Poster Abstract: Frequency-Hopped Chirp Spread Spectrum for Collision Resolution in LoRa Network

Mingyu Park
Department of Computer Science & Engineering
Chung-Ang University
Republic of Korea

Jeongyeup Paek
Department of Computer Science & Engineering
Chung-Ang University
Republic of Korea

ABSTRACT

LoRa technology enables low power long range communication thanks to its modulation scheme, the *'Chirp Spread Spectrum (CSS)'*. However, it is extremely susceptible and vulnerable to chirp collisions. To resolve this problem, we propose a novel modulation scheme, *'Frequency Hopped Chirp Spread Spectrum (FH-CSS)'*. *FH-CSS* slices a chirp into several sub-chirps and places each to different frequency offsets. This enables demodulated signals of misaligned chirps to spread out in the frequency domain due to the characteristic of CSS. We implement *FH-CSS* on GNU Radio and compare its performance against standard CSS through real experiments on USRP B200 software defined radios. Preliminary results show that *FH-CSS* achieves around 90% decoding success rate despite collision while standard CSS has less than 10% on average.

KEYWORDS

LoRa, Frequency Hopping, Packet Collision Resolution

1 INTRODUCTION

LoRa[1] attracts attention as one of the key wireless technology for Internet of Things applications owing to its capability to cover a wide area with low power consumption. LoRa enables low-power long-range communication via its modulation scheme, the *'Chirp Spread Spectrum (CSS)'*, which has exceptional receiver sensitivity. It allows LoRa to be robust and resilient to noise (or low signal-to-noise ratio (SNR)) compare to other wireless technologies such as Bluetooth or ZigBee. However, this advantage becomes a disadvantage when two or more chirps collide. In other words, high sensitivity of CSS modulation suffers more performance degradation with packet collision.

Several recent studies have proposed collision resolution schemes to enable concurrent transmissions in LoRa [2, 4–6]. Our work aims for the similar goal, but with a novel approach: We propose *'Frequency Hopped Chirp Spread Spectrum (FH-CSS)'* to decode packets despite collisions. *FH-CSS* slices a chirp into several sub-chirps and assigns them to different frequency indexes. This spreads out the demodulated signals of misaligned chirps in the frequency domain due to the characteristic of CSS, allowing correct decoding of the aligned chirp even if the collision has extremely low collision offset.

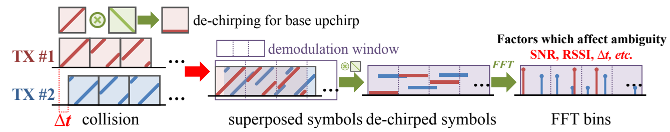
We implement *FH-CSS* in GNU Radio with USRP B200 software defined radio (SDR), and evaluate through real experiments. Preliminary evaluation results show that *FH-CSS* achieves approximately 90% packet reception success rate despite collision while standard CSS has less than 10% on average.

[1]LoRa Alliance®, https://lora-alliance.org/

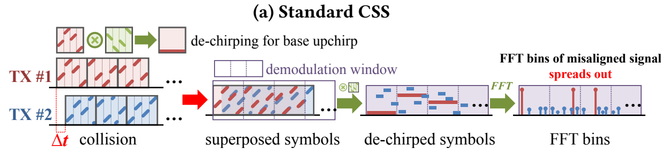
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A4A5034130 & No. 2021R1A2C1008840), and also by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

SenSys '23, November 12–17, 2023, Istanbul, Turkiye

ACM ISBN 979-8-4007-0414-7/23/11…$15.00
https://doi.org/10.1145/3625687.3628406



(a) Standard CSS



(b) FH-CSS

Figure 1: Demodulation for collision packets

2 *FH-CSS* DESIGN

We first provide a primer of LoRa CSS modulation with some intuitions that leads to our design of *FH-CSS*.

Standard CSS uses a 'chirp', which increases/decreases frequency linearly, to embed and extract data symbols. Each is called upchirp and downchirp respectively. Modulation and demodulation are done according to the following steps;

(1) Start frequency index of an upchirp represents a data symbol.
(2) Transmitter sets the start frequency according to the data to send, and transmits the packet.
(3) Receiver multiplies received signal with a downchirp. (*de-chirp*)
(4) Fast Fourier Transform (FFT) is applied to the de-chirped signal.
(5) Index of FFT bin peak represents a demodulated data symbol.

Since CSS chooses a FFT bin peak of de-chirped signal as the data symbol, decoding success depends on how clean the de-chirped signal is. However, since a chirp *consecutively increases frequency*, a collision of two chirps leads to less distinction in FFT bin height between superposed symbols as shown in Fig. 1a. The ambiguity will be more severe as the collision offset (δt) gets narrower.

To alleviate the ambiguity due to collision, *FH-CSS* exploits *frequency hopping* within the chirps based on the observation that the

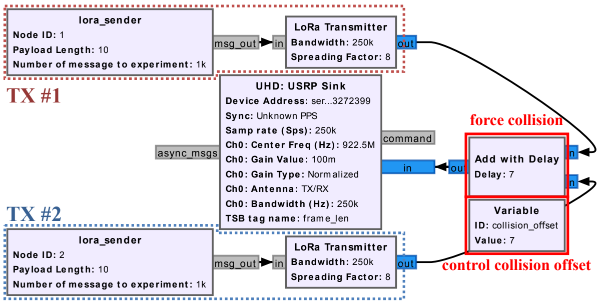


Figure 2: A LoRa transmitter to generate collision signal

energy of de-chirped signal spreads out if chirps are not a counter part of each other [6]. Fig. 1b illustrates an example of how *FH-CSS* works. Each chirp is sliced into several sub-chirps, and each sub-chirp starts to increase frequency from different frequency indexes according to the *pre-shared frequency hopping pattern*. Upon reception, a *FH-CSS* receiver multiplies the received signal with a frequency-hopped downchirp to enable de-chirped signals to have a FFT bin peak similar to the standard CSS. While doing this, the receiver aligns the demodulation window to fit a downchirp perfectly to a frequency-hopped upchirp. Then the de-chirped signals of a chirp misaligned with demodulation window has less FFT bin height while the FFT bin of the aligned signal peaks. Therefore, a *FH-CSS* LoRa receiver is capable of recovering packets from collisions without any additional signal processing technique.

However, *FH-CSS* has several challenges to support real-time PHY-rate decoding with high reliability. We plan to further investigate and refine below challenges in our future work.

Synchronization/Alignment. A key idea of *FH-CSS* is that 'misaligned chirps will spread out'. The receiver can decode a preamble only if the demodulation window aligns with a frequency-hopped chirp. Therefore, the correctness and time-complexity of alignment is the most important challenge of *FH-CSS*. Our initial proof-of-concept implementation uses a brute-force linear search.

Number of sub-chirps per chirp introduces a trade-off. Higher number of sub-chirps provide better robustness to collisions, but it is more susceptible to noise, and more importantly, harder to synchronize/align the demodulation window. The number of sub-chirps was set to 16 in our preliminary experiments.

Frequency hopping pattern is momentous for collision resolution reliability. It should ensure that *FH-CSS* has *'different level of changes'* between adjacent hopping patterns so that they are better distinguishable. We have carefully hand-picked a pseudo-random hopping pattern that follow this rule for current implementation.

3 EVALUATION

We implement *FH-CSS* and standard CSS in GNU Radio [1] using an open source LoRa library [3], and conduct experiments on USRP B200 SDR[2]. To conduct packet collision experiments with precisely controlled collision offsets, we built a LoRa transmitter to mix two modulated signals with deterministic delay in software (Fig. 2).

Each LoRa packet is modulated with the spreading factor SF8 with 250 kHz bandwidth, thus the number of samples in a chirp is $2^8 = 256$. The number of sliced sub-chirps is set to 16, meaning

[2] USRP B200 SDR - Ettus Research, https://www.ettus.com/all-products/ub200-kit/

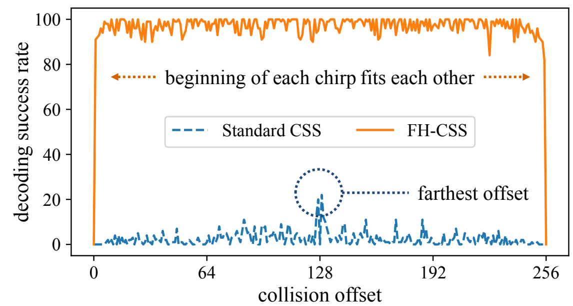


Figure 3: Decoding success ratio with varying collision offset

frequency hopping occurs 16 times in a chirp. An SDR LoRa transmitter transmits 100 packets every 100 milliseconds. The receiver demodulates and decodes the received signals, and checks validity. Then it counts the number of successfully decoded packets to calculate the decoding success rate.

Fig. 3 plots the decoding success rate according to collision offset. Standard CSS has poor performance, less than 10% decoding rate on average, when it encounters collision. An interesting point for standard CSS is that the decoding rate increases slightly as chirps move apart farther, and decreases again after 128^{th} offset. This is because the first chirp of second transmitter becomes close to the second chirp of the first transmitter.

Nevertheless, *FH-CSS* maintains ~90% decoding success rate for almost any collision offsets. The performance drops to 0% whenever collision offset is multiples of 256 (the number of samples in a chirp with SF8). This is an expected behavior given that the signals have identical transmission power, leading to ambiguity in selecting FFT bin peaks. Overall, *FH-CSS* has significantly better performance than standard CSS in terms of resolving collided packets.

4 DISCUSSION AND SUMMARY

We proposed *FH-CSS* that applies frequency hopping to standard chirp spread spectrum (CSS). *FH-CSS* is based on the idea that a de-chirped signal spreads out when a chirp is multiplied with a non-counter part chirp. Preliminary results demonstrate that *FH-CSS* is able to decode collided packets better than standard CSS. There are still several challenges for real-time processing and reliability improvement, which we will explore as our future work.

REFERENCES

[1] Eric Blossom. 2004. GNU radio: tools for exploring the radio frequency spectrum. *Linux journal* 2004, 122 (2004), 4.

[2] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. 2021. Concurrent interference cancellation: Decoding multi-packet collisions in LoRa. In *Proceedings of the ACM SIGCOMM Conference*. 503–515.

[3] Joachim Tapparel, Orion Afisiadis, Paul Mayoraz, Alexios Balatsoukas-Stimming, and Andreas Burg. 2020. An open-source LoRa physical layer prototype on GNU radio. In *IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 1–5.

[4] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. 2023. Pcube: scaling lora concurrent transmissions with reception diversities. *ACM Transactions on Sensor Networks* 18, 4 (2023), 1–25.

[5] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: Parallel decoding for LoRa transmissions. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 192–204.

[6] Zhenqiang Xu, Shuai Tong, Pengjin Xie, and Jiliang Wang. 2020. FlipLoRa: Resolving collisions with up-down quasi-orthogonality. In *17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9.