mLoRa: A Multi-Packet Reception Protocol in LoRa networks

Xiong Wang[1], Linghe Kong[1], Liang He[2], Guihai Chen[1]
[1]Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China,
[2]University of Colorado Denver, USA,
Corresponding Email:linghe.kong@sjtu.edu.cn

***Abstract*—We present mLoRa in this paper, a novel protocol that can decode multiple collided packets simultaneously from different transmitters in LoRa networks. As a recently proposed wireless technology designed for low-power wide-area networks, LoRa has been proverbially employed in many fields, such as smart cities, intelligent agriculture, and environmental monitoring. In LoRa networks, a star-of-stars topology is conventionally implemented, in which thousands of nodes connect to a single gateway. Accordingly, the convergecast scenario becomes common. For example, in intelligent agriculture, multiple sensor nodes send information with respect to the soil temperature and humidity to a LoRa gateway. Regularly, simultaneous transmissions result in the severe collision problem. Meanwhile, the ALOHA protocol is widely applied in LoRa networks, which further aggravates the collision problem. To conquer this challenge, we propose a protocol named mLoRa for multi-packet reception in LoRa networks, leveraging unique features inherent in LoRa's physical layer including chirp spread spectrum (CSS), M-FSK modulation, and demodulation. In addition, design enhancements are developed to mitigate the noise and frequency offset influence. We implement mLoRa on a six-node testbed with USRPs. Experiment results demonstrate that mLoRa enables up to three concurrent transmissions. Correspondingly, mLoRa based throughput is around 3 times more than the conventional LoRa.**

***Index Terms*—LoRa Networks, Collision Resolution, Chirp, M-FSK Modulation**

I. Introduction

As a critical technique to satisfy the strict requirements (e.g., low power, long range, and ubiquitous connectivity) for Internet of things, LoRa has received wide-spread attention from both academia and industry [1], [2]. Up to now, a wide range of areas have witnessed the practical implementation of LoRa, such as smart cities, intelligent agriculture and logistics and so on. The star-of-stars topology is usually applied in LoRa networks, where thousands of LoRa nodes connect to a single LoRa gateway. The aggregated network structure leads to the severe packet collision, incurring extensive packet loss and throughput degradation.

Even worse, LoRa networks adopt the ALOHA protocol for simplicity and energy conservation. According to the protocol's specifications, LoRa nodes transmit packets if needed without channel detection. If the LoRa gateway fails to decode one packet transmitted from the node due to collisions, the packet will be retransmitted after a random back-off time, thus further exacerbating the collision problem in LoRa networks.

In the meantime, existing literature has been proposed to deal with the collision problem in traditional wireless techniques (e.g., WiFi, Zigbee) [3], [4], [5], and it can be categorized into the collision avoidance and resolution strategies. The collision avoidance scheme specifies that nodes should first probe the channel state, and then schedule links according to the channel state. In this case, real-time detection and accurate synchronization are two critical factors, yet causing additional overhead. For the collision resolution, unique time domain features (e.g., fixed waveform shape) in wireless techniques or power difference between packets are exploited to decode the concurrent packets in one collision [6], [7], [8], hence enhancing the throughput and efficiency. However, these decoding schemes can not be applied to decode the collided packets due to various time domain waveform shapes in LoRa.

In this paper, we propose a protocol named mLoRa to decode the collided packets in LoRa networks from another new perspective–frequency domain. mLoRa leverages the unique features in physical layer to resolve the collision, such as CSS, M-FSK modulation, and demodulation, the goal of which is to decompose m collided packets into m collision-free packets directly. Naturally, the theoretical throughput based on mLoRa is m-fold than the conventional LoRa network. However, there exist three main challenges to practically implement mLoRa, as summarized below.

First, it is non-trivial to decode the collided packets in one collision, since baseband signals from different nodes will add up at the LoRa gateway.

Second, the inevitable noise in real wireless channels poses a great challenge to mLoRa's design. The noise mitigation is of great concern to the overall decoding performance of mLoRa.

Third, LoRa has a high-order M-ary FSK property compared to traditional wireless techniques, which causes narrow subcarrier spacing. Therefore, even a small carrier frequency offset (CFO) can render the decoded result completely deviated from the ground truth.

As demonstrated in Fig. 1, two LoRa nodes (e.g., node A and node B) send packets to the LoRa gateway simultaneously and then cause a collision. The baseband samples from these two nodes will add up at the gateway, completely differing from the original signals. The receiver fails to decode the severely distorted signals.

To overcome above challenges, we first decode the collided packets according to the CSS modulation, where these collided packets can be successfully decoded in a sample-by-sample, and then chirp-by-chirp manner. Meanwhile, we employ the moving average scheme to mitigate the noise because the

978-1-7281-2700-2/19/$31.00 2019

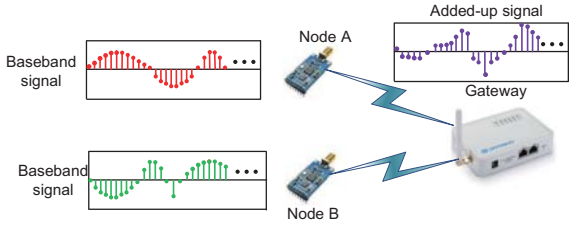


Fig. 1. The convergecast scenario when two LoRa nodes (e.g., node A and node B) transmit packets to the LoRa gateway.

signal's amplitude remains constant based on M-FSK modulation. LoRa packets consist of the preamble and payload parts, especially the preamble is composed of several standard up-chirps. Theoretically, the derived bin index should be equal to zero after de-chirping the standard up-chirp. However, it does not hold in practice due to the inevitable CFO in the decoding process, which can be regarded as the bin offset. Finally, we can obtain the actual bin values in the payload part through the subtraction of the bin offset from derived bin values.

Specifically, there exists a chirp-level time offset between two collided packets, as shown in Fig. 2(a). To resolve the collision, mLoRa derives the time offset between the two collided packets based on the designed preamble detection strategy, and then obtains the chirp-level collision-free samples and corresponding frequencies, as illustrated in Fig. 2(b). According to these collision-free frequencies and samples, mLoRa can derive the following samples within the same chirp that have been overlapped with the packet from node B called packet B, leveraging the constant linear chirp modulation and constant amplitude. Subsequently, subtracting these estimated samples from added-up baseband samples, mLoRa can derive partial samples in the first chirp of packet B. For the next iteration, these derived samples in packet B can be regarded as collision-free samples, which can be utilized for the rest samples estimation within the same chirp in packet B. Repeating the estimation and subtraction operations, mLoRa can successfully decode the collided packets in a sample-by-sample and then chirp-by-chirp manner.

The main contributions of this work can be summarized as follows.

- As described above, the convergecast scenario appears frequently in LoRa networks, resulting in the severe collision. To resolve the collision problem, we devise a novel protocol named mLoRa to decode the collided packets leveraging the unique features in LoRa's physical layer.
- Design enhancements for mLoRa have been introduced to improve mLoRa's performance, including the noise mitigation and CFO elimination schemes.
- We have implemented mLoRa on USRPs and built a six-node testbed. Experiment results demonstrate that mLoRa can decode up to three concurrent packets, the throughput of which is around 3 times more than the conventional LoRa.

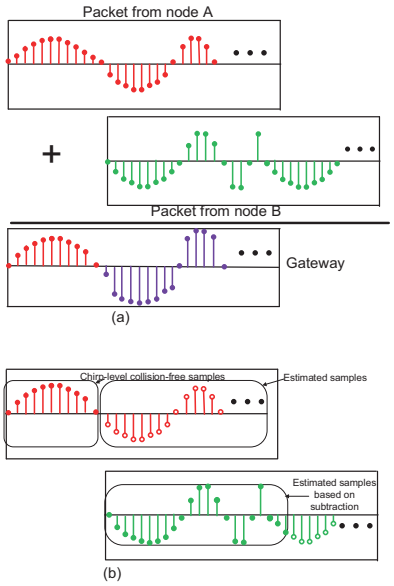


Fig. 2. The decoding process for the collided packets based on the derived time offset.

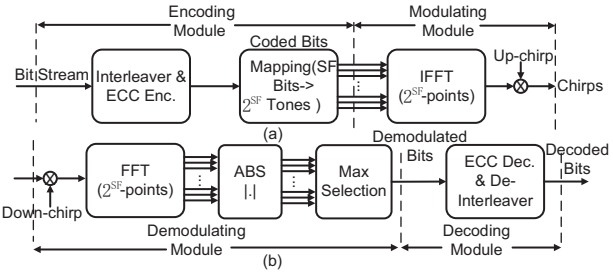


Fig. 3. The transmitter and receiver realization in LoRa's physical layer.

II. PRELIMINARY

Before introducing the detailed design of mLoRa, we provide an overview of LoRa's physical layer. Then, we illustrate the unique features in physical layer, which are the foundations in the design of mLoRa.

A. LoRa's physical layer

As a recently proposed wireless technique for low-power wide-area networks, LoRa operates at different frequencies in different regions. For example, LoRa operates at 915MHz in America, while LoRa in Europe specifies its operation in 868MHz ISM band. As demonstrated in Fig. 3(a), the LoRa transmitter mainly consists of the encoding and modulating modules. The bit stream is first interleaved and encoded, in which every SF bits are spread to another 2^{SF} bits. The spreading bits are further delivered to the inverse fast Fourier transform (IFFT) engine for the 2^{SF}-ary FSK modulation, where SF represents the spreading factor. Finally, the time-domain signal is modulated by the generated chirp signal and then transmitted.

Correspondingly, the LoRa receiver includes the demodulation and decoding modules, as described in Fig. 3(b). The

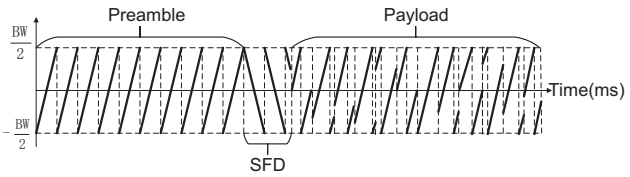


Fig. 4. The LoRa packet structure.

received signal is firstly de-chirped and then transmitted to a fast Fourier transform (FFT) engine. The demodulation can be accomplished by simply selecting the subcarrier with the maximum power at the FFT output. In the decoding module, the demodulated bits are decoded and de-interleaved. Until now, a complete process for the LoRa packet transmission and reception is accomplished.

B. Unique features in physical layer

As shown in Fig. 4, LoRa packets start with the preamble, which consists of several standard up-chirps (e.g., 8). LoRa adopts the CSS modulation in physical layer, which enables a trade-off between the data rate and sensitivity by selecting different SFs ranging from 7 to 12. SF determines the data rate and dictates the signal's sensitivity. According to the CSS modulation and packet structure, we observe that there exist three unique features in LoRa's physical layer.

Chirp spread spectrum. The payload in LoRa packets consists of several consecutive up-chirps in frequency domain, with differences only in the initial position. Within a chirp, the frequency increases linearly over time until reaching the upper frequency bound. Subsequently, it falls down to the lower frequency bound and then continues to ascend. The slope gradient for all up-chirps is constant. Although these up-chirps may be interfered by the noise in real wireless channels, their basic slope shapes are maintained.

Uniform amplitude. Unlike ASK and QAM [9], which modulate data information through different amplitude levels, LoRa adopts the M-FSK modulation. Consequently, the amplitude within one LoRa packet remains uniform.

Unique packet structure. LoRa packets start with several standard up-chirps, which linearly sweep from the lower frequency bound $-\frac{BW}{2}$ to the upper frequency band $\frac{BW}{2}$ without frequency drop. Here, BW represents the channel bandwidth.

Next, we introduce the core design of mLoRa based on above features.

III. Design of mLoRa

In this section, we present the theoretical foundation for multi-packet reception, as well as the design details of mLoRa.

A. Multi-packet collision primer

In physical layer, a wireless signal can be represented as discrete complex samples in baseband. Specifically, LoRa chirps are shaped to be a sequence of complex samples, the number of which equals to 2^{SF}. Consequently, the received signal at the LoRa gateway can also be denoted as a series of baseband samples, which are isolated at the sampling interval. Assume $X[n]$ represents the n-th sample at the LoRa node, the corresponding received sample $Y[n]$ can be defined by

$$Y[n] = HX[n] + W[n], \tag{1}$$

where $H = he^{\gamma}$ denotes the channel parameter. In particular, h and γ respectively indicate the channel attenuation and phase shift relying on the distance and environments between the transmitter and receiver.

If nodes A and B send packets to the LoRa gateway simultaneously, their baseband signals will superimpose at the gateway side, which can be expressed as

$$Y[n] = H_A X_A[n] + H_B X_B + W[n], \tag{2}$$

where H_A and H_B represent the channel parameters for nodes A and B, respectively. Meanwhile, X_A and X_B denote the baseband samples respectively from nodes A and B.

In the one-transmitter one-receiver scenario, the channel parameter H can be estimated under relatively large signal-to-noise ratio (SNR). According to Equation 1 and H, the receiver can successfully decode the received packets.

Nevertheless, when two transmitters send packets to one receiver concurrently, the decoding process becomes much more complicated. The receiver can not decode the collided packets according to Equation 2 despite H_A, H_B, and $Y[n]$ are already known. This is because two unknown variables $X_A[n]$ and $X_B[n]$ can not be resolved simultaneously only based on one equation.

In conclusion, it is tough to decode the collided packets in LoRa networks without extra information. Therefore, we design a novel protocol to decode the collided packets leveraging the unique features in LoRa's physical layer.

B. Overview of mLoRa

Since mLoRa is a physical layer protocol that enables multi-packet reception, the implementation details of mLoRa related to the physical layer including both transmitter and receiver sides are presented.

Transmitter side: We make no changes in physical layer of transmitters for mLoRa. That is to say, mLoRa adopts the conventional physical layer design of LoRa to send packets.

Receiver side: At the receiver side, mLoRa has made some lightweight changes in physical layer compared to the conventional LoRa. Since the aim of mLoRa is to decode the collided packets, these added-up signals first go through the designed preamble detection introduced in Section IV-A, rather than demodulating directly. Therefore, some extra modules should be added in the flowchart of mLoRa, as shown in Fig. 5. mLoRa is added as an independent module between the received baseband signals and conventional LoRa receiver. Combined with the design enhancements, the functionality of mLoRa is to decompose the m-packet collision into m separate sample sequences without interference from other packets. When there is no collision detected, mLoRa outputs only one sequence of baseband samples to the conventional LoRa receiver.

From above descriptions, it can be seen that mLoRa makes little change to existing LoRa protocol in physical layer, which can be easily implemented in practice.

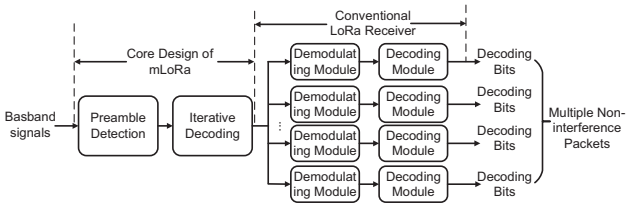


Fig. 5. The physical layer of LoRa receiver combined with mLoRa.

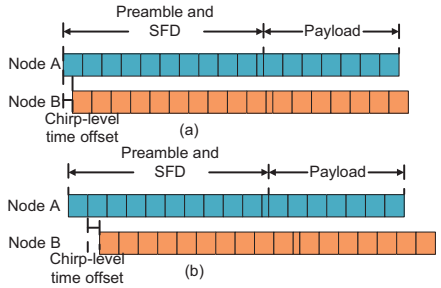


Fig. 6. Examples with chirp-level collision in the case of preamble-with-preamble collision.

C. Core Design of mLoRa

The preamble in LoRa packets is followed by the SFD–2.25 down-chirps, which complicates the decoding design. In order to simplify the implementation, we divide the collision scenario into two categories according to the time offset between the collided packets, respectively as preamble-with-preamble collision and preamble-with-payload collision. The preamble-with-preamble collision implies that at the beginning, the preamble and SFD parts in packet B collide with the corresponding parts in packet A. Meanwhile, the preamble-with-payload collision means that the preamble and SFD parts in packet B collide with the payload part in packet A. For simplicity, we use the two-packet collision case to illustrate the core design of mLoRa.

1) Preamble-with-preamble collision: The preamble-with-preamble collision case arises when the following inequality is established.

$$T_{A,B} < (N_c + 2.25) \times T_c, \tag{3}$$

where $T_{A,B}$ denotes the time offset between these two packets, and T_c represents the chirp duration. Meanwhile, N_c indicates the number of up-chirps in the preamble. The collision scenario in this case can be further classified into two categories to devise a robust decoding protocol, as demonstrated below.

Collision with chirp-level time offset. In this category, the chirp in the preamble of packet B does not align with any chirp in the preamble and sync parts of packet A, i.e., the following inequality is satisfied.

$$\delta_{A,B} = |T_{A,B}| \bmod T_c \neq 0. \tag{4}$$

Some examples for this case are listed in Fig. 6.

Collision without chirp-level time offset. The collision without chirp-level time offset implies that the chirp in the preamble of packet A completely aligns with the chirp in the preamble or SFD of packet B. This case happens if the following equation is satisfied.

$$\delta_{A,B} = |T_{A,B}| \bmod T_c = 0. \tag{5}$$

Some specific examples under this case are given in Fig. 7.

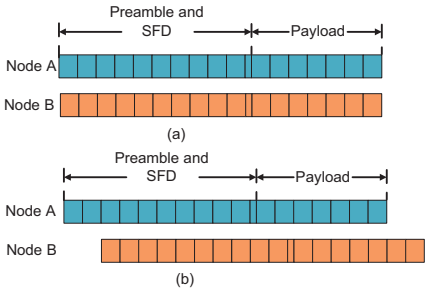


Fig. 7. Examples without chirp-level collision in the case of preamble-with-preamble collision.

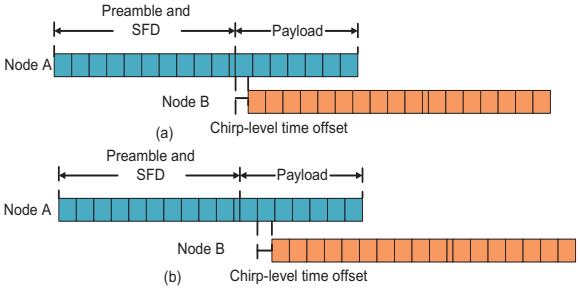


Fig. 8. Examples with chirp-level collision in the case of preamble-with-payload collision.

2) Preamble-with-payload collision: Another collision scenario is the preamble-with-payload collision, which can be defined by

$$T_{A,B} > (N_c + 2.25) \times T_c, \tag{6}$$

In this category, the unique structure of LoRa packets, which contains 2.25 down-chirps, should be considered to calculate the time offset between the collided packets. Similarly, the preamble-with-payload collision can also be divided into two cases based on the time offset $T_{A,B}$.

Collision with chirp-level time offset. Collisions under this case indicate that the chirp in the preamble of packet B does not align with the chirp in the payload part of packet A. The case occurs when the following inequality is established.

$$\delta_{A,B} = |T_{A,B} - 0.25 \times T_c| \bmod T_c \neq 0. \tag{7}$$

Typical examples in this case are illustrated in Fig. 8.

Collision without chirp-level time offset. The collision in this case implies that the chirp in the preamble part of packet B completely aligns with the payload part of packet A. Such a case can be judged by

$$\delta_{A,B} = |T_{A,B} - 0.25 \times T_c| \bmod T_c = 0. \tag{8}$$

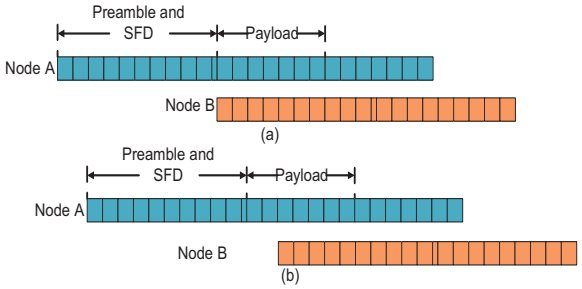


Fig. 9. Examples without chirp-level collision in the case of preamble-with-payload collision.

Examples in Fig. 9 demonstrate the collision instances without chirp-level time offset.

In this paper, we focus on the analysis and resolution when the preamble-with-payload collision happens, since the preamble-with-preamble collision resolution is similar to the preamble-with-payload collision.

D. Iterative Decoding

The core of mLoRa is to decode the collided packets using the unique features like CSS and constant amplitude. Specifically, mLoRa combines these two features with collision-free samples within the chirp-level time offset to estimate the collided samples in the same chirp.

Assume d_{num} denotes the number of samples in one chirp, equaling to 2^{SF}. The chirp-level time offset $\delta_{A,B}$ can be derived in Subsection III-C. Hence, we can obtain the number of collision-free samples k within $\delta_{A,B}$, which can be computed by $k = \lfloor d_{num} \times \frac{\delta_{A,B}}{T_c} \rfloor$. As described in Subsection III-A, we can not acquire two unknown variables $X_A[n]$ and $X_B[n]$ concurrently only based on one equation. Therefore, mLoRa attempts to introduce another novel information–chirp spread spectrum (CSS) as the relationship between samples. For example, the k collision-free samples $X_A[n-1]$, $X_A[n-2]$,..., $X_A[n-k]$ are utilized to estimate the rest $(d_{num}-k)$ samples $X_A[n]$, $X_A[n+1]$,..., $X_A[n+d_{num}-k-1]$, which collide with the baseband samples in packet B.

In the preamble-with-payload collision case, the preamble and SFD parts in packet B collide with the payload of packet A at the beginning. Unlike the preamble and payload parts only consisting of up-chirps, the SFD composed of 2.25 down-chirps challenges the decoding design. To deal with this challenge, mLoRa categorizes the decoding process into two different levels, respectively as the chirp-level decoding and sample-level decoding.

Chirp-level decoding. In the chirp-level decoding, mLoRa obtains the first full chirp in packet B through the sample-level decoding, which will be introduced later. As mentioned above, the preamble consists of several standard up-chirps, and the SFD is comprised of 2.25 standard down-chirps with constant amplitude. Consequently, mLoRa can acquire the overall set of samples constituting the preamble and SFD parts, the number of which equals to $(N_c + 2.25)d_{num}$. Then, subtracting the acquired sample set from $Y[n]$ in Equation 2, we can obtain the corresponding $(N_c + 2.25)d_{num}$ samples in packet A.

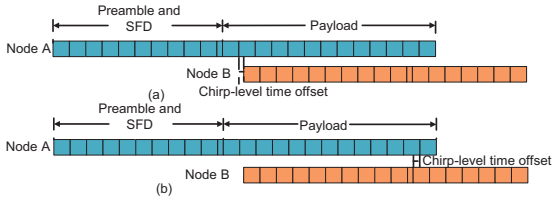


Fig. 10. Examples with chirp-level decoding in the case of preamble-with-payload collision.

Naturally, these $(N_c+2.25)d_{num}$ collided samples that have been decoded can be considered as collision-free samples for both packets A and B. In conclusion, mLoRa first decodes the collided packets in the chirp-level in order to overcome the challenge brought by 2.25 down-chirps in the SFD part. Examples illustrating this decoding process are described in Fig. 10.

Sample-level decoding. After the chirp-level decoding, we can obtain a new set of collision-free samples within a recalculated chirp-level time offset $\delta_{A,B}$ according to Equation 9. Then, the preamble-with-payload collision is transformed into the payload-with-payload collision case. For the sample-level decoding, mLoRa can decode the collided samples in a sample-by-sample and then chirp-by-chirp manner because the payload is composed of consecutive up-chirps and the length of these up-chirps is the same.

$$\delta_{A,B} = |T_{A,B} - 0.25 \times T_c - (N_c + 2.25) \times T_c| \bmod T_c \neq 0. \quad (9)$$

First, mLoRa computes the frequency value f_{n-1} corresponding to the last collision-free sample $X_A[n-1]$ within $\delta_{A,B}$. The computation process is shown in Algorithm 1.

Before introducing Algorithm 1, we present the theoretical foundation for the frequency estimation at one certain sample. Taking the sample $X_A[n-i]$ for example, the phase difference between the sample $X_A[n-i]$ and its previous sample $X_A[n-i-1]$ can be calculated by

$$p_{n-i} = \arctan\left(\frac{im\{X_A[n-i] \times \overline{X_A[n-i-1]}\}}{re\{X_A[n-i] \times \overline{X_A[n-i-1]}\}}\right), \quad (10)$$

where $im\{X_A[n-i] \times \overline{X_A[n-i-1]}\}$ and $re\{X_A[n-i] \times \overline{X_A[n-i-1]}\}$ represent the real and imaginary parts of the multiplication result, respectively, and $\overline{X_A[n-i-1]}$ denotes the conjugate value of $X_A[n-i-1]$. Accordingly, we can derive the frequency at the sample $X_A[n-i]$ according to the following equation.

$$f_{n-i} = \frac{p_{n-i} \times (2^{SF}-1)}{T_c}, \quad (11)$$

where $\frac{T_c}{2^{SF}-1}$ denotes the time interval between two adjacent samples. Therefore, we can obtain the frequency value at any collision-free sample.

Next, we illustrate the frequency acquisition process at the last collision-free sample $X_A[n-1]$ in Algorithm 1. Specifically, $\frac{BW}{2^{SF}-1}$ represents the frequency increment between two

adjacent samples because of the linearly increasing frequency set in up-chirps. mLoRa starts the frequency estimation from the 2-th collision-free sample, as shown in line 7. The process is repeated until reaching the last collision-free sample. However, it should be noted that the deduced frequency may exceed the upper frequency bound $\frac{BW}{2}$ during the iterative process. According to the CSS specification, the current frequency should be set as the lower frequency bound $-\frac{BW}{2}$ when the deduced one exceeds the upper frequency bound, as depicted in line 9.

Algorithm 1: The frequency estimation at the last collision-free sample.

input : k: The number of collision-free samples; $X_A[n-k], ..., X_A[n-1]$: The set of collision-free samples
output: f: The frequency value at the last collision-free sample;

1 $i \leftarrow 1$; $f \leftarrow 0$;
2 **if** *(i<k)* **then**
3 **while** *(i<k)* **do**
4 **if** *(i == 1)* **then**
5 $f \leftarrow$ Frequency estimation according to samples $X_A[n-k+i-1], X_A[n-k+i]$, and Equations (10)-(11);
6 **else**
7 $f \leftarrow f + \frac{BW}{2^{SF}-1}$;
8 **if** *(f>$\frac{BW}{2}$)* **then**
9 $f \leftarrow -\frac{BW}{2}$;
10 **else**
11 $i \leftarrow i$+1;
12 **else**
13 **return** f;

After obtaining the frequency value f_{n-1} corresponding to the sample $X_A[n-1]$ in Algorithm 1, we introduce the estimation algorithm for the next sample $X_A[n]$ collided with packet B. Specifically, according to the CSS modulation and f_{n-1}, we can obtain the frequency value f_n corresponding to the sample $X_A[n]$, which can be defined by

$$f_n = f_{n-1} + \frac{BW}{(2^{SF}-1)}. \tag{12}$$

Using f_n, Equation 13 can be established with only one unknown variable $X_A[n]$, according to the relationship between phase and frequency.

$$f_n = \arctan(\frac{im\{X_A[n] \times \overline{X_A[n-1]}\}}{re\{X_A[n] \times \overline{X_A[n-1]}\}}) \times \frac{(2^{SF}-1)}{T_c}. \tag{13}$$

Meanwhile, we can also get the following equation with the unknown variable $X_A[n]$ because the amplitude within the same packet remains constant.

$$E_{aver} = \{im\{X_A[n]\}^2 + \{re\{X_A[n]\}^2, \tag{14}$$

where E_{aver} represents the average energy of collision-free samples. Combining Equations (12)-(14) with the frequency value f_{n-1}, mLoRa can derive the first collided sample $X_A[n]$. Iteratively, it can estimate the second collided sample value $X_A[n+1]$ based on $X_A[n]$. Repeating this operation, mLoRa can obtain the collided samples $X_A[n]$, $X_A[n+1]$,..., $X_A[n+d_{num}-k-1]$ of packet A in a sample-by-sample manner. In addition, it can derive the corresponding collided samples $X_B[n]$, $X_B[n+1]$,..., $X_B[n+d_{num}-k-1]$ in packet B through subtracting X_A from Y, as described in Algorithm 2. Similar to Algorithm 1, the derived frequency may exceed the upper frequency bound $\frac{BW}{2}$, hence it should be set as $-\frac{BW}{2}$ when this case happens, as shown in line 13.

Algorithm 2: The collided samples estimation.

input : f_{n-1}: $X_A[n-1]$:The last collision-free sample value; The frequency value at the sample $X_A[n-1]$; $Y_A[n], ..., Y_A[n+d_{num}-k-1]$: The added-up samples of packets A and B
output: S_1: The set of collided samples in the node 1 packet; S_2: The set of collided samples in the node 2 packet

1 $i \leftarrow 0$; $f \leftarrow f_{n-1}$; $a \leftarrow 0$; $b \leftarrow 0$;
2 $S_1 \leftarrow \emptyset$;
3 $S_2 \leftarrow \emptyset$;
4 **if** *(i<($d_{num}-k$))* **then**
5 **while** *(i<($d_{num}-k$))* **do**
6 $f \leftarrow f + \frac{BW}{(2^{SF}-1)}$;
7 $f_{n+i} \leftarrow f$;
8 $a \leftarrow$ The estimation of $X_A[n+i]$ according to Equations (12)-(14), the previous sample $X_A[n+i-1]$, and its corresponding frequency f_{n+i-1};
9 $b \leftarrow Y[n+i] - a$;
10 $S_1 \leftarrow S_1 \cup a$;
11 $S_2 \leftarrow S_2 \cup b$;
12 **if** *(f>$\frac{BW}{2}$)* **then**
13 $f \leftarrow -\frac{BW}{2}$;
14 **else**
15 $i \leftarrow i$+1;
16 **else**
17 **return** S_1;
18 **return** S_2;

In conclusion, mLoRa can acquire a complete chirp of packet A in the sample-level, and partial collided samples in packet B, referred as to the sample set S_2 in Algorithm 2. For the next iteration, mLoRa leverages the collision-free samples in the set S_2 to decode the following collided samples within the same chirp, according to both Algorithms 1 and 2. Therefore, mLoRa can decode the collided packets in a chirp-by-chirp manner until no collision exists.

For the collision case without chirp-level time offset, we exploit the amplitude-level difference to decode the collided packets, which is similar with the resolution strategy in [6]. In order to mitigate the noise influence and enhance mLoRa's accuracy, the backward decoding starting from the last collided chirp has also been implemented in mLoRa. Consequently, mLoRa can obtain two symmetric decoded sample sequences, which is a double check to reduce the estimation errors.

IV. Design Enhancements for mLoRa

However, mLoRa can not provide a satisfactory performance only based on the core design, since there exist many uncertain factors in wireless communications. For example, the noise interference and CFO are inevitable problems in the practical implementation of mLoRa. This section presents the design enhancements to mitigate these effects in the decoding process.

A. Time offset detection

In mLoRa, the precise time offset $T_{A,B}$ detection between the collided packets plays a crucial role.

To achieve this goal, mLoRa leverages the unique preamble structure. Unlike other traditional wireless techniques, LoRa packets start with N_c consecutive standard up-chirps. At the receiver end, the signal can be de-chirped by multiplying the generated standard down-chirp and then performing FFT. The demodulation can be accomplished by simply selecting the bin with max power. Motivated by the demodulation procedure, we develop a novel preamble detection scheme. Specifically, align the first d_{num} received samples with the standard down-chirp, and then multiplying these two different sample series and performing FFT. Subsequently, shift the alignment of received samples to the next sample and re-perform the multiplication and FFT until reaching the packet's end. The bin index with max power in each FFT result is stored.

Different from the preamble detection for one single packet, we observe that two peaks exist among the FFT results in the case of two collided packets, as demonstrated in Fig. 11. This is because the first chirp in the collision part are the superposition of the payload of packet A and the preamble of packet B. Note that when the standard up-chirp aligns with the standard down-chirp, the peak with the maximum power after FFT is at bin 0. If shifting the beginning of the standard up-chirp to the next sample, the peak will also shift to bin 1 with less power. Hence, in the collision case, mLoRa selects the peak's bin with the maximum power as the preamble's demodulation result, which is referred to as peak 1 in Fig. 11.

After obtaining n bin indexes based on the proposed scheme, denoted as $b_0, b_1, ..., b_{n-1}$. In the ideal case, the bin index when the standard up-chirps in the preamble part align with the standard down-chirp should be 0, the number of which is equal to N_c. However, in real wireless communications, these bin indexes practically deviate from the ground truth–0.

Then, the designed preamble detection scheme can be defined by

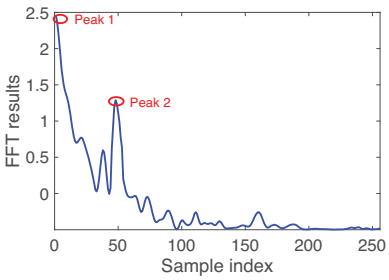


Fig. 11. The FFT results within one chirp in the collided packets.

$$|b_i - b_{i+j\times d_{num}}| < b_{offset}, \forall j \in [1, N] \tag{15}$$

where b_{offset} represents the tolerant bin offset due to the noise and CFO effects, and N denotes the required number of standard up-chirps. Through numerous field studies, b_{offset} and N are set as 2 and 4, respectively. To illustrate, if one certain bin b_i, and the following bins $b_{i+d_{num}}$, $b_{i+2d_{num}}$, $b_{i+3d_{num}}$, $b_{i+4d_{num}}$ can satisfy the above constraint, the preamble of a LoRa packet is detected. The beginning sample index of the detected packet is b_i. Finally, we can obtain the time offset $\delta_{A,B}$ using the preamble detection for each individual packet.

B. Noise effect mitigation

Noises are inevitable in real wireless communications and distort the raw signals. In mLoRa, we propose an anti-noise design to mitigate the noise interference.

Traditionally, noises in wireless channels are time-varying, hence the noise effect on every chirp differs from each other. Meanwhile, the noise model follows the Gaussian distribution $W \sim N(0, \delta^2)$. Hence, the noise effect on different chirps trends to be the same. Leveraging the concept of average noise $\overline{W}$, mLoRa averages the amplitudes of all chirps, during which the noise W can be averaged to be $\overline{W}$. The noise interference can be considerably mediated when re-operating the core design of mLoRa with the average amplitude.

C. Frequency offset elimination

When performing FFT, ideally, the receiver witnesses a peak in the first bin when demodulating the standard up-chirps of the preamble. However, CFO always exists induced by the environments and inaccurate synchronization. In the meantime, LoRa adopts a high-order M-ary modulation in order to increase the spectrum efficiency. For example, if LoRa is configured as the lowest data rate mode, the FSK is 4096-ary and thus the frequency interval between two samples is only 30.5 Hz. Compared to the small frequency interval, relatively large CFO can easily decrease the overall performance of mLoRa.

Based on one key observation–a time delay in the chirp signal translates to a frequency shift, we can eliminate the CFO when decoding these collision-free packets that have been parsed by mLoRa. Firstly, mLoRa can obtain the bin value b_0 when demodulating the preamble part. The difference

between b_0 and 0 can be regarded as the bin offset caused by CFO. Then, bin indexes in the following payload part only needs to be subtracted by b_0, hence achieving the accurate demodulation results.

V. Analysis of mLoRa

This section analyzes the application scope of mLoRa and the error propagation in the decoding process.

A. Beyond two-packet collision

mLoRa can be easily extended from the two-packet collision to the case beyond two-packet collision. Here, the extension case takes the three-packet collision for example, which can be adopted for other multi-packet collision cases.

According to the preamble detection scheme in Section IV-A, the chirp-level time offsets $\delta_{A,B}$, $\delta_{B,C}$, and $\delta_{A,C}$ between packets A, B, and C can be calculated. Assume that samples within $\delta_{A,B}$ are collision-free at the beginning of the first collision. Therefore, the first chirp in packet A can be obtained based on mLoRa. Subsequently, the first chirp in packet B can be obtained firstly through subtraction and then estimation based on mLoRa. For the second collision, the first chirp in packet C can be derived by subtracting from the added-up samples in packets A and B, and then estimated based on mLoRa. Finally, iterate through above operations until no collision exists.

B. Error propagation analysis

Up to now, mLoRa can successfully decode the collided packets on the premise of correct decoding. However, the wrong estimation of one certain sample in mLoRa probably incurs the decoding failure. Since mLoRa adopts the iterative decoding strategy, the wrongly decoding chirp affects the decoding accuracy of later chirps. Using the error propagation analysis related to the iterative decoding in [7], it can be concluded that the error propagation in mLoRa dies exponentially.

VI. Implementation and Experiments

We have implemented mLoRa on USRPs and build a six-node testbed. In this section, we first introduce the experiment settings, followed by the experiment results and analysis.

1) Experiment settings: The sampling rate is set as 2MHz, and the transmission gain is set as 89dB due to the power constraint set in GNURadio. The carrier frequency specified by transmitters and receiver operates at 915MHz. Therefore, transmitters can send LoRa packets including a preamble with 10 up-chirps, a SFD with 2.25 down-chirps, and the payload with a random number of up-chirps, to the receiver.

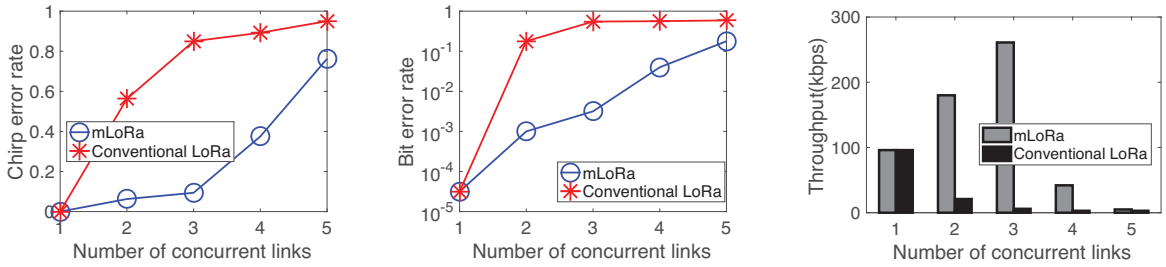
However, the length of LoRa packets highly depends on SF. For example, the maximum packet length can reach up to 222 bytes when setting SF as 7, while 51 bytes are the maximum allowable packet length when SF is set as 13. Based on the experiment setup shown in Fig. 12, we first investigate the application scope of mLoRa, with respect to metrics such as chirp error rate(CER), bit error rate(BER), and throughput. Meanwhile, we conduct massive experiments with various packet lengths, when setting SF as 8. Finally, we have also explored mLoRa's performance with different SFs.

Fig. 12. The experiment setup.

2) Experiment results and analysis: First of all, experiments are carried out to demonstrate the application scope of mLoRa with the number of concurrent transmission links ranging from 1 to 5. In the experiment, when the CER is below 10% and the BER is lower than 5×10^{-3}, the packet is considered to be correctly received. Fig. 13(a) shows the CER results when setting the packet size as 40 bytes and SF as 8, respectively. It verifies that conventional LoRa can successfully decode one single packet based on existing demodulating and decoding strategies. However, when two packets collide at the receiver side, it cannot decode the collided packets since the baseband samples are added up. Specifically, the CER increases from less than 10% to more than 60%. Compared to conventional LoRa, mLoRa can successfully decode the collided packets, the number of which can reach up to 3. In particular, when the number of collided packets increases from 2 to 3, the CER also increases. When the number of collided packets becomes 4, mLoRa based CER increases rapidly compared to the case when the number of collided packets equals to 3, which is much higher than 10%. The reason is that coupled with more collided packets, the greater cumulative noise in the decoding process incurs the collided packets decoding failure.

Similarly, the BER also increases with the increase in the number of concurrent links, as described in Fig. 13(b). Specifically, the BER based on conventional LoRa becomes more than 5×10^{-3} when only two packets collide at the receiver side. Consequently, conventional LoRa fails to deal with the collision problem even in the simplest collision case. With regard to mLoRa, the BER remains lower than 5×10^{-3} in the cases of both 2-packet and 3-packet collisions. However, when the 4-packet collision happens, the BER rises to be more than 5×10^{-3}, thus implying the decoding failure. The variation trend of BER is highly similar to CER with the increase in the number of collided packets. This is because the noise interference increases with the number of concurrent transmissions in the decoding process of mLoRa. In conclusion, mLoRa can successfully decode up to 3 collided packets.

Fig. 13(c) shows the total throughput with different number of concurrent transmission links. At the beginning, it can be observed that the throughput based on mLoRa increases with the number of concurrent transmissions, reaching up to be more than 250kbps when 3-packet collision occurs. However, when the number of collided packets increases to 4, the throughput declines significantly, as low as around 40kbps,



(a) (b) (c)

Fig. 13. mLoRa based experimental results with respect to CER, BER, and throughput when setting the packet size as 40 bytes and the SF as 8, respectively.

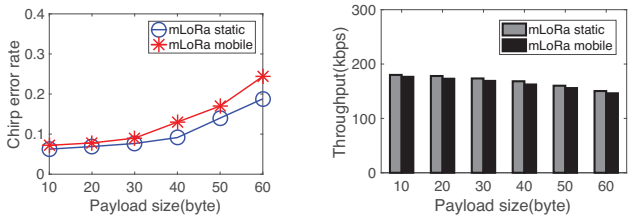
since mLoRa cannot decode beyond 3-packet collisions.

We have also investigated the performance of mLoRa when setting SF as 12. In this case, the number of samples within one chirp is equal to 4096, which is much more than the case of SF=8. We compare the CER, BER, and throughput performance between these two different SF cases. Experiment results are depicted in Fig. 14. Specifically, Fig. 14(a) compares the BER results when setting different SFs, from which we can observe that the CER in the case of SF=12 is lower than that when SF is set as 8. Meanwhile, the BER in the case of SF=12 remains lower than the case when setting SF as 8, as demonstrated in Fig. 14(b). The reason is that LoRa can expand its extended channel through adjusting SF, which in turn contributes to both processing gain and anti-jamming ability improvement. With SF becoming larger, the processing gain and anti-jamming ability for LoRa signals increase.

On the contrary, the throughput under the case of SF=12 is much lower than that when setting SF as 8, as shown in Fig. 14(c). In LoRa modulation, data rates will decrease when SF increases, since an effective bit is composed of $\frac{2^{SF}}{SF}$ samples in LoRa communications. Therefore, the total throughput declines when SF increases.

Next, we have evaluated mLoRa's performance with different payload sizes when setting the number of concurrent transmissions as 2. As illustrated in Fig. 15(a) and Fig. 15(b), when increasing the packet length from 10 to 60 bytes, both CER and BER based on mLoRa gradually increases simultaneously, and finally beyond 10%. Fig. 15(c) describes the total throughput based on mLoRa under different payload sizes. In particular, the total throughput slightly decreases from 7.8kbps to 5.4kbps with the payload size increases. This is because that with the increase of payload size, the noise accumulation problem becomes more serious in the chirp-by-chirp decoding process of mLoRa. Fortunately, LoRa packets are conventionally small since it is widely applied in fields such as logistics, agriculture, smart city and so on.

Above experiments are conducted in a static state. Next, we explore the performance of mLoRa in a mobile state when setting SF as 8, and the number of concurrent transmission links as 2. In this case, two USRPs move slowly while one USRP acting as the receiver keeps static. As depicted in Fig. 16(a), the CER based on mLoRa in a mobile state is a little higher than the static state. In addition, from Fig. 16(b) we can find that the total throughput in a mobile state is a little lower than the static one. The reason is that the mobility results in the occasional link weakness, thus rendering the signal intensity variable. Furthermore, the signal intensity variation affects the decoding accuracy in mLoRa.



(a) (b)

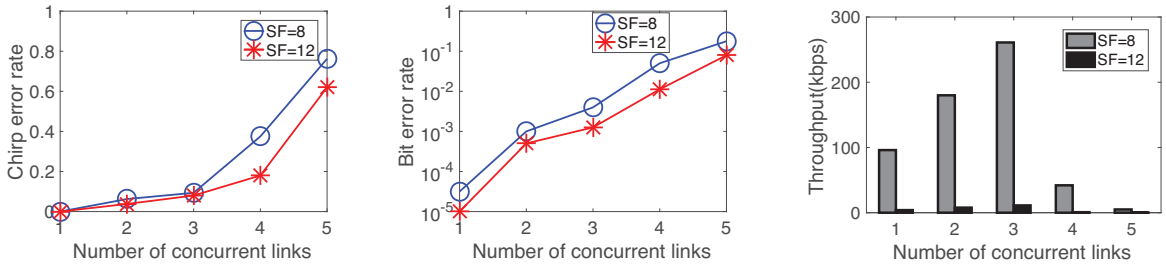
Fig. 16. The experimental results comparison with respect to CER and throughput in mobile and static states.

VII. Related Work

LoRa adopts the ALOHA protocol in MAC layer to schedule links. In LoRa networks, the convergecast scenario frequently occurs because thousands of nodes connect to one single gateway. Above two factors result in the collision problem. However, there exists little research work on dealing with this problem in LoRa networks up to now. Existing literature about conquering the collision problem in traditional wireless techniques (e.g., Zigbee and WiFi) can be divided into two categories, respectively as the collision-avoidance and collision-resolution schemes.

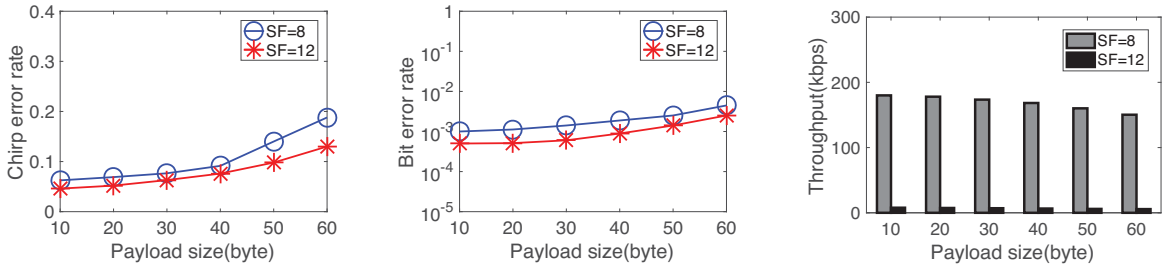
Collision-avoidance Extensive research work focuses on the collision-avoidance in wireless sensor networks. The most traditional method is to adopt CSMA/CA [10], [11], [12], the core of which is to utilize random backoff time and retransmission when collision happens. For example, ZigBee adopts CSMA [13]. However, this method fails to work in some scenarios such as hidden terminals. In addition, the random delay slows down the transmission and then decreases the efficiency.

Hence, RTS-CTS is proposed to enhance the performance of CSMA/CA [14]. It can overcome the hidden terminal problem by handshake. In order to improve the efficiency of MAC layer with the increase of PHY data rates, Tan et al. [15] devise a fine-grained channel access method, which can be



(a) (b) (c)

Fig. 14. The experimental results comparison with respect to CER, BER, and throughput when setting the SF as 8 and 12, respectively.



(a) (b) (c)

Fig. 15. The experimental results comparison with respect to CER, BER, and throughput under different payload sizes.

commensurate with PHY data rate and typical frame size. The core of the proposed scheme is to utilize the physical layer RTS/CTS signal and frequency domain backoff to efficiently coordinate subchannel access. However, experiments demonstrate that RTS/CTS significantly decreases the overall network efficiency [16]. Consequently, conventional wireless techniques has disabled the RTS/CTS strategy.

Collision-resolution Advanced technologies propose to resolve packet collisions at the receiver side rather than avoiding. The capture effect is one unique feature utilized by multi-packet decoding. For example, Liao et al. [8] construct a multi-hop LoRa network. At each receiver, one of the collided packets can be successfully decoded based on the capture effect.

Successive interference cancellation (SIC) is another conventionally adopted scheme for multi-packet decoding [17], [18]. It also requires distinct power difference or known precoded information. However, these strict requirements are hard to be satisfied in most cases for wireless sensor networks.

Therefore, construct interference is proposed to realize multiple simultaneous transmissions with less prior settings [19], [20]. Yet, this strategy requires the same content for all concurrent transmissions. It is not suitable for being adopted in LoRa networks since packets transmitted from different nodes differ from each other.

In order to eliminate these strict requirements, researchers have make determined efforts to realize multi-packet reception. For example, research work in [7], [6] can successfully decode the multi-packet collision without any prior settings or other stringent requirements. Nevertheless, these schemes are designed for conventional wireless techniques, and can not be applied to LoRa networks.

Different from these wireless techniques, LoRa is a unique wireless technique which adopts CSS in the physical layer. Meanwhile, the preamble in LoRa packets includes several up-chirps and 2.25 down-chirps. All of these differences pose a huge challenge to the collision resolution in LoRa networks.

VIII. Conclusion

In this paper, we present mLoRa, a novel protocol designed to decode the collided packets in LoRa networks. mLoRa mainly leverages the unique features in LoRa's physical layer, including CSS, M-FSK modulation, and demodulation. Specifically, the decoding process is divided into two categories to simplify the implementation, respectively as the chirp-level and sample-level decodings. If the preamble/payload-with-preamble collision case happens, mLoRa transforms them into the payload-with-payload collision through the chirp-level decoding. For the payload-with-payload collision, the collided packets can be successfully decoded in a sample-by-sample and then chirp-by-chirp manner. Experimental results have demonstrated that mLoRa can decode up to 3 concurrent transmission links and improve the total throughput in LoRa networks.

IX. Acknowledgements

This work was supported in part by the National Key R&D Program of China 2018YFB1004703, NSFC grant 61972253, 61672349, 61672353.

REFERENCES

[1] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "PLoRa: a passive long-range data network from ambient LoRa transmissions," in *SIGCOMM*. ACM, 2018, pp. 147–160.

[2] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa backscatter: Enabling the vision of ubiquitous connectivity," *IMWUT*, vol. 1, no. 3, p. 105, 2017.

[3] S. Wang, S. M. Kim, L. Kong, and T. He, "Concurrent transmission aware routing in wireless networks," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6275–6286, 2018.

[4] L. Rocha and D. Sasaki, "The backbone packet radio network coloring for time division multiple access link scheduling in wireless multihop networks," *Networks*, vol. 71, no. 4, pp. 403–411, 2018.

[5] F. Bouabdallah, R. Boutaba, A. Mehaoua *et al.*, "Collision avoidance energy efficient multi-channel MAC protocol for underwater acoustic sensor networks," *IEEE Transactions on Mobile Computing*, 2018.

[6] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in zigbee," in *Proceedings of the 21st annual international conference on mobile computing and networking*. ACM, 2015, pp. 552–565.

[7] S. Gollakota and D. Katabi, *Zigzag decoding: combating hidden terminals in wireless networks*. ACM, 2008, vol. 38, no. 4.

[8] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-hop LoRa networks enabled by concurrent transmission," *IEEE Access*, vol. 5, pp. 21 430–21 446, 2017.

[9] M. T. Abuelma'atti, "New ASK/FSK/PSK/QAM wave generator using multiple-output operational transconductance amplifiers," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 4, pp. 487–490, 2001.

[10] R. Laufer and L. Kleinrock, "The capacity of wireless CSMA/CA networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1518–1532, 2016.

[11] M. Michalopoulou and P. Mähönen, "A mean field analysis of CSMA/CA throughput," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2093–2104, 2017.

[12] B. Van Houdt, "Explicit back-off rates for achieving target throughputs in CSMA/CA networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 765–778, 2017.

[13] C. Wang, T. Jiang, and Q. Zhang, *ZigBee® network protocols and applications*. Auerbach Publications, 2016.

[14] P. Lin and L. Zhang, "Full-duplex rts/cts aided csma/ca mechanism for visible light communication network with hidden nodes under saturated traffic," in *ICC*. IEEE, 2018, pp. 1–6.

[15] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless LAN," *SIGCOMM*, vol. 41, no. 4, pp. 147–158, 2011.

[16] G. Judd and P. Steenkiste, "Using emulation to understand and improve wireless networks and applications," in *NSDI*. USENIX, 2005, pp. 203–216.

[17] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: interference cancellation for wireless LANs," in *MOBICOM*. ACM, 2008, pp. 339–350.

[18] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: Carving out MAC layer opportunities," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 346–357, 2013.

[19] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *NSDI*, 2013, pp. 269–282.

[20] Y. Wang, Y. Liu, Y. He, X.-Y. Li, and D. Cheng, "Disco: Improving packet delivery via deliberate synchronized constructive interference," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 713–723, 2015.