH and SAA. The medium PRR of DFH and SAA is 0.72 and 0.75. The average PRR of DFH is 0.65, which is 2.5% lower than the SAA. The result shows that our interleaving-based method basically achieves the same effect as the SAA.

F. Performance in the Real Deployed Network

We further evaluate DFH in a real deployed indoor network. We deploy five nodes in five different locations as shown in Fig. 19. According to the specification of LoRaWAN, five nodes adopt naive ALOHA that randomly accesses the channel. We also deploy one interferer which uses SDR to generate dynamic white noise interference. The interferer randomly interferes 10 sub-channels among 35 sub-channels and sets the transmission power in each sub-channel varying in $[-3, 3]dBm$. The period of the interferer switching to different sub-channels and transmission power is 30 seconds. The SNR of received packets in five locations after adding interference is shown in Fig. 20(a).

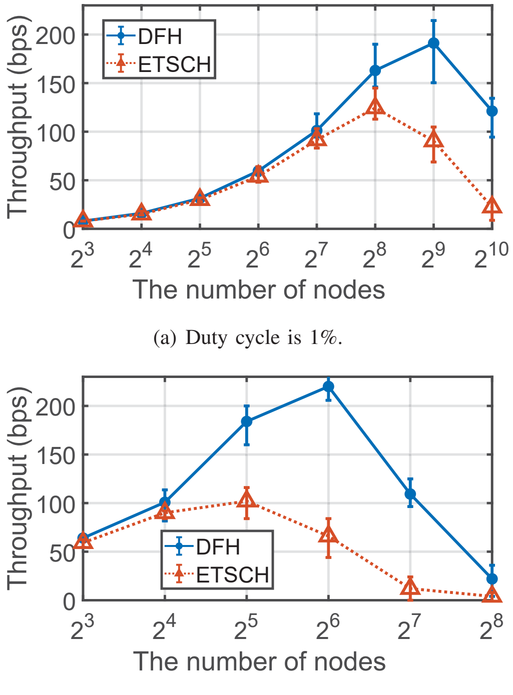
We first show the performance of three methods using the maximum duty cycle in LoRaWAN, 10%. As illustrated in Fig. 20(b), for all methods, the node in location 3 has the lowest PRR among five nodes because it suffers from the serious hindrance of 3 walls and has the lowest SNR. The average PRR of DFH and ETSCH is 0.86 and 0.85 which is 26.7% and 25.1% higher than Std. FHS. The above result shows that DFH and ETSCH have similar performance in low concurrency scenarios.

Furthermore, we demonstrate the performance of three methods under high concurrency scenarios. We increase the duty cycle of each node to 70%, which can significantly increase the probability of packet collisions. As shown in Fig. 20(c), the average PRR of DFH is 0.73, which is 60.5% and 42.7% higher than ETSCH and Std. FHS. The performance of ETSCH suffers from serious degradation because it adopts blacklist-based method which only considers the channel quality and pushes multiple nodes to select the similar good sub-channels for transmission, aggravating the packet collisions. Then serious transmission collisions of the used hops occur and cause more decoding errors.

Fig. 21 shows PRR changes for node 4 under dynamic interference. Under medium interference where we set the transmission power of the interferer to $0dBm$, the average PRR of DFH is 0.89, which is 59.9% higher than Std. FHS. When the interference decreases to the low level where the transmission power of the interferer equals $-3dBm$. The average PRR of DFH and Std. FHS is improved to 0.95 and 0.86 due to the high SNR within sub-channels. When the interference increases to a high level where the transmission power of the interferer equals $3dBm$, the average PRR of Std. FHS is reduced to 0.32. Thanks to the rearrangement of FHS in DFH to adapt to dynamic interference, DFH can still achieve the average PRR of 0.57, which is 78.1% higher than Std. FHS. The result shows that DFH can improve the reliability of LR-FHSS under dynamic interference.

G. Performance in Large-scale Networks

Furthermore, we compare the performance of DFH and existing methods in large-scale networks where extensive devices connect with one gateway, which is a typical application



(a) Duty cycle is 1%.

(b) Duty cycle is 10%.

Fig. 22: Performance under Large-scale Network.

scenario of LPWAN. We conduct trace-driven simulations to investigate the performance of different methods in large-scale networks. We first collect traces of real signals by SDR. We collected 30 traces in 30 different locations. The SNR of received packets in different locations varies in $[1, 15]dB$. Each trace contains 50 LR-FHSS packets with random 20-byte payloads. Each packet first repeats transmitting the headers three times in three randomly selected sub-channels, and then randomly selects 12 sub-channels within 35 sub-channels as the 12 payload hops for payload transmission.

In the evaluation, when the number of nodes is smaller than 30, we only use the real collected trace to obtain the experimental results; when the number of nodes exceeds 30, we perform the trace-driven simulation to investigate the performance. When more nodes are added to the network, we generate more traces based on our collected traces. For packets transmitted from each location, We divide the original signals of the header and payload hops into two repositories respectively. The newly added node randomly selects one location for transmission. Then we randomly select header and payload hops from corresponding repositories to generate the packet and randomly configure the initial FHS of the packet. According to the duty cycle, we can calculate the number of packets expected to be transmitted within a given period. We also randomly set the arrival time of each packet to simulate naive ALOHA. Then we execute different methods to decide the next FHS for each node to obtain the transmission performance of different methods.

Fig. 22(a) illustrates the throughput of DFH and ETSCH under duty cycle of 1%. When the number of nodes is smaller than 2^7, DFH and ETSCH can achieve similar PRR. However, when more nodes concurrently transmit, the throughput of ETSCH suffers from serious degradation. When 2^8 nodes transmit in the network, DFH can achieve the throughput of $163.0bps$, which is 31.0% higher than the throughput of ETSCH. The reason is that ETSCH directly blocks the sub-channels with poor quality. Then the number of available sub-channels significantly decreases and the probability of packet collisions increases. Serious collisions cause more packet loss. Different from ETSCH, DFH only adjusts the order of using sub-channels. The sub-channels with poor link quality are also utilized. The number of available sub-channels is not reduced. Even when the number of nodes increases to 2^{10}, DFH can still achieve a throughput of $121.2bps$, which is $4.41\times$ higher than the throughput of ETSCH.

Furthermore, when the duty cycle of each node increases to 10%, DFH can achieve even better performance than ETSCH, as shown in Fig. 22(b). The higher duty cycle causes more collisions under the same number of nodes. When the duty cycle increases from 1% to 10%, the number of supported nodes when achieving maximum throughput of DFH and ETSCH decreases from 2^9 and 2^8 to 2^6 and 2^5. The throughput improvement of DFH increases from $1.54\times$ to $2.26\times$ when the duty cycle increases from 1% to 10%. The reason is that ETSCH only considers the channel conditions and blocks poor channels, which makes multiple nodes use similar sub-channels. The packet collisions are aggravated under the higher duty cycle. DFH combines both channel quality and FHS to select sub-channels, which reduces the impact of collisions.

VI. Conclusion

In this paper, we propose DFH, a dynamic frequency hopping method of LR-FHSS to improve transmission reliability in dynamic environments. Our in-depth study reveals that the reliability of FHS is not only affected by channel quality but also depends on the FHS. To select the optimal FHS under dynamic link quality, we first define the hop SNR as the new metric to indicate fine-grained channel quality. Then, we established a model based on both hop SNR and FHS to estimate the transmission performance of LR-FHSS. To online select optimal FHS, we design an interleaving-based search algorithm that reduces the time consumption in searching optimal FHS. The experimental results show that DFH can improve the PRR by up to 2.76×, compared to the standard LR-FHSS.

Acknowledgment

We thank the anonymous reviewers and shepherd of this paper for their extensive technical feedback. This work is supported in part by the Funds for Creative Research Groups of China (No. 61921003), the National Natural Science Foundation of China (No. 62072050 and No. 62225204), the A3 Foresight Program of NSFC (No. 62061146002), and Xiaomi Young Talents Program of Xiaomi Foundation.

References

[1] F. Adib and D. Katabi, “See through walls with wifi!” in *Proceedings of ACM SIGCOMM*, 2013.

[2] C. Gao, Y. Li, and X. Zhang, “Livetag: Sensing human-object interaction through passive chipless wifi tags,” in *Proceedings of ACM NSDI*, 2018.

[3] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, “Saiyan: Design and implementation of a low-power demodulator for lora backscatter systems,” in *Proceedings of ACM NSDI*, 2022.

[4] W. Chen, C. He, J. Lu, K. Yan, J. Liu, F. Zhou, X. Xu, and X. Hao, “Research and design of distributed fire alarm system of indoor internet of things based on lora,” *Scientific Programming*, vol. 2021, pp. 1–12, 2021.

[5] L. Zhao, W. Wu, and S. Li, “Design and implementation of an iot-based indoor air quality detector with multiple communication interfaces,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9621–9632, 2019.

[6] R. Nandakumar, V. Iyer, and S. Gollakota, “3d localization for sub-centimeter sized devices,” in *Proceedings of ACM Sensys*, 2018.

[7] J. Liu, J. Gao, S. Jha, and W. Hu, “Seirios: leveraging multiple channels for lorawan indoor and outdoor localization,” in *Proceedings of ACM MobiCom*, 2021.

[8] L. Chen, J. Xiong, X. Chen, S. I. Lee, K. Chen, D. Han, D. Fang, Z. Tang, and Z. Wang, “Widesee: Towards wide-area contactless wireless sensing,” in *Proceedings of ACM Sensys*, 2019.

[9] N. Poddar, S. Z. Khan, J. Mass, and S. N. Srirama, “Coverage analysis of nb-iot and sigfox: two estonian university campuses as a case study,” in *Proceedings of IEEE IWCMC*, 2020.

[10] M. Lauridsen, H. Nguyen, B. Vejlgaard, I. Z. Kovács, P. Mogensen, and M. Sorensen, “Coverage comparison of gprs, nb-iot, lora, and sigfox in a 7800 km^2 area,” in *Proceedings of IEEE VTC*, 2017.

[11] S.-Y. Wang, Y.-R. Chen, T.-Y. Chen, C.-H. Chang, Y.-H. Cheng, C.-C. Hsu, and Y.-B. Lin, “Performance of lora-based iot applications on campus,” in *Proceedings of IEEE VTC*, 2017.

[12] Semtech, “Application note: Lr-fhss system performance,” Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1262.

[13] B. S. Std, “Specification of the bluetooth system version 4.0,” Available: https://www.bluetooth.com/specifications/specs/core-specification-4-0/.

[14] M. Spörk, J. Classen, C. A. Boano, M. Hollick, and K. Römer, “Improving the reliability of bluetooth low energy connections.” in *Proceedings of ACM EWSN*, 2020.

[15] P. Popovski, H. Yomo, and R. Prasad, “Strategies for adaptive frequency hopping in the unlicensed bands,” *IEEE Wireless Communications*, vol. 13, no. 6, pp. 60–67, 2006.

[16] M. Hänninen, J. Suhonen, T. D. Hämäläinen, and M. Hännikäinen, “Link quality-based channel selection for resource constrained wsns,” in *Proceedings of Springer GPC*, 2011.

[17] P. Du and G. Roussos, “Adaptive time slotted channel hopping for wireless sensor networks,” in *Proceedings of IEEE CEEC*, 2012.

[18] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens, “Dependable interference-aware time-slotted channel hopping for wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 14, no. 1, pp. 1–35, 2018.

[19] A. Elsts, X. Fafoutis, R. Piechocki, and I. Craddock, “Adaptive channel selection in ieee 802.15. 4 tsch networks,” in *Proceedings of IEEE GIoTS*, 2017.

[20] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, “Label: Link-based adaptive blacklisting technique for 6tisch wireless industrial networks,” in *Proceedings of ACM MSWiM*, 2017.

[21] M. Sha, G. Hackmann, and C. Lu, “Arch: Practical channel hopping for reliable home-area sensor networks,” in *Proceedings of IEEE RTAS*, 2011.

[22] S.-H. Lee and Y.-H. Lee, “Adaptive frequency hopping for bluetooth robust to wlan interference,” *IEEE Communications Letters*, vol. 13, no. 9, pp. 628–630, 2009.

[23] M. Farahmandand and M. Nabi, “Channel quality prediction for tsch blacklisting in highly dynamic networks: A self-supervised deep learning approach,” *IEEE Sensors Journal*, vol. 21, no. 18, pp. 21 059–21 068, 2021.

[24] Y. Wang, X. Zheng, L. Liu, and H. Ma, “Polartracker: Attitude-aware channel access for floating low power wide area networks,” in *Proceedings of IEEE INFOCOM*, 2021.

[25] R. Li, X. Zheng, Y. Wang, L. Liu, and H. Ma, “Polarscheduler: Dynamic transmission control for floating lora networks,” in *Proceedings of IEEE INFOCOM*, 2022.

[26] 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.

[27] F. Yu, X. Zheng, L. Liu, and H. Ma, “Loradar: An efficient lora channel occupancy acquirer based on cross-channel scanning,” in *Proceedings of IEEE INFOCOM*, 2022.

[28] X. Xia, N. Hou, Y. Zheng, and T. Gu, “Pcube: scaling lora concurrent transmissions with reception diversities,” in *Proceedings of ACM MobiCom*, 2021.

[29] S. Tong, J. Wang, and Y. Liu, “Combating packet collisions using non-stationary signal scaling in lpwans,” in *Proceedings of ACM MobiSys*, 2020.

[30] Y. Wang, F. Zhang, X. Zheng, L. Liu, and H. Ma, “Decoding lora collisions via parallel alignment,” *ACM Transactions on Sensor Networks*, vol. 19, no. 3, pp. 1–25, 2023.

[31] F. Yu, X. Zheng, L. Liu, and H. Ma, “Enabling concurrency for non-orthogonal lora channels,” in *Proceedings of ACM MobiCom*, 2023.

[32] L. Alliance, “Ts001-1.0.4 lorawan® l2 1.0.4 specification,” Available: https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification.

[33] Y. Li, J. Yang, and J. Wang, “Dylora: Towards energy efficient dynamic lora transmission control,” in *Proceedings of IEEE INFOCOM*, 2020.

[34] V. Hauser and T. Hégr, “Proposal of adaptive data rate algorithm for lorawan-based infrastructure,” in *Proceedings of IEEE FiCloud*, 2017.

[35] W. Gao, Z. Zhao, and G. Min, “Adaplora: Resource adaptation for maximizing network lifetime in lora networks,” in *Proceedings of IEEE ICNP*, 2020.

[36] J. Park, K. Park, H. Bae, and C.-K. Kim, “Earn: Enhanced adr with coding rate adaptation in lorawan,” *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 873–11 883, 2020.

[37] I. Korn, “Gmsk with limiter discriminator detection in satellite mobile channel,” *IEEE Transactions on Communications*, vol. 39, no. 1, pp. 94–101, 1991.

[38] M. Bayati, D. Shah, and M. Sharma, “Max-product for maximum weight matching: Convergence, correctness, and lp duality,” *IEEE Transactions on Information Theory*, vol. 54, no. 3, pp. 1241–1251, 2008.

[39] G. Sharma, N. B. Shroff, and R. R. Mazumdar, “Maximum weighted matching with interference constraints,” in *Proceedings of IEEE PERCOMW*, 2006.

[40] Y. Berstein and S. Onn, “Nonlinear bipartite matching,” *Discrete Optimization*, vol. 5, no. 1, pp. 53–65, 2008.

[41] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[42] K.-L. Du, M. Swamy, K.-L. Du, and M. Swamy, “Simulated annealing,” *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*, pp. 29–36, 2016.

[43] J. Mingjun and T. Huanwen, “Application of chaos in simulated annealing,” *Chaos, Solitons & Fractals*, vol. 21, no. 4, pp. 933–941, 2004.

[44] A. Lapidoth, “On the probability of symbol error in viterbi decoders,” *IEEE Transactions on Communications*, vol. 45, no. 2, pp. 152–155, 1997.

[45] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[46] ——, “Convolutional codes and their performance in communication systems,” *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 751–772, 1971.

[47] I. Sason, S. Shamai *et al.*, “Performance analysis of linear codes under maximum-likelihood decoding: A tutorial,” *Foundations and Trends® in Communications and Information Theory*, vol. 3, no. 1–2, pp. 1–222, 2006.

[48] N. Metropolis and S. Ulam, “The monte carlo method,” *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[49] J. Haddock and J. Mittenthal, “Simulation optimization using simulated annealing,” *Computers & industrial engineering*, vol. 22, no. 4, pp. 387–395, 1992.

[50] B. Abbasi, A. H. E. Jahromi, J. Arkat, and M. Hosseinkouchack, “Estimating the parameters of weibull distribution using simulated annealing algorithm,” *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 85–93, 2006.