This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2022.3178208, IEEE Communications Letters

LiCo: Toward Lightweight Resolution of Signal Collisions in LoRaWAN

Chenglong Shao, *Member, IEEE* and Osamu Muta, *Member, IEEE*

***Abstract*—This letter proposes a novel technique named LiCo to resolve LoRa signal collisions. Unlike existing approaches that introduce considerable computational overhead due to redundant LoRa demodulation operations, LiCo aims at a more lightweight way by simply performing one-shot demodulation regarding all colliding signals. For the separation of colliding signals, LiCo further analyzes the demodulation results in a comparative manner with no need to apply any complicated arithmetic. Theoretical and experimental results show that LiCo can reduce the computational overhead while achieving similar collision resolution performance as the state of the art.**

***Index Terms*—LPWANs, LoRaWAN, LoRa, collision resolution, chirp spread spectrum.**

I. Introduction

Low Power Wide Area Networks (LPWANs) are one of the most compelling and fastest growing wireless networking technologies for the Internet of Things (IoT). Among various LPWAN technologies, unlicensed LPWANs have gained more attention since the use of the license-free frequency bands (*e.g.,* 902-928 MHz) further facilitates their installation ease for arbitrary users. Specifically, Long Range Wide Area Network (LoRaWAN) represents a leading choice for unlicensed LPWAN-based IoT solutions. They are proved simple to use and quick to deploy in a number of IoT applications such as smart agriculture and asset tracking. Typically, a LoRaWAN is composed of four elements: end device, gateway, network server, and application server. The end devices are distributed to transmit sensor data to the servers with the gateway as a transparent bridge in between. The network server performs LoRaWAN management and the application server collects the data to enable related applications.

While this type of network architecture is workable, current LoRaWAN suffers from severe LoRa signal collisions when massive end devices coexist with each other. This is essentially due to the use of pure ALOHA protocol for wireless channel access. This protocol allows an end device to occupy a channel and start its own signal transmission as long as it has data to send. In this context, the past few years have witnessed the emergence of research efforts for LoRa signal collision resolution [1]–[8]. Despite their feasibility for the collision resolution, they introduce considerable computational overhead due to the redundant execution of complicated signal processing algorithms (*e.g.,* discrete Fourier transform (DFT)).

This work was supported in part by the Japan Society for the Promotion of Science Postdoctoral Fellowships for Research in Japan (No. 21F21074).

C. Shao and O. Muta are with the Center for Japan-Egypt Cooperation in Science and Technology, Kyushu University, Fukuoka, Japan (E-mail: shao@mobcom.ait.kyushu-u.ac.jp, muta@ait.kyushu-u.ac.jp).

To mitigate this drawback, this letter proposes LiCo, a more lightweight approach to resolve LoRa signal collisions. LiCo includes a one-shot demodulation procedure regarding all colliding signals. It then separates the colliding signals by simply comparing their demodulation results based on their magnitudes and frequencies. Our evaluation results show that LiCo avoids the complicated signal processing operations, reduces the computational overhead, and achieves similar collision resolution performance as the existing studies.

II. State of The Art

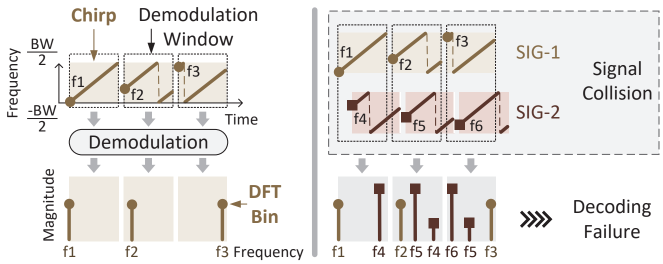
We provide a brief summary of several representative and noteworthy solutions for physical-layer collision resolution of LoRa signals [1]–[8]. As with LiCo, they do not need any modifications at LoRa transmitter and consider a single-antenna LoRa receiver. Choir exploits the transmitter-receiver frequency offsets for colliding signal separation via a larger-size (10$\times$) DFT operation than the standard one used in LoRa [1]. To directly distinguish the demodulation results of colliding signals, FTrack [2], SCLoRa [3], Pyramid [4], CIC [5], and AlignTrack [6] repeatedly perform DFT regarding multiple portions of each symbol in the colliding signals. In addition to the standard LoRa demodulation process, NScale further applies a customized demodulation scheme [7]. Based on the foreknown waveforms of LoRa signals, mLoRa presents a sample-wise signal recovery technique [8]. While these approaches can resolve LoRa signal collisions, they also incur considerable computational overhead due to the use of complicated signal processing algorithms. In comparison with them, LiCo is designed as a physical-layer solution to provide similar performance of collision resolution with much less computational overhead.

Unlike the physical-layer solutions above, researchers have also sought to design new medium access control (MAC)-layer protocols to avoid the occurrence of LoRa signal collisions. They focus on refining the pure ALOHA protocol [9], adopting Time-Division Multiple Access (TDMA) [10], or implementing Carrier-Sense Multiple Access (CSMA) [11] in LoRaWAN. However, since LiCo targets collision resolution in the physical layer, the detailed discussion of the MAC-layer approaches is beyond the scope of this work.

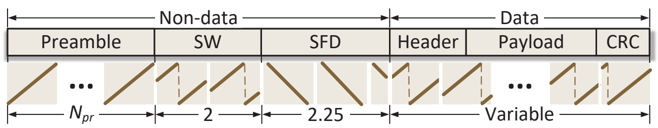
III. LoRa Primer

As depicted in the left subfigure of Fig. 1 (a), a LoRa signal is composed of multiple *chirps*. Say that the bandwidth of the LoRa signal is BW. Each chirp increases linearly in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2022.3178208, IEEE Communications Letters



(a)



(b)

Fig. 1. Overview of LoRa. (a) LoRa demodulation without (left) and with (right) signal collision. As an example, the power of SIG-2 is assumed to be larger than that of SIG-1. (b) LoRa packet format.

frequency domain. Chirp duration is determined by a parameter called spreading factor (SF) that indicates the number of data bits included in a chirp. A chirp encodes its carried data with its initial frequency (*e.g.*, $f1$–$f3$). To demodulate a chirp, LoRa leverages multiple demodulation windows each of which aligns with a chirp. Within each demodulation window, LoRa performs a multiplication between the received chirp and a predefined chirp. With down-sampling and DFT applied to the multiplication result, a clear DFT peak bin appears and its index corresponds to the data carried in the chirp. However, collision between LoRa signals may lead to data decoding failure. As illustrated in the right subfigure of Fig. 1 (a), this is because multiple DFT bins appear within each demodulation window and LoRa is unable to recognize their associated signals. Fig. 1 (b) shows the typical LoRa packet format. At the beginning, a preamble exists for signal detection with $N_{pr} = 8$ chirps included by default [12], [13]. The Sync Word (SW) (2 chirps) and Start Frame Delimiter (SFD) (2.25 chirps) are used for packet synchronization and frequency offset compensation, respectively. In LoRa uplink transmission under the explicit mode, the packet data portion consists of a Header, a payload, and a cyclic redundancy check (CRC) field.

IV. Design of LiCo

Generally, LiCo aims to effectively decode colliding signals while reducing the number of operations in comparison with the existing solutions.

A. LiCo Overview

Fig. 2 shows the basic building blocks of LiCo regarding a collision among three signals (SIG-1, SIG-2, and SIG-3) modulated with the same SF. Given a clear non-data portion of SIG-1, LiCo firstly performs demodulation window alignment accordingly[1]. It then harnesses the magnitude information

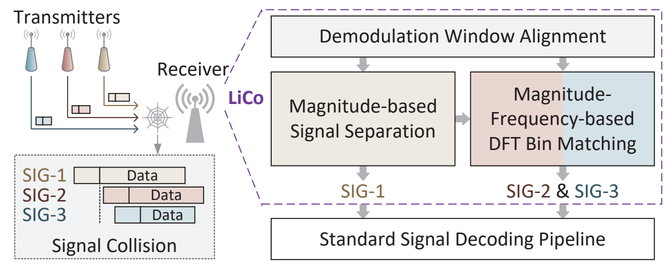


Fig. 2. Building blocks of LiCo against a three-signal collision.

of the demodulation results regarding the preamble of SIG-1 to separate SIG-1 from the collision. Afterwards, LiCo concurrently recognizes SIG-2 and SIG-3 by analyzing both the magnitude and frequency of their demodulation results. Note that LiCo does not need to know the number of colliding signals in advance. To resolve the collisions including more signals, LiCo performs similar operations by firstly handling SIG-1 and concurrently demodulating the others thereafter.

B. Demodulation Window Alignment

Instead of performing repeated LoRa demodulation operations regarding a single chirp as adopted in [2]–[7], LiCo resorts to one-shot demodulation to avoid the redundant use of the complicated DFT. Given that the preamble, SW, and SFD of SIG-1 are clear, LiCo achieves this by firstly applying the standard LoRa demodulation operation. After detecting the SW field, LiCo skips the following SFD field and sets successive demodulation windows thereafter. In this way, each chirp of SIG-1 is aligned with a demodulation window where the one-shot demodulation is applied to the three signals.

C. Magnitude-based Signal Separation

Regarding the collision case among SIG-1, SIG-2, and SIG-3, LiCo firstly seeks to separate SIG-1 from the collision and decode it thereafter. LiCo achieves this based on the fact that the preamble field of SIG-1 is not contaminated by the other signals. To be more specific, after detecting the preamble of SIG-1, LiCo further utilizes the magnitude information of the DFT bins that are generated by the demodulation operation regarding the preamble chirps. Since the chirps of a LoRa signal theoretically feature the same power, the magnitudes of the corresponding DFT bins are identical. Accordingly, as depicted in Fig. 3, LiCo can use the DFT bin magnitude (m3) corresponding to a preamble chirp of SIG-1 as a reference magnitude. It then searches for the data-portion DFT bins ((m3, f2), (m3, f3), (m3, f4)) whose magnitudes are equal to the reference one as the demodulation results of SIG-1.

In practice, however, it is hard to guarantee that the resulting DFT bins of a LoRa signal have exactly the same magnitude. This is because LoRa signals may experience varying degrees of signal sample distortions due to multipath effect, frequency selective fading, and so forth. As a result, LiCo may fail to simply use the reference magnitude of m3 to recognize the DFT bins of (m3, f2), (m3, f3), and (m3, f4) whose magnitudes may be slightly different from m3. Worse still, the DFT bin

[1] If the non-data portion of SIG-1 is not clear, we can address this case by detecting the preamble of SIG-1 based on correlation as adopted in [2]. Briefly, we generate a standard preamble and perform correlation operation regarding received signal samples. If a peak correlation result is observed, the detection of the preamble of SIG-1 becomes successful.

magnitudes corresponding to the preamble chirps of SIG-1 may also have tiny differences from each other, which makes it difficult to choose the reference magnitude.

As a simple add-on function to address the above challenge, our workaround is to firstly calculate the averaged magnitude of multiple preamble-related DFT bins as the reference one ($\overline{\text{m3}}$ in Fig. 3). Mathematically, the reference magnitude is calculated as $\frac{1}{N_{bin}}\Sigma_{i=N_{pr}-1}^{N_{pr}-N_{bin}} m_i$ where m_i is the bin magnitude and $N_{bin} \in [1, N_{pr}-1]$ is the number of used DFT bins. The calculation starts from the last DFT bin ($i = N_{pr} - 1$) in the preamble. The reason to set the maximum value of N_{bin} as $N_{pr} - 1$ is that the magnitude of the first preamble-related DFT bin may have a huge difference from those of the other $N_{pr} - 1$ DFT bins due to premature signal detection. Given a reference magnitude, LiCo targets the colliding chirps within each demodulation window. Among the resulting DFT bins, the one with the minimum magnitude difference from the reference is deemed associated with SIG-1. In this way, all the DFT bins of SIG-1 can be extracted from the collision.

D. Magnitude-Frequency-based DFT Bin Matching

After extracting the DFT bins of SIG-1, LiCo removes them from the collision and then embarks on distinguishing the remaining DFT bins of SIG-2 and SIG-3. Note that there is no need to align each chirp of SIG-2 and SIG-3 with a demodulation window. This is because the misalignment between a chirp and a demodulation window does not significantly change the frequency of the corresponding DFT bin. LiCo firstly observes the magnitudes of the remaining DFT bins. As shown in Fig. 3, when SIG-2 and SIG-3 interfere with each other, the demodulation result within each demodulation window contains four DFT bins with different magnitudes (m1, m2, m4, and m5). This is because SIG-2 and SIG-3 typically feature different signal strengths and distinct misalignment pattern with the demodulation window. Based on this fact, LiCo classifies these DFT bins into four bin groups according to their magnitudes. In this way, the magnitudes of the DFT bins (*e.g.,* (m1, f5), (m1, f6), and (m1, f7)) in a bin group are identical and differ from those of other bin groups. In addition, the DFT bins in a bin group are arranged in the order of their appearance in the time domain. Since the DFT bins in a bin group may not have exactly the same magnitude as explained in Section IV-C, the DFT bin grouping operation is performed in a comparative manner rather than based on the true values of the magnitudes. This is achieved by ranking the magnitudes of the resulting DFT bins in each demodulation window. Accordingly, the DFT bins whose magnitudes have the same rank are grouped together.

Based on the obtained bin groups, LiCo then aims to perform group matching to recognize the DFT bins of SIG-2 and SIG-3, respectively. In the collision case between SIG-2 and SIG-3 shown in Fig. 3, SIG-2 and SIG-3 have distinct signal duration. Accordingly, the numbers of their associated DFT bins are different from each other. LiCo can hence simply match the bin groups (*e.g.,* the groups of (m1, ·) and (m2, ·)) with the same number of DFT bins as a bin cluster. In this way, the collision between SIG-2 and SIG-3 is disentangled

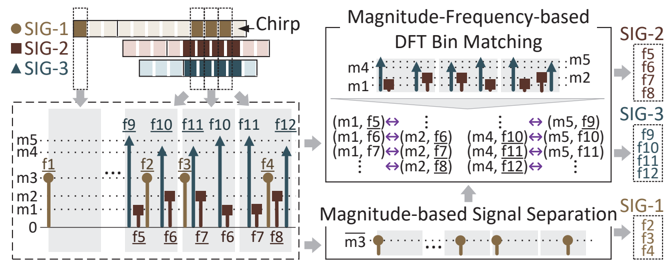


Fig. 3. Collision resolution regarding SIG-1, SIG-2, and SIG-3. Each DFT bin is denoted as (bin magnitude, bin frequency).

into two separated bin clusters. More importantly, when the two bin groups in a bin cluster are aligned, the two DFT bins (*e.g.,* (m1, f6) and (m2, f6) or (m4, f11) and (m5, f11)) with the same order in their associated bin groups have an identical frequency index since they belong to a single chirp. Therefore, LiCo simply neglects the DFT bins in a bin group and determines the DFT bin frequencies in another bin group to be the contained data in the corresponding signal.

V. Discussion

As with the existing solutions [1]–[8] that are invalid in some specific collision cases, LiCo also fails to correctly resolve signal collisions when i) no clear signal preamble exists, ii) two contiguous chirps of a signal contain identical data, iii) the DFT bins associated with different signals have the same magnitude, or iv) several signals in a collision have the same duration. Despite these special cases, LiCo provides an effective way to resolve typical signal collisions as targeted in the existing studies. Since the main goal of this work is to show that it is possible to resolve LoRa signal collisions in a more lightweight way, LiCo enhancement against these limitations will be considered in our future work.

VI. Performance Evaluation

We evaluate LiCo experimentally based on commodity LoRa end devices and software-defined radio (SDR). Specifically, 10 Dragino LoRa Shields installed on Arduino Uno and a USRP N210 SDR are used to mimic 10 transmitters and a receiver, respectively. This experimental setting is consistent with the ones adopted in the existing approaches that typically consider 2–12 concurrent transmitters [1]–[8]. The experiment is conducted in a typical laboratory environment that contains severer channel fading mainly caused by object blocking/reflection and human movement. To mimic a near-far situation, we distribute the 10 transmitters at arbitrary locations around the receiver with the transmitter-receiver distance ranging in $[1, 10]$ meters. Based on the pure ALOHA protocol, each transmitter sends random data with the average transmission interval of 1 second. Transmission power and payload length are randomized in the ranges of $[-3, 0]$ dBm and $[40, 50]$ Bytes, respectively. We target LoRa uplink channel 63 (914.9 MHz) with the bandwidth of 125 KHz. The default SF is 8. For performance comparison, the benchmark solutions are Choir [1], FTrack [2], SCLoRa [3], Pyramid [4], CIC [5], AlignTrack [6], NScale [7], and mLoRa [8].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2022.3178208, IEEE Communications Letters

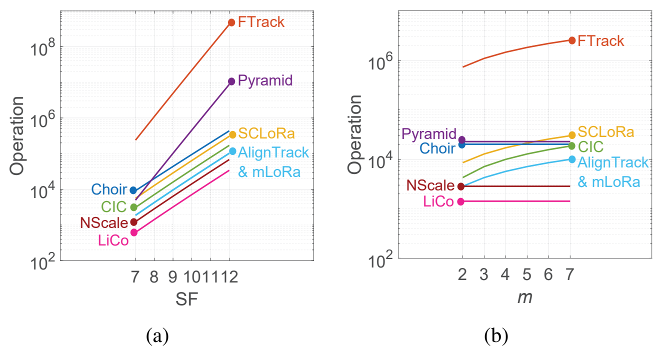


Fig. 4. Computational overhead of each solution under different settings of (a) SF and (b) m. The y-axes are in log scale.

A. Computational Overhead

Since the benchmark studies (except FTrack [2] and SCLoRa [3]) do not focus on the lightweightness for signal collision resolution, they do not provide any experimental results related to the computational overhead. While computational time is measured in FTrack and SCLoRa, it highly depends on the used experiment hardware, which makes it unfair to use the provided computational overhead results for the comparison with LiCo. Thus, we compare the computational overhead of the solutions in a theoretical manner.

Particularly, we focus on the dominant computational overhead that comes from the DFT operation within each demodulation window as explained in [2]–[4]. Since the compared solutions target different LoRa collision cases and hence include distinct case-specific operations (*e.g.,* signal preamble detection) additionally, it is clearer and fairer to derive the computational overhead only based on the common DFT operation adopted in all the solutions. Accordingly, given a collision among m signals whose every chirp contains $n = 2^{\mathrm{SF}}$ signal samples, the computational overhead of each solution is: Choir [1]: $10n\mathrm{log}10n$; FTrack [2]: $mn^2\mathrm{log}n$; SCLoRa [3]: $3mn\mathrm{log}n$; Pyramid [4]: $(n^2\mathrm{log}n)/16$; CIC [5]: $(2m-1)n\mathrm{log}n$; AlignTrack [6]: $mn\mathrm{log}n$; NScale [7]: $2n\mathrm{log}n$; mLoRa [8]: $mn\mathrm{log}n$; LiCo: $n\mathrm{log}n$. Given $m = 3$ for example, Fig. 4 (a) illustrates the computational overhead of each solution under different settings of SF. We figure out that LiCo needs less operations for collision resolution even than NScale, the most lightweight solution among all the benchmark ones. This results from the one-shot demodulation operation regarding all the colliding signals. More importantly, Fig. 4 (b) shows that in comparison with FTrack, SCLoRa, CIC, AlignTrack, and mLoRa, the amount of reduced computational overhead via LiCo increases as the collision includes more signals. As with Choir, Pyramid, and NScale, LiCo is not restricted to the number (m) of signals in a collision in principle. Note that m used here is for our theoretical analysis of the computational overhead and not related to our experiment settings.

B. Collision Resolution Capability

We further assess LiCo from the perspective of resolving individual signals from collisions. The targeted metrics are chirp error rate (CER) and throughput. CER indicates the ratio between incorrectly decoded chirps and the total number of received chirps. Throughput is defined as the average number of correctly decoded chirps per second. We collect received signals for 30 minutes. Regarding the four limitations of LiCo presented in Section V, we find that i) 95% of the received collisions contain a clear non-data portion, ii) the case of identical data being modulated in two contiguous chirps does not occur, iii) the DFT bin magnitudes of different signals in a collision are distinguishable, and iv) only 4% of the received collisions include signals with the same duration. We also observe different overlapping patterns among colliding signals. These observations may change as more transmitters are used, which can lead to performance degradation of LiCo. Moreover, LiCo may fail to resolve the collisions where signal strength varies over time. Note that LiCo is triggered only when a signal collision is detected. This is achieved by adopting the standard LoRa demodulation operation. If multiple DFT bins are observed within several demodulation windows, a collision is recognized and LiCo is applied accordingly.

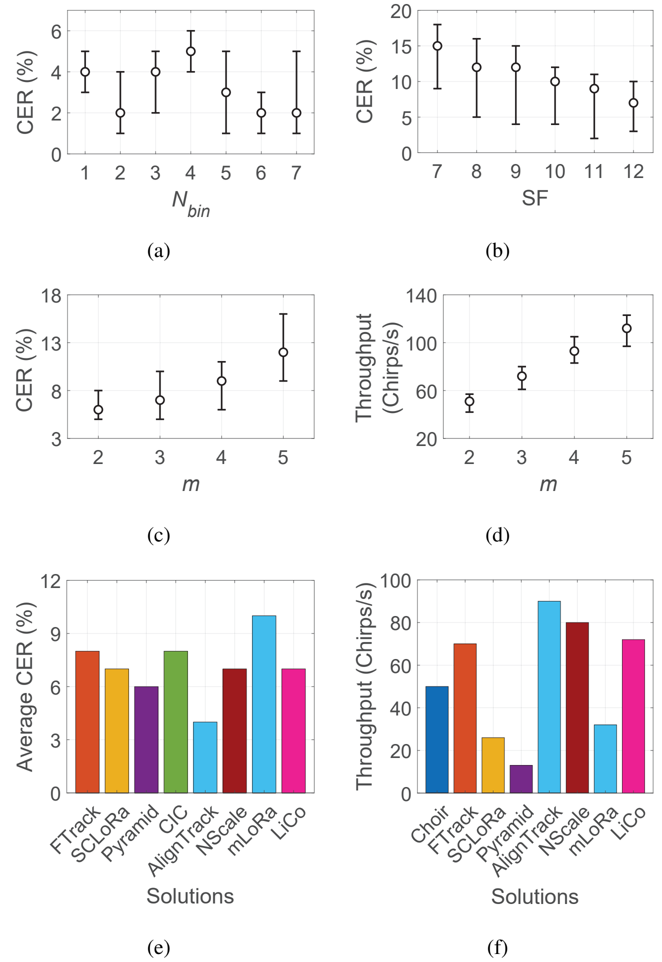


Fig. 5. Collision resolution performance. (a) CER of SIG-1 with varying N_{bin} regarding each collision. (b) CER in each collision with varying SF. (c) CER in each collision containing m signals. (d) Throughput regarding each collision containing m signals. (e) Average CER via different solutions. (f) Overall throughput via different solutions.

Firstly, we need to determine the value of N_{bin} used to separate the signal of SIG-1 as described in Section IV-C.

Fig. 5 (a) shows the CER of SIG-1 with varying N_{bin}. Given $N_{pr} = 8$, N_{bin} is in the range of $[1, 7]$. We figure out that the setting of N_{bin} does not significantly affect the CER of SIG-1. In this context, we choose $N_{bin} = 1$ for the following evaluation. Under different settings of SF, Fig. 5 (b) gives the overall CER regarding each received collision. We observe that adopting a larger SF slightly reduces the CER. This is because a LoRa signal modulated with a a larger SF has a higher signal-to-noise ratio (SNR). Note that these chirp errors mainly result from multipath fading and delay spread in wireless channel.

Under our experiment setting, we find that the SNR of the received signals ranges in $[-10, 5]$ dB and that the number (m) of signals in a collision is up to 5. Note that this SNR range is smaller than that ($[-15, 20]$ dB) in the existing solutions [1]–[8]. This indicates a more challenging experimental setting since the colliding signals in our experiment do not significantly differ from each other in terms of their signal strength. Figs. 5 (c) and (d) show the impact of m on the CER and the throughput regarding each received collision, respectively. We can see that the CER increases generally as a collision contains more signals. This accords with our expectation that it becomes harder for LiCo to distinguish more DFT bins based on their magnitudes. While not shown in Fig. 5 (c), we also find that the CER regarding collision-free signals is between 1% and 7%, which is similar to the CER regarding colliding signals. Despite more erroneous chirps given a larger m, the throughput grows when more signals collide. This is because the total number of received chirps increases significantly though more chirp errors are observed. Note that in our experiment, 12% colliding chirps cannot be successfully resolved by LiCo due to several reasons such as identical signal duration and similar signal strength.

In addition to the LiCo-oriented evaluation above, we also compare LiCo with our benchmark solutions. For fair comparison, we focus on the collisions with a certain number (m) of signals included. Given $m = 3$ for example, Figs. 5 (e) and (f) provide the average CER and the achieved throughput, respectively. The benchmark results are obtained directly from the corresponding studies. Note that Choir [1] and CIC [5] are not evaluated in terms of the average CER and the throughput, respectively. From Fig. 5 (e), we can see that AlignTrack [6] achieves the minimum CER and that LiCo introduces only a slightly larger CER than AlignTrack. Similarly, the achieved throughputs via these solutions are of the same order of magnitude as illustrated in Fig. 5 (f). While the throughputs via AlignTrack and NScale [7] are slightly higher than that via LiCo, this mainly originates from the differences of the experimental settings (*e.g.,* signal transmission interval, collision status, SF, SNR, and experimental environment) in these solutions. It does not mean that their core techniques significantly outperform LiCo. Together with the reduced computational overhead via LiCo shown in Fig. 4 which is the focal point of this work, this proves the feasibility and superiority of LiCo for LoRa collision resolution.

VII. Conclusion

This letter has proposed LiCo to resolve LoRa signal collisions in a more lightweight way than the existing solutions. LiCo performs signal demodulation only once regarding all colliding signals and separate them without any complicated arithmetic. The evaluation results show that in comparison with the state of the art, LiCo achieves similar collision resolution performance with much less computational overhead.

References

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

[2] X. Xia, Y. Zheng, and T. Gu, “FTrack: Parallel decoding for LoRa transmissions,” *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, pp. 2573–2586, Dec. 2020.

[3] B. Hu, Z. Yin, S. Wang, S. Wang, Z. Xu, and T. He, “SCLoRa: Leveraging multi-dimensionality in decoding collided LoRa transmissions,” in *Proc. of IEEE ICNP*, 2020.

[4] Z. Xu, P. Xie, and J. Wang, “Pyramid: Real-time LoRa collision decoding with peak tracking,” in *Proc. of IEEE INFOCOM*, 2021.

[5] M. O. Shahid, M. Philipose, K. Chintalapudi, S. Banerjee, and B. Krishnaswamy, “Concurrent interference cancellation: Decoding multi-packet collisions in LoRa,” in *Proc. of ACM SIGCOMM*, 2021.

[6] Q. Chen and J. Wang, “AlignTrack: Push the limit of LoRa collision decoding,” in *Proc. of IEEE ICNP*, 2021.

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

[8] X. Wang, L. Kong, L. He, and G. Chen, “mLoRa: A multi-packet reception protocol in LoRa networks,” in *Proc. of IEEE ICNP*, 2019.

[9] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, “Slotted ALOHA on LoRaWAN-design, analysis, and deployment,” *Sensors*, vol. 19, no. 4, 838, pp. 1–19, Feb. 2019.

[10] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, “TS-LoRa: Time-slotted LoRaWAN for the industrial internet of things,” *Comput. Commun.*, vol. 153, pp. 1–10, Mar. 2020.

[11] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, “LMAC: Efficient carrier-sense multiple access for LoRa,” in *Proc. of ACM MobiCom*, 2020.

[12] Semtech SX1261/62 LoRa Transceiver. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-core/sx1262.

[13] Semtech SX1272/73 LoRa Transceiver. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-core/sx1272.