



HAL
open science

Enhancing LR-FHSS Scalability Through Advanced Sequence Design and Demodulator Allocation

Diego Maldonado, Megumi Kaneko, Juan Fraire, Alexandre Guitton, Oana Iova,
Hervé Rivano

► To cite this version:

Diego Maldonado, Megumi Kaneko, Juan Fraire, Alexandre Guitton, Oana Iova, et al.. Enhancing LR-FHSS Scalability Through Advanced Sequence Design and Demodulator Allocation. IEEE Transactions on Green Communications and Networking, 2025, <10.1109/TGCN.2025.3570185>. <hal-05087729>

HAL Id: hal-05087729

<https://inria.hal.science/hal-05087729v1>

Submitted on 28 May 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Enhancing LR-FHSS Scalability Through Advanced Sequence Design and Demodulator Allocation

Diego Maldonado, Megumi Kaneko, *Senior Member, IEEE*, Juan A. Fraire, *Senior Member, IEEE*, Alexandre Guitton, Oana Iova, Hervé Rivano

Abstract—The accelerating growth of the Internet of Things (IoT) and its integration with Low-Earth Orbit (LEO) satellites demand efficient, reliable, and scalable communication protocols. Among these, the Long-Range Frequency Hopping Spread Spectrum (LR-FHSS) modulation, tailored for LEO satellite IoT communications, sparks keen interest. This work presents a joint approach to enhancing the scalability of LR-FHSS, addressing the demand for massive connectivity. We deepen into Frequency Hopping Sequence (FHS) mechanisms within LR-FHSS, spotlighting the potential of leveraging Wide-Gap sequences. Concurrently, we introduce two novel demodulator allocation strategies, namely, “Early-Decode” and “Early-Drop,” to optimize the utilization of LoRa-specific gateway decoding resources. Our research validates these findings through extensive simulations, providing a detailed analysis of the scalability potential of LR-FHSS in IoT environments. The results demonstrate that the proposed strategies can boost decoding capacity by up to 50% in certain configurations.

Index Terms—LR-FHSS, Frequency Hopping Spread Spectrum, Hamming Correlation, Optimal Sequences

I. INTRODUCTION

A. Background

LOW-Earth Orbit (LEO) satellites have become instrumental in extending the coverage of various Internet of Things (IoT) applications, where terrestrial infrastructures are unsuitable due to the challenging geographical locations [1].

Manuscript received 30 January 2024; revised 20 August 2024 and 26 February 2025; accepted 07 May 2025. Date of publication xxx; date of current version 09 May 2025. This research has received support from the NII-INRIA MoU Grant funded by NII, the Grant-in-Aid for Scientific Research Kakenhi 22KK0156 from the Ministry of Education, Science, Sports, and Culture of Japan, the JST ASPIRE grant for Top Teams number JPMJAP2325, the EU’s H2020 R&D program under the Marie Skłodowska-Curie grant agreement No 101008233 (MISSION project) and the French National Research Agency (ANR) projects ANR-22-CE25-0014-01 and ANR-21-CE25-0002-01.

D. Maldonado is with INSA Lyon, Inria, CITI, UR3720, 69621 Villeurbanne, France, (diego.maldonado-munoz@inria.fr).

M. Kaneko is with the Information Systems Architecture Science Research Division, National Institute of Informatics (NII), Chiyoda 101-8430, Japan, and with the Dept. of Computer Science, The University of Tokyo, Tokyo 113-8654, Japan (megkaneko@nii.ac.jp).

J. A. Fraire is with Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France, also with the Facultad de Ciencias Exactas Físicas y Naturales (FCEFN), Argentinian Research Council (CONICET), Córdoba 5000, Argentina, and the Dependable Systems and Software, Saarland University, 66123 Saarbrücken, Germany. (juan.fraire@inria.fr).

A. Guitton is with Université Clermont-Auvergne, CNRS, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France. (alexandre.guitton@uca.fr).

O. Iova is with INSA Lyon, Inria, CITI, UR3720, 69621 Villeurbanne, France, (oana.iova@insa-lyon.fr).

H. Rivano is with INSA Lyon, Inria, CITI, UR3720, 69621 Villeurbanne, France, (herve.rivano@insa-lyon.fr).

These include the environmental and ecological observation of oceans, poles, expansive forests, and extensive natural parks [2]. In these cases, LEO satellites are more cost-effective than geostationary satellites mainly because of their closer proximity to Earth, reducing transmission delays and power requirements. As the number of IoT devices surges, the demand for scalable and reliable communication protocols that address terrestrial and direct-to-space access (a.k.a. Direct-to-Satellite IoT [3], see Fig. 1) increases. Satellite IoT is a more environmentally friendly option for global connectivity. By positioning the application within an IoT framework (characterized by low power and low data rate), we enhance its green credentials [4]. However, achieving this requires making IoT scalable, which is the focus of this paper. One solution that has gained substantial attention is Long-Range (LoRa) modulation, which stands out for its low power consumption and expansive coverage. LoRa has been successfully ported to the satellite context with increasing interest from the academic and industrial communities (e.g., FossaSat, LacunaSat). However, LoRa is too bandwidth-consuming, and it is challenging for a receiving device located on a satellite to decode many incoming LoRa frames. These scalability issues hinder the use of LoRa communications from Earth to satellites [5], as the coverage area of a satellite is typically extensive.

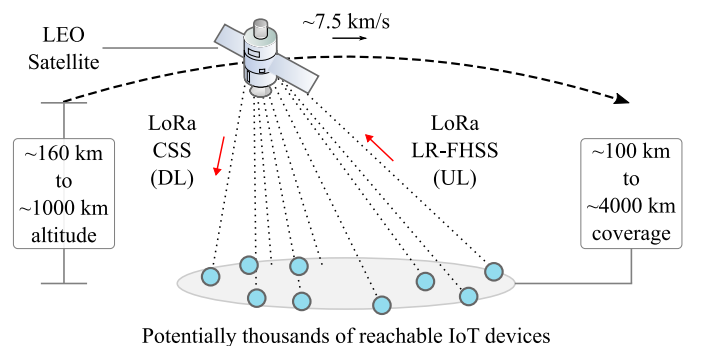


Fig. 1. Direct-to-Satellite IoT Scenario.

To address the scalability issue while maintaining the high link budget required for direct-to-space communications, Semtech introduced a new modulation technique called Long-Range Frequency Hopping Spread Spectrum (LR-FHSS) [6]. The main advantage of LR-FHSS is that it can evade self-interference by using many narrow-band channels and a frequency hopping technique based on a Frequency Hopping Sequence (FHS). This is done while maintaining a link budget similar to LoRa’s most robust settings. Moreover, most recent

LoRa devices are compatible with the LR-FHSS modulation, streamlining LR-FHSS adoption and integration with terrestrial IoT infrastructures¹. Semtech showed LR-FHSS modulation greatly increases LoRa’s scalability [8].

Although the introduction of LR-FHSS is relatively recent, it has swiftly garnered attention from the research community in satellite communications. Areas of focus include performance evaluation [9], [10], outage probabilities [11], [12], headerless recovery [13], and transceiver design [14]. These works assume that the FHS is random or defined by the standard. Yet, two pivotal topics that could jointly extend LR-FHSS scalability await further research and refinement: the sequence determining the frequency hop pattern and the concurrent methodology for allocating demodulator processors to decode multiple incoming frames in parallel.

B. Related Work

a) Sequence Design: FHS techniques are employed to enhance the security and reliability of wireless transmissions [15]. These sequences quickly alternate the carrier frequency during data transmission, neutralizing interference and jamming threats [16]. This is achieved by dispersing the transmitted frame over an extensive frequency range, which makes it notably challenging for potential adversaries to intercept or interrupt the ongoing communication [17]. The introduction of Wide-Gap FHS (WGFHS) is a recent advancement in this domain. This innovative sequence ensures a guaranteed minimum gap between successive frequency hops and boasts optimal attributes in terms of Hamming correlation [18]. What makes WGFHS particularly noteworthy is its compliance with standard requirements set by regional regulators for maintaining gaps between adjacent frequency channels. This is precisely the case of LR-FHSS. This adheres to regulations and fortifies the anti-jamming capabilities of FHS systems [19]–[21]. Despite the promising potential of this new FHS construction in the context of LR-FHSS modulation, to the authors’ best knowledge, it remains an under-explored topic in existing literature as most of it only mentions the FHS sequences used in LR-FHSS modulation is just a randomly generated sequence by using a hash function [10], [11], [12] or do not even mention its construction techniques [9], [14]. Moreover, recently published papers mention that FHS sequences for LR-FHSS are randomly generated [22]. This further highlights our detailed description of the FHS construction mechanisms used in LR-FHSS devices as an important contribution that has not yet been discussed in the relevant literature.

b) Demodulator Allocation: Existing LoRaWAN gateways, primarily powered by Semtech’s SX1301 chip [23], are equipped to concurrently demodulate several LoRa signals as long as they operate on unique channels or different Spreading Factors (SF). This SX1301 chip, a standard in most LoRaWAN gateways, houses eight distinct demodulator chips. It continuously scans for preambles across all channels and SFs, and upon detection, the integrated packet arbiter decides which

of the available demodulators should demodulate the corresponding frame. The limited number of demodulators can be a potential bottleneck, impairing the network’s overall performance [24]–[26]. Addressing this limitation, authors in [27], [28] proposed better algorithms to harness these demodulators in individual gateway configurations. The precise workings of the SX1301’s packet arbiter for LoRa remain undocumented, and the architecture of LR-FHSS-capable chips remains unknown to the community. Indeed, LR-FHSS will likely rely on a more advanced chipset designed explicitly for demodulating FHS signals. The capabilities of these newer chipsets are still unclear, but estimates suggest that accommodating up to 1000 simultaneous demodulation might be plausible [8]. To the authors’ knowledge, no demodulator allocation technique for LR-FHSS has been proposed thus far.

C. Contributions

This paper contributes by expanding LR-FHSS scalability in two dimensions.

On the one hand, we propose improving the FHS construction to enhance device-to-gateway extraction rates in view of massive connectivity. In particular, we propose leveraging Wide-Gap sequences’ existing properties while making them applicable to LR-FHSS requirements. This property emerges as a superior candidate, outclassing current LR-FHSS driver implementations in specific scenarios. To the best of our knowledge, this is the first work to conduct an in-depth study of the design of LR-FHSS sequences and further exploit Wide-Gap sequences in the context of LR-FHSS modulation. In particular, we investigate the parameter design and settings of Wide-Gap sequences, so as to make them fit practical LR-FHSS requirements.

On the other hand, we introduce two pioneering strategies for demodulator allocation designed to amplify the scalability of LR-FHSS-based networks further. At the heart of these strategies is the efficient use of redundancies inherent in the payload’s convolutional encoding and decoding processes. The first, termed “Early-Decode,” frees up demodulators once they’ve gathered sufficient convolutional blocks for frame decoding. The second, “Early-Drop,” proactively dismisses a packet if a certain number of its blocks fail, liberating the processor for subsequent demodulation.

The detailed contributions made in this paper are listed as follows.

- 1) **Description of the FHS used in the driver of LR-FHSS Devices:** Based on an analysis of the source code of the driver, we describe the FHS implemented in LR-FHSS devices and show that it is based on m -sequences, and not on pseudo-random sequences nor hash functions, unlike what was stated by previous research [9]–[12], [22].
- 2) **The proposed unified framework combines novel Wide-Gap FHS designs and innovative demodulator allocation strategies,** significantly enhancing LR-FHSS scalability as detailed below:
 - a) **Novel Wide-Gap FHS construction tailored to LR-FHSS:** We propose to leverage the Wide-Gap FHS concept, and design an alternative FHS

¹As of the current writing, LR-FHSS modulation is supported in devices featuring Semtech’s SX1261, SX1262, and LR1110 chipsets, and is compatible with V2.1 gateways employing the Semtech SX1301 chips [7]

construction optimized for the specifics of LR-FHSS modulation, thus enhancing its scalability. To the best of our knowledge, the proposal of a new sequence design is the first in the literature. This contribution is backed up by theoretical research on wide-gap sequences, which better fit to the LR-FHSS constraints than m -sequences.

- b) **Design of demodulator allocation strategies, tailored to LR-FHSS:** We propose two novel approaches, termed as the Early-Decode and Early-Drop methods. Tailored to amplify the capacity of LR-FHSS demodulators, they are designed to maximize the attainable scalability. Note that to the best of our knowledge, no research paper has currently focused on the computation complexity for the demodulation of LR-FHSS frames.
- 3) **Simulation-Driven Validation:** We evaluate the proposed techniques through an extensive simulation campaign, quantifying the practical benefits and potential implications for LR-FHSS systems. We show that our proposed FHS yields better performance than the FHS currently used in the driver of LR-FHSS devices.
- 4) We propose an **open-source tool, available at [29], to generate the m -sequences used by the LR-FHSS drivers, as well as our wide-gap sequences.** This tool will help researchers evaluate, compare, and improve the choice of FHS for LR-FHSS.

The remainder of the paper is organized as follows. Section II presents the system model, the LR-FHSS modulation, and the state-of-the-art design of FHS. Section IV delineates our proposed methods, encompassing FHS family construction and demodulator optimization. Section V presents and discusses our simulation results. Section VI draws Conclusions and avenues for future research.

II. LR-FHSS MODULATION AND FHS OVERVIEW

A. Basics of LR-FHSS

LR-FHSS is a modulation technique that leverages many narrow-band channels to perform rapid frequency hopping within a single frame. Primarily tailored for uplink communications, LR-FHSS proves particularly efficient when many energy-limited devices transmit simultaneously to LEO satellites, thanks to the large number of narrow-band channels LR-FHSS uses.

LR-FHSS segments the frequency band into Occupied Channel Width (OCW) channels. Each OCW is further partitioned into multiple narrow-band channels called Occupied Bandwidth (OBW) [10]. As illustrated in Fig. 2, an individual OBW acts as a physical subcarrier with a bandwidth set at 488 Hz. Within each OBW, LR-FHSS utilizes a Gaussian Minimum-Shift Keying (GMSK) modulation, prized for its exceptional spectral and energy efficiency [30]. To align with regional regulations stipulating the minimum spacing between subcarriers in the FHS, OBWs are systematically arranged into grids. Grids comprise non-sequential OBWs spaced apart by a mandated minimum distance, whose value depends on regional regulatory standards (e.g., 3.9 kHz in the EU). For instance,

in the EU, Grid 1 contains OBWs 1, 9, 17, ..., while Grid 2 contains OBWs 2, 10, 18, ... These grids play a pivotal role in LR-FHSS: a hopping sequence within any given frame is restricted to using OBWs within the same grid. A summary of the LR-FHSS parameters for chosen channels, including Data Rate (DR) modes and Frequency Hopping (FH) channels, together with details on the OCW channel bandwidth, OBW channel count, and the number of grids specified for LR-FHSS, are provided in [8]. Note that the large number of OBW channels (between 280 and 3120) is the core of the scalability of LR-FHSS.

Frame transmission initiates with multiple header copies, called replicas, each lasting 233 ms in airtime. Depending on the chosen data rate scheme, the header is replicated two or three times to ensure reliable reception. Note that the reception of a single replica is needed to retrieve the header. Then, replicas are followed by the segmented payload, where each segment, called a fragment, lasts 102.4 ms in airtime. These replicas and fragments are relayed over varying OBWs within a single grid, according to the FHS. Every header replica conveys the index of the chosen FHS. The diversity of sequences, typically 512, combined with the large number of OBWs, reduces the possibility of multiple frames adopting an identical hopping sequence. Payload encoding is designed to keep data integrity intact even with substantial fragment loss [10]. The LoRaWAN standard prescribes two LR-FHSS configurations:

- **Robust Configuration:** Headers are transmitted three times, and the payload is encoded at CR = 1/3 (denoted as CR1 in this work), meaning that successful frame recovery demands a minimum of 33% fragment decoding.
- **Fast Configuration:** Headers are transmitted twice, and the payload is encoded at CR = 2/3 (denoted as CR2 in this work), meaning that successful frame recovery demands a minimum of 67% fragment decoding.

A crucial element impacting the scalability of LR-FHSS is the FHS, which fundamentally determines how LR-FHSS frames with different sequences overlap and, subsequently, the extraction rate of colliding LR-FHSS frames. This relationship encompasses the intricate interplay between LR-FHSS's error correction mechanisms and the mandatory frequency shifts dictated by grid definitions. However, to our knowledge, there is a lack of studies on the design of appropriate FHS techniques for LR-FHSS, such as those in the Wide-Gap FHS (WGFHS) family.

B. Frequency Hopping Sequences

The FHS defines the sequence and timing of frequency shifts during a transmission. Often, these sequences are generated from mathematical algorithms or pseudo-random number generators. The key feature of these sequences is their unpredictability and non-repetitive nature, ensuring that potential adversaries find it challenging to intercept or jam the transmission [31]. Moreover, FHS-based modulations exhibit significant resilience to intense interference scenarios, typical in satellite communications, due to the expansive areas they cover on the Earth's surface.

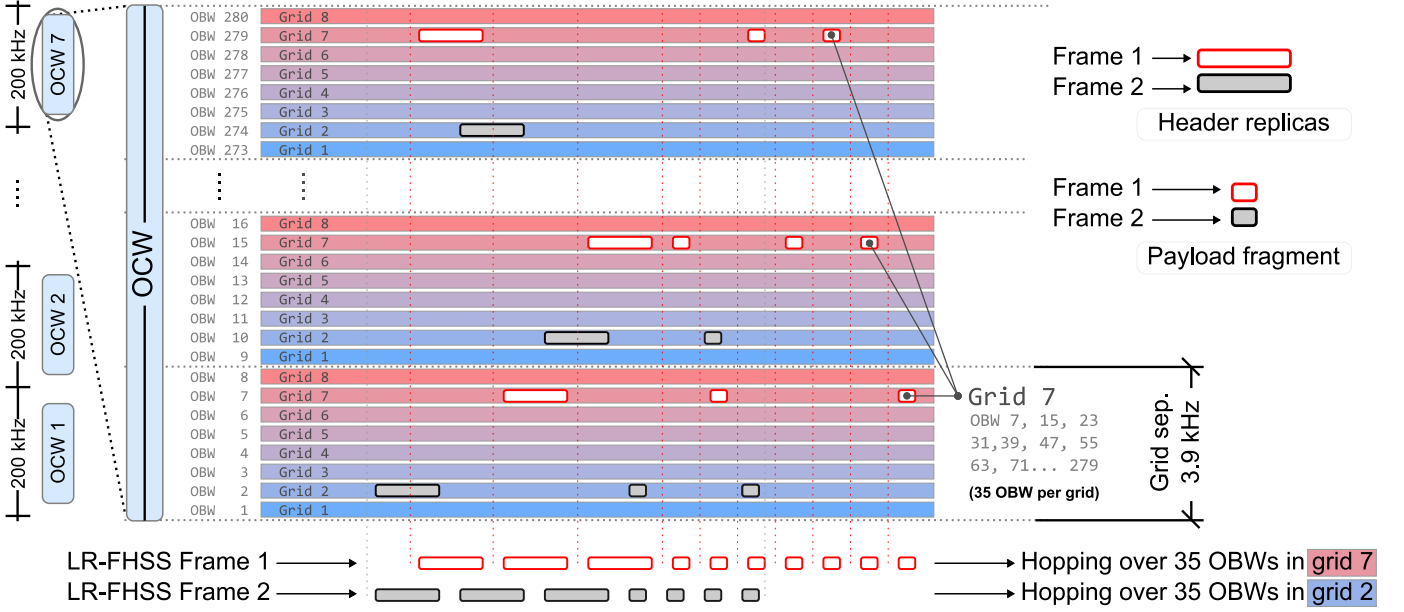


Fig. 2. Division into OCWs, OBWs, and grids for 1 OCW in the European 868 MHz frequency (simplification from [13])

In the following, the set of accessible frequency channels in an FHSS system is denoted $\mathcal{F} = \{f_0, f_1, \dots, f_{\ell-1}\}$. A FHS X is a periodic sequence $\{x_i, i = 0 \dots L\}$ of frequencies within \mathcal{F} . In the example of Fig. 2, $\ell = 280$, and the period is $L = 35$.

Hamming Correlation is a measure of the similarity or correlation between two FHSs [32]. In our setting, it quantifies the number of collisions between two FHS potentially shifted in time. Let $X = \{x_i\}$ and $Y = \{y_j\}$ be two FHS, their Hamming Correlation at a time offset τ is:

$$H(X, Y; \tau) = \sum_{i=0}^{L-1} h(x_i, y_{i+\tau \pmod{L}}), \quad (1)$$

where $h = \mathbb{1}_{\{f_i=f_j\}}$, with $f_i, f_j \in \mathcal{F}$.

III. CONSTRUCTION METHOD OF LR-FHSS SEQUENCES WITH SEMTECH'S DRIVER

The construction technique for FHS tailored for LR-FHSS devices, as found in an official LoRa repository on Github², diverges from the method detailed in [10] or [9]. Rather than relying on a hash function or a pseudo-random function, it harnesses the potential of m -sequences [33].

Table I provides an exhaustive breakdown of the LR-FHSS sequence generator's configurations assumed in the driver. Five distinct configurations arise based on the number of grids they support. Table 1 from [8] lists the available LR-FHSS Channel Bandwidths, indicating the number of grids each channel supports. Several grids between 10 and 47 fall in case 1 from Table I, 60 and 62 grids correspond to cases 2, 86 and 99 grids to case 3, 185 and 198 grids to case 4, and 390 and 403 grids to case 5. It is worth noting that for cases 1 and 2, a 6-register LFSR is employed. Yet, with this configuration, only six primitive polynomials exist, meaning the FHS families

TABLE I
LR-FHSS FHS CONSTRUCTION PARAMETERS

case	1	2	3	4	5
initial state	6	56		6	
LFSR size	6	7	8	9	
polynomial bits	3	2	1	0	
polynomial set	{33,45,48, 51,54,57}	{65,68, 71,72}	{142,149}	{264}	
seed bits	6	7	8	9	
total seeds	64	128	256	512	
FHS family size	384		512		

produced for LR-FHSS devices in these scenarios are limited to 384 sequences.

The generator block for the m -sequence, positioned at the top of the figure, signifies an LFSR comprising m registers. The primitive polynomial, instrumental in the m -sequence's creation, is encoded within the x most significant bits of the 9-bit Hopping Sequence field conveyed in the Header, where x can take values from the set $\{0, 1, 2, 3\}$. The modulation parameters inform the determination of the initial state. At the bottom, a constant *xoring* seed is coded into the remaining $y = 9 - x$ least significant bits of the 9-bit Hopping Sequence field. The LR-FHSS sequence is derived by performing a bit-wise exclusive OR (often referred to as *xoring*, synonymous with modulo two addition) between each component of the generated m -sequence and the *xoring* seed. The sequence's length is bounded by the necessary hops to encapsulate the payload data and header replicas. Furthermore, if the resulting values after the operation surpass the count of supported grids in a given scenario, they are omitted. This selective process trims down the LR-FHSS sequence's length from the original m -sequence length (i.e., $2^m - 1$) to the exact count of the grids.

As a result, the *driver* family is composed of the $M = 384$

²SX126X radio driver: https://github.com/Lora-net/sx126x_driver

possible sequences that can be generated by the LR-FHSS Driver method for data rates DR8 and DR9 (case 1 from Table I), all of length $L = 31$. A tool for generating these sequences is available in our public simulator [29].

IV. PROPOSED METHODS FOR LR-FHSS

This section describes our two main contributions as building blocks of the proposed general framework to push LR-FHSS scalability limits. First, it introduces sequences crafted by harnessing the WGFHS principle delineated in Section II-B. These sequences are tailored to align with the stipulations of the LR-FHSS regulations. Second, we propose various strategies to enhance the LoRa demodulators' decoding proficiency in the LEO satellite scenario.

A. Proposed general framework

The proposed framework, illustrated in Fig. 3, integrates two key contributions to enhance the scalability of LR-FHSS. The first component, represented by the red block, introduces advanced Frequency Hopping Sequence (FHS) designs leveraging Wide-Gap sequences to optimize collision resilience. The second component, shown in the blue block, incorporates novel demodulator allocation strategies—"Early-Decode" and "Early-Drop"—to maximize decoding efficiency by dynamically managing gateway resources. Together, these innovations form a unified approach to addressing the challenges of massive IoT connectivity in satellite-based LR-FHSS networks.

B. FHS Families for LR-FHSS

This section delves into constructing FHS families tailored for the LR-FHSS protocol, namely the red block of Fig. 3. The families constructed can be categorized based on the design principles derived from Lempel-Greenberger and Li-Fan.

1) *Lempel-Greenberger FHS Families*: Lempel and Greenberger [34] proposed FHS construction techniques based on m -sequences that optimize the Hamming Correlation for generating optimal FHS families. The Lempel-Greenberger FHS construction involves selecting parameters $\{p, k, n\}$ and a primitive polynomial for an LFSR of a given order, generating an m -sequence that serves as the base sequence to build optimal sequences. This method enables the generation of $M = p^k$ sequences of length $q = p^n - 1$.

Two Lempel-Greenberger FHS families were built. The first one has the construction parameters $\{p = 2, k = 5, n = 5\}$ with the primitive polynomial 0x12 in hexadecimal. The resulting FHS family named **lem-green** has 32 sequences of length 31.

A second family is built with the same previous parameters but using the primitive polynomial 0x12. Then, merging the previous families generates a second Lempel-Greenberger FHS family named **lem-green-2x**. The **lem-green-2x** FHS family has 64 sequences of length 31.

2) *Wide-gap FHS*: The smallest gap or distance e between successive frequencies of X is delineated as:

$$e = \min_{0 \leq i \leq L-2} \{|x_{i+1} - x_i|, |x_{L-1} - x_0|\}. \quad (2)$$

In situations where $e > 0$, there are no two successive frequencies that are similar, and sequence X is classified as a wide-gap FHS (WGFHS) [18], [35]. For LR-FHSS, the concept of grids is a means to achieve such a wide gap property.

3) *Li-Fan FHS Families*: In [18] the authors propose a new type of Wide-gap FHS. The authors then provide a construction method for such sequences in [35], which we name here Li-Fan FHS. The Li-Fan FHS construction provides families based on the methods of period 2ℓ or 3ℓ . Depending on the chosen parameters, families might need further splitting or merging to attain desired frequency values and lengths. For all of the following Li-Fan, the frequency values above 280 are discarded and the minimum gap is adjusted to 8.

The **li-fan-2l** Family is based on 2ℓ -method with parameters $\ell = 281, d = 8$. A single FHS of length $2\ell = 562$, further split into 18 sequences of length 31. We repeat this process with the parameters $\ell = 277, \ell = 283$ and $\ell = 287$, all with $d = 8$. Then we merge these 4 families to form the **li-fan-2l-4x** Family.

The **li-fan-3l** Family is similar to the **li-fan-2l** Family but it uses the 3ℓ method. Lastly, **li-fan-3l-4x** Family construction is analogous to the **li-fan-2l-4x** Family.

We tailored FHS families optimized for the LR-FHSS protocol by employing the above design principles and methods. These are crucial as they lay the foundation for enhancing the decoding proficiency of LoRa demodulators, a significant step forward for satellite-based communications.

C. Early Drop and Early Decode Mechanisms for LR-FHSS

We next describe our second proposed component of our general framework, namely the blue block of Fig. 3.

The LR-FHSS satellite gateway's primary functionality can be conceptualized as follows:

- *Frame Reception*: Upon LoRa satellite gateway's reception of a transmission from a ground IoT device, an available demodulator springs into action. Its core responsibilities encompass collision monitoring and the capture of headers and any fragmented data in the transmission.
- *Frame Demodulation*: After completing the reception process of all fragments, the actual demodulation occurs. A packet is successfully decoded if at least one header replica and a minimum number of necessary fragments are free from collisions.
- *Demodulator Availability*: When no demodulators are on standby or accessible, the transmission is sidestepped. As a result, the corresponding packets are discarded.

This structured approach streamlines the gateway's operation, ensuring efficiency and minimal data loss.

Given the payload's encoding structure, it's possible to decode it successfully, even if only a portion of the fragments are available. This suggests an opportunity to utilize each demodulator's capability more efficiently. Towards this end,

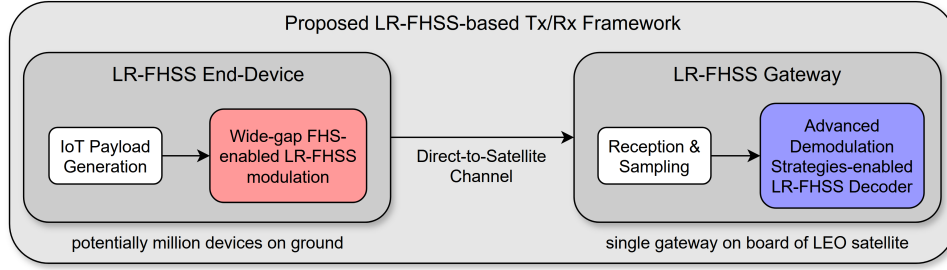


Fig. 3. Proposed framework for enhancing LR-FHSS scalability

we propose adopting the subsequent smart LoRa gateway mechanisms.

1) *Early Drop*: The Early Drop demodulation strategy is specifically designed to optimize the efficiency of the satellite gateway's demodulation resources. This mechanism comes into play when the demodulator determines that the minimum number of fragments required for successful packet decoding cannot be achieved. Under such circumstances, the demodulator preemptively terminates the packet's reception. To elucidate the concept, consider a payload consisting of 13 fragments. Depending on the encoding scheme:

- Code Rate 1 (CR1): A minimum of 5 fragments is essential for successful decoding.
- Code Rate 2 (CR2): 9 fragments are required.

Figure 4 delineates these configurations, where each fragment corresponds to a frequency channel (indicated with numbers). Red highlights collisions (unsuccessfully received fragments), while green signifies successfully received fragments.

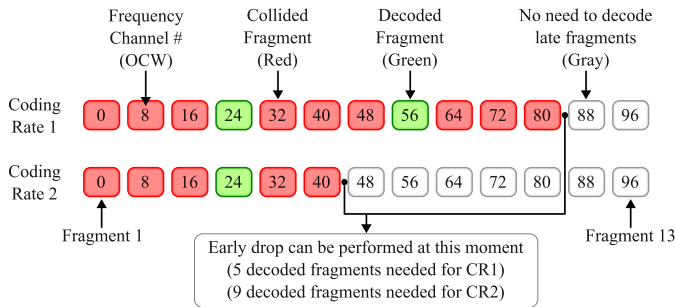


Fig. 4. Illustration of Early Drop for a 13-fragment payload

In a CR1 scenario, after recognizing the 9th collision, it becomes clear that only a maximum of 4 successful fragments can be accumulated, which is inadequate for decoding. Recognizing this, the demodulator can promptly categorize the payload as 'collided' and free itself for other tasks. Analogously, for CR2, the demodulator would adopt a similar course of action after the 5th collision. The steps involved in this process are detailed in Algorithm 1.

2) *Early Decode*: This mechanism enables the demodulator to commence decoding once it has collected the minimum number of fragments required for successful payload decoding. This strategy lets the demodulator quickly process each packet, reducing its LoRa gateway occupancy and enhancing the overall throughput and efficiency. Using the same fragment configurations detailed above:

Algorithm 1: Early Drop

```

1 totalFragments ← 31
2 threshold ← ceil(totalFragments / 3)
3 if CR is 2/3 then
4   | threshold ← ceil(2 · totalFragments / 3)
5 end
6 collidedFragments ← 0
7 for fragment in payload do
8   | if fragment is collided then
9     | | collidedFragments ← collidedFragments + 1
10  | end
11  | if collidedFragments > (totalFragments - threshold)
12  | | Return
13  | end
14 end

```

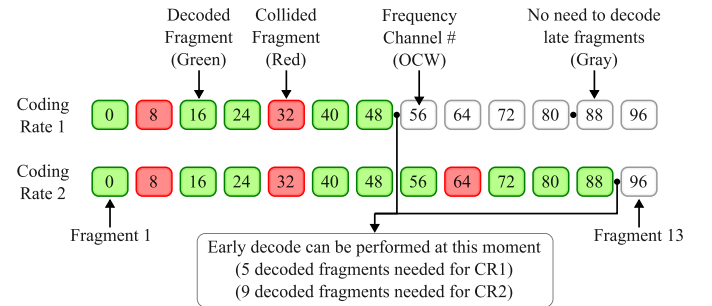


Fig. 5. Representation of Early Decode for a 13-fragment payload

- Code Rate 1 (CR1): The decoding process initiates immediately after the successful reception of the 5th fragment.
- Code Rate 2 (CR2): Decoding begins after the successful reception of the 9th fragment.

Fig. 5 showcases these configurations with various fragment collision statuses. Once the required fragments are received successfully, the demodulator has adequate data to decode the payload. It can then ready itself for the reception of subsequent packets. Algorithm 2 offers an in-depth procedural description of the Early Decode mechanism.

V. RESULTS AND ANALYSIS

A. Simulation Settings and Parameters

We constructed a simplified but representative simulation model to assess the performance implications of various FHS

Algorithm 2: Early Decode

```

1 totalFragments  $\leftarrow$  31
2 threshold  $\leftarrow$  ceil(totalFragments / 3)
3 if CR is 2/3 then
4   | threshold  $\leftarrow$  ceil(2 · totalFragments / 3)
5 end
6 successfulFragments  $\leftarrow$  0
7 for fragment in payload do
8   | if fragment is successful then
9     | successfulFragments  $\leftarrow$  successfulFragments +
10    | 1
11   | end
12   | if successfulFragments  $\geq$  threshold then
13     | Return
14   | end
15 end

```

families in an LR-FHSS network. This model encapsulates fragments’ and headers’ time-slotted communication dynamics within finite frequency channels.

1) *Simulation Model Overview*: Each fragment in the simulation persists for g time slots, where g denotes the model’s granularity. Correspondingly, headers last for $\lceil 233g/102.4 \rceil$ slots. Throughout the simulation duration, N nodes transmit a packet of fixed length L , broken down into fragments. These transmissions are initiated at random time slots. The choice of an FHS for each node is randomized from the available FHS family. The gateway employed in the model is equipped with p demodulators, which are essential for decoding the packets.

A collision is deemed to have occurred if two or more fragments or headers share at least one-time slot on an identical frequency channel. Payloads receive a ‘collided’ designation if their collided fragments breach a certain threshold, contingent on the CR utilized for encoding. Additionally, the *Header Tolerance* parameter determines the header’s collision tolerance, defining the maximum collided slots permissible for a successful decode.

Our model leverages the free space channel model, which is especially fitting for LR-FHSS in satellite communications due to outer space’s open and unobstructed nature. Contrary to the cluttered terrestrial environments, space provides a clear signal path, validating the model’s straightforwardness as an effective means of characterizing signal interactions between satellites and terrestrial stations [36]. Such a channel becomes even more pertinent when factoring in the vast distances involved, which are not typically accounted for in standard urban and suburban IoT [37]. Furthermore, frequency-selective fading is negligible, given the narrow-band channels in LR-FHSS. While physical-layer impairments such as space radiation interference and multipath fading exist in LEO satellite IoT communications, their impact remains significantly lower than the dominant factors studied in this work, including sequence design and demodulator allocator.

Our simulator’s codebase comprising the scenario configurations used in this section is publicly accessible on Github [29].

2) *Early Header Drop*: In our extended gateway model, we introduce the ‘‘Early Header Drop’’ feature to accurately portray an LR-FHSS gateway’s operation. The importance of a header in packet communication cannot be overstated, as it carries the FHS information crucial for decoding the corresponding packet. Thus, if a header is not successfully decoded, the processor remains uninformed about the payload fragment sequence. In such situations, the gateway’s logical course of action would be to release the demodulator, enabling it to track subsequent packets. The previous operation mode is how the LR-FHSS works in reality, so the benefit of including this feature in our simulation is to assess the potential of decoding headerless frames. This is the case when, hypothetically, the gateway receives the minimum number of fragments for a successful payload decoding.

While our research primarily focuses on assessing FHS performance within LR-FHSS networks, evaluating the ‘‘Early Header Drop’’ mechanism is largely excluded from most tests in Section V. Still, the study of decoding payloads without their associated headers holds considerable potential and significance in LR-FHSS research, as noted in [13]. The ‘‘Early Header Drop’’ mechanism is executed after the reception of the last header. The ‘‘Early Header Drop’’ mechanism is no more than the decoder behavior. An LR-FHSS decoder will only attempt to decode the payload data if it recovers at least one of the header replicas.

3) *Simulation Configuration*: The simulations were primed using the parameters outlined in Table II, following the parameters for DR8 and DR9. In a gateway containing a single DSP and employing a channel spacing of 200 kHz, end devices can utilize a maximum of 7 OCW channels when employing a 136.70 kHz channel bandwidth [10]. The payload size was set to 31 fragments, the maximum allowed for DR8. The fragment granularity is six time slots. Then, the maximum sequence duration arises during CR1 encoding (factoring in 3 header repetitions), culminating in a transmission span of 228 time-slots. In contrast, CR2-encoded transmissions wrap up in 214 time slots. We chose the simulation time as a fourfold duration of the longest sequence, totaling 912 time-slots.

The ratio of fragment-to-header durations in our simulation, which stands at 14 slots / 6 slots \approx 2.33, is a close reflection of real-world values approximated at 233 ms / 102.4 ms \approx 2.27.

4) *Parameters Under Study*: Table III delineates the parameters that underwent scrutiny in our research, each having two feasible values. Both the *Early Decode* and *Early Drop* features can be toggled on or off individually. Nevertheless, both are activated or deactivated simultaneously in our experiments. When *Early Header Drop* is turned on, the demodulator will drop the payload if no header is successfully received. For most experiments, the *Header Tolerance* is set to 0, meaning that a single collided time slot will result in a collided header. The last set of experiments is set to 4 time slots. Given that the payload fragment granularity requires six time-slots, a header will only tolerate collisions at the header reception’s start, the end, or both. Our control parameter, the node count, is the transmission tally since each node transmits once within the simulation timeframe. Across different scenarios in the ensuing sections, the node (or transmission) count oscillates

TABLE II
SIMULATION CONFIGURATION

Parameter	Value
Number of repetitions	10
Simulation time	912 [slots]
OCW channels	7
OBW channels	280
Payload size	31 [fragments]
Fragment size	6 [slots]
Header size	14 [slots]

TABLE III
STUDY PARAMETERS

Parameter	Possible values
Coding Rate	CR1, CR2
FHS family	li-fan, driver
Number of demodulators	100, 1000
Early Decode	True, False
Early Drop	True, False
Early Header Drop	True, False
Header tolerance in slots	0, 4

between 10 and 10,000. Given the results presented in section V-B1, the families of interest are the driver and the Li-Fan.

In their work, Boquet et al. [10] pointed out that the FHS generation method in LR-FHSS pivots around a hash function. Drawing inspiration from this methodology, we formulated a *hash* family for FHS based on the sha256 function. This curated FHS family encompasses $M = 384$ families, each with a length of $L = 31$.

B. Preliminary Frequency Hopping Sequence Analysis

1) *Hamming Distance Analysis*: The summary of the evaluated FHS families can be found in Table IV, where they are distinguished based on their size and the family's maximum Hamming Cross-Correlation $\bar{H}_{\max}(S, S)$.

The family maximum Hamming Correlation (*maxHC*) is illustrated in Figure 6a for both the *li-fan* and the *driver* families. The *maxHC* is the maximum Hamming Correlation value between any pair of sequences belonging to the FHS family. The correlation is assessed against increasing sequence lengths. For shorter sequences spanning up to 10 hops, the *li-fan* family exhibits a reduced correlation compared to the *driver* family. However, as the sequence length extends beyond this range, the *driver* family takes precedence in terms of lower correlation. It is noteworthy that even though the *li-fan*

TABLE IV
FHS FAMILY SIZE AND CORRELATION COMPARISON

family	size	$\bar{H}_{\max}(S, S)$
lem-green	32	0.939
lem-green-2x	64	2.108
li-fan-2l	18	0.480
li-fan-2l-4x	71	1.769
li-fan-3l	26	0.704
li-fan-3l-4x	107	1.566
hash	384	0.407
driver	384	0.416

FHS is designed to yield an optimal sequence, this property is guaranteed only for the singular sequence generated. This explains the observed degradation in correlation performance when this primary sequence is partitioned into more minor sequences to constitute an LR-FHSS-compatible FHS family.

Fig. 6b (*avgHC*) presents the family average Hamming Correlation $\bar{H}_{\text{avg}}(S, S)$. The *avgHC* is the mean Hamming Correlation value between all pairs of sequences belonging to the FHS family. This correlation is demonstrated for the families of both *driver* and *li-fan*, each formulated under three distinct configuration parameters. The three configurations are: 1. *EU137* with a maximum FHS length of 35 hops, 2. *EU336* with a maximum of 86 hops, and 3. *US1523* with 60 hops. Interestingly, across all these configurations, the *li-fan* FHS families consistently outperform their *driver* counterparts, showcasing superior correlation metrics.

2) *Collision Rate Analysis*: For an initial assessment, we used a preliminary test to study the collision rate at the payload level for different FHS families, as detailed in Table IV. This model is more basic than the one discussed above, encompassing fewer parameters. The collision rate is calculated as the number of collided slots from overlapping transmissions, i.e., with more than one fragment, divided by the total number of slots available. This design eliminated granularity, meaning each fragment was equivalent to a single time slot. Throughout the simulation, each node was responsible for transmitting a single payload of 31 fragments devoid of headers. The total time allocated for simulation was fourfold the time required for payload transmission, summing up to 124 time slots. Another distinguishing feature of this model was its lack of a decoding threshold for simultaneous receptions.

The core aim of this basic model was to analyze the FHS's correlation performance under conditions of multiple concurrent transmissions. Fig. 7 presents the relative collision rate, which is compared against the baseline established by the *driver* family. Here, the collision rate of each family is relative to the one achieved by the *driver* family for every transmission count. The *lem-green* family initially demonstrated a better performance than the *driver* family up to a certain number of transmissions but later faced a decline, especially for CR2. Observations highlighted that the *li-fan* families exhibited superior performance compared to the reference point after reaching a specific number of nodes. This trend was consistent for both Coding Rates.

Given the insights derived from this preliminary evaluation, we select the *li-fan-2l* family as the most performing in LR-FHSS. The upcoming comprehensive simulation campaigns will further investigate the combined performance of *li-fan-2l* with the proposed demodulation optimization techniques.

C. Early Drop and Early Decode Analysis

Building on our selection of the *li-fan* family, our analysis proceeds with the comprehensive model detailed in Section V-A. Under this framework, a packet can experience one of three outcomes:

- 1) *Discarded*: When no demodulators are available.
- 2) *Successful*: A demodulator decodes the payload.

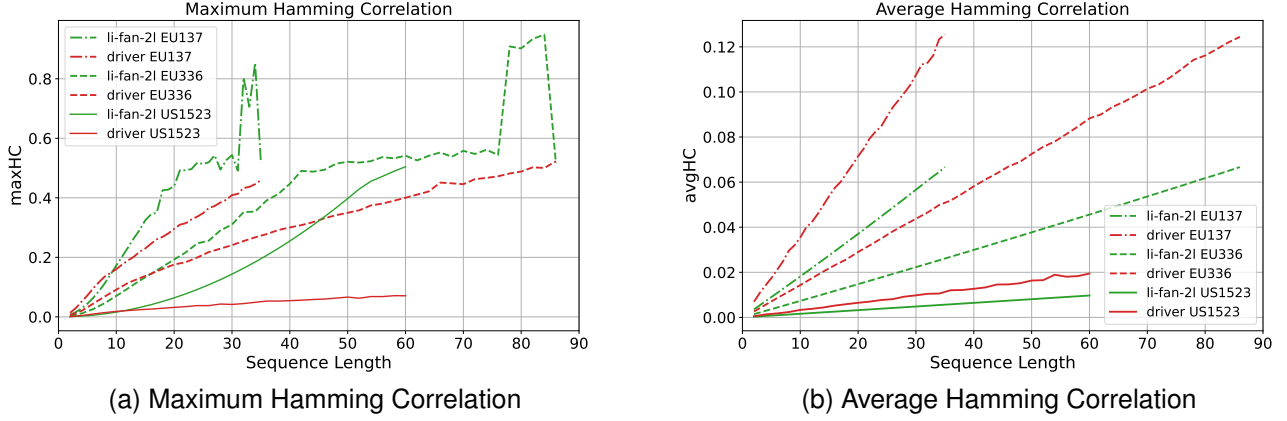


Fig. 6. Maximum and average Hamming Correlation for different LR-FHSS FHS families (*li-fan* and *driver*).

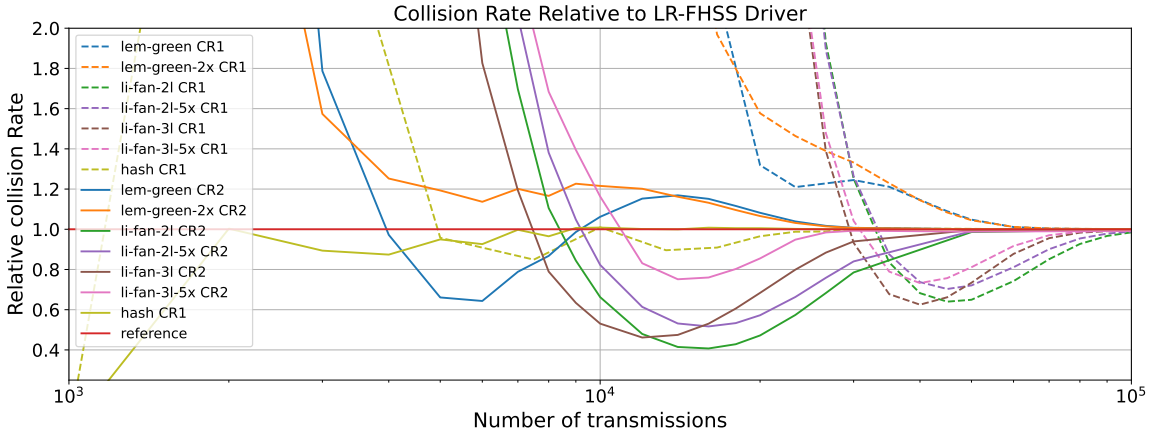


Fig. 7. Relative collision rate for different LR-FHSS FHS families against the reference Driver family, considering no granularity, no headers, and unlimited decoding capacity.

- 3) *Collided*: The payload fails to be decoded as the minimum number of fragments is unmet.

Although headers are not essential for payload decoding in this scenario, their presence still influences collision dynamics. We distinguish between *payload* decoding, when the minimum amount of fragments is received, and the payload can be successfully decoded, and *packet* decoding when at least one header replica is received along with the payload.

Fig. 8a plots the cumulative decoded payloads against the total packets transmitted using CR1 and 100 demodulators, while Fig. 9a is similar but using 1000 demodulators. The comparison is multifaceted:

- Families Comparison: The solid lines represent the *li-fan* family, while the dashed lines depict the *driver* family.
- Decoding Mechanisms: The results derived from the *Early Drop* and *Early Decode* mechanisms, tagged as *earlydd* and shown in blue and purple shades, are set against the baseline demodulator model, shown in orange and green shades.
- Demodulators Count: The data segregates between setups with 100 demodulators in Fig. 8a and 1000 demodulators in Fig. 9a. For reference, the unlimited decoding capacity

scenario is illustrated with the label *infp* in red.

With the configuration of 100 demodulators (100p), activating the *earlydd* mechanism brings about a significant enhancement in decoding performance for both the *li-fan* and *driver* families. Specifically, the number of successfully decoded payloads can see an increase of up to 100%, effectively doubling the decoding output. However, when the demodulator count is raised to 1000 (1000p), the benefits of the *earlydd* mechanism become less pronounced. Even so, the decoding capability curve is nudged closer to that of the model with infinite decoding capacity. A notable observation is that, under this configuration, the *li-fan* family witnesses an improvement of up to 50% in successfully decoded payloads when there are around 4000 transmitted packet.

When contrasting the performance of the *li-fan* and *driver* FHS families, the *driver* FHS consistently outperforms the *li-fan* up to a specific threshold of transmissions. This superiority becomes even more pronounced when operating with 1000 demodulators (1000p). Beyond this threshold, however, a transition from the *driver* FHS to the *li-fan* FHS has the potential to yield better network throughput.

As illustrated in Fig. 8b for the case using CR2, the *earlydd*

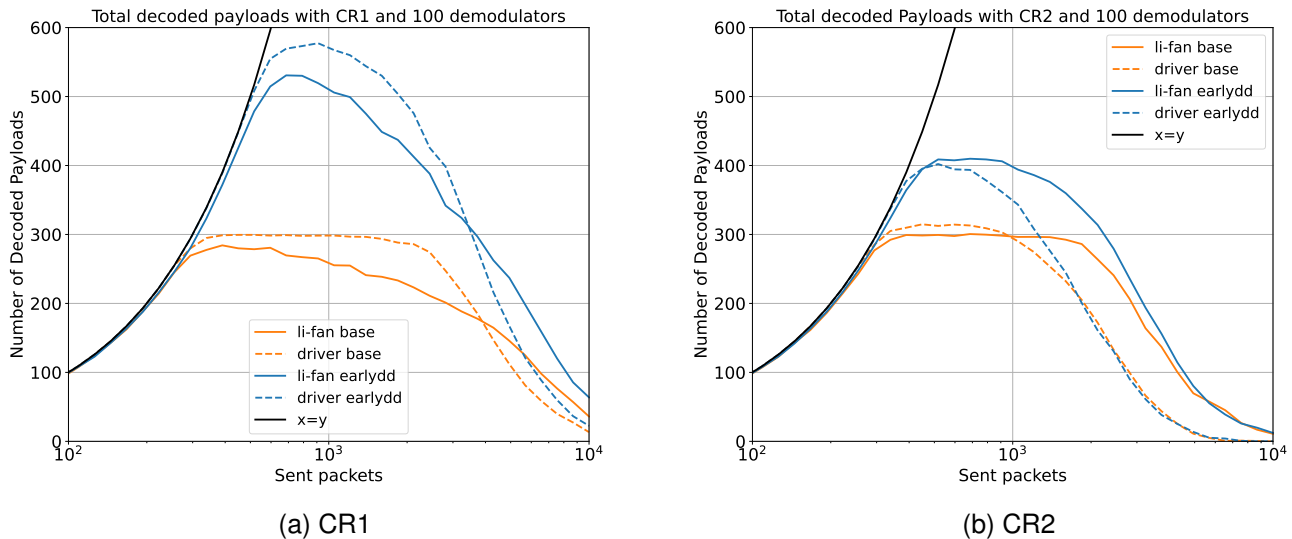


Fig. 8. Total decoded payloads with 100 demodulators, comparing CR1 with CR2.

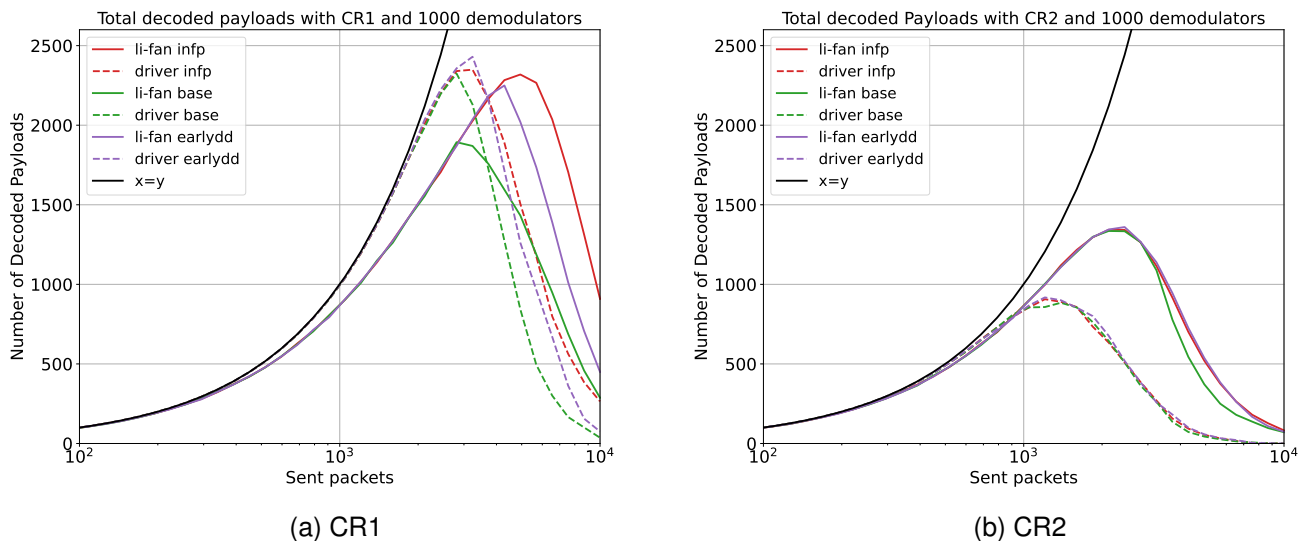


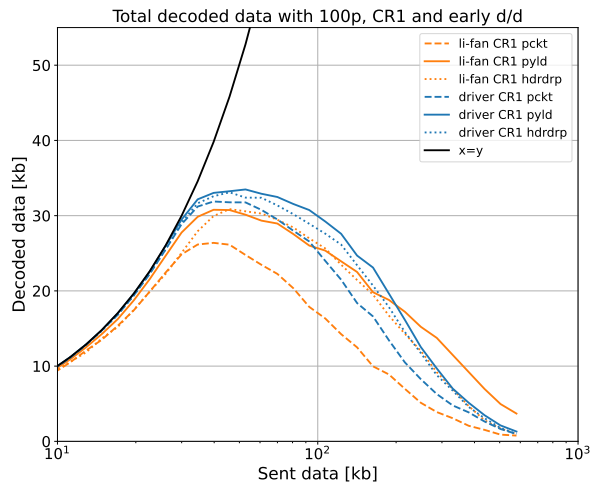
Fig. 9. Total decoded payloads with 1000 demodulators, comparing CR1 with CR2.

mechanism consistently enhances the decoding capacity for both the *li-fan* and *driver* FHS families. The improvement is particularly significant when utilizing 100 demodulators (100p), but it tapers off as the channel nears saturation. The benefits are less pronounced when employing 1000 demodulators (1000p), illustrated in Fig. 9b.

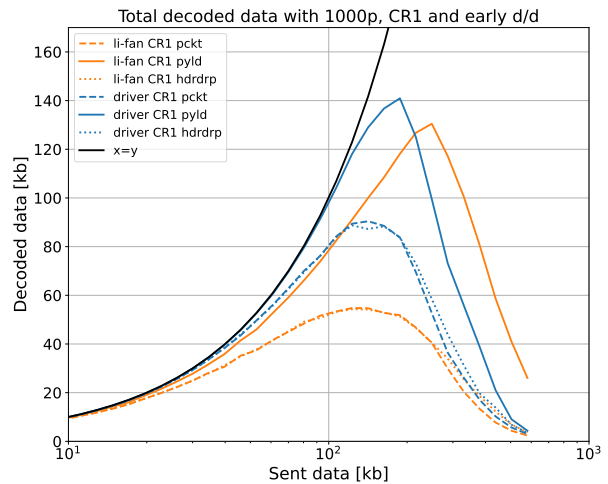
A comparative evaluation of the *li-fan* and *driver* families reveals a considerable advantage for the *li-fan* family. This superiority is especially evident in the 1000p setup, where the decoding capacity of the *li-fan* family surpasses that of the *driver* family by a remarkable 170%. While the *driver* family does exhibit marginally superior performance under lighter network loads, the disparity at higher loads suggests that the *li-fan* family is the best choice when CR2 is in use.

D. Packet versus Payload Analysis

This section compares packet decoding and payload decoding. While the latter was the primary focus of the preceding section, the emphasis here shifts to packet decoding. Packet decoding implies that at least one header replica is received in addition to a successful payload decode. For the context of this analysis, both the *Early Drop* and *Early Decode* mechanisms remain activated. Our metric of interest is the total decoded data compared to the total data sent in kilobytes. This approach offers a balanced comparison between CR1 and CR2. Notably, despite requiring the same number of hops, CR1 encodes less data than CR2. Thus, this metric facilitates an unbiased assessment of their performance relative to the data throughput they deliver.

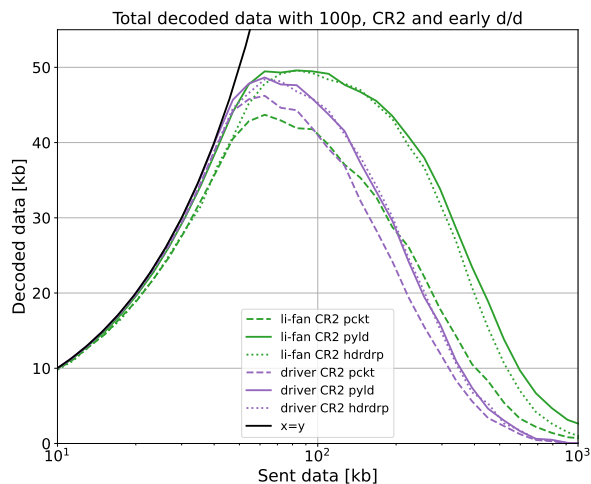


(a) 100 demodulators

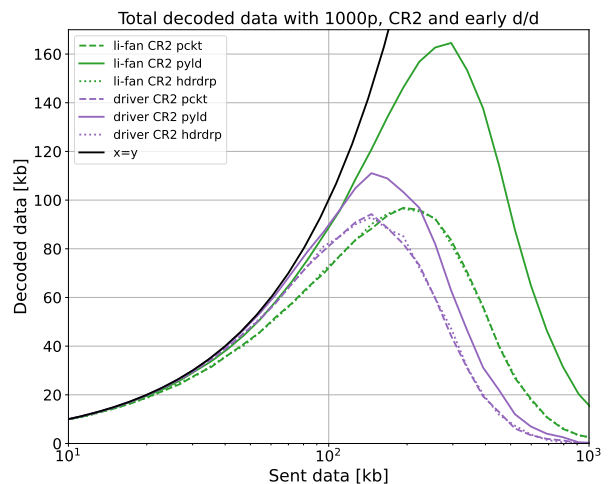


(b) 1000 demodulators

Fig. 10. Total decoded data with 100 and 1000 demodulators, CR1, Early-Drop and Early-Decode.



(a) 100 demodulators



(b) 1000 demodulators

Fig. 11. Total decoded data with 100 and 1000 demodulators, CR2, Early-Drop and Early-Decode.

1) *100 Processor Analysis*: The distinction between packet and payload decoding is brought to the fore in Figures 10a and 11a. The *li-fan* and *driver* FHS families are portrayed with different color codes: orange and green for *li-fan*, and blue and purple for *driver*. A further distinction is made between CR1 (depicted in shades of orange and blue in Fig. 10a) and CR2 (represented in shades of green and purple in 11a). Dashed lines signify packet (*pckt*) decoding, while solid lines represent payload (*pyld*) decoding.

The observation from the figure confirms the intuitive expectation: the rate of packet decoding, which requires both a successful payload and at least one header replica, is invariably lower than the payload-only decoding. This is depicted by the consistently higher solid lines (*pyld*) compared to their dashed

counterparts (*pckt*) across the graph.

When considering CR1, this decrease in the packet decoding performance relative to payload decoding also manifests in the competitive dynamics between the *li-fan* and *driver* families. Here, the *li-fan* family fails to outperform the *driver* family across the entirety of the depicted range.

Turning our attention to CR2, a similar trend unfolds. Although *li-fan* had previously displayed a higher efficiency, this distinction diminishes when pitted against *driver* in the context of packet decoding. The relative performance advantage of *li-fan* observed in the payload-only decoding scenario wanes, cementing the assertion that the header replica decoding requirement poses a challenge to both families, altering their relative efficiencies. For a dense network, *li-fan* with CR2

achieves the best performance.

2) *1000 Processor Analysis*: With the enhanced processing capacity depicted in Figure 10b, we observe distinct trends compared to the prior analysis with 100p.

The increased processing capability does not shield the *li-fan* family from the added complexity of *pckt* decoding, as the family remains noticeably more affected than when only considering *pyld* decoding. For CR1, the performance gap between the two families is quite pronounced. The *driver* family outpaces the *li-fan* by a substantial margin, decoding up to 60% more data. This demonstrates the *driver*'s efficiency and highlights the challenges *li-fan* encounters when packet decoding is in play, even with enhanced processing capability.

For CR2, in Figure 11b, the dynamics between the two families become more intricate. The *driver* family retains dominance over a broader range of transmissions before *li-fan* takes the lead. This extension of the *driver*'s dominant zone underscores the impact of packet decoding on the *li-fan*'s performance, even in the context of CR2.

Taking a step back to compare the broader impact of the coding rates, it is evident that CR2 offers superior throughput compared to CR1. This is particularly true when the available OCW channels are kept constant for both coding rates. The data, therefore, lends itself to the idea that, in terms of raw throughput and irrespective of the FHS family in question, CR2 presents a more efficient encoding choice. Again, for a dense network, *li-fan* with CR2 achieves the best performance.

E. Header Drop Analysis

Incorporating the *Early Header Drop* mechanism brings about a significant shift in the dynamics of the decoding process, casting a spotlight on the vital role headers play in LR-FHSS networks.

In Figures 10a and 11a, which detail the outcomes with 100p, a noteworthy observation is the proximity of the *hdrdrp* (*Early Header Drop* activated) curves to the *pyld* (payload decode with *Early Header Drop* off) curves. This indicates that when relieved from payloads whose headers were not successfully received, the demodulators can maintain a decoding rate almost as efficiently as if headers were non-essential. This suggests that in more constrained processing environments, the *Early Header Drop* mechanism assists in focusing resources on more probable decoding successes.

However, the dynamics change as the processing power is ramped up in Figures 10b and 11b. Here, the *hdrdrp* curves converge with the *pckt* curves. The vast gap between the performance of the *hdrdrp* and *pyld* signifies missed decoding opportunities. The divergence between these curves indicates that with more processing resources, there is substantial potential for successfully decoding payloads even when missing headers. This observation underscores the limitation of being overly reliant on headers in LR-FHSS systems.

Given the contrast observed between the *hdrdrp* and *pyld* results in the 1000p scenario, the findings hint at the prospective advantages of a headerless decoding mechanisms [13]. If LR-FHSS networks were to develop a reliable method of decoding payloads without needing headers, it could substantially

amplify the effective throughput and overall performance. With its apparent untapped potential, this direction warrants further research and could radically change LR-FHSS network methodologies.

F. Header Tolerance Analysis

This section offers an extended perspective on decoding dynamics by introducing a "collision tolerance" for the header to model its more robust modulation properly. In Fig. 12a for the 100p scenario, the inclusion of the *Header Tolerance* (*hdrtol*) displays the immediate benefits of the flexibility offered by this model. While both FHS families witness an improvement, the *li-fan* family reaps a more pronounced uptick in decoding performance. For CR2, even though the *driver* family starts with a performance edge over *li-fan*, this dominance is short-lived. Beyond the 60 Kb transmission threshold, *li-fan* surges ahead with its decoding capacity, outperforming the *driver* by as much as doubling it. Similarly, in the CR1 scenario, the earlier observations from Fig. 8a come to mind, where *li-fan* eventually surpasses the *driver* in decoding performance after a specific transmission volume.

Scaling the processing power to 1000p, as represented in Fig. 12b, the benefits of the *Header Tolerance* model become even more conspicuous. The resilience against header collisions boosts the decoding capacities for both families, but once again, the *li-fan* stands out. The CR2 results reiterate the shift in dominance, with *li-fan* pulling far ahead of the *driver* family, especially after a specific transmission volume. The differential reaches a staggering 100%, highlighting the merits of the *li-fan* family in this specific scenario. The CR1 results echo a similar outcome, marking an apparent comeback for *li-fan* over *driver* with increasing transmission counts.

This analysis accentuates the importance of adaptability in network protocols and how slight adjustments, such as allowing for some collision tolerance in headers, can have outsized impacts on performance metrics. It underscores the potential of optimizing network parameters to adapt to specific requirements, thus enhancing overall network efficiency.

Finally, Table V presents a comprehensive summary of the most effective FHS families for each variant of LR-FHSS discussed in this study relative to different scales of node counts. The findings strongly suggest that the strategic integration of sequence design and resource allocation methods explored in this research pushes LR-FHSS to its scalability limits. Consequently, LR-FHSS stands out as a highly viable solution for managing the extensive IoT connectivity demands posed by the evolving direct-to-satellite IoT paradigm.

VI. CONCLUSIONS

The emerging integration of IoT with LEO satellites necessitates the evolution of robust, efficient, and scalable communication protocols. Our in-depth exploration into LR-FHSS modulation, mainly focusing on the design of FHS and efficient demodulator allocation strategies, has highlighted the significant promise of Wide-Gap sequences for massive LEO satellite IoT communications. Our primary findings and contributions are summarized as follows:

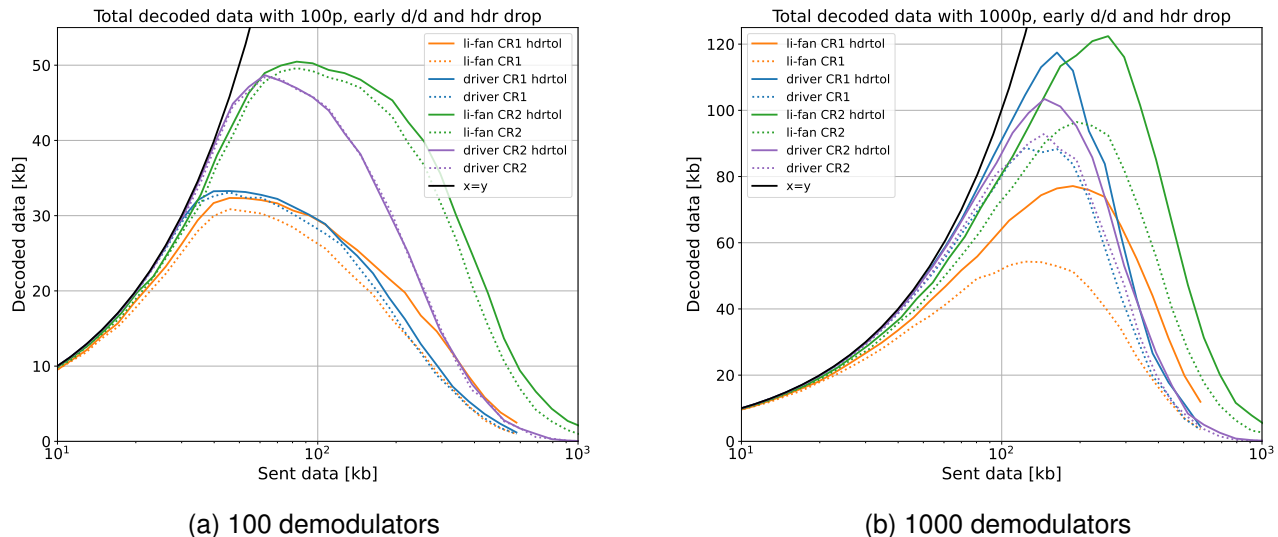


Fig. 12. Total decoded data volume with Early-Drop, Early-Decode, Header Drop, and Header Tolerance of 4 time slots.

TABLE V
BEST FHS FAMILY FOR DIFFERENT LR-FHSS CONFIGURATIONS

100 Decoding Processors				
Node Count	Data Volume CR1 in Kb	CR1	Data Volume CR2 in Kb	CR2
0 - 500	0 - 29	driver	0 - 61	driver
500 - 1725	29 - 100	driver	61 - 209	li-fan
1725 +	100 +	li-fan	209 +	li-fan
1k Decoding Processors				
Node Count	Data Volume CR1 in Kb	CR1	Data Volume CR2 in Kb	CR2
0 - 1240	0 - 72	driver	0 - 150	driver
1240 - 6550	72 - 380	driver	150 - 793	li-fan
6550 +	380 +	li-fan	793 +	li-fan

a) *Actual sequences used by LR-FHSS devices:* We have highlighted in Section III how the frequency hopping sequences of LR-FHSS are constructed in real LR-FHSS devices. This knowledge can help the community improve the sequence construction in the future.

b) *Wide-Gap Sequence Design:* The research highlights the superior performance of the *li-fan* FHS sequence family over conventional sequences within LR-FHSS contexts, notably in setups involving advanced demodulators.

c) *Demodulator Allocation Strategies:* The introduction of “Early-Decode” and “Early-Drop” strategies represents a transformative approach to optimizing decoding resources for LR-FHSS gateways.

Some guidelines for future work to further enhance the scalability of the LR-FHSS network include the possibility of decoding LR-FHSS payloads without the header information. Also, future work will extend this study with refined channel models incorporating additional physical-layer effects.

REFERENCES

- [1] J. A. Fraire, O. Iova, and F. Valois, “Space-terrestrial integrated Internet of Things: Challenges and opportunities,” *IEEE Communications Magazine*, 2022.
- [2] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, “Satellite communications supporting Internet of remote Things,” *IEEE Internet of Things J.*, vol. 3, no. 1, pp. 113–123, 2015.
- [3] J. A. Fraire, S. Céspedes, and N. Accettura, “Direct-to-satellite IoT—a survey of the state of the art and future research perspectives: Backhauling the IoT through LEO satellites,” in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2019, pp. 241–258.
- [4] M. Capuzzo, C. Delgado, A. K. Sultania, J. Famaey, and A. Zanella, “Enabling green iot: Energy-aware communication protocols for battery-less lorawan devices,” in *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2021, pp. 95–98.
- [5] S. Herrería-Alonso, M. Rodríguez-Pérez, R. Rodríguez-Rubio, and F. Pérez-Fontán, “Improving uplink scalability of lora-based direct-to-satellite net networks,” *IEEE Internet of Things J.*, 2023.
- [6] L. Alliance, “LoRaWAN 1.0. 3 specification,” *Technical Specification*, 2018.
- [7] Semtech, “LoRaWAN® protocol expands network capacity with new long range – frequency hopping spread spectrum technology,” 11 2020, accessed: 2023-10-17. [Online]. Available: <https://tinyurl.com/2s3ue7e7>
- [8] —, “Application note: LR-FHSS system performance,” SX1261/62/LR1110, pp. 1–28, February 2022.
- [9] M. A. Ullah, K. Mikhaylov, and H. Alves, “Analysis and simulation of LoRaWAN LR-FHSS for direct-to-satellite scenario,” *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 548–552, 2021.
- [10] G. Boquet, P. Tuset-Peiró, F. Adelantado, T. Watteyne, and X. Vilajosana, “LR-FHSS: Overview and performance analysis,” *IEEE Communications Magazine*, vol. 59, no. 3, pp. 30–36, 2021.
- [11] A. Maleki, H. H. Nguyen, E. Bedeer, and R. Barton, “D2D-aided LoRaWAN LR-FHSS in direct-to-satellite IoT networks,” *arXiv preprint arXiv:2212.04331*, 2022.
- [12] A. Maleki, H. H. Nguyen, and R. Barton, “Outage probability analysis of LR-FHSS in satellite IoT networks,” *IEEE Communications Letters*, vol. 27, no. 3, pp. 946–950, 2022.
- [13] J. A. Fraire, A. Guitton, and O. Iova, “Recovering headerless frames in LR-FHSS,” *arXiv preprint arXiv:2306.08360*, 2023.
- [14] S. Jung, S. Jeong, J. Kang, J. G. Ryu, and J. Kang, “Transceiver design and performance analysis for LR-FHSS-based direct-to-satellite IoT,” *arXiv preprint arXiv:2305.13779*, 2023.
- [15] J. Bao and L. Ji, “Frequency hopping sequences with optimal partial hamming correlation,” *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3768–3783, 2016.
- [16] G. Ge, Y. Miao, and Z. Yao, “Optimal frequency hopping sequences: Auto-and cross-correlation properties,” *IEEE transactions on information theory*, vol. 55, no. 2, pp. 867–879, 2009.
- [17] D. Peng and P. Fan, “Lower bounds on the hamming auto and cross

- correlations of frequency-hopping sequences,” *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2149–2154, 2004.
- [18] P. Li, C. Fan, Y. Yang, and Y. Wang, “New bounds on wide-gap frequency-hopping sequences,” *IEEE Communications Letters*, vol. 23, no. 6, pp. 1050–1053, 2019.
- [19] Z. Huaqing, “Design and performance analysis of frequency hopping sequences with given minimum gap,” in *2010 International Conference on Microwave and Millimeter Wave Technology*. IEEE, 2010, pp. 1271–1274.
- [20] L. Bin, “One-coincidence sequences with specified distance between adjacent symbols for frequency-hopping multiple access,” *IEEE Transactions on Communications*, vol. 45, no. 4, pp. 408–410, 1997.
- [21] L. Guan, Z. Li, J. Si, and R. Gao, “Generation and characteristics analysis of cognitive-based high-performance wide-gap FH sequences,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5056–5069, 2014.
- [22] M. A. Ullah, K. Mikhaylov, and H. Alves, “Experiment-based models for air time and current consumption of lorawan lr-fhss,” *arXiv preprint arXiv:2408.09954*, 2024.
- [23] Semtech, “SX1301 - digital baseband chip LoRaWAN macro gateways,” Wireless & Sensing Products, Datasheet, 06 2017.
- [24] R. B. Sorensen, N. Razmi, J. J. Nielsen, and P. Popovski, “Analysis of LoRaWAN uplink with multiple demodulating paths and capture effect,” in *IEEE ICC (International Conference on Communications)*, 2019.
- [25] P. K. Dalela, S. Sachdev, and V. Tyagi, “LoRaWAN network capacity for practical network planning in India,” in *URSI AP-RASC (Asia-Pacific Radio Science Conference)*, 2019.
- [26] D. Magrin, M. Capuzzo, and A. Zanella, “A thorough study of LoRaWAN performance under different parameter settings,” *IEEE Internet of Things J.*, vol. 7, no. 1, pp. 116–127, 01 2020.
- [27] A. Guitton and M. Kaneko, “Improving LoRa scalability by a recursive reuse of demodulators,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.
- [28] —, “Multi-gateway demodulation in LoRa,” in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 2008–2013.
- [29] D. Maldonado, “Lr-fhss network simulation for collision analysis,” 2025, accessed on March 17, 2025. [Online]. Available: https://github.com/diegomm6/lr-fhss_seq-families
- [30] T. Turetli, “GMSK in a nutshell,” *Telemedia Networks and Systems Group LCS, MIT-TR*, 1996.
- [31] D. Torrieri, *Frequency-Hopping Systems*. Cham: Springer International Publishing, 2018, pp. 155–212. [Online]. Available: https://doi.org/10.1007/978-3-319-70569-9_3
- [32] D. Sarwate, “Reed-solomon codes and the design of sequences for spread-spectrum multiple-access communications,” *Reed-Solomon Codes and Their Applications*, 1994.
- [33] M. Goresky and A. Klapper, *Algebraic shift register sequences*. Cambridge University Press, 2012.
- [34] A. Lempel and H. Greenberger, “Families of sequences with optimal hamming-correlation properties,” *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 90–94, 1974.
- [35] P. Li, C. Fan, S. Mesnager, Y. Yang, and Z. Zhou, “Constructions of optimal uniform wide-gap frequency-hopping sequences,” *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 692–700, 2021.
- [36] K. Vogelgesang, J. A. Fraire, and H. Hermanns, “Uplink transmission probability functions for LoRa-based direct-to-satellite IoT: A case study,” in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 01–06.
- [37] T. N. González, J. L. Salamanca, S. M. Sánchez, C. A. Meza, and S. Céspedes, “Analysis of channel models for LoRa-based direct-to-satellite IoT networks served by LEO nanosatellites,” in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.