Recovering Headerless Frames in LR-FHSS

Juan A. Fraire
Univ Lyon, Inria, INSA Lyon, CITI, EA3720, 69621 Villeurbanne, France
Saarland University, Saarland Informatics Campus E1 3, 66123 Saarbrücken, Germany
juan.fraire@inria.fr

Alexandre Guitton
Univ Lyon, Inria, INSA Lyon, CITI, EA3720, 69621 Villeurbanne, France
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France
alexandre.guitton@uca.fr

Oana Iova
Univ Lyon, INSA Lyon, Inria, CITI, EA3720, 69621 Villeurbanne, France
oana.iova@inria.fr

Abstract

Long-Range Frequency Hopping Spread Spectrum (LR-FHSS) is a recent modulation designed for communications from low-power ground end-devices to Low-Earth Orbit (LEO) satellites. To decode a frame, an LR-FHSS gateway must receive at least one header replica and a large proportion of payload fragments. However, LR-FHSS headers will likely be lost when there are many concurrent transmissions. In this paper, we motivate the header loss problem with an analytical model, propose a linear programming model to extract headerless frames and design a cost-effective sliding-window heuristic. Simulation results show that our approach exhibits near-optimal headerless detection and extraction results while ensuring a low computational cost. The proposed method is, therefore, suitable for future LR-FHSS gateways located onboard resource-constrained LEO satellites.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Link-layer protocols

General Terms

Frame recovery, headerless frames, satellite IoT

Keywords

LR-FHSS, ILP, heuristic

1 Introduction

Low-Earth Orbit (LEO) satellites enable a large variety of promising monitoring applications (e.g., environmental or ecological monitoring of oceans, poles, vast forests, extensive natural parks [4]) with a cost significantly reduced compared to geostationary satellites. Such applications are otherwise difficult or impossible to serve with terrestrial infrastructure due to their geographical location [6].

Semtech recently introduced a new modulation called Long-Range Frequency Hopping Spread Spectrum (LR-FHSS) [9], specifically designed for communications from low-power end-devices on the ground to LEO satellites. This modulation is supported by existing LoRaWAN gateways with a simple firmware update and works alongside legacy LoRa modulation, which will ease its adoption. Contrary to legacy LoRa, LR-FHSS is based on frequency hopping within several narrow-band sub-channels (of 488 Hz bandwidth), where the payload is divided into multiple short fragments, each sent on random sub-channels. The index of the chosen frequency hopping sequence is encoded in the header of the LR-FHSS frame, which is sent multiple times on random sub-channels in order to improve its robustness. An LR-FHSS frame is received if at least one header replica is received, as well as a large proportion of the fragments (typically one-third or two-thirds). Thanks to the combination of narrow-band sub-channels and robust coding rates, a LoRaWAN gateway hosted onboard an LEO satellite can decode hundreds of simultaneous LR-FHSS transmissions, which fits the requirements of direct-to-satellite IoT communications [5].

However, recent research showed that header loss is a significant cause of LR-FHSS frame loss [14, 11]. This is due to both the small number of header replicas (typically 2 or 3) and the relatively larger time-on-air of the header (233 ms) compared to the time-on-air of a fragment (102.4 ms). Thus, **addressing the recovery of headerless LR-FHSS frames is crucial to improve the reliability of LR-FHSS communication.**

We propose in this paper a method to recover headerless LR-FHSS frames. To our knowledge, this is the first method of its kind. The core idea is to deduce the random sequence used by the fragments of a given frame. To do this, we assume that it is possible to detect whether a sub-channel is busy or not at a given time slot. This knowledge enables us to detect the sequence that matches the observed channel occupancy among all frequency hopping sequences since all possible sequences are known by the LoRaWAN gateway.

Our specific contributions are:

1. A mathematical derivation of the cause of LR-FHSS frame loss that motivates the recovery of header loss frames;

Table 1: Regional parameters for LR-FHSS OCWs, OBWs, and grids.

Region	OCW bandwidth (MHz)	OBW bandwidth (Hz)	Grid sep. (kHz)	No. of grids	No. of OBWs per grid	No. of OBWs	Coding rate (CR)	Bit rate (bits/s)	Data rate
EU	137	488	3.9	8	35	280 (=8x35)	1/3	162	DR8
							2/3	325	DR9
	366	488	3.9	8	86	688 (=8x86)	1/3	162	DR10
							2/3	325	DR11
US	1523	488	25.4	52	60	3125 (=52x60)	1/3	162	DR5
							2/3	325	DR6

2. A formalization of the sequence detection problem as an integer linear program (ILP) model;
3. A low-cost heuristic to recover headerless LR-FHSS frames leveraging resource-constrained processors.

Finally, by means of an extensive simulation campaign, we present compelling evidence of the efficiency and effectiveness of the proposed headerless decoding approach for LR-FHSS.

The remainder of this paper is organized as follows. Section 2 introduces the LR-FHSS modulation as well as recent related works. Section 3 motivates our work through a probabilistic analysis that exposes the vulnerability of LR-FHSS headers. Section 4 first presents an optimal ILP to recover headerless frames, followed by a low-cost heuristic. Section 5 compares the results of the ILP and the heuristic through extensive simulations. Section 6 raises a discussion on our assumptions and on the mitigation of the false positives produced by our algorithms. Finally, Section 7 concludes our work.

2 LR-FHSS in a Nutshell

LR-FHSS is a frequency hopping modulation that enables fast intra-frame frequency hopping with similar long-range capabilities as LoRa. It is one of the underlying modulations supported by LoRaWAN-compliant devices, together with LoRa and FSK [9]. While LR-FHSS is used for uplink communications (when energy-constrained end-devices communicate with gateways located in LEO satellites), LoRa continues to be used for downlink communication. Indeed, with a simple firmware update, LoRaWAN gateways can decode LR-FHSS frames without disturbing the reception and decoding of LoRa frames. In the reference gateway design, a gateway can listen to all available sub-channels at the same time. Its system capacity for decoding LR-FHSS frames is limited only by its digital signal processing (DSP) capacity [13]. The real-world limit is 700k packets per day with the current software.

LR-FHSS uses Gaussian Minimum-Shift Keying (GMSK) modulation [8] in each sub-channel, which is highly spectral-efficient and energy-efficient. GMSK encodes the data by discrete frequency changes. The use of a Gaussian filter reduces the sidebands caused by the modulation, which is desired for narrow-band communications. GMSK is famous for being used in Global System for Mobile (GSM) communications, as well as for satellite communications.

To increase the recovery of frames in dense environments that are prone to collisions and interference, LR-FHSS also has built-in error correction mechanisms. The used coding rate (CR) is 1/3 or 2/3, depending on the data rate.

In LR-FHSS, the frequency band is split into several contiguous Occupied Channel Width (OCWs). These OCWs are then split into multiple Occupied Bandwidth (OBWs) [2], where each OBW corresponds to a physical sub-carrier with a 488 Hz bandwidth. To enforce the minimum distance between sub-carriers used in the frequency hopping sequence, as imposed by regional regulations, the OBWs are organized into grids. Grids are a fundamental concept in LR-FHSS, as a hopping sequence of a given frame can only use OBWs corresponding to the same grid. Grids are groups of nonconsecutive OBWs separated by a minimum distance, depending on the regional regulation. The number of OCWs, OBWs, and grids for the EU and the US is shown in Table 1.

We exemplify the use of LR-FHSS in Europe in Figure 1, following regional regulations. The European 868 MHz band is divided into 7 OCWs of 137 kHz, each containing 280 OBWs of 488 Hz. In Europe, the minimum separation between OBWs during the frequency hopping is 3.9 kHz; thus, the set of 280 OBWs is organized into 8 grids of 35 OBWs each. This ensures that every two OBWs inside the grid are separated by a multiple of 3.9 kHz. For example, grid 1 contains OBWs 1, 9, 17, ... , 273 (i.e., all the blue-colored OBWs).

The transmission of a frame starts by sending several replicas of the header, followed by the payload divided into short fragments, each sent on a different OBW inside a grid. For example, we can see in Figure 1 that all the headers and the fragments of the red frame are transmitted only in pink-colored OBWs, which are part of grid 7. There are a total of $2^9 = 512$ possible hopping frequencies, known in advance by both end devices and gateways. Each replica of the header contains the index of the chosen frequency hopping sequence. To ensure robustness at reception, each replica is repeated two or three times (depending on the data rate scheme). The large number of possible sequences, which is 512, makes it unlikely to have two or more frames share the same hopping sequence. The payload is encoded such that it can still be recovered even if a large proportion of the fragments are lost [2]. The two configurations specified in the LoRaWAN standard [9] are:

- Robust configuration: each header is sent 3 times, and the payload is encoded using CR = 1/3 (at least 33% fragments must be decoded to recover such a frame).

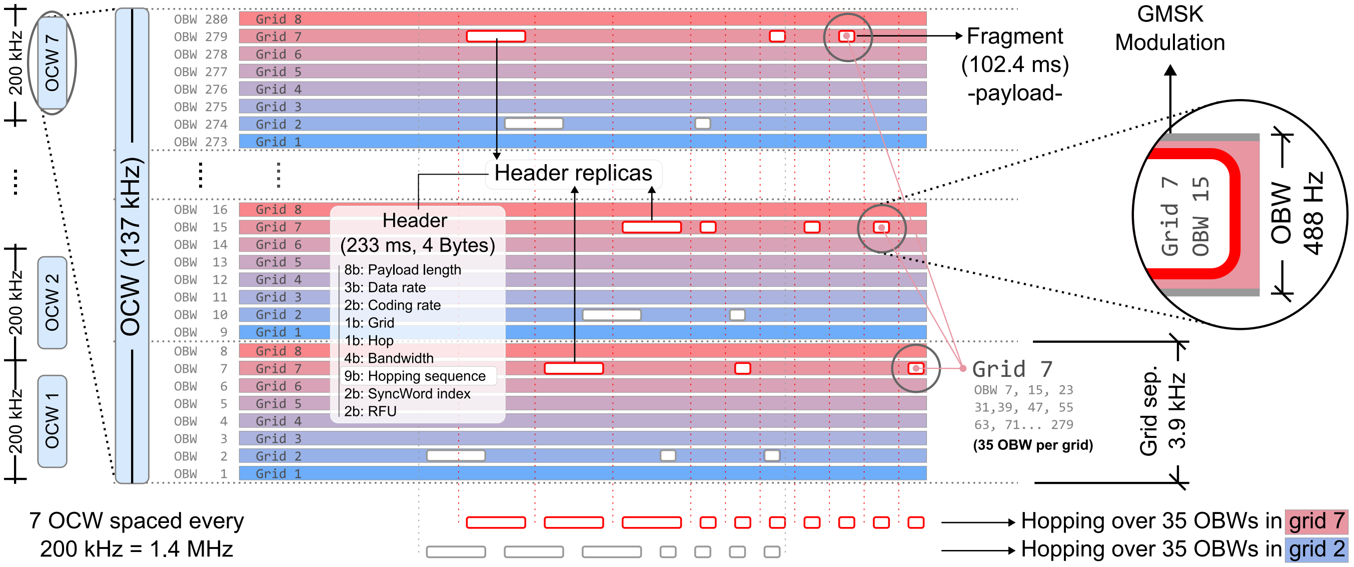


Figure 1: In LR-FHSS, the European 868 MHz frequency is divided into 7 channels called Occupied Channel Width (OCWs). Each OCW is divided into 280 sub-channels called Occupied Bandwidth (OBWs). To ensure that the mandatory minimum separation of 3.9kHz between two consecutive sub-channels, the notion of a grid has been introduced. Each OCW is divided into 8 grids, each grouping 35 OBWs (where two consecutive OBWs are separated by 3.9kHz). A frequency hopping sequence can use only OBWs inside a given grid.

- Fast configuration: each header is sent 2 times, and the payload is encoded using CR = 2/3 (at least 67% fragments must be decoded to recover such a frame).

With LR-FHSS, the gateway located onboard the LEO satellite constantly monitors all OBWs (for all OCWs) in order to detect header replicas. Once a replica is found, the gateway locks onto this transmission, extracts the index of the hopping sequence and the number of fragments from the replica, and follows the sequence in order to obtain the fragments. Since the gateway is mains powered, it is constantly listening for incoming frames.

The novelty of the LR-FHSS technology results in a relatively sparse state of the art, with results obtained analytically or in simulation. Boquet et al. [2] were the first to evaluate the performance of LR-FHSS, showing its considerable improvement in network capacity (up to two orders of magnitude) when compared to LoRa. Ullah et al. [14] showed through simulation that the primary reason for frame loss in LR-FHSS is the loss of the headers. Maleki et al. [11] evaluated the outage probability in LR-FHSS analytically with a realistic channel model. This work was later extended in [10] to study the outage probability in direct-to-satellite-IoT.

As we can see, there is a lack of experimental work that shows how LR-FHSS works in practice, and how its communication is impacted by the environment. While we have a clear view of the effect of interference and collisions in legacy LoRa (e.g., non-orthogonality of spreading factors [3, 1], capture effect [12]), we lack this information for LR-FHSS communication.

3 Motivation: Fragility of LR-FHSS Headers

In the following, we analytically compare the cause of frame loss in LR-FHSS, which can come from both header or payload loss. We consider the robust configuration (3 header replicas, CR=1/3), although the translation to the fast configuration is straightforward.

We denote p_{hdr} the probability of receiving at least one header replica. If $coll(233)$ is the probability that a collision occurs during the 233 ms of any of the three replicas sent in the robust mode, then

$$p_{hdr} = 1 - coll(233)^3.$$

On the other hand, the payload is received if at least one-third of the fragments are received in the robust configuration. We denote p_{pld} to the probability of receiving the required number of fragments. Therefore, we have:

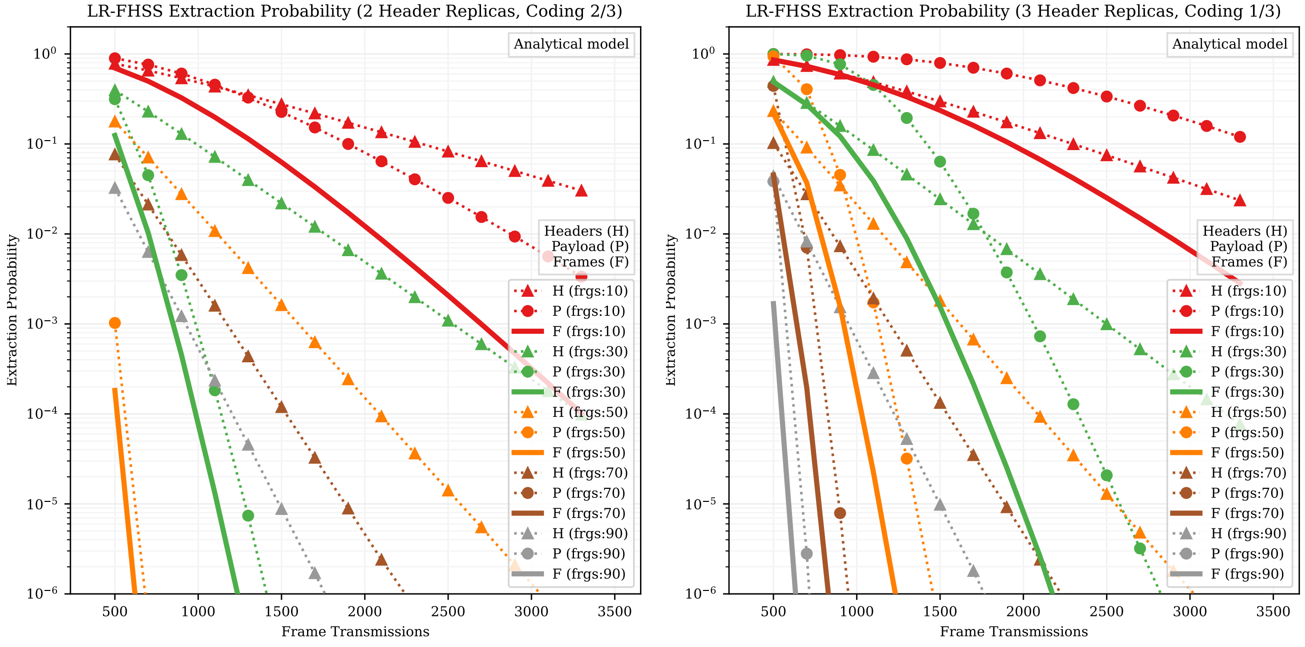
$$p_{pld} = \sum_{i=\lceil \mathcal{P}/3 \rceil}^{\mathcal{P}} \binom{\mathcal{P}}{i} . (1 - coll(102))^i . coll(102)^{\mathcal{P}-i},$$

where $\mathcal{P}$ is the number of fragments and $coll(102)$ denotes the probability that a collision occurs during the 102 ms of each fragment. Note that the term within the sum refers to the probability that exactly i fragments out of $\mathcal{P}$ are received.

Finally, an LR-FHSS frame is received if both the header and the payload are received. The probability of receiving a complete frame is thus:

$$p_{frame} = p_{hdr} . p_{pld}.$$

The probability of collision $coll(d)$ on a given OBW during a period d depends on the number of ongoing transmissions n_{og} at a given time slot. Let us assume for simplicity that all transmissions are synchronized at the slot level, where each slot spans 102 ms. If we denote by $\mathcal{C}$ the number of OBWs, the average number of free OBWs at a given time



(a) (b)

Figure 2: Analytical extraction model. Successfully extracted headers (H), payload fragments (P), and complete frames (F) for varying frame transmission count for the more aggressive configuration (a) and the most robust LR-FHSS configuration (b).

slot is equal to:

$$free(\mathcal{C}) = \mathcal{C}.(1-1/\mathcal{C})^{n_{og}}.$$

Thus, the probability that a collision occurs during a duration d is:

$$coll(d) = 1-(free(\mathcal{C})/\mathcal{C})^{\lceil d/102 \rceil}.$$

Note that if there are exactly n_{tx} full transmissions during a number $\mathcal{T}$ of 102 ms time slots, then we have:

$$n_{og} \approx (3*233+\mathcal{P}*102)n_{tx}/(102\cdot\mathcal{T}).$$

Figure 2 shows the probability of header, payload, and full frame reception for the fast configuration (Figure 2(a)) and for the robust configuration (Figure 2(b)). The probabilities are given as a function of the number of concurrent transmissions n_{tx}, during $\mathcal{T} = 1000$ time slots, and for a varying number of fragments $\mathcal{P} \in \{10,30,50,70,90\}$. In all cases, the probability of losing the header plays a significant role in the overall frame loss. This is particularly true in the robust configuration, where the header extraction probability can be lower than the payload recovery probability. For instance, for 1000 frame transmissions and a frame length of 30 fragments, headers are lost for about 10% of the frames in both configurations. However, the payload is lost for 99.9% of the frames in the fast configuration and for 60% of the frames in the robust configuration. The observed relative fragility of LR-FHSS headers is consistent with the results from [14] for DR8[1], and from [11][2].

4 Headerless Frames Recovery Methods

To address the problem of recovering headerless LR-FHSS transmissions, this section describes two solutions. The first uses an ILP model, and the second is a cost-effective heuristic. We start by describing our assumptions and framing a model on which to base these methods.

4.1 Assumptions

Channel: We focus on a single OCW and a single grid to simplify the approach. Extending the solutions to multiple OCWs and grids is trivial (e.g., one model instance per OCW/grid). Thanks to the robust GMSK modulation used for fragments, we assume that fragments always reach the gateway (there is no loss due to weak signal) and there is no noise in the channel: in other words, an OBW is always detected free when there is no transmission using it, and always detected busy if there is at least one transmission using it. We consider that fragments collide if they overlap in time and use the same OBW, that is, we consider the pessimistic case where there is no capture effect. Fragment collisions can be detected using channel sensing, but they can never be resolved with our assumption.

Frames: We assume that the number $\mathcal{P}$ of fragments per frame is known and fixed, that is, all frames have the same length. We also assume that all transmissions start and end

[1] see Figure 6 and the conclusion of [14].
[2] see Figure 8 of [11].

Table 2: LR-FHSS Constants and Variables.

Constants	
$\mathcal{C}$	Number of OBWs per grid
$\mathcal{T}$	Number of time slots
$\mathcal{S}$	Number of sequences
$\mathcal{F}$	Number of frames
$\mathcal{P}$	Number of fragments per frame
Variables	
$S = \{s \mid s \in \mathcal{S}\}$	Set of $\mathcal{S}$ sequences
$M[t][c] = \{0,1\}$	Matrix ($\mathcal{T} \times \mathcal{C}$) of observed transmissions equal to 1 if one or more transmissions were observed on OBW $c \in [1;\mathcal{C}]$ at time $t \in [1;\mathcal{T}]$
$T = \{(s_i, t_i, p_i)\}$	List of $\mathcal{F}$ 3-tuples of frame transmissions each defined by a sequence $s_i \in S$, a time $t_i \in [1;\mathcal{T}]$ and its length (in number of fragments) $p_i = \mathcal{P}$
$\hat{T}$	The frame transmission list (same structure as T) that the gateway was able to derive from observed fragment receptions in $M[t][c]$
$y_{t,s} \in \{0,1\}$	A matrix ($\mathcal{T} \times \mathcal{S}$) representation of $\hat{T}$

within the $\mathcal{T}$ time slots of the whole simulation, which is a weak assumption when $\mathcal{T}$ is much larger than the duration of a single frame. Finally, we assume a time-slotted model where each slot lasts for the duration of one fragment, and each transmission starts at the beginning of a slot.

We discuss the effect of these assumptions on the proposed solutions in Subsection 6.1. We can already mention here that some of them (e.g., no capture effect) correspond to a worst-case scenario, and are therefore pessimistic. By relaxing such assumptions, the performance of our algorithms would improve, but this requires obtaining real evaluation data on the LR-FHSS modulation.

4.2 Problem model

Our model is comprised of the following elements summarized in Table 2 and described below.

We define a series of constants to model the LR-FHSS environment. We define the number of OBWs per grid $\mathcal{C}$, the number of time slots $\mathcal{T}$, the number of possible pseudo-random sequences $\mathcal{S}$, the number of transmitted frames $\mathcal{F}$, and the number of fragments used on each frame $\mathcal{P}$.

We also define the following variables. S is a set of $\mathcal{S}$ random sequences, each representing the frequency hopping pattern (see the example of sequences from Fig. 3). Furthermore, we define a two-dimensional boolean matrix M called the observed traffic matrix, such that $M[t][c] = 1$ if and only if one or more fragment transmissions were observed on OBW c at time t. A matrix M contains all OBW in a given grid. We assume no fragment losses due to weak signal reception, but we consider the possibility of collisions. Therefore, the matrix M is composed of a series of fragment transmissions for each frame using a given sequence from $\mathcal{S}$. As a result, each frame transmission i is uniquely defined by a sequence $s_i \in S$, a starting time slot $t_i \in [1;\mathcal{T}]$, and its length $p_i = \mathcal{P}$.

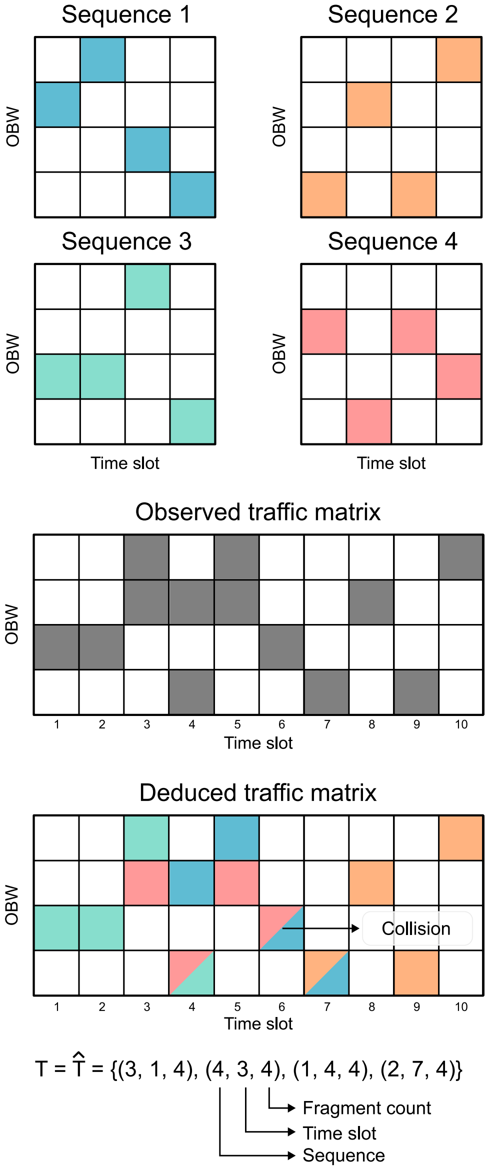


Figure 3: A simplified example of a grid with 4 OBWs and 4 frequency hopping sequences of the same length. Each sequence is represented by a different color. The observed traffic matrix $M[t][c]$ represents the channel activity detected by the gateway: each square corresponds to an $M[t][c]$ element and is colored gray if $M[t][c] = 1$, meaning an activity (one or more fragment transmissions) has been observed on OBW c at time t. The deduced traffic matrix $\hat{T} = \{(s_i, t_i, p_i)\}$) contains all the sequences that have been identified from the observed traffic matrix M. As we can see in the example each square that was gray in the observed traffic matrix M has been changed in the color of the identified sequence. If a square has more than one color, it means two fragments from different frames (and hence different sequences) collided. If all the gray squares can be colored in a given sequence it means, that all frames have been correctly identified, and hence $T = \hat{T}$. Our objective in this paper is to propose a solution that maximizes the number of identified sequences (and hence frames).

We coin T as the set of 3-tuple elements $\{(s_i, t_i, p_i)\}$, each 3-tuple defining a single frame transmission.

While T is the actual transmission set, the receiving gateway can only observe the matrix M. Indeed, M comprises all fragment transmissions without the underlying frame structure (see the observed traffic matrix in Fig. 3). In fact, the core problem of decoding headerless LR-FHSS transmissions is to correctly map observed fragment receptions from M to the actual frame transmissions of T, including the sequence, time, and length (s_i, t_i, p_i) for each frame i (see the deduced traffic matrix in Fig. 3). The goal is for the gateway to derive a frame transmission set $\hat{T}$ as close as possible to the original T. In the ideal case, all transmitted frames can be recovered when $T = \hat{T}$.

4.3 ILP Model

Based on the constants and variables from Table 2, the headerless decoding problem for the LR-FHSS modulation can be formalized as the following optimization problem.

$$\text{min.} \quad \sum_{t=1}^{\mathcal{T}} \sum_{s=1}^{\mathcal{S}} y_{t,s}$$

$$\text{s.t.} \quad \mathcal{P} \cdot y_{t,s} \geq \sum_{k=1}^{\mathcal{P}} M_{t+k-1, s[k]} \quad \forall t \in [1;\mathcal{T}], \forall s \in S.$$

For ease of manipulation by the ILP, we have introduced a new set of variables $y_{t,s}$, $\forall t \in [1;\mathcal{T}]$ and $\forall s \in S$, such that $y_{t,s} = 1$ if and only if $(s,t,\mathcal{P}) \in \hat{T}$. The resulting model aims at finding the minimum number of transmissions in $\hat{T}$ needed to cover the matrix M. To this end, the objective function aims to minimize the number of sequences in $\hat{T}$ ($y_{t,s}$) for all time slots t and sequences s. However, the set of transmissions in $\hat{T}$ must comply with a constraint that forces each frame's $\mathcal{P}$ fragments to correspond to an observed OBW occupancy of 1 in M. As observed in the previous equations, a variable number of fragments would render a non-linear model. The linearization of this non-linear model is left as future work. To solve the theoretical model with state-of-the-art linear solvers, we assume $\mathcal{P}$ is constant for all transmissions in the subsequent evaluation.

4.4 Heuristic Solution

Besides the previous exact model, we introduce a simple yet effective heuristic to find $\hat{T}$ from the input matrix M. The step-by-step process is listed in Algorithm 1. The heuristic iterates over each time slot t in an increasing manner. Then, for each sequence s among the possible set of 512 known sequences, it greedily determines whether M contains a transmission starting at this time slot t and with sequence s. If this is the case, $(s,t,\mathcal{P})$ is added to $\hat{T}$. This greedy heuristic has a time complexity of $O(\mathcal{T} \cdot \mathcal{S} \cdot \mathcal{P})$ when the full matrix M is known statically. Indeed, the main operation of line 6 is performed within three nested `for` loops, the first (line 2) being executed $\mathcal{T}$ times, the second (line 3) being executed $\mathcal{S}$ times, and the third (line 5) being executed $\mathcal{P}$ times.

Since $[1;\mathcal{T}] increasingtimeslotsexplore$, this static heuristic can be easily translated into an online heuristic with a sliding window of $\mathcal{P}$ time slots, thus resulting in a time and space complexity of $O(\mathcal{S} \cdot \mathcal{P})$ at each time slot. Note that according to the specification of LR-FHSS, we have $\mathcal{S} = 2^9$ and $\mathcal{P} \leq 32$. The sliding-window capability, combined with the reduced complexity, makes this heuristic a perfect candidate to be implemented in future LR-FHSS gateways onboard resource-constrained LEO satellites to enable the recovery of LR-FHSS frames in which headers are lost.

Algorithm 1 Greedy Headerless Decoding Heuristic.

```
Require: $\mathcal{T}$, $S$, $\mathcal{P}$ and $M$
 1: $\hat{T} \leftarrow \{\}$
 2: for $t \in [1;\mathcal{T}]$ do
 3:     for $s \in S$ do
 4:         $found \leftarrow true$
 5:         for $p \in [1;\mathcal{P}]$ do
 6:             if $M[t+p-1][s[p-1]] \neq 1$ then
 7:                 $found \leftarrow false$
 8:             end if
 9:         end for
10:         if $found = true$ then
11:             $\hat{T} \leftarrow \hat{T} \cup \{(s,t,\mathcal{P})\}$
12:         end if
13:     end for
14:     return $\hat{T}$
15: end for
```

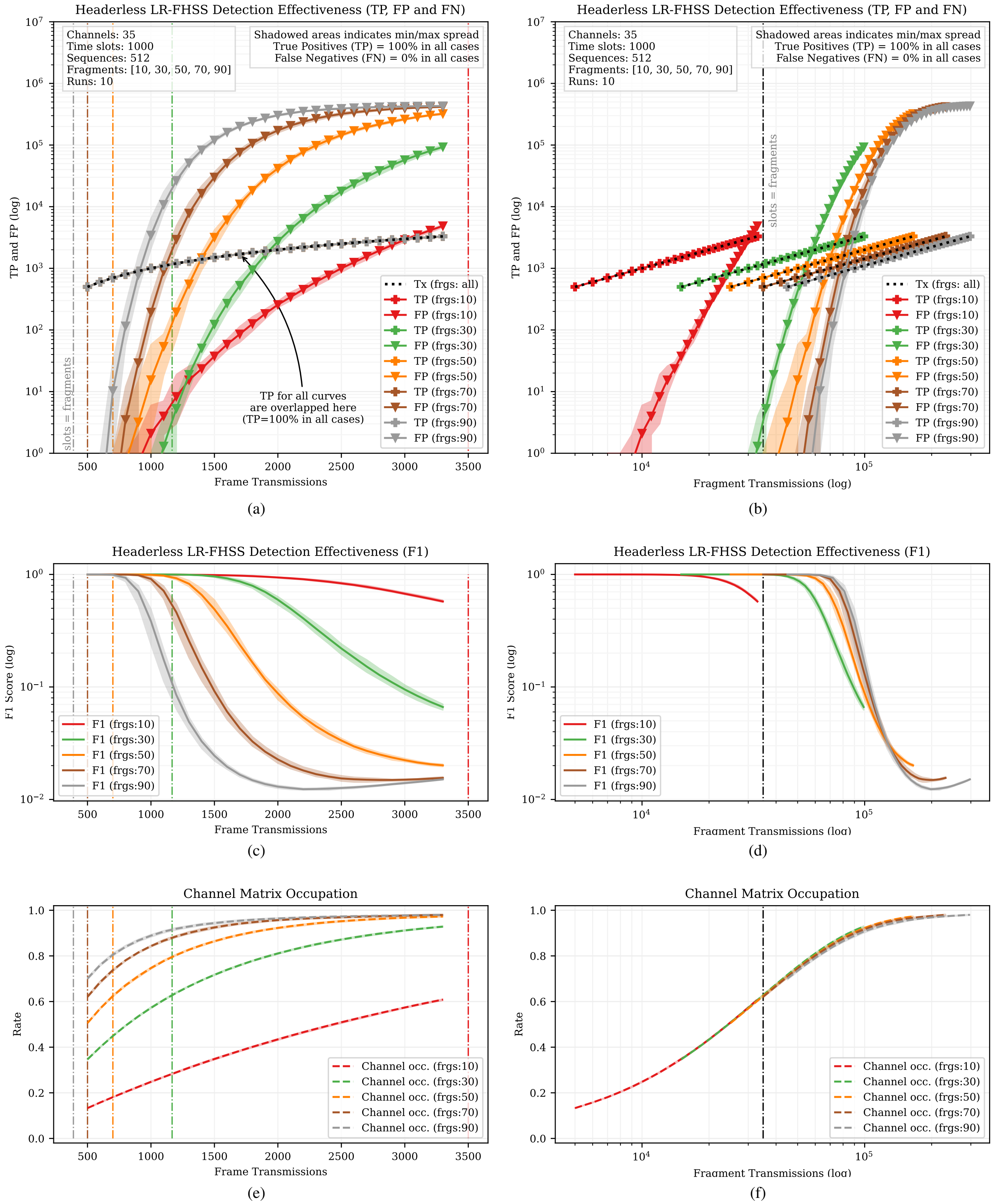
5 Simulation results

We conducted an extensive simulation campaign to evaluate the effectiveness and efficiency of the ILP model and the heuristic approach. We implemented a reference Python library that leverages an interface with Gurobi [7] to solve the ILP model. The parameters used in the following analysis are summarized in Table 3. Since the channel hopping of LR-FHSS takes place on a single grid, we considered here one grid and one OCW. The performance results can be easily generalized for LR-FHSS systems with more grids or more OCWs. We used the European settings for the number of OBWs per grid (35) and the number of sequences (512). We randomly generate a new set of sequences S and a new list of frame transmissions $\mathcal{T}$ (uniform distribution) for each simulation run. For each metric, we plot the average with lines and the minimum and maximum values with a shaded area. We allow repeated transmission elements in T (same (s,t,p) 3-tuple can occur multiple times), but not in S.

5.1 Detection Effectiveness

The metrics to assess the decoding success are based on the difference between the real traffic T and the inferred traffic $\hat{T}$. Recalling that both traffic matrices are comprised by (s_i, t_i, p_i) tuples where t_i is the time at which a frame starts with sequence s_i and with $p_i = \mathcal{P}$ fragments, the effectiveness metrics are:

- True positive (TP): $(s,t) \in T$ and $(s,t) \in \hat{T}$,
- False positive (FP): $(s,t) \notin T$ but $(s,t) \in \hat{T}$,
- False negative (FN): $(s,t) \in T$ but $(s,t) \notin \hat{T}$.
- F1-Score: $F1 = 2 \times TP/(2 \times TP + FP + FN)$.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Detection effectiveness results. True Positives (TP), False Positives (FP), and False Negatives (FN) for varying fragment count expressed vs. (a) frame transmissions and (b) total fragment count. The F1 score and OBW matrix occupation are shown in (c), (d), (e), and (f) for frame transmissions and total fragment count, respectively. Plots are valid for the ILP model and the heuristic algorithm as they provide exactly the same effectiveness results in all cases.

Table 3: Simulation Parameters.

Description	Value(s)
Number of OCW	1
Number of grids	1
Number of OBWs per grid	35
Number of time slots $\mathcal{T}$	1000
Number of sequences $\mathcal{S}$	512
Number of frames $\mathcal{F}$	500 to 3200 in steps of 100
Number of fragments $\mathcal{P}$ per frame	10 to 90 in steps of 20
Runs per step	10

The F1 Score is the harmonic mean of *precision* (accuracy of the model in identifying positive instances) and *recall* (completeness of the model in identifying positive instances), combining both metrics into a single value. Thus, high decoding effectiveness is achieved by high TP and low FP and FN. Also, a high F1 score indicates better performance and accuracy.

We analyze these metrics for varying traffic loads. Traffic load can be expressed in two ways: by the number of transmitted frames or the total fragment count. Frame decoding rate is a better indicator of the user experience, while fragment count is a direct measure of the OBW occupation of matrix M. Thus, we will use both traffic loads in the following results.

Figure 4 shows the effectiveness (TP, FP, and FN), of both the ILP and the heuristic, as a function of the traffic load. Figure 4 (a) uses frame transmission as traffic load, and Figure 4 uses fragment count (in logarithmic scale). Figure 4 (c) and (d) offer F1, a more synthetic metric for all numbers of fragments. In these figures, to indicate the channel occupation, the vertical lines represent a channel occupancy such that the number of used slots in M equals the number of fragments among all transmissions in T. As a more precise channel utilization reference, Figure 4 (e) and (f) present the channel matrix occupation measured in M. This last curve confirms that plots expressed against fragment transmissions are compared over similar channel conditions.

Results are obtained from simulations of 500 to 3300 frame transmissions (with steps of 100) for 10, 30, 50, 70, and 90 fragments each (total fragment count ranges from 5000 to 297000). The time horizon is 1000 time slots, and we assume 512 sequences over 35 OBWs as specified for LR-FHSS [9]. The resulting configuration corresponds to robust (DR8) and fast (DR9) data rates from Table 1 (coding rate and bit rate will become relevant in the extraction efficiency discussed in Section 5.2). For each step, we perform 10 simulations (random s and t for each run) and record the mean, maximum, and minimum for each metric (shadowed area in Figure 4).

The first observation from the simulation results is that the ILP model and the heuristic algorithm provide precisely the same TP, FP, and FN results. This is a promising outcome considering the algorithm can operate without relying on an ILP solver while delivering optimal performance in the aforementioned settings.

The second observation is that the (s_i, t_i, p_i) tuples from T are always contained in $\hat{T}$. In other words, TP is always 100 %, which indicates that all transmitted headerless frames can be detected at the receiver end in all evaluated cases. However, FP increases exponentially with the traffic load to the point that FP exceeds the number of TP. Consequently, the gateway will detect many incorrect frames when the channel is heavily loaded. Obtained F1 scores support this statement (Figure 4 (c) and (d)), as optimum precision/recall trade-off is provided until the channel matrix becomes heavily occupied. However, as discussed in Section 6, the CRC checks of LoRaWAN can detect and filter most of these FP frames.

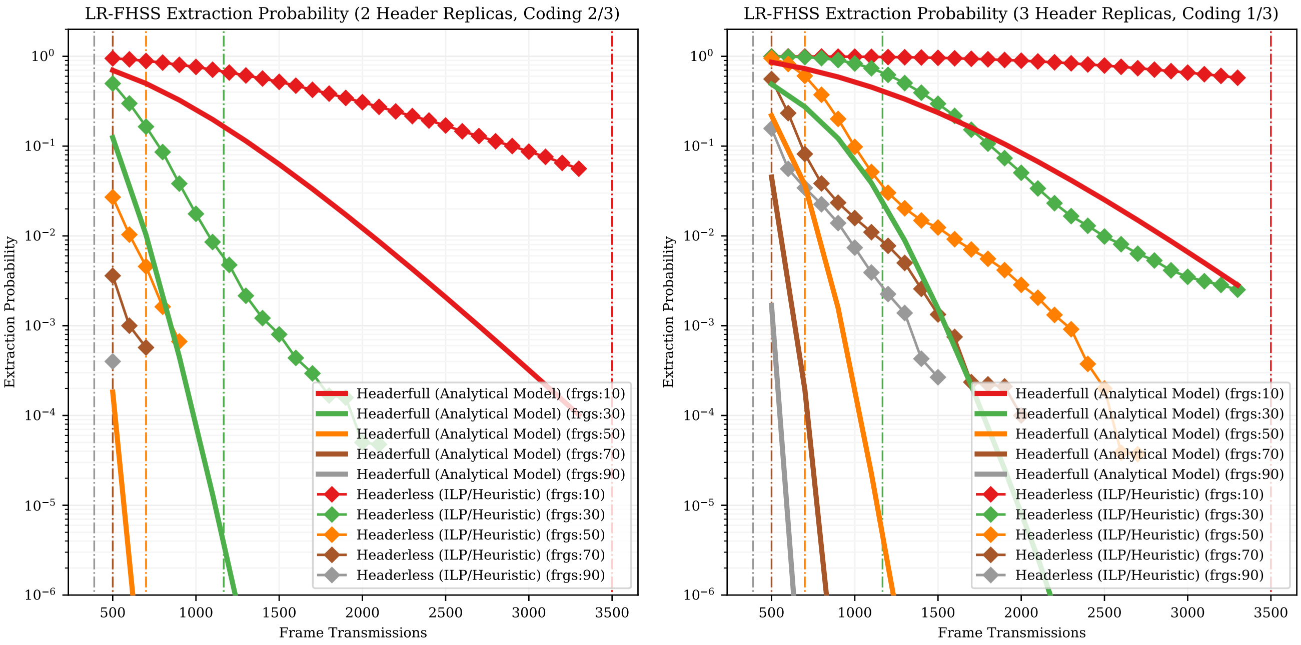
The third observation involves the impact of fragment count. At first glance, results show that transmissions with smaller frames (that is, fewer fragments) provide higher efficiency thanks to a reduced FP count (Figure 4 (a)). However, this is a direct consequence of a less occupied channel, as frames with fewer fragments require fewer slots in M. If we observe the FP measured in fragment transmissions (Figure 4 (b)), we find that for equal OBW occupation, larger frames are more effective. Indeed, larger frames exploit more extensive sequences s and have better chances of being correctly identified at the receiver. This finding motivates the utilization of frame aggregation features in the device-to-gateway link.

A better appraisal of OBW occupation can be achieved by observing the vertical lines in both Figures 4(a) and (b). These lines indicate the point at which there are as many transmitted fragments as total slots in M. Namely, if no collisions exist, M would be fully occupied at these lines. But due to collisions, M is filled at approximately 60% at this line. Of course, in Figure 4 (a), the x-position depends on the number of fragments per frame. But, in Figure 4 (b), the maximum fragment count to fill the channel is equal to 35000 ($\mathcal{C} \times \mathcal{T} = 35 \times 1000 = 35000$). Figures 4 (c) and (d) show the occupancy of the channel matrix (proportion of cells in M equal to 1) corresponding to respective plots (a) and (b).

It is thus reassuring that all transmissions in T can be identified at the receiver (TP = 100%) even on the most challenging channel loads. However, this comes at the expense of an increasingly large number of incorrect frames (FP) that must be filtered out at the gateway.

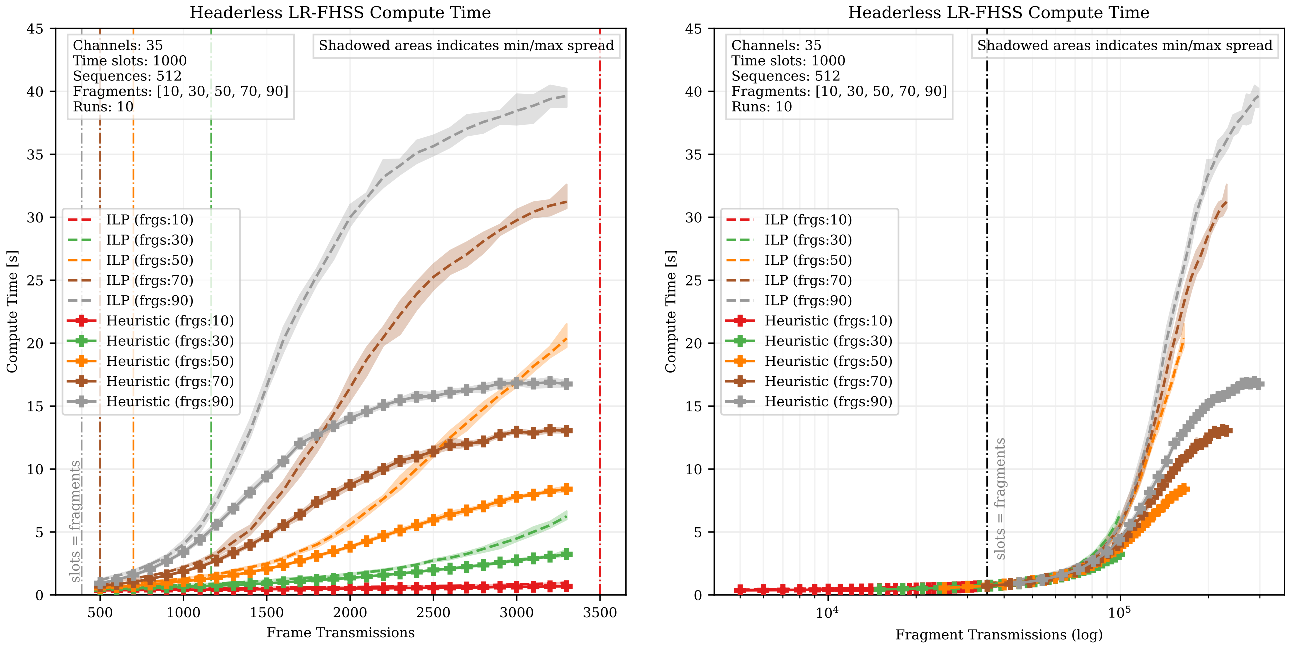
5.2 Extraction Effectiveness

Another measure of effectiveness is the number of actual transmissions that can be recovered at the gateway, also



(a) (b)

Figure 5: Extraction effectiveness results. Plots compare the headerfull extraction rate model from Section 3 with the headerless extraction rate using the ILP and the heuristic approach presented in Section 4. Results are for (a) the fast configuration and (b) the robust configuration.



(a) (b)

Figure 6: Computation efficiency results. Compute time for the ILP model and the heuristic for varying fragment count expressed vs. (a) frame transmissions and (b) total fragment count.

called the extraction probability. We consider two distinct cases in this section:

- Headerfull case: at least one header replica has to be received, and a sufficient proportion of the fragments must be received without collision. This is how LR-FHSS is specified. Data is obtained from the model in Section 3 (complete frames in Figure 2).
- Headerless case: no header replica reception is required (thanks to our algorithms achieving 100% TP), and a sufficient proportion of the fragments must be received without collision. Data is obtained from the simulation campaign introduced in Section 5.1 for the ILP model and the heuristic.

Figure 5 shows the extraction probability as a function of the number of frame transmissions for the fast configuration (see Figure 5(a)) and the robust configuration (see Figure 5(b)). For both configurations and for all values of the number of fragments per frame, the gain brought by the ILP or heuristic algorithms (termed headerless) over the legacy LR-FHSS is significant. For instance, in the fast configuration and for 2000 frame transmissions, the extraction ratio is 1% for legacy LR-FHSS and 30% for our heuristic. This shows that decoding headerless LR-FHSS frames can bring huge gains.

5.3 Computation Efficiency

Figure 6 indicates the computation time required to execute the ILP model and heuristic algorithm in practice.

The primary metric we analyze is the compute (wall-clock) time that spans the execution of each solution. To this end, we executed a series of benchmarking runs on the same hardware. Specifically, we leveraged a WSL2 system running Python 3.8.10 and Gurobi 10.0.0 on an AMD Ryzen 9 5900HS clocking at 3.30 GHz with 16 GB of RAM.

To ease the comparison, Figure 6 presents the compute time for the same scenarios as in the previous sub-section (parameters in Table 3). We present the compute time for both the ILP model (dashed lines) and the heuristic algorithm (solid lines) as a function of the number of frame transmissions (Figure 6(a)) and total fragment count (Figure 6(b)).

The first observation is that the ILP model can take up to 40 seconds in compute time to solve the most challenging cases with 1000 time slots and 3300 concurrent transmissions. However, as expected, the heuristic algorithm delivers the same effectiveness with 17 seconds in the worst case. Therefore, we observe that the algorithmic solution is 2.3 times more efficient than state-of-the-art solvers.

The second observation is that the compute time scales primarily with the fragment count rather than the transmitted frames. This is reasonable, considering that the combinatorial problem complexity depends on the number of occupied slots in M. However, we observe a secondary (weaker) correlation with the number of fragments per frame. A larger number of fragments per frame seems to create more challenging problems for the same number of fragments (but also higher effectiveness, as discussed in the previous section).

6 Discussion

In this section, we first discuss the limits of the proposed solutions considering the assumptions we made. Then, we propose a way to mitigate false positives.

6.1 Model Assumptions

6.1.1 *Known Fragment Count*

In LR-FHSS, the fragment count of a frame is stored in each replica of the header. In this study, we assumed that the number of fragments $\mathcal{P}$ is known and constant for each frame. Thus, this number does not have to be guessed when all replicas are lost. However, LR-FHSS is robust when handling frames of under-estimated (or even over-estimated) length, as the payload can be decoded even if 33% (CR=2/3) or 67% (CR=1/3) of the fragments are lost. We distinguish two cases:

1. The identified sequence is too short: as long as a sufficient proportion of the first fragments are received and identified, the frame can still be decoded.
2. The identified sequence is too long: the trailing fragments will be ignored, and the frame will be decoded without any problems.

This assumption simplifies our model and keeps it linear. Removing this assumption would require the model and the heuristic to deduce the fragment count. For the heuristic, this requires stopping the innermost `for` loop (see line 5 of Algorithm 1) whenever the channel is detected free. For the ILP model, a linearization strategy is needed for the term $\mathcal{P} \cdot y_{t,s}$ which would involve the product of a binary variable and a bounded variable in $\mathbb{N}$.

6.1.2 *No Capture Effect and Good Fragment Detection*

We did not consider the capture effect: when two fragments collide, we consider that none can be retrieved. This is particularly relevant for the extraction effectiveness results summarized in Figure 5. In practice, however, likely, a transmission with a significantly larger signal strength than a weak colliding transmission will be retrievable, thus improving the number of recoverable frames compared to our results. Determining the actual capture effect threshold requires further investigation of the LR-FHSS modulation.

We also assumed that fragments can always be detected (but not recovered in case of collision) at the gateway, even if they collide. For this assumption, we rely on the good robustness of the GMSK modulation, which is the underlying modulation for the fragments [8]. For the heuristic, bad fragment detection (including due to noise) can be taken into account by adding $(s,t,\mathcal{P})$ in the set $\hat{T}$ if a sufficient proportion of the corresponding OBWs are detected busy, rather than all corresponding OBWs.

6.1.3 *Time Slotting*

Our strongest assumption is that transmissions are slotted at the fragment level, that is, every 102.4 ms. Transmissions will not be slotted in practice, and our algorithms must deal with arbitrary transmission times.

When a transmission is not aligned with the slots, each fragment will overlap with two slots. The receiving energy of the fragment is distributed into the two slots based on the overlap time. Suppose the overlap between an unaligned fragment and a slot is small. In that case, the energy of this fragment on the slot becomes negligible, and our assumption applies as the fragments are nearly aligned. If the overlap between an unaligned fragment and a slot is large (up to half of the duration of the slot), the model could detect the residual energy during two consecutive slots and incorrectly assume that transmission occurred during these two slots. Dealing with arbitrary transmission times remains challenging and requires the research on LR-FHSS to progress on characterizing its capture effect capabilities and its tolerance to desynchronization.

6.2 False Positive Filtering

The two algorithms produce many FP when the channel is heavily loaded. The gateway will consider each FP as a frame with an arbitrary payload. Two practical strategies can mitigate the impact of these fake frames.

On the one hand, it is possible to use the cyclic redundancy check (CRC) present in the LoRaWAN frame to check the *validity* of the received payload. Each LoRaWAN frame contains a CRC calculated on the whole payload. If the CRC value in the received frame differs from the CRC computed on the decoded payload, the gateway will drop the frame. Thus, many fake FPs will be removed due to incorrect CRC checks at the gateway. With a 16-bit CRC, on average, only one FP every 2^{16} would still be accepted. For one million FP, the average number of *fake* frames would be about 15.

On the other hand, FP occurs when a sequence fully overlaps with existing transmissions. If this occurs, it means that most of the sequence slots will collide, and thus the gateway will not be able to retrieve actual data from the colliding fragments, and the sequence will not translate into a valid frame. Thus, it is likely that FP will produce corrupted (fake) frames.

7 Conclusion

In this paper, we aim to improve LR-FHSS throughput by enabling the recovery of frames whose header was lost. First, we showed that header loss is a frequent cause of frame loss in LR-FHSS. Then, we proposed an exact ILP and a low-cost heuristic to detect and extract headerless frames profiting from the *a priori* known sequence set. We showed through extensive simulations that our heuristic consistently achieves optimal results in many configurations while incurring tractable computation costs. Thus, we believe that our heuristic can successfully be implemented on LR-FHSS gateways located onboard of LEO satellite and help deal with the massive amount of incoming concurrent transmissions. Our future work involves weakening some of the assumptions of this initial work, including a more realistic channel model and transmission schedule.

This work encourages more experimental work on LR-FHSS to help improve the current models, e.g., giving thresholds for capture effect.

Acknowledgement

This research has received support from the European Union's Horizon 2020 R&D program under the Marie Skłodowska-Curie grant agreement No 101008233 (MISSION project), the French National Research Agency (ANR) under the projects ANR-22-CE25-0014-01 and ANR-21-CE25-0002-01.

8 References

[1] F. Benkhelifa, Y. Bouazizi, and J. A. McCann. How orthogonal is LoRa modulation? *IEEE Internet of Things Journal*, 9(20):19928–19944, 2022.

[2] G. Boquet, P. Tuset-Peiró, F. Adelantado, T. Watteyne, and X. Vilajosana. LR-FHSS: Overview and performance analysis. *IEEE Communications Magazine*, 59(3):30–36, 2021.

[3] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. Impact of LoRa imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters*, 22(4):796–799, 2018.

[4] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad. Satellite communications supporting Internet of Remote Things. *IEEE Internet of Things Journal*, 3(1):113–123, 2016.

[5] 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 *ADHOC-NOW*, pages 241–258, 2019.

[6] J. A. Fraire, O. Iova, and F. Valois. Space-terrestrial integrated internet of things: Challenges and opportunities. *IEEE Communications Magazine*, 2022.

[7] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[8] S. Lebreton. A crash course in long-range frequency hopping (LR-FHSS) spread spectrum. `https://www.youtube.com/watch?v=QbH28P6c-qM`, 2022. Accessed: 12-12-2022.

[9] LoRa Alliance Technical Committee Regional Parameters Workgroup. Lorawan regional parameters. Technical Report RP002-1.0.3, LoRa Alliance, 5 2021.

[10] A. Maleki, H. H. Nguyen, and R. Barton. Outage probability analysis of LR-FHSS in satellite IoT networks. *IEEE Communications Letters*, 2022.

[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] U. Noreen, L. Clavier, and A. Bounceur. LoRa-like CSS-based PHY layer, capture effect and serial interference cancellation. In *European Wireless Conference*, pages 1–6, 2018.

[13] Semtech. Application note: LR-FHSS system performance, SX1261/62/ LR1110. Technical Report AN1200.64, Rev. 1.2, Semtech, 2 2022.

[14] M. A. Ullah, K. Mikhaylov, and H. Alves. Analysis and simulation of LoRaWAN LR-FHSS for direct-to-satellite scenario. *IEEE Wireless Communications Letters*, 11(3):548–552, 2022.