# Integrating LoRa Collision Decoding and MAC Protocols for Enabling IoT Massive Connectivity

Weixuan XIAO [1], Megumi Kaneko [1], Nancy El Rachkidy [1], and Alexandre Guitton [2]

[1]Affiliation not available
[2]UCA - LIMOS CNRS and CITI Lyon - INRIA

October 30, 2023

# Integrating LoRa Collision Decoding and MAC Protocols for Enabling IoT Massive Connectivity

Weixuan Xiao*, Megumi Kaneko† *IEEE Senior Member*, Nancy El Rachkidy*, Alexandre Guitton*‡

* Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France.

† National Institute of Informatics, 101-8430, Tokyo, Japan

‡ Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France.

Emails: †megkaneko@nii.ac.jp,

*{weixuan.xiao,nancy.el_rachkidy,alexandre.guitton}@uca.fr

**Abstract**

One major goal of Beyond 5G and 6G networks is to provide connectivity for a massive number of Internet-of-Things (IoT) devices. Towards that goal, Long Range (LoRa) is a promising physical layer technology which features low data-rates and large communication ranges, while requiring only low power. However, as the number of devices increases, more and more collisions occur, hence severely degrading LoRa system performances. To cope with this critical drawback, several LoRa collision decoding algorithms and MAC protocols have been proposed. The purpose of this article is to present how collision decoding algorithms interact with MAC layer protocols, and to discuss the potential of such integrated approaches. To do so, we first classify the collision decoding algorithms according to their principles and distinctive features, and compare some reference algorithms in a single simulation setup, using a Software Defined Radio (SDR) hardware. Then, we analyze how each class of MAC protocols can benefit from each category of collision decoding algorithms. Finally, we discuss long-term perspectives and open issues in this active research area.

**Index Terms**

LoRa, Packet Collisions, Interferences, MAC protocols, Massive Connectivity, IoT, Beyond 5G

# I. Introduction

Along with the relentless growth of current Internet-of-Things (IoT) technologies, ever more diversified and specialized IoT use cases are expected in the near future. Faced by the exponential surge of IoT mobile data traffic, Beyond 5G (B5G) and 6G networks have the daunting task of providing seamless and ubiquitous coverage to billions of users and devices, yet in a sustainable manner.

Long Range (LoRa) has been regarded as one of the most prominent technologies to cope with these issues. LoRa is a physical layer for Low Power Wide Area Networks (LPWANs), and is most suited for IoT applications that only need sporadic, limited data rate and delay-tolerant transmissions, but require very wide coverage, up to tens of kilometers in rural scenarios. LoRa is envisioned to become a key enabler of some essential IoT services for supporting future society, among which smart environments and energies, smart cities, smart homes, etc., as depicted in Fig. 1. Despite the ability of LoRa to trade-off data rate and coverage by means of its modulation based on Chirp Spread Spectrum (CSS) with various Spreading Factors (SFs), LoRa as such is unable to cope with the B5G joint demands of massive connectivity, long range and higher data rates. Additionally, current LoRa systems are based on LoRaWAN, a simplistic MAC layer that barely handles packet collisions. Current LoRa systems are thus unable to face the aforementioned scalability issues.

One essential aspect is hence the design of LoRa-specific collision decoding algorithms. Various approaches for LoRa collision decoding at gateways have been proposed so far [1], an example of which is illustrated in Fig. 1. These techniques may improve LoRa performance whenever the node density increases, or when each node generates higher traffic. Despite their improved performance, current methods are limited to the decoding of only few concurrent packets, and are still insufficient to cope with the imminent surge of IoT data traffic, as envisioned in future IoT applications such as massive 3D sensing. More performant yet cost-efficient LoRa-specific collision decoding methods are crucially needed.

Another essential aspect is the design of suitable MAC protocols which can perform well with various densities of devices and traffic patterns. While some MAC protocols aim at avoiding collisions, dense scenarios will always produce them. It becomes primordial to integrate collision decoding algorithms into MAC protocols. Yet, very few works in the literature have addressed this issue so far.

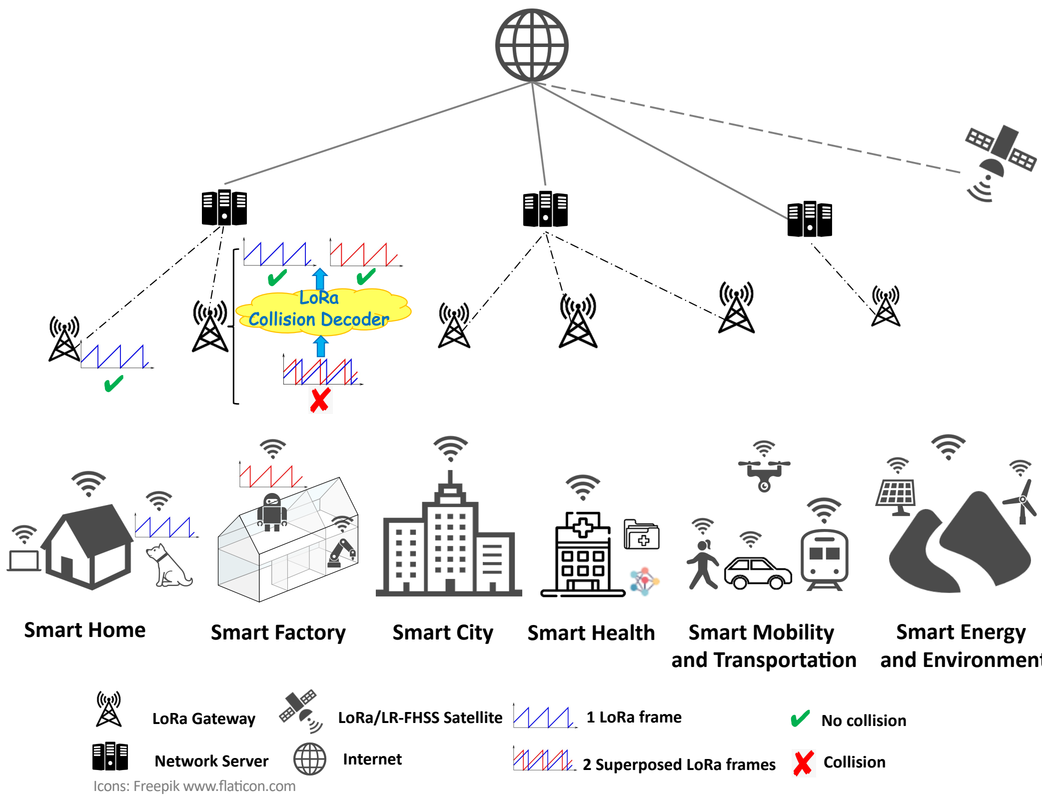


Figure 1. Examples of use cases supported by LoRa and generating billions of IoT data packets. A blue LoRa frame is sent by a connected dog and a red frame by a connected robot. The leftmost gateway receives the blue frame without collision, while both frames collide at the neighboring gateway. By means of LoRa collision decoding algorithms, all frames in collision may be received successfully and be forwarded to the network server, and in turn to the application server.

Therefore, the purpose of this article is to clarify how LoRa collision algorithms can be integrated with MAC protocols. In particular, we classify existing collision decoding algorithms into categories according to their decoding principles and usage conditions. For each category, we discuss the integration with MAC protocols. Then, we unveil some key open research issues and future perspectives. The main contributions of this overview article are summarized as follows.

1) We first recall the basic knowledge and fundamentals about LoRa physical layer technology, with the main LoRa parameters.
2) We categorize the major existing approaches for decoding collided LoRa signals, and explain the key principles and differences of each technique.
3) We experimentally compare the performances of the most representative collision decoding algorithms, through simulations based on Software Defined Radio (SDR) hardware implementation.

4) We classify the major scheduling and MAC layer protocols amenable to LoRa, and give promising insights about integrating collision decoding algorithms into MAC schedulers and protocols.
5) We finally discuss crucial open research issues, and describe the future challenges pertaining to the expansion towards more complex network topologies, energy efficiency and mobility issues, as well as satellite-terrestrial communications.

## II. Fundamentals of LoRa and LoRaWAN

LoRa leverages a CSS modulation designed to yield large communication ranges, of about 5 km in urban areas and 20 km in rural areas. The modulation is based on chirps, also called symbols, which are linear frequency sweeps of a given bandwidth. Upchirps correspond to increasing frequencies, while downchirps correspond to decreasing frequencies. Data is encoded into chirps by changing the initial frequency. A receiver correctly synchronized with the transmitter listens during the whole chirp, multiplies the received upchirp (resp. downchirp) by a downchirp (resp. upchirp) of value 0, performs a Fast Fourier Transform (FFT), and identifies the FFT bin with the highest received power, which is equal to the transmitted value.

LoRa uplink frames are composed of a preamble, an optional header, and a payload. The preamble consists of a series of 8 upchirps of value 0 used for synchronization, a network identification encoded with 2 upchirps, and a start of frame delimiter of 2.25 downchirps. The header is encoded as 8 upchirps and contains metadata including the payload length. Finally, the payload contains the data encoded using upchirps.

The LoRa modulation is designed to be robust. It includes several coding techniques: the data is whitened, encoded with a Hamming code based on a parameter called Coding Rate (CR), interleaved in order to spread bits over several symbols, and indexed with a Gray code. LoRa also uses the SF parameter to trade-off bitrate with communication range: a higher SF enables a lower receiver sensitivity threshold, inducing an increased communication range, but producing longer symbols, thereby reducing the bitrate.

Signals sent on different SFs are quasi-orthogonal. Nevertheless, this imperfect SF quasi-orthogonality may significantly deteriorate the overall network performance as experimentally verified in various works and analyzed in [2]. LoRa also exhibits the capture effect for transmissions using the same SF, which allows a strong signal to be received even if a weaker interference signal is present, provided that the difference of power is at least 6 dB.

However, the level of orthogonality (orthogonality of channels, quasi-orthogonality for different SFs, capture effect for same SFs) that can be reached with LoRa is not sufficient to accommodate the upcoming surge of IoT data traffic. In dense networks, collisions are likely to occur on all channels and for all SFs, and conditions for capture effect might not hold. Such collisions reduce network throughput, increase frame delays, and increase overall energy consumption. In order to lessen all these negative impacts, collision decoding approaches will be primordial.

LoRaWAN (Long Range Wide Area Network) is the most common MAC protocol based on LoRa. In the LoRaWAN topology, end-devices send their frames to gateways using LoRa, and gateways communicate to a network server (and eventually, to an application server) via IP. LoRaWAN is based on the ALOHA medium access mechanism: an end-device can transmit its data as soon as it is available. After each transmission, an end-device opens two short receive windows in order to receive potential data from the network server through gateways. Such data includes acknowledgments, which are sent in response to confirmed frames. Confirmed frames that are not acknowledged are retransmitted.

Being based on ALOHA, LoRaWAN does not work well under heavy load. Indeed, large traffic causes many collisions, which drastically reduces the performance of the protocol. Several other MAC protocols have been proposed in order to deal with a heavy load.

## III. Collision Decoding at the PHY layer

Decoding collided LoRa signals is a key asset for increasing LoRa scalability, thus several mechanisms have been proposed so far to perform this decoding. First, we classify the main collision decoding mechanisms according to their major features. Then, we compare several mechanisms within the same environment.

### *A. Classification of the algorithms*

All collision decoding mechanisms require the identification of the features of colliding signals in order to separate them. Such features include power difference, time (including frequency and phase) difference, or modulation difference. We propose to classify the main mechanisms according to the features they use, as shown in Table I. Note that it is possible to use several features.

| Name | Reference | Power-based | | Time-based | | Modulation-based |
|---|---|---|---|---|---|---|
| | | SIC | Power-difference | Sub-FFT-bin | Multi-FFT-bins | |
| CHOIR | [3] | | | ◯ | | |
| CIC | [4] | | △ | | ◯ | |
| CoLoRa | [5] | ◯ | | | ◯ | |
| FlipLoRa | [6] | | | | | ◯ |
| FTrack | [7] | | | | ◯ | |
| SF-DS | [8] | | | | ◯ | |

Table I

CATEGORIZATION OF SEVERAL LORA COLLISION DECODING MECHANISMS ACCORDING TO THE MAIN SIGNAL FEATURE THEY USE. ◯ STANDS FOR PRIMARY USE, AND △ STANDS FOR SECONDARY USE.

*1) Power-based approach:* This approach uses the difference of received power in order to separate colliding signals. Such approaches include Successive Interference Cancellation (SIC), or reception power tracking.

SIC is a well-established technique, applied in various wireless systems. With SIC, the receiver first extracts and demodulates the strongest signal, removes it from the superposed signals, and applies itself iteratively in order to demodulate the next strongest signal. SIC usually requires that the strongest signal is about 3 dB larger than the other signals. Its main drawback is that it runs several passes over the whole signal, thereby requiring large memory storage, and is prone to decoding error propagation. Apart from CoLoRa, few collision decoding algorithms in the literature actually use SIC.

Reception power tracking focuses on identifying the amplitude of all FFT peaks across several symbols, and mapping the peaks of similar amplitude to the same transmitter. Figure 2 shows the superposition of a strong signal (in red) with a weaker signal (in blue). When performing FFT, the strong signal produces high peaks, while the weak signal produces low peaks. The demodulation of both signals can be performed by computing the two peaks at each symbol, and by mapping them to the correct transmitter based on their amplitude. The computational complexity of this approach is much lower than for SIC, and the required difference of power is typically lower. Most power-based algorithms, such as CIC, use reception power tracking.

*2) Time-based approach:* This approach uses the delay between the colliding signals in order to separate them.

Sub-FFT-bin desynchronization originates from tiny frequency drifts due to hardware imper-

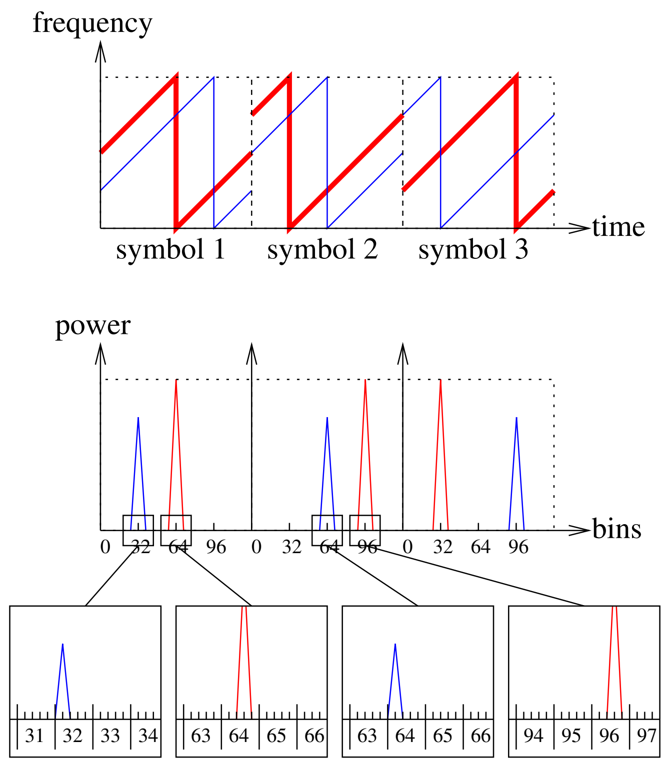


Figure 2. There are two ways to separate the two signals. (1) The stronger signal (in red) produces high FFT peaks, while the weaker signal (in blue) produces low FFT peaks. Thus, the two signals can be identified and demodulated. (2) It is possible to map symbols to transmitters using a very small-scale desynchronization which can be seen on the sub-FFT-bin level. In this example, the symbols of the blue frame are always sent in the sub-FFT-bin number 1/5 (e.g., values of 32.2 or 64.2), while the symbols of the red frame are always sent in the sub-FFT-bin number 3/5 (e.g., values of 64.6 or 96.6).

fection of transceivers. These drifts are typically small enough to let the corresponding symbol values still be placed in the correct FFT bins. However, it is possible to use the fractional part of the FFT bins in order to map symbols to transmitters. This is achieved through an oversampling with typically ten times the number of samples. Figure 2 shows an example with two superposed frames from two transmitters. By performing an FFT (here with five times more samples), the desynchronization of signals can be observed at the sub-FFT-bin level. Indeed, in the example, symbols of the blue frame always have a fractional value of 0.2, while the symbols of the red frame always have a fractional value of 0.6. The conditions of application of this approach are: the hardware offset needs to be constant over the whole frame duration, and it needs to be separable from the time offsets of the other colliding transmitters. To the best of our knowledge, CHOIR is the only algorithm using this approach.

Multi-FFT-bins desynchronization enables disentangling superposed symbols when the desynchronization is sufficient. It uses the fact that a reception window synchronized with one frame overlaps with two symbols of desynchronized frames. In this case, one high peak is produced for the synchronized frame, and two small peaks are produced for each desynchronized frame. Figure 3(a) shows the superposition of two frames, where the receiver is synchronized with the red frame, and the blue frame is delayed by 1/4-th of a symbol. During the first symbol, the values of the blue symbols change: this results in a high FFT peak for the red symbol, a small peak for the end of the blue symbol, and a medium peak for the beginning of the next blue symbol. During the second symbol, the values of the blue symbols do not change, which results in two high peaks. To determine which peak is the red peak, the algorithms often use peaks from previous or future symbols. The conditions of application of this approach are: the delay between symbols should be between 10% and 90% of a symbol duration, consecutive symbols should have a different value, and superposed symbols of different frames should not have close values in order to avoid inter-symbol interferences causing issues in the detection of individual values. Most algorithms of the literature use this approach, such as CIC, CoLoRa, FTrack and SF-DS.

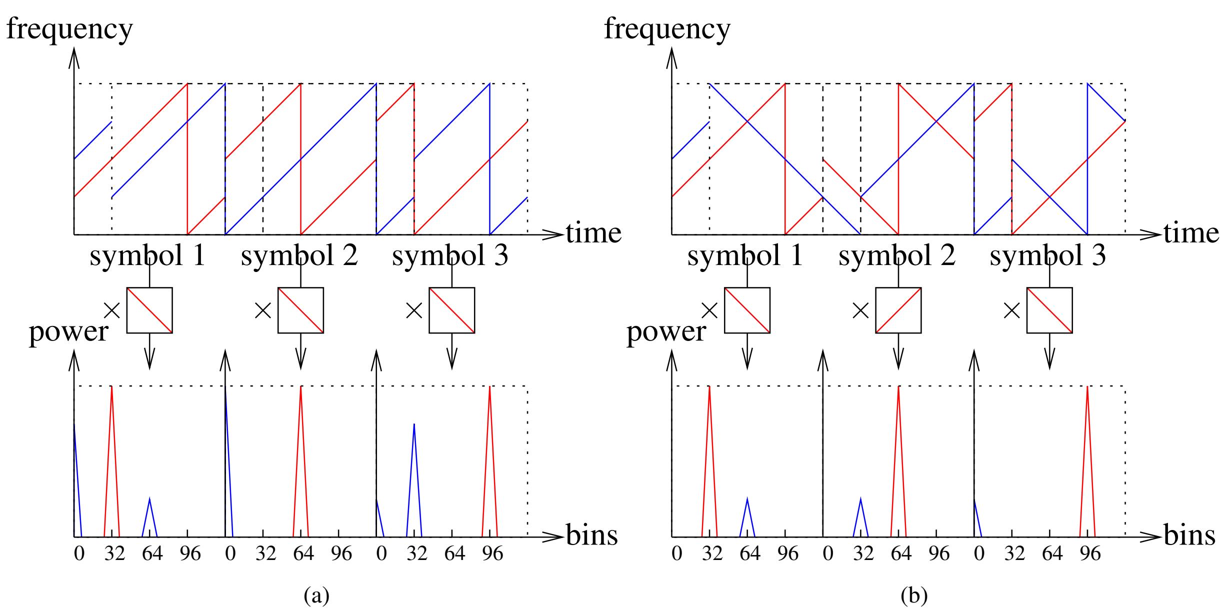


Figure 3. **(a)** Time-based approach, with multi-FFT-bins desynchronization: when a receiver is synchronized on the red frame and receives a superposition of slightly desynchronized signals, it identifies FFT peaks of various sizes. The peaks corresponding to the red frame are among the highest peaks. However, the correct peak is not always simple to identify, as there might be other high peaks (see symbol 2). **(b)** Modulation-based approach, using the orthogonality of upchirps and downchirps: when two frames with alternating upchirps and downchirps are slightly desynchronized, the receiver synchronized on a frame is able to detect the correct peaks easily.

*3) Modulation-based approach:* This approach uses the fact that upchirps and downchirps are quasi-orthogonal. By alternating upchirps and downchirps when modulating the payload of a frame, the probability that two colliding frames cannot be decoded is reduced. Figure 3(b) shows the superposition of a red frame and a blue frame, both alternating upchirps and downchirps, with the receiver synchronized on the red frame, and the blue frame delayed by 1/4-th of a symbol. For the first red symbol, which is assumed by the receiver to be an upchirp, the receiver performs a multiplication with a downchirp and extracts two FFT peaks: a high peak corresponding to the red upchirp symbol, and a small peak corresponding to the partial upchirp of the blue frame. For the second red symbol, the receiver performs the multiplication with an upchirp and extracts two FFT peaks: the high peak for the red symbol, and a small peak for the part of the blue downchirp. The conditions of application of this approach are: the time difference between frames should be such that the overlap between the upchirp (resp. downchirp) symbols of different frames is as small as possible, and that simultaneous downlink communications are limited as they become non-orthogonal to uplink communications (unlike with the LoRa modulation). To the best of our knowledge, FlipLoRa [6] is the only protocol using this feature.

### B. Experimentations

We set up an experimentation environment based on GNU Radio to evaluate the performance of representative collision decoding algorithms, under high SNR (20 dB) and low SNR (-5 dB) scenarios. In each experimentation, 2 to 10 nodes are uniformly deployed around a gateway. The devices periodically send packets with a random interval between 1000 and 2200 ms. The packet contains a random payload of 22 bytes. When encoding with CR=1, each frame contains 38 payload symbols for SF8 and 28 payload symbols for SF12. With a 125 kHz bandwidth, the time-on-air is 100 ms for SF8 and 1320 ms for SF12. To simulate heavy traffic and maximize the number of collisions, the nodes are configured to use the same channel and the same SF.

To show the theoretical performance of collision decoding algorithms on the PHY layer, we directly apply the core algorithms to the superposed payload signals without the preamble detection (instead, we give the necessary information of the frames), in order to alleviate the impact of preamble detection. Figure 4 shows the achieved network throughput. Note that the Max plot gives the ideal throughput if all frames are correctly decoded.

With SF8, the network is not saturated. Although some frames are lost due to collisions, the network throughput increases with the number of nodes. In the high SNR scenario, all collision

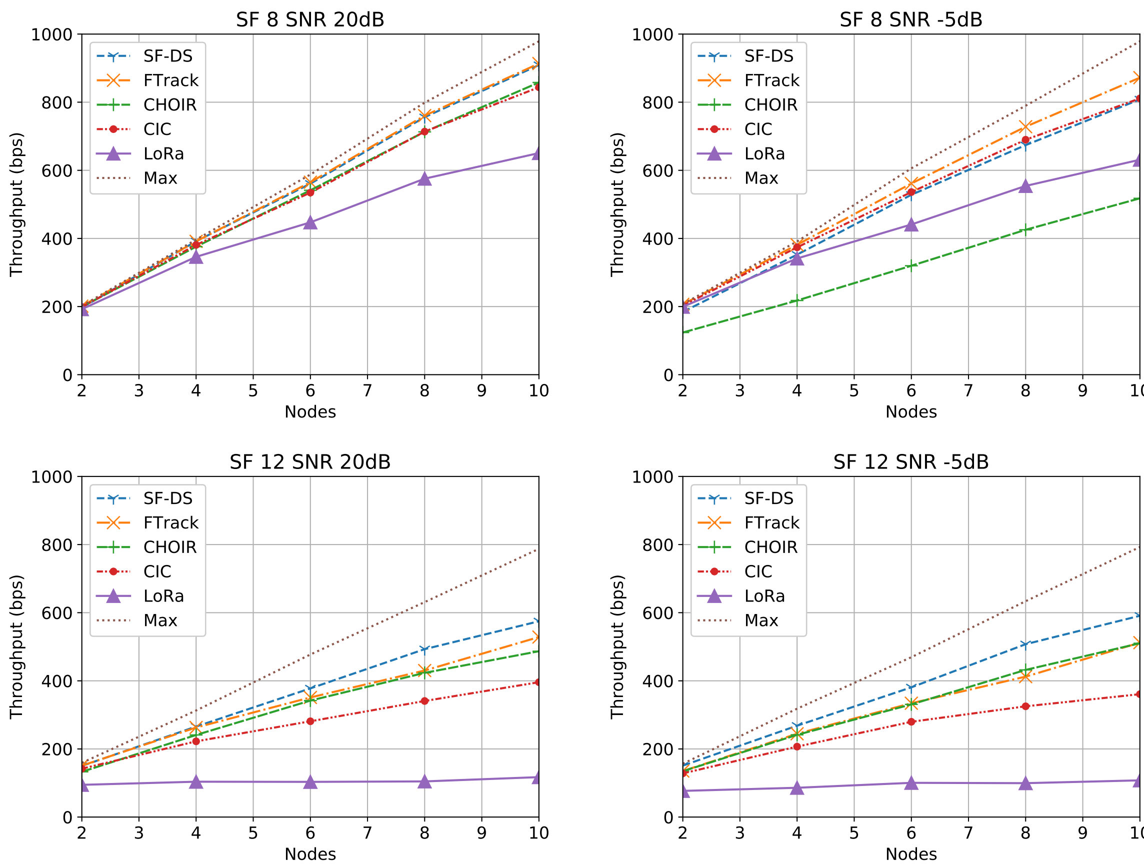


Figure 4. Throughput of decoding algorithm under different SNRs. The sending interval saturates the network with SF12. The SNR also changes the relative performance among the algorithms.

decoding algorithms outperform basic LoRa. However, in the low SNR case, CHOIR performs worse than LoRa, because the tiny frequency offsets are buried in noise and become invisible to the gateway. Under low SNR, SF-DS slightly degrades as the noise may shift the FFT bins due to incomplete samples and padding in each FFT: indeed, all protocols performing FFTs that are shorter than the symbol length suffer from spectrum leakage, thereby reducing the accuracy of the FFT peaks' detection.

With SF12, LoRa yields a relatively constant throughput as the number of nodes grows. This is because it captures at most one frame at a time. By contrast, other algorithms do not exhibit a large throughput decrease compared to SF8. Indeed, SF12 symbols are much longer (typically 16 times) than SF8 symbols, thereby increasing the power levels on FFT bins. This helps differentiating the LoRa peaks from noise. Interestingly, SF-DS achieves the best performance for SF12 under high and low SNR scenarios.

## IV. Collision avoidance at the MAC layer

Before discussing the integration issues of LoRa collision decoding algorithms and MAC protocols, we first classify existing scheduling and MAC approaches tailored to LoRa-based networks.

### A. Classification

*1) Centralized scheduling optimization:* The problem of centralized LoRa resource allocation optimization has been investigated in a number of works in the literature, among which those surveyed in [9]. The basic LoRa radio resources consist of channels, SFs, transmit power and codes. The global approach is to formulate a mathematical optimization problem in terms of maximizing sum-rate, fairness, or energy efficiency objectives, given constraints such as power budget, maximum number of allocated devices per SF, minimum rate requirements, etc. The problem is solved centrally at the NS for single or multiple gateways. Although this approach enables a global optimization of the overall LoRa network, many works assume ideal conditions that are not always suitable for LoRa use cases, such as (1) assumption of perfect and instantaneous Channel State Information (CSI) knowledge for all device-gateway channels at the NS, (2) no overhead costs for signalling the scheduling decisions to each device, (3) perfect synchronization of uplink LoRa frames.

To cope with some of these drawbacks, the proposed SF and transmit power allocation approach in [10] only requires prior information of the large-scale channel fading (path loss), and discusses a MAC layer and its induced overheads under a realistic LoRa framework. Moreover, this method achieves a three-fold increase in the number of supported devices, while improving energy efficiency and fairness, by allocating simultaneous devices transmissions to each SF. This is realized by optimizing LoRa resource allocation based on the proposed mathematical modeling of the target system, where the impairments of co-SF and inter-SF interferences are fully taken into account.

*2) Distributed access protocols:* LoRaWAN is based on the ALOHA access method, which is suitable when the traffic load is very low. In ALOHA-based protocols, devices can transmit frames as soon as data is available. ALOHA access methods are very simple to implement, but do not scale with the number of nodes. Among them, LoRaWAN is the fundamental protocol for

LoRa. However, as the demand for capacity increases, the ALOHA method quickly saturates. CSMA-CA-based protocols aim to avoid signal collisions. To do so, the devices sense the channel before sending a frame: if the channel is detected busy, the transmission is postponed. If the channel sensing is imprecise, collisions will occur as with ALOHA (that is, in the absence of channel sensing), hence most CSMA-CA-based methods also secondarily rely on ALOHA. LMAC is an example of such protocols. TDMA-based protocols also aim to avoid signal collisions by computing a schedule usually designed by a central entity. The schedule ensures that no collisions occur, or defines synchronized slots (and possibly channel sensing at the beginning of a slot), in order to reduce collisions. In both cases, collisions are reduced at the cost of delay. Time Slotted LoRa and all the centralized scheduling algorithms are examples of TDMA-based protocols.

Overall, the main MAC protocols for LoRa can be classified as in Table II. Note that it is possible for a protocol to use several features.

| **Name** | **Reference** | ALOHA-based | CSMA-CA-based | TDMA-based |
|---|---|---|---|---|
| Centralized scheduling | [9] [10] | | | ◯ |
| LMAC | [11] | △ | ◯ | |
| LoRaWAN | [12] | ◯ | | |
| Time Slotted LoRa | [13] | | | ◯ |

Table II

CATEGORIZATION OF SEVERAL PROTOCOLS THAT AIM TO AVOID OR BENEFIT FROM COLLISIONS. ◯ STANDS FOR PRIMARY USE, AND △ STANDS FOR SECONDARY USE.

### *B. Integration*

*1) Centralized scheduling optimization and mathematical modeling of collision decoding effects:* Although optimization methods such as [10] considered the effects of co-SF and inter-SF frame collisions, they have assumed the capture of at most one frame in case of collision. However, the presented collision decoding algorithms enable the simultaneous decoding of multiple packets, which directly impacts the performance of the centralized schedulers built on top of them. Hence, the overall LoRa network performance could be boosted by fully integrating the PHY layer collision decoding algorithms in the resource allocation optimization. For that, viable mathematical modeling of the PHY layer decoding process should be conceived for each type of decoding algorithm, for instance, an analytical rate expression given the statistical collisions among $n$ packets, each with a specific SF. In addition to the potential performance enhancements, this analysis would enable to better predict the achievable LoRa network performances, and to make the best integration choice among the different PHY decoding and allocation methods, depending on the application-specific targets.

*2) Distributed access protocols and opportunities for collision decoding integration:* Distributed access protocols have varying potential for collision decoding algorithms, depending on their category.

ALOHA-based protocols typically generate many collisions, especially under heavy load, as they do not attempt to avoid collisions. Due to the random nature of their transmissions, they are a good basis for collision decoding algorithms, especially those based on multi-FFT-bins time desynchronization or on orthogonal modulations. However, the generated collisions do not necessarily meet the conditions for collision decoding, such as a minimum delay between the symbol frontiers of colliding frames.

CSMA-CA-based protocols aim to avoid collisions based on a channel sensing procedure. Thus, their performance improvement by most collision decoding algorithms may be rather limited. Still, they can benefit from sub-FFT-bins time desynchronization, as the channel sensing procedure does not avoid tiny desynchronizations. Power-based decoding can still be beneficial in case of simultaneous collisions, provided that the channel sensing threshold is carefully set.

TDMA-based protocols are typically collision-free. However, by changing the way the schedule is performed, it is possible to force collisions with the right properties, in terms of power, time or modulation. When these protocols are based on time slots, the slot size can also be adapted to increase the probability of specific types of collisions, namely multi-FFT-bins collisions. Indeed, by setting a time slot equal to the minimum required delay to decode overlapping frames, collisions with the correct delay are likely to occur. Overall, TDMA-based MAC protocols are very good candidates for collision decoding algorithms.

These discussions disclose promising directions to pursue, for realizing an efficient and scalable integration of PHY layer collision decoding and MAC layer protocols for LoRa. Next, we generalize this discussion by identifying further long-term perspectives.

## V. Open Research Directions and Perspectives

### A. Extension to Multi-Gateway Deployment

Although LoRa-based networks are expected to contribute to ultra-massive device connectivity, there is major hurdle imposed by the fundamental constraint on the number of available demodulators at each gateway. Indeed, gateway chips integrate a limited number of demodulators, each capable of demodulating a single frame at a time, but are unable to demodulate too many concurrent LoRa signals (typically, eight). Increasing the number of demodulators in a chip is possible but increases hardware complexity and costs. This constraint imposes a drastic limitation on the number of frames that can be demodulated in parallel, and consequently, on LoRa scalability. Nevertheless, this problem had been largely overlooked in the literature. In [14], we have proposed the new approach of recursive reuse of demodulators, enabling to jointly improve throughput and fairness even under high device density scenarios.

Another possible solution is to integrate multi-gateway features in the previously presented decoding algorithms, which are mono-gateway by nature. To ensure efficient use of demodulators among gateways, new decoding algorithms should ensure that redundant packets are demodulated by a single gateway and discarded by the others prior to demodulation, possibly by having gateways exchange control messages about the detected preambles. The overhead of such protocols should be minimized in order to limit the induced latency and control signaling costs. There are surprisingly very few works investigating these promising open research directions.

### B. Effect of Mobility

Most LoRa networks so far have been designed for static or low mobility IoT devices, as the main applications were monitoring of buildings or environments. However, to realize future applications pertaining to vehicular or aerial IoT applications, developing efficient mobile IoT networks has become inevitable. LoRa represents a fundamental solution, as its CSS modulation offers inherent robustness to Doppler spreads. Among the presented decoding algorithms, those exploiting timing offsets to map symbols to devices are thus promising to cope with device mobility. However, one important requirement is a constant timing offset for each device over the whole frame. This assumption does not hold if the signal path difference undergoes a significant variation during the frame duration. Therefore, a key direction is the design of new collision decoding methods combined with mobility-aware scheduling and LoRa access approaches. Given that different SFs are impacted differently by a given device speed, such SF-dependent effects should be fully taken into account by the decoding and the SF allocation procedures.

### *C. Energy Efficiency*

Although LoRa is one of the main PHY layer technologies enabling LPWANs, it still induces a significant amount of energy consumption at gateways. Indeed, current protocols require gateways to listen to devices constantly, in order to intercept possible communications. This can be particularly problematic for outdoor LoRa network deployments in isolated or extreme conditions, including forests, deserts or volcanoes, where even the gateways are remote. Therefore, there is a need for effective and energy-efficient decoding protocols that minimize the computational complexity and energy burden at gateways.

Future LoRa networks for outdoor monitoring should also integrate energy-harvesting gateways, for which new collision decoding protocols should be optimized, jointly with the gateway energy harvesting protocol. That is, the decoding capabilities of a gateway could be a function of its fluctuating harvested energy level. One direction to pursue would be to coordinate uplink transmissions through an energy-efficient scheduler, possibly with topology control, such that fewer collisions occur whenever the gateway has low energy supplies.

On the other side, to support severely battery-limited IoT devices or passive LoRa tags within close-range of the gateway, Wireless Power Transfer (WPT) technology could be exploited and integrated into LoRa-specific MAC protocols. For instance, the power transfer could be controlled by means of, e.g., power beamforming, such that the subsequent uplink transmissions entail collision patterns that facilitate decoding at gateways.

### *D. Satellite IoT Networks*

Finally, another highly promising research direction is that of Satellite IoT networks, a topic that has sparked worldwide interests lately with the advent of Low Earth Orbit (LEO) satellite constellations, boosting terrestrial-satellite communications. Many studies have discussed the integration of Narrowband IoT (NB-IoT) into LEO satellite systems, with the goals of expanding the communication range, mitigating coverage holes, or providing ubiquitous IoT services. Along this line, a new modulation coined as Long-Range Frequency Hopping Spread Spectrum (LR-FHSS) has recently emerged and has been included in LoRaWAN. This modulation is designed to tackle the low scalability inherent to LoRaWAN. As detailed in [15], the robustness and scalability offered by LR-FHSS stem from its mechanism of replicated header transmissions, fragmented and redundant payload transmissions, and pseudo-random frequency hopping over a very large number of subcarriers with high granularity. These features render this modulation particularly amenable to support long-range, large-scale and high-capacity satellite IoT networks. Even though this modulation enables much lower collision probabilities as compared to LoRaWAN, its network capacity is still severely limited given the millions of concurrent transmissions that would occur, owing to the vastness of satellite coverage zones. Furthermore, the scalability improvements of LR-FHSS over LoRa come at the cost of much-reduced data rate per device. Hence, one of the major challenges is to enhance the device-wise QoS satisfaction levels required by each IoT application, while achieving the tremendous coverage and scalability increase offered by LR-FHSS. Towards this end, the LoRa collision decoding and access protocol methods discussed so far should be extended by including collisions among LR-FHSS packets, as well as between LoRa and LR-FHSS packets. Additionally, methods for jointly optimizing SF and frequency hopping sequences so as to minimize collisions and interferences can also bring considerable benefits, not only in large-scale, but also at device-scale.

## VI. Conclusion

This article has focused on the promising solutions for enhancing LoRa system performances, aiming at IoT massive connectivity in B5G. Firstly, the most representative LoRa collision decoding algorithms have been presented and classified according to their features. Their performances have been experimentally evaluated through simulations based on real hardware. After presenting and classifying the major scheduling and MAC protocols tailored to LoRa, we have discussed the overall framework for the awaited integration of collision decoding and MAC scheduling. Finally, key technological challenges have been identified, and compelling future research avenues have been unfolded.

## References

[1] C. Shao, O. Muta, W. Wang, and W. Lee, “Toward ubiquitous connectivity via LoRaWAN: An overview of signal collision resolving solutions,” *IEEE Internet of Things*, Dec. 2021.

[2] A. Waret, M. Kaneko, A. Guitton, and N. El Rachkidy, “LoRa Throughput Analysis with Imperfect Spreading Factor Orthogonality,” *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 408–411, Apr. 2019.

[3] R. Eletreby, D. Zhang, S. Kumar, and O. Yagan, "Empowering low-power wide area networks in urban settings," in *ACM SIGCOMM*, Aug. 2017.

[4] M. O. Shahid, M. Philipose, K. Chintalapudi, S. Banerjee, and B. Krishnaswamy, "Concurrent interference cancellation: Decoding multi-packet collisions in LoRa," in *ACM SIGCOMM*, 2021, pp. 503–515.

[5] S. Tong, Z. Xu, and J. Wang, "CoLoRa: Enabling multi-packet reception in LoRa," in *INFOCOM (IEEE Conference on Computer Communications)*, 2020, pp. 2303–2311.

[6] Z. Xu, S. Tong, P. Xie, and J. Wang, "FlipLoRa: Resolving collisions with up-down quasi-orthogonality," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2020, pp. 1–9. DOI: 10.1109/SECON48991.2020.9158432.

[7] X. Xia, Y. Zheng, and T. Gu, "FTrack: Parallel Decoding for LoRa Transmissions," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2573–2586, 2020. DOI: 10.1109/TNET.2020.3018020.

[8] W. Xiao, N. El Rachkidy, and A. Guitton, "SF-DS: A slot-free decoding scheme for collided LoRa transmissions," in *IEEE VTC Spring (Vehicular Technology Conference)*, 2022.

[9] P. Gkotsiopoulos, D. Zorbas and C. Douligeris, "Performance Determinants in LoRa Networks: a Literature Review," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1721–1758, 2021.

[10] L. Amichi, M. Kaneko, E. H. Fukuda, N. El Rachkidy and A. Guitton, "Joint Allocation Strategies of Power and Spreading Factors with Imperfect Orthogonality in LoRa Networks," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3750–3765, Jun. 2020.

[11] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for LoRa," in *ACM MoBiCom*, https://doi.org/10.1145/3372224.3419200, 2020, pp. 1–13.

[12] Semtech Corporation, "LoRaWAN Specification v1.1," Semtech, Tech. Rep. Revision B, 2017. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf.

[13] D. Zorbas and X. Fafoutis, "Time-slotted LoRa networks: Design considerations, implementations, and perspectives," *IEEE Internet of Things Magazine*, vol. 4, no. 1, pp. 84–89, 2021.

[14] A. Guitton and M. Kaneko, "Improving LoRa Scalability by a Recursive Reuse of Demodulators," *IEEE Globecom*, pp. 1–6, Dec. 2020.

[15] G. Boquet, P. Tuset-Peiro, F. Adelantado, T. Watteyne and X. Vilajosana, "LR-FHSS: Overview and Performance Analysis," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 30–36, 2021.