A Channel Selection Game for Multi-Operator LoRaWAN Deployments

Kinda Khawam[1], Hassan Fawaz[1], Samer Lahoud[2], Odalric-Ambrym Maillard[3], and Steven Martin[4]

[1]Université Paris-Saclay, UVSQ, 78000, Versailles, France
[2]Université Saint-Joseph de Beyrouth, ESIB, CIMTI, Beirut, Lebanon
[3]INRIA Lille - Nord Europe, Team SequeL, France
[4]Université Paris-Saclay, LRI, 91190, Gif-sur-Yvette, France

***Abstract*—With over 75 billion Internet of Things (IoT) devices expected worldwide by the year 2025, inaugural MAC layer solutions for long-range IoT deployments no longer suffice. LoRaWAN, the principal technology for comprehensive IoT deployments, enables low power and long range communications. However, synchronous transmissions on the same spreading factors and within the same frequency channel, augmented by the necessary clustering of IoT devices, will cause collisions and drops in efficiency. The performance is further impacted by the shortage of radio resources and by multiple operators utilizing the same unlicensed frequency bands. In this paper, we propose a game theoretic based channel selection algorithm for LoRaWAN in a multi-operator deployment scenario. We begin by proposing an optimal formulation for the selection process with the objective of maximizing the total normalized throughput per spreading factor, per channel. Then, we propose centralized optimal approaches, as well as distributed algorithms, based on reinforcement learning and regret matching dynamics, to finding both the Nash and Correlated equilibria of the proposed game. We simulate our proposals and compare them to the legacy approach of randomized channel access, stressing their efficiency in improving the total normalized throughput as well as the packet delivery ratios.**

***Index Terms*—LoRaWAN, Game Theory, Nash Equilibrium, Correlated Equilibrium, Learning**

I. Introduction

LoRaWAN is the one of the principle IoT technologies in the unlicensed radio band. It aims to provide low-cost, large-scale, and ultra-durable connectivity. It is designed to allow low-powered devices to communicate with the access network over long-range wireless connections. Transmission is possible on selected channels with the usage of spread-spectrum modulations derived from the chirp spread spectrum (CSS) technique. A collision will occur when two or more devices select, at the same time, the same channel and the same spreading factor (SF). Such collisions are inescapable due to the use of Aloha-based random access and the scarcity in radio resources. This shortcoming is further magnified by the cohabitation of multiple LoRaWAN operators leading to a massive deployment of LoRa devices. To limit the interference on the industrial, scientific, and medical (ISM) license-free band, LoRaWAN transmissions are restricted to a 1% duty cycle, but this solution is not enough to curtail inter-network interferences.

In this work, we address the scalability issues of LoRaWAN by astutely apportioning the traffic of various operators over the available channels. We propose a game for channel assignment in a multi-operator LoRaWAN deployment. We portray the channel allocation problem as a non-cooperative game among *selfish but rational operators*, each seeking to improve its own normalized Aloha throughput. We show that the game possesses a potential function. As a result, a pure Nash equilibrium (NE) can be attained using either repeated best response or replicator dynamics. In particular, our replicator dynamics learning-based algorithm is completely distributed and avoids signaling burdens, allowing thus for a prolonged device battery life.

We additionally investigate the correlated equilibrium (CE) of our game proposal. A CE is a concept that is more general than an NE. It is the idea that each player of the game will choose their action according to their observation of a public signal shared among all players [1]. Roger Myerson, an economics Nobelist and game theory expert, wrote "If there is intelligent life on other planets, in a majority of them they would have discovered correlated equilibrium before Nash equilibrium". The CE is a relevant solution for non-cooperative games, and it is a particularly interesting one as it is much easier to obtain than its Nash counterpart and sometimes even results in better payoffs.

As with our approach in finding the NE, we introduce both centralized and distributed approaches to computing the CE. First, we use linear optimization for computing the correlated equilibrium as well as for finding the social welfare CE, a Pareto optimal solution. Second, we use regret matching–a distributed learning mechanism–to find the CE in a decentralized manner. We study our different approaches to computing the NE and the CEs and compare them to the concurrent method for LoRaWAN channel selection.

The remainder of this work is structured as follows: section II includes the related works from the state-of-the-art. In section III, we detail our system model. We introduce our optimal formulation for channel assignment in section IV and our non-cooperative game proposal for channel selection in section V. After proving its existence, we propose a repeated best response algorithm for finding the NE in section V-D, as well as a distributed learning based algorithm with the same purpose in section V-E. We introduce the notion of correlated equilibrium in section VI, propose an optimal method of finding it in section VI-A, and a learning based algorithm to compute it in section VI-C. Simulations and results are shown in section VII. This paper is concluded with section VIII.

II. Related Works and Contributions

A lot of papers in the state-of-the-art discuss the capabilities, as well as limitations, of LoRa and LoRaWAN networks. In [2], the authors address the scalability of LoRaWAN by implementing the legacy approach using an ns3 simulator. They conclude that downstream traffic degrades the packet delivery ratio and that increasing the number of gateways can improve performance. The article in [3] illustrates that while adaptive data rate (ADR) SF selection reduces IoT device power consumption, it deters the scalability of LoRaWAN. In [4], the authors put forward a monitoring system for LoRa architecture. They use a smart gateway to study the performance of LoRa under multiple scenarios. Finally, in [5] the authors assess the impact of ISM interference on the bit error rate. They establish a signal-to-interference ratio threshold after which the impact of narrow-band interference becomes minimal.

Many articles in the state-of-the-art have proposals aimed at improving the resilience, efficiency, and overall scalability of LoRaWAN networks. They do so by focusing on a range of LoRaWAN features and aspects. The majority of these works as in [6]–[8], focus on spreading factor assignment algorithms beyond the legacy approaches proposing ideas based on decision trees, interference-awareness, and device radio conditions, respectively.

While not as common in the state-of-the-art, other approaches tackle LoRaWAN problems from the perspective of the ISM channels. In [9], the authors propose a lightweight scheduling algorithm wherein the devices choose the channels they transmit on based on what they deem to be best. They show via simulations that in a single cell scenario with 1000 nodes, their proposal can reduce the packet error ratio of the legacy LoRaWAN by 20%.

In [10], the authors propose a channel control scheme for LoRa networks. Their objective is to balance the traffic loads across all available channels lowering thus the amount of collisions. They claim that their algorithm outperforms traditional and state-of-the-art proposals in dense deployment scenarios.

The authors in [11] propose a scheduling algorithm to improve the scalability of LoRaWAN. Their algorithm schedules the spreading factors, frequency channels, and the timeslots for the wireless links connecting end-devices and gateways. They compare their proposal to the Aloha access scheme and highlight its advantages.

Finally, in [12], the authors study a Multi-Armed Bandit proposal for channel assignment in LoRaWAN. Their experimental results show that each IoT device selects a channel based on a reinforcement learning approach that aims to improve the packet delivery rate.

In our work, we simulate multiple co-located network operators unlike the vast majority of the state-of-the-art, wherein papers consider single operator deployments [3], [6], [7], [9], [11], [13]. As we are dealing with competing operators, we propose a game theory based channel selection algorithm to load balance the traffic. The different operators are the players of our game. Our proposal for channel assignment does not require device intelligence as in [9], [12] and can as such be implemented on all LoRa device classes. Furthermore, it is not based on classical scheduling as in [11], an unrealistic ask of LoRa networks which are limited to a 1% duty cycle.
We highlight our main contributions:

(a) We introduce an algorithm, based on non-cooperative game theory, for channel selection in a multi-operator LoRaWAN setting. We show that the game possesses a potential function and thus admits both Nash and Correlated equilibria.
(b) We simulate a semi-distributed approach, based on best response dynamics, for finding the NE. Additionally, we propose a reinforcement learning approach to finding the NE in a distributed manner.
(c) We propose centralized and distributed algorithms for finding the CE. Using linear optimization, we introduce an approach for finding the social welfare CE, a Pareto optimal solution. Using regret matching, we propose a learning algorithm to find the CE in a distributed manner.
(d) We simulate our different algorithms and compare them to the legacy state-of-the-art approach, highlighting the gains they achieve with respect to the latter and to each other.

III. System Model and Specifications

In our model, we assume the presence of multiple LoRaWAN deployments, each belonging to a different operator. Every LoRaWAN deployment $i \in \mathcal{N}$ has N^i IoT LoRa nodes and r^i gateways. The gateways for each LoRaWAN deployment are coincident and set at the same locations. The IoT devices are spread over a square shaped zone as shown in Fig. 1.

The IoT nodes attempt to transmit packets at a rate λ^i. These attempts are done following a Gaussian distribution. T_s is the time required to transmit a packet of size l on SF s. LoRa supports SFs ranging from 7 to 12. The SFs institute a trade-off between device coverage and data rate. SF7 results in the highest data rate but at the shortest distances. The SFs are assumed to be orthogonal. Packets which are transmitted during the same time frame but on different spreading factors can be successfully received. LoRa utilizes forward error correction to detect and correct transmission errors with the coding rate set to 4/(C+4) where $C \in \{1,2,3,4\}$. Table I shows how the data rate, sensitivity and the SNR thresholds for reception vary as a function of the spreading factors utilized within the 868 MHz band for C = 1.

LoRaWAN is the upper layer protocol for LoRa. It uses a star topology. Each network deployment will connect to its own network server using one hop communications, where all the gateways forward the packets to the server. An example

Table I: LoRa Technology in the 868 MHz band

SF	**Data Rate** [kbps]	**Sensitivity** [dBm]	**SNR** [dB]
7	5.458	-123	[-7.5,∞[
8	3.125	-126	[-10,-7.5[
9	1.757	-129	[-12.5,-10[
10	0.976	-132	[-15,-12.5[
11	0.537	-134.5	[-17.5,-15[
12	0.293	-137	[-20,-17.5[
ϕ	0	Not covered	$<$-20

architecture of this network can be seen in Fig. 2.

LoRaWAN uses pure Aloha for channel access. The duty cycle for LoRa within the ISM band is limited to d=1%. The rate of packet generation should as such verify $\lambda T_s \leq d$. Multiple frequency bands are supported by LoRaWAN within the unlicensed ISM bands of 433, 868, and 915 MHz. The 868 MHz band is used in Europe alongside the 125, 250, and 500 kHz bandwidth channels.

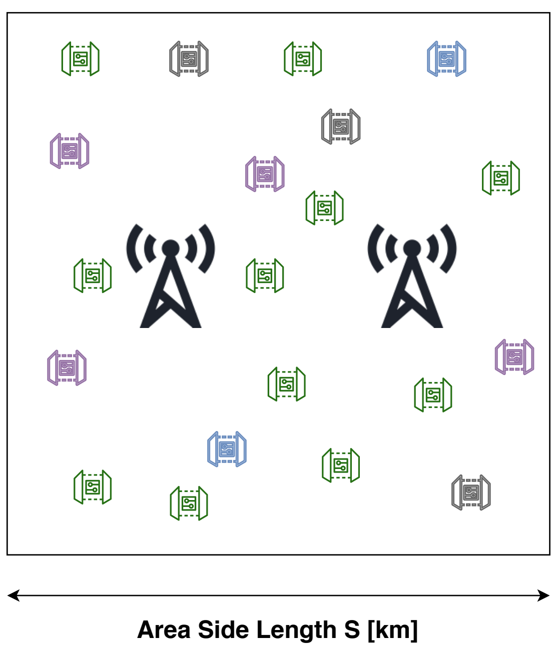


Figure 1: A 2 gateway 4 LoRaWAN scenario

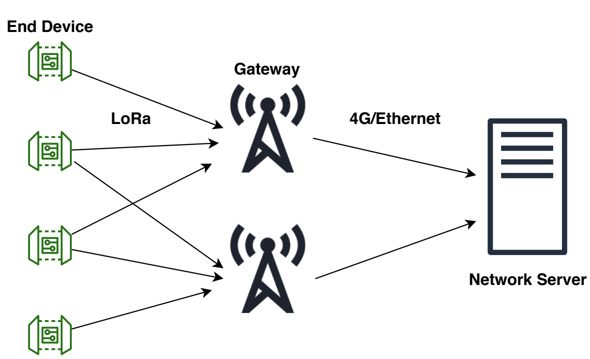


Figure 2: LoRaWAN architecture

In Fig. 3, we show a network stack of a typical LoRa node. The latter can be divided into three classes based on how often they are listening for updates:

- Class A: Consuming the least amount of energy, these devices wake up to send a packet and afterwards listen for a short duration of time.
- Class B: Moderate on energy consumption, these devices have regularly scheduled listening windows.
- Class C: Causing maximum energy consumption, these types of devices are always listening.

Unlike the vast majority of the state-of-the-art proposals for channel management in LoRaWAN, our proposals do not require any additional signaling with respect to current class A specifications. The LoRa devices can simply receive a notification with their channel assignment during their receive windows. Finally, Table II has a summary of the notations concerning our system model.

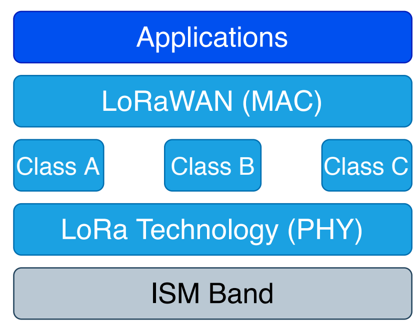


Figure 3: LoRa node network stack

Table II: Notation Summary

Notation	**Definition**
λ^i	Rate of device packet generation in network i
λ^e_{sc}	External traffic on channel c and spreading factor s
T_s	Time needed to transmit a packet on SF s
l	Packet length in bytes
d	Duty cycle
N^i_s	Number of covered nodes that can using SF s for operator i
G_{sc}	Total normalized traffic load on SF s and channel c
$\mathcal{N}$	Set of operators
$\mathcal{C}$	Set of available channels
$\mathcal{T}$	Total normalized throughput

IV. Optimal Problem for Channel Selection

In our work, we simulate a set of co-located operators. The devices belonging to each operator use one of the available channels of the set $\mathcal{C}$. The normalized channel traffic load per SF, and per channel, G_{sc} is defined as:

$$G_{sc} = \lambda^e_{sc} \cdot T_s + \sum_{i=1}^{\mathcal{N}} x^c_i (\lambda^i \cdot N^i_s \cdot T_s), \qquad (1)$$

where x^c_i is a binary variable that is equal to one if network operator i is using channel c, and zero otherwise. With the success rate, per SF and per channel, being expressed as

$\exp(-2G_{sc})$ following the Aloha protocol, the total normalized channel traffic load for a certain network i, G^i_{sc}, can be defined as:

$$G^i_{sc} = x^i_c(\lambda^i \cdot N^i_s \cdot T_s). \quad (2)$$

The total normalized throughput in the network is as such:

$$\mathcal{T} = \sum_{i=1}^{\mathcal{N}}\sum_{c=1}^{\mathcal{C}}\sum_{s=1}^{S} G^i_{sc}\exp(-2G_{sc}). \quad (3)$$

The global optimal formulation to maximize the total normalized throughput through channel allocation, given a predefined SF selection process, can thereafter be written as:

$$\underset{x^i_c}{\text{Maximize}} \sum_{i=1}^{\mathcal{N}}\sum_{c=1}^{\mathcal{C}}\sum_{s=1}^{S} G^i_{sc}\exp(-2G_{sc}) \quad (4a)$$

Subject to

$$\sum_{c=1}^{C} x^i_c \leq 1, \quad \forall i \in \mathcal{N}, \quad (4b)$$

$$x^i_c \in \{0,1\} \quad \forall i \in \mathcal{N}, \forall c \in \mathcal{C} \quad (4c)$$

(4a) is the objective of the optimal problem: to maximize the total normalized throughput in the entire system. The constraints in (4b) and (4c) enforce that a network operator chooses a particular channel c to utilize for transmissions.

Unfortunately, the optimal problem in its current state can only be solved using a centralized entity. The latter has to have all the information on all the different LoRa devices regardless of which operator they belong to. With the IoT devices and gateways belonging to different operators, it is unrealistic to expect different network operators to share the such sensitive data. In what follows, we study, assess, and solve the problem at the operator level.

V. Multi-Operator Game for Channel Selection

Our objective is to utilize non-cooperative game theory to formulate an algorithm for channel assignment in multi-operator LoRaWAN networks. Each network seeks to select the channel(s) that maximizes its own normalized total throughput. The different network operators, *i.e.*, the players of our game proposal, are competing for seemingly contradicting objectives. As such, non-cooperative game theory is well adapted to channel selection in multi-operator LoRaWAN deployments.

A. Game Formulation

We define a multi-player game $\mathcal{G}$ among the different network operators present. The formulation of this game $\mathcal{G} = \langle \mathcal{N}, \prod_i S_i, U_i \rangle$ can be described as follows:

- A finite set of players $i \in \mathcal{N}$, the set of operators.
- The action of a certain player is the channel chosen, the strategy chosen by an operator i is then $\boldsymbol{x}^i = (x^i_1, ..., x^i_C)$, indicating whether it utilizes channel c or not.
- For each player i, the space of pure strategies is S_i, such that $S_i = \{\boldsymbol{x}^i \in \{0,1\}^C | \text{ (4b)}, \ \forall c = 1, ..., C\}$.
- A set of utility functions $(U_{i\in\mathcal{N}})$ that quantify the players' profit for a given strategy profile.

B. Player Utilities

Each player, or operator, seeks to maximize greedily the total of its own normalized throughput. The utility U_i for every player is formulated as:

$$U_i = \sum_{c=1}^{\mathcal{C}}\sum_{s=1}^{S} G^i_{sc}\exp(-2G_{sc}) \quad (5)$$

where $\exp(-2G_{sc})$ is the success rate per SF and per channel for all the different co-located operators, and G^i_{sc} represents the traffic load for network i. The former can be determined by the operators through packet acknowledgments, while the latter is computed using the network's knowledge of its own devices and their transmission rates. Each player is greedily seeking to maximize its own normalized throughput.

C. Existence of a Nash Equilibrium

In game theory, the players aim to find a solution to which they all adhere. The latter is known as a Nash equilibrium (NE) [14]. An NE represents a profile of player strategies wherein no player can take advantage of the others by changing its own strategy in a unilateral fashion. As such, the primary task in game theoretics is to put froward algorithms capable of reaching such equilibrium. A classical, and yet simple, approach to computing the NEs is known as the repeated best response dynamics: each player will select its locally optimal strategy in reply to the other players, until the dynamic response algorithm converges. In what follows we aim to prove that an NE exists for our game.

For simplicity, we denote $w^i_s = \lambda^i \cdot N^i_s \cdot T_s$. The utility of a player i can then be expressed as:

$$U_i = \sum_{c=1}^{\mathcal{C}}\sum_{s=1}^{S} x^i_c \cdot w^i_s \cdot \exp(-2(w^i_s + \sum_{j\neq i} x^j_c \cdot w^j_s + \lambda^e_{sc} \cdot T_s)). \quad (6)$$

Lemma V.1. *Our game proposal is exact potential.*

Proof. To verify that an NE exists for this game, we assume $j \in \mathcal{N}$ and define the vector of functions $\vec{\phi}$ of elements ϕ_s such that:

$$\phi_s = -\sum_j\sum_c \frac{x^j_c}{\sum_j x^j_c} \cdot \exp(-2(\sum_j x^j_c \cdot w^j_s + \lambda^e_{sc} \cdot T_s)). \quad (7)$$

We additionally define the vector $\vec{\alpha}^i$ of elements α^i_s such that:

$$\alpha^i_s = w^i_s \cdot \frac{\exp(-2w^i_s)}{1 - \exp(-2w^i_s)}, \quad (8)$$

and the value l_{sc}, the load of all the networks except i:

$$l_{sc} = \sum_{j\neq i} w^j_s \cdot x^j_c. \quad (9)$$

We aim to compute the value $\vec{\alpha}^i \cdot \vec{\phi}(\boldsymbol{x}^i, \boldsymbol{x}^{-i}) - \vec{\phi}(\boldsymbol{x}'^i, \boldsymbol{x}'^{-i}) \cdot \vec{\alpha}^i$ to prove that it is essentially the difference in the payout resulting from player i changing its decision alone from channel c to channel c'. x^{-i} is the strategy of all other players different than i.

$$\vec{\alpha}^i \cdot \vec{\phi}(\boldsymbol{x}^i, \boldsymbol{x}^{-i}) - \vec{\alpha}^i \cdot \vec{\phi}(\boldsymbol{x}'^i, \boldsymbol{x}'^{-i}) =$$

$$
\begin{aligned}
&-\sum_j \sum_s \alpha_s^i \cdot \frac{x_c^j}{\sum_j x_c^j} \cdot \exp(-2(l_{sc} + w_s^i + \lambda_{sc}^e \cdot T_s)) \\
&-\sum_j \sum_s \alpha_s^i \cdot \frac{x_{c'}^j}{\sum_j x_{c'}^j} \cdot \exp(-2(l_{sc'} + \lambda_{sc'}^e \cdot T_s)) \\
&+\sum_j \sum_s \alpha_s^i \cdot \frac{x_c^j}{\sum_j x_c^j} \cdot \exp(-2(l_{sc} + \lambda_{sc}^e \cdot T_s)) \\
&+\sum_j \sum_s \alpha_s^i \cdot \frac{x_{c'}^j}{\sum_j x_{c'}^j} \cdot \exp(-2(l_{sc'} + w_s^i + \lambda_{sc'}^e \cdot T_s)) \\
&= -\sum_s \alpha_s^i \cdot \exp(-2(l_{sc} + \lambda_{sc}^e \cdot T_s))(\exp(-2w_s^i) - 1) \\
&+\sum_s \alpha_s^i \cdot \exp(-2(l_{sc'} + \lambda_{sc'}^e \cdot T_s))(\exp(-2w_s^i) - 1) \\
&= \sum_s \frac{(1-\exp(-2w_s^i))}{(1-\exp(-2w_s^i))} \cdot \Big(w_s^i \big[\exp(-2(l_{sc} + w_s^i + \lambda_{sc}^e \cdot T_s)) \\
&\qquad - \exp(-2(l_{sc'} + w_s^i + \lambda_{sc'}^e \cdot T_s)) \big] \Big) \\
&= \sum_s \Big(w_s^i \big[\exp(-2(l_{sc} + w_s^i + \lambda_{sc}^e \cdot T_s)) \\
&\qquad - \exp(-2(l_{sc'} + w_s^i + \lambda_{sc'}^e \cdot T_s)) \big] \Big) \\
&= U_i(\boldsymbol{x}^i, \boldsymbol{x}^{-i}) - U_i(\boldsymbol{x}'^i, \boldsymbol{x}'^{-i})
\end{aligned}
\tag{10}
$$

□

As a potential function exists for our proposed game, pure Nash equilibrium exist and best response dynamics are guaranteed to converge [15]. We prove, via simulations, their efficiency in terms of total normalized throughput and packet delivery ratio.

D. Best Response to Computing the Nash Equilibrium

As we proved that our proposal possesses a potential function, pure NEs exist and we can attain them using repeated best response dynamics. We implement a best response approach where in every iteration t_g, a network i seeks to find its locally optimal channel selection as a response to $\boldsymbol{x}_c^{-i}(t_g - 1)$, the decisions of other operators, by solving the following problem:

$$\underset{x_c^i}{\text{Maximize}}\ U_i(\boldsymbol{x}^i, \boldsymbol{x}^{-i}) \tag{11a}$$

Subject to

$$x_c^i \in \{0,1\} \ \ \forall i \in \mathcal{N}, \forall c \in \mathcal{C} \tag{11b}$$

The best response algorithm we proposed can be seen in Algorithm 1. The approach requires less than 5 iterations to converge. In order to solve the optimization problems in our work, we made use of mixed integer disciplined convex programming from CVX [16], a Matlab based optimization tool. CVX itself, in this instance, uses branch and bound to get the optimal solution.

Algorithm 1 Game for Channel Selection

```
Requires: Maximum tolerance ϵ ← 10^-5
Input: Initial channel assignment
Initialize: t_g = 0
Do
    t_g ← t_g + 1
    For i=1,...,N
        Solve the problem in (11) for i
        Update channel selection
        Compute δ^i = ||x_c^i(t_g) − x_c^i(t_g − 1)||
    End For
While ∃ i such that δ^i ≥ ϵ
```

While the best response algorithm represents a comprehensive semi-distributed approach to computing the NE, we are also interested in a completely distributed approach that limits the need for major signaling or inter-operator cooperation.

E. Distributed Learning-based Approach to Computing the Nash Equilibrium

We aim to use reinforcement learning, namely replicator dynamics [17], to help solve the problem in a distributed manner. Because our game possesses a potential function, we know that replicator dynamics will converge to a pure Nash equilibrium [18]. Let $p_c^i(t)$ be the probability that network i utilizes channel c for transmissions. The sum of all the probabilities is equal to 1, for each network i *i.e*, $\sum_{c \in \mathcal{C}} p_c^i(t) = 1\ \forall\, i \in \mathcal{N}, \forall\, t$. At $t = 1$, the beginning of the learning process, all possible channel selections have equal probabilities. At every time step t, the network chooses a particular channel based on the probabilities draw, and is subsequently issued a reward. Following this reward, the probability of choosing the same channel in a subsequent time slot is updated. The probabilities of channel selection are updated after every selection as follows:

$$p_c^i(t+1) = \begin{cases} p_c^i(t) + \beta \mathcal{R}(1 - p_c^i(t)), & \text{if} \quad \theta_{ic} = 1 \\ p_c^i(t) - \beta \mathcal{R} p_c^i(t), & \text{otherwise} \end{cases} \tag{12}$$

θ_{ic} maps between a channel selection and an operator. It is equal to one if network i chooses channel c, and zero otherwise. β is the learning rate, selected between 0 and 1, and $\mathcal{R}$ is the reward. The reward will quantify how good the channel choice was and is equal to the normalized throughput of the network on the utilized channel (T_c^i), divided by a theoretical upper bound for that same throughput T_{max}.

The learning rate β defines the speed with which the algorithm converges towards a channel decision for the different network operators, as well as the efficiency of this decision. A large value of β would lead to quicker decisions, but might not converge towards the Nash equilibrium leading to inefficient decisions. Ideally, we aim to select the largest value of β that would certainly converge towards an efficient Nash equilibrium. While we settle on a value of β through trials, we know it is efficient because our proposal does indeed converge to a Nash equilibrium.

In order to compute the reward, each network needs to be able to estimate the packet delivery ratio (success rate) at its gateways. We assume that each network knows its own number of devices, and their packet generation rates. After utilizing a channel for a certain amount of time, a network can generate an estimate of the success rate per spreading factor on the channel. For instance, if it has 1000 devices transmitting at the rate of one packet per hour, it expects to receive 1000 packets one hour after utilizing a given channel c. If it receives 600, then the success rate is 0.6. The larger the waiting period, the more accurate this estimation is. However, we show via simulations that any error in estimation does not prevent the algorithm from reaching a pure NE for the different operators. The pseudo-code for the learning approach can be seen in Algorithm 2.

Algorithm 2 Learning the NE

Requires: Set of states $\mathcal{S}$, actions $\mathcal{A}$, and rewards $\mathcal{R}$
Input: Learning rate $\beta \in$ [0,1]
Initialize: $p_c^i(1) \leftarrow \frac{1}{C}$, $\forall\ i \in \mathcal{N}$
Do
$t \leftarrow t+1$
    For i=1,…,$\mathcal{N}$
        Select a channel following $p_c^i(t)$
        Network i utilizes the selected channel enough
        time to estimate the success rate per SF
        Compute $\mathcal{R} = T_c^i / T_{max}$
        For $c \in \mathcal{C}$
            If θ_{ic}== 1
                $p_c^i(t+1) \leftarrow p_c^i(t) + \beta\mathcal{R}(1 - p_c^i(t))$
            Else
                $p_c^i(t+1) \leftarrow p_c^i(t) - \beta\mathcal{R}p_c^i(t)$
            End If
        End For
    End For
Until $\exists\ c \in \mathcal{C}$ such that $p_c^i(1) \approx 1$, $\forall\ i \in \mathcal{N}$

VI. Correlated Equilibrium

A correlated equilibrium, introduced by Aumann (1974) [19], is the game theory notion wherein each player of the game receives a private signal that does not affect the payoff of their decisions. The players of the game then choose their actions based on this signal. When the players have no motives to change their decisions, knowing that the other players would not, they arrive at a correlated equilibrium. From a practical point of view, the correlated equilibrium is the most relevant non-cooperative solution for a game [1]. By definition, every Nash equilibrium is a correlated equilibrium. What makes the latter so interesting is that they are much easier to obtain [20] and can sometimes result in better payoffs than their Nash counterparts.

In our work, we examine three different approaches, centralized and distributed, for finding the correlated equilibrium. The first, the social-welfare method, is a Pareto optimal linear programming method aimed at maximizing the objective function. The second method, which we refer to as the general CE, is an optimization problem aimed at finding the equilibrium without the social-welfare constraints. Note that both of these problems output globally optimal solutions. And thirdly, we propose a distributed learning approach to finding the CE, known as matching-regret, where players may depart from their current play with probabilities proportional to measures of regret for not having used other strategies in the past.

A. Social-Welfare Correlated Equilibrium

Our objective in this linear programming approach is to find an efficient correlated equilibrium which maximizes the normalized throughput of the players (operators). Let a_i and a_i' be two strategies of the set of pure strategies for player i, denoted A_i. Let A=(A_i, A_{-i}) be the global strategy profile. Finally, let $p(a)$ be the probability of choosing the profile a. $P = (p(a)_{a\in A}) \in \Delta A$, the set of the probability distributions. P is a form of recommendation for each player and is a CE if it holds that:

$$\sum_{a\in A|a_i\in a} p(a)U_i(a) \geq \sum_{a\in A|a_i\in a} p(a)U_i(a_i', a_{-i}) \qquad \forall i \in \mathcal{N},\ \ \forall a_i, a_i' \in A_i. \tag{13}$$

The intention behind this condition is that for every player i, the payoff for its strategy a_i in certain general profile of strategies $a \in A$, with a certain probability $p(a)$, remains higher than the payoff of any other individual action it might take. As such we can formulate the linear optimal problem for computing a social-welfare CE as:

$$\underset{p(a)}{\text{Maximize}} \ \sum_{a\in A} p(a) \sum_{i\in\mathcal{N}} U_i(a) \tag{14a}$$

Subject to

$$\sum_{a\in A|a_i\in a} p(a)U_i(a) \geq \sum_{a\in A|a_i\in a} p(a)U_i(a_i', a_{-i}), \qquad \forall i \in \mathcal{N},\ \ \forall a_i, a_i' \in A_i, \tag{14b}$$

$$p(a) \geq 0 \qquad \forall a \in A, \tag{14c}$$

$$\sum_{a\in A} p(a) \leq 1. \tag{14d}$$

Equation (14a) has the objective of the problem, that aims to increase the probabilities of the strategy profiles which maximize the social-welfare. Constraints (14c) and (14d) are in relation to the probability distribution, where the probabilities need to be positive and sum up to one. Finally, the social-welfare maximizing correlated equilibrium is Pareto optimal as illustrated in [21].

B. General Linear Problem

A more general approach to computing the correlated equilibrium drops the social-welfare constraints. This increases the size of the candidate set of correlated equilibria, but might at the same time lead to less efficient solutions. The linear optimal problem to compute the CEs can be reformulated as follows:

$$\underset{p(a)}{\text{Maximize}} \sum_{a \in A} p(a) \tag{15a}$$

Subject to

$$\sum_{a \in A | a_i \in a} p(a) U_i(a) \geq \sum_{a \in A | a_i \in a} p(a) U_i(a'_i, a_{-i}), \quad \forall i \in \mathcal{N}, \;\; \forall a_i, a'_i \in A_i, \tag{15b}$$

$$p(a) \geq 0 \qquad \forall a \in A, \tag{15c}$$

$$\sum_{a \in A} p(a) \leq 1. \tag{15d}$$

We note that linear programs are easy and straightforward to solve in practice. We again used CVX to solve this problem.

C. Matching Regret

In this section, we implement a distributed learning algorithm to find the correlated equilibrium, known as the matching regret [1]. In essence, each operator i at a certain instant t, makes a decision of choosing a given channel c according to a certain probability. This is dubbed its strategy. Afterwards, it computes its *regret i.e.,* the difference in payoff it would have received had it made a different decision. Following this regret, it updates its probability of making the same choice in the subsequent time slot.

Given the history of play, we suppose that each player (network) $i \in \mathcal{N}$ chooses a strategy $a_i^{t+1} \in A_i$ according to a probability distribution $p_a^i(t) \in \Delta A_i$. For every two strategies a and $a' \in A_i$ of player i, suppose that the latter was to replace the former every time it was played in the past. The payoff at time τ, $\tau < t$, would become:

$$W_i^\tau(a, a') = \begin{cases} U_i(a', a_{-i}^\tau), & \text{if} \quad a_i^\tau = a \\ U_i(a^\tau), & \text{otherwise} \end{cases} \tag{16}$$

where a_i^τ is the strategy of player i at time τ. The resulting difference in the average payoff up to time t can thereafter be expressed as:

$$\begin{aligned} D_i^t(a, a') &= \frac{1}{t} \sum_{\tau=1}^{t} W_i^\tau(a, a') - \frac{1}{t} \sum_{\tau=1}^{t} U_i(a^\tau) \\ &= \frac{1}{t} \sum_{\tau \leq t : a_i^t = a} [U_i(a', a_{-i}^\tau) - U_i(a^\tau)]. \end{aligned} \tag{17}$$

Finally, let

$$R_i^t(a, a') = [D_i^t(a, a')]^+ = max\{D_i^t(a, a'), 0\}, \tag{18}$$

be the regret player i experiences at time t for choosing strategy a instead of a'. The probabilities of choosing a given strategy are then updated as follows:

$$\begin{cases} p_{a'}^i(t+1) = \frac{1}{\mu} R_i^t(a, a'), & \forall a' \neq a \\ p_a^i(t+1) = 1 - \sum_{a' \in A_i : a' \neq a} p_{a'}^i(t+1), & \text{otherwise} \end{cases} \tag{19}$$

where μ is a number chosen large enough to ensure convergence. The probability of switching strategies is then proportional to the regret the player experiences when selecting them. After the algorithm converges, at time slot t the distribution of the N-tuples of strategies played up to time t can be expressed as:

$$z^t(a) = \frac{1}{t} |\{\tau \leq t : a^\tau = a\}|. \tag{20}$$

That is to the say the final probability of player i selecting a strategy a is the percentage of times it was played throughout the process. The pseudo-code for the process can be seen in Algorithm 3.

In the context of our matching regret proposal for finding the correlated equilibrium, it is important to highlight the following:

- The value of μ is chosen to be greater than $2M_i(m_i - 1)$ $\forall\ i \in \mathcal{N}$, where M_i is an upper bound for $|U_i(.)|$ and m_i is the number of strategies of player i (in our case the number of channels it can choose). Convergence of the algorithm is not guaranteed for smaller values of μ.
- If every player i plays according to the procedure illustrated in Algorithm 3, the latter is guaranteed to converge towards the set of CE as $t \to \infty$ [1].
- In regards to the complexity of the matching regret proposal, at each iteration, every player performs one table lookup to calculate its utility. They also perform two additions and two multiplications to update the regret value, and one random number draw, one multiplication and one comparison to calculate the next strategy.

Algorithm 3 Learning the CE

```
Requires: Set of states S, actions A, and rewards R
Input: μ > 0
Initialize: p_a^i(1), ∀ i ∈ N, ∀ a ∈ A_i
For t=1,...,T
t←t + 1
    For i=1,...,N
        Select an action (channel) following p_a^i(t)
        Compute the regret according to (18)
        Update the probability distributions such that
        For a,a' ∈ A_i
            If a' ≠ a
                p_{a'}^i(t + 1) = (1/μ) R_i^t(a, a')
            Else
                p_a^i(t + 1) = 1 − Σ_{a'∈A_i:a'≠a} p_{a'}^i(t + 1)
            End If
        End For
    End For
End For
```

D. Complexity With Respect to Nash Equilibrium

As we showed in section VI-A, computing the correlated equilibrium requires solving a problem with a linear objective and linear constraints. Intuitively speaking, the correlated equilibrium has only a single randomization over outcomes and its problem is as such solvable in polynomial time. However, the NE is constructed as a product of independent probabilities [22]. If we want to solve the same problem for the Nash equilibrium, the constraint in equation (13) becomes:

$$\sum_{a\in A} U_i(a) \prod_{j\in\mathcal{N}} p_j(a_j) \geq \sum_{a\in A} U_i(a_i', a_{-i}) \prod_{j\in\mathcal{N}\setminus\{i\}} p_j(a_j) \quad \forall i \in \mathcal{N},\ \ \forall a_i, a_i' \in A_i. \tag{21}$$

This constraint is non-linear, and as such the optimal problem for finding a Nash equilibrium becomes a mixed integer non-linear program, which could become extremely difficult to compute as the number of variables increases.

VII. Simulations and Results

Through simulations, we aim to test our algorithms and their effectiveness in different scenarios. Unless specified otherwise, the parameters are as shown in Table III. Using a Matlab-based environment, we simulate four co-located operators. The number of devices per operator is ascending and equal to 750, 1000, 1250, and 1500, respectively. The traffic per device is also different per network and increases from 1 packet per hour to 4, for operators 1 to 4, respectively. We simulate our channel selection algorithm, solved for different equilibria through multiple approaches, alongside the traditional adaptive data rate spreading factor selection approach. We compare it to the average result of a randomized channel selection process, plotted in the results alongside a 99.99 % confidence interval. In the simulation results, the best response approach to computing the NE is labeled "NE", and the general approach to computing the CE is labeled "CE". The learning approach to finding the NE is labeled "RL NE", while the Regret Matching and Social Welfare approaches to finding the CE are referenced by their names.

Table III: Simulation Parameters

Parameter	**Value**
Number of operators	4
Number of LoRa devices per operator	[750,1000,1250,1500] (4500 total)
Number of available channels	3
Number of gateways per operator	2
Network layout	Square with side S = 8 km
Path loss model	Okumura-Hata Model
Spreading factors SF	7-12
Tx power	14 dBm
Carrier frequency	868 MHz
Gateway height	30 m
Device antenna height	1.5 m
Packet generation rate per network λ^i	[1,2,3,4] packets/hour
Packet length	50 Bytes

A. Impact of Packet Size

In this section, we assess the impact ot the packet size on the performance of our proposals. As such, we simulate our algorithms for the packet lengths l of: 10, 20, 30, 40, and 50 bytes. Figure 4 has a plot illustrating the resulting total normalized network throughput for our different proposals as a function of the packet size.

Figure 4 (a) compares between random channel selection and the social welfare CE. The latter outperforms random selection regardless of the packet size with total normalized throughput values between 0.36 and 0.8, compared to average values between 0.34 and 0.71 for random selection. The gap between the upper and lower bounds of the confidence interval for the random selection process increases with the packet size, resulting in higher possibilities for inefficient choices. In Fig. 4 (b), we plot the results for the NE, attained via the best response algorithm, and the CE, attained via the linear optimization problem. The two approaches attain near identical results, with the CE ever so slightly outperforming its Nash counterpart.

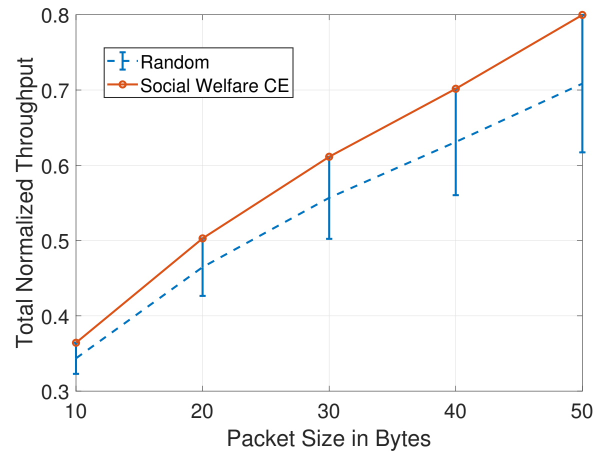
We additionally compare between the different approaches in terms of the total packet delivery (success) ratio. Figure 5 has the results. In Fig. 5 (a), we show the gains of the game theoretic approach, solved for the NE via repeated best response dynamics, with respect to random channel selection. For the former, the packet delivery ratio decreases as the pack size increases with values between 0.82 and about 0.75. Randomized channel selection always produces lower packet delivery ratios. In Fig. 5 (b), we notice that though small, the performance gap between the NE and the CE is more visible with the latter producing success rates higher by about 1% for l=10 bytes, and 0.5% at l=50 bytes.

B. Impact of the Area Size

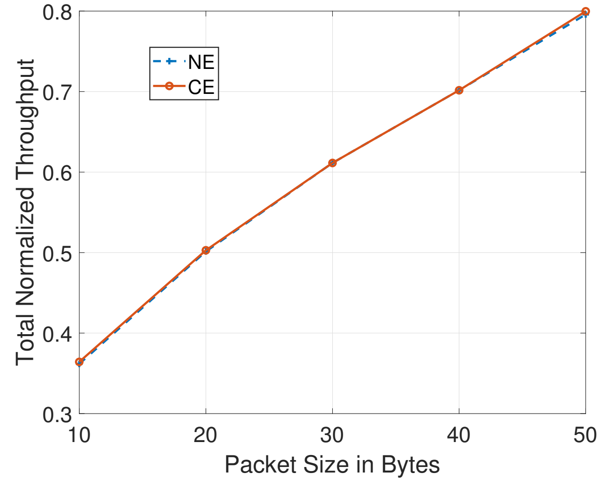
In this section, we study the impact of varying the simulation area size on the performance of our game proposal and its different equilibrium finding solutions. The area side length S is varied between 2, 4, 8, 12, and 16 km. Figure 6 has the results in terms of the total normalized throughput.

First, in Fig. 6 (a), we compare again between random channel selection and the regret matching approach to finding the CE. The latter always outperforms the former with ascending values ranging from about 0.25 for S=2 km to a maximum close to 1 at S=12 km. The performance of the game approach, solved for the CE, decreases afterwards. This increase and decrease in the total normalized throughput is in relation to the ADR spreading factor selection process. When the area size is small, the density of the devices on the lower SFs is increased and collisions thereafter increase, reducing thus the total normalized throughput. When the area size is very large, the radio conditions of the devices degrade reducing thus their throughput capabilities. Inversely to before, the size of the confidence interval decreases with the increase in the area size, indicating that it is harder to make inefficient decisions.

In addition, we aim to assess the cost in efficiency (for the total normalized throughput) for computing the NE and the CE via the proposed distributed learning algorithms. Figure 6 (b) has a box plot showing the difference in total normalized throughput between the game solved for the NE via repeated best response to the NE computed via the learning proposal. We notice that the difference is insignificant with maximum normalized throughput losses in the vicinity of 0.25 %. As for the CE, Fig. 6 (c) has a box plot showing the difference in the

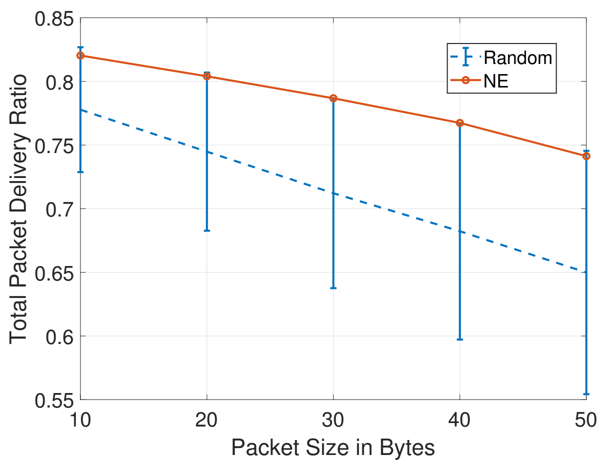


(a) Game approach gains

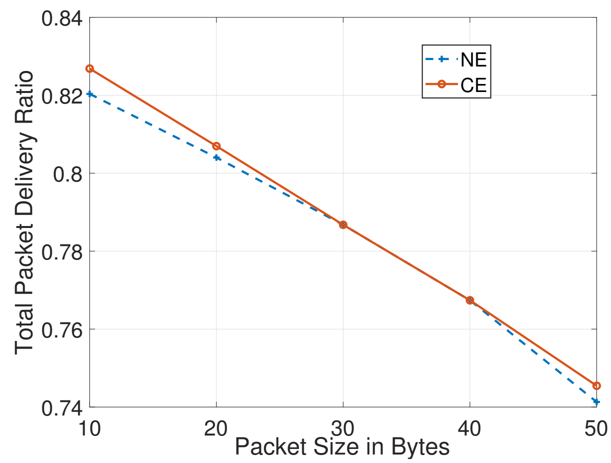


(b) NE vs. CE

Figure 4: Impact of packet size on the total normalized throughput

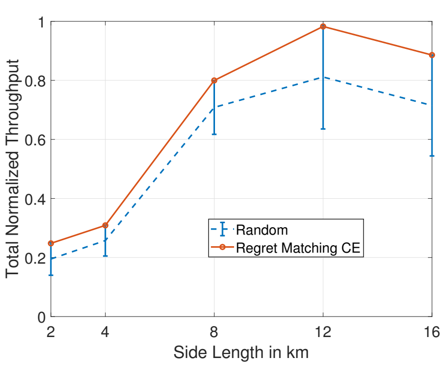


(a) Game approach gains

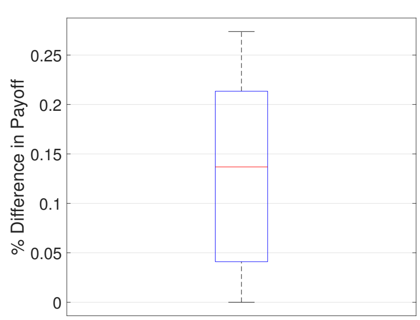


(b) NE vs. CE

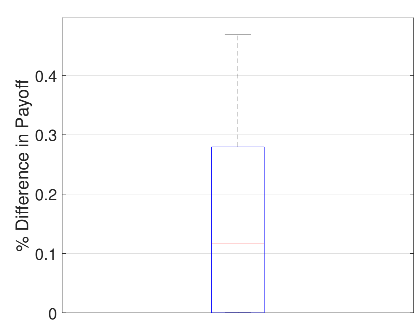
Figure 5: Impact of packet size on the total packet delivery ratio



(a) Game approach gains

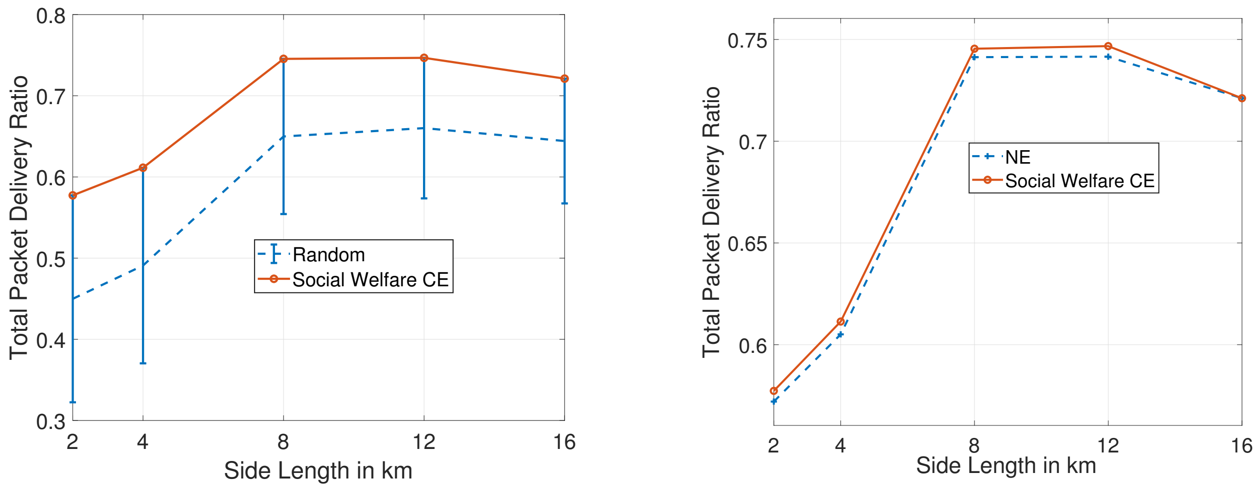


(b) Cost of learning the NE



(c) Cost of learning the CE

Figure 6: Impact of the area size on the total normalized throughput



(a) Game approach gains

(b) NE vs. the social welfare CE

Figure 7: Impact of the area size on the total packet delivery ratio

payoff for the CE solution solved using the linear problem to the case where it is computed via the regret matching learning algorithm. As with the NE, learning the CE does not incur any significant losses. The box plot shows a maximum difference close to 0.45 %. We are thus able to solve the game in a decentralized manner without a loss in efficiency.

We further study the impact of this variation in the area size on the total packet delivery ratio. The results can be seen in Fig. 7. In the plot of Fig. 7 (a), we once again show that our game proposal can significantly outperform randomized channel selection in terms of total packet delivery ratio. Similar to before, the delivery ratio increases from 0.45 at S=2 km to a maximum close to 0.75 at S=12 km. It starts to drop afterwards. In Fig. 7 (b), we compare between the NE solution and the corresponding social welfare CE. With little room for improvement considering the scenario, the optimality of the social welfare approach can still be noted. The latter always produces better packet delivery ratios regardless of the area size.

C. Impact of an Increase in the Number of Available Channels

We increase the number of available channels, from which the network operators can choose, from 3 up to 8, the maximum allowed under LoRa and ISM band specifications. In each simulation, each added channel has 10% more external traffic. This means that some channels will be more loaded than others. Figure 8 has the results in terms of total normalized throughput and the total packet delivery ratio.

An increase in the number of possible choices, especially in the case of crowded channels, will lead random channel selection to make more erroneous decisions, increasing thus the size of the confidence interval. Figure 8 (a) shows that the gains of our game proposal, solved for the social welfare CE, become more pertinent. The plot shows that the total normalized throughput produced by the corresponding NE is always above 0.81, whilst the average random approach payoff remains capped below 0.74.

In Fig. 8 (b), we plot the difference in the resulting normalized throughput between the NE, solved via repeated best response, and the social welfare CE. After the throughput improves when the number of channels is increased from three to four, the payoff for the social welfare CE remains constant no matter the number of channels available after that. This proves that the social welfare CE is indeed Pareto optimal. On the other hand, the throughput resulting from the NE fluctuates over a small margin depending on which NE the problem lands on. As in the previous simulations, the social welfare approach to computing the CE leads to slightly higher normalized throughput values than its NE counterpart.

Finally, in Fig. 8 (c) we look at the performance of our game proposal in terms of the total packet delivery ratio. The plot compares between the social welfare CE and randomized channel selection. The former fairly outperforms the latter with the total packet delivery ratio for the CE capped at 0.755.

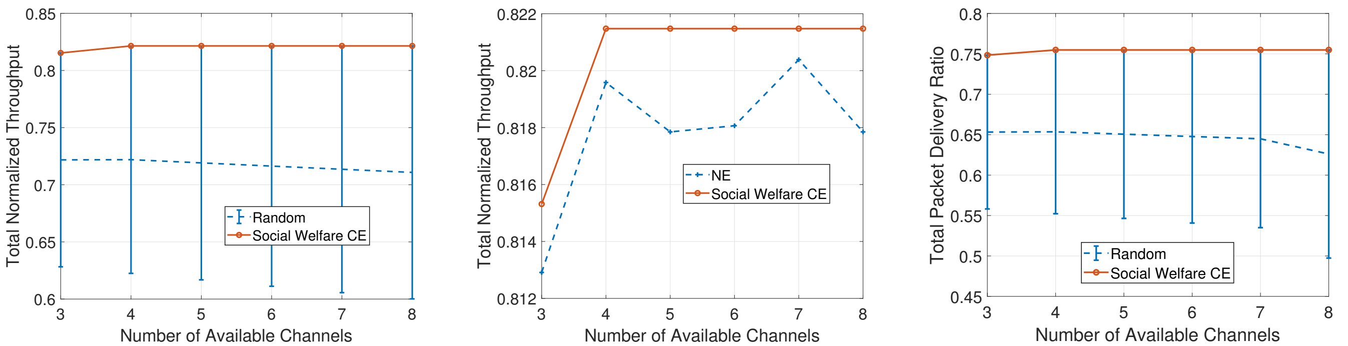
D. Energy Consumed per Delivery

We want to study the average energy expended per successful delivery in our considered scenario. For this simulation, we change side length S of the square area and compare between our reinforcement learning (RL) NE proposal and the randomized approach. Following [23], the energy cost of data delivery can be expressed as:

$$EC_{delivery} \approx \frac{a \cdot \exp(2G_{sc})}{l_{pay}}, \tag{22}$$

where a is the energy expended per packet transmission attempt, G_{sc} is the total normalized traffic load per SF, per channel, and l_{pay} is the payload size. The results are shown in Fig. 9.

The results show that our learning proposal always consumes less energy per delivered byte, in comparison to the



(a) Game approach gains in throughput (b) NE vs. Social Welfare CE (c) Game approach gains in delivery ratio

Figure 8: Impact of increasing the number of channels on performance

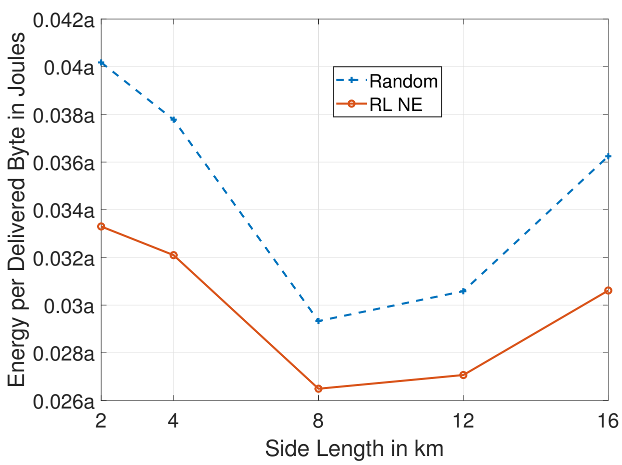


Figure 9: Energy per byte as a function of S

randomized approach. Unsurprisingly, the least energy expenditure for the learning algorithm came at S=8 km when it produced the highest packet delivery ratio.

E. Comments on the Results

In this paper, we proposed both centralized and distributed approaches to computing both the NE and the CE of our game theoretic proposal for channel selection in multi-operator LoRaWAN scenarios. In line with the simulation results, we highlight the following:

- Both the NE and the CE solutions for our games produce similar results in terms of the resulting normalized throughput.
- Both our NE and CE solutions provide better results with respect to random channel allocation, the currently utilized method for LoRaWAN, in terms of throughput, packet delivery ration, and energy efficiency.
- The Social Welfare CE produces, ever so slightly, the best results in terms of throughput and packet delivery ratio as well. Its main drawback is that it needs to be solved for in a centralized manner.
- Generally speaking, and as shown in the paper, the CE is easier to attain than the NE. The usual drawback is that the CE space is larger and less constrained. This could lead to less favorable solutions. However, as attested for by the results, this is not the case for our game.
- The choice of one method over the other thus comes down to preference and ease of implementation. If a centralized server is available, the Social Welfare CE would be preferred. In case of non-cooperative operators, a learning approach to finding the NE would be more suited.
- Finally, our comparison to a randomized approach shows the significance of our NE and CE proposals. They are optimal, or near-optimal, solutions that could be sometimes attained with concurrent approaches, but in most cases would not be.

VIII. Conclusion

In this paper, we proposed a game theory based algorithm for channel selection in a multi-operator LoRaWAN deployment. We solve for its Nash equilibrium via repeated best response dynamics, and through a distributed learning algorithm as well. In addition, we propose both a centralized optimal algorithm and a distributed regret-matching based learning approach for finding a Correlated equilibrium for the game. The latter is a more general and a mathematically easier method to obtain solution for the game. We compare and contrast between the two concepts and highlight, via simulations, the gains they achieve with respect to concurrent approaches to channel selection.

References

[1] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[2] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale lorawan networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.

[3] A. Tiurlikova, N. Stepanov, and K. Mikhaylov, "Method of assigning spreading factor to improve the scalability of the lorawan wide area network," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2018, pp. 1–4.

[4] D. Eridani, E. D. Widianto, R. D. O. Augustinus, and A. A. Faizal, “Monitoring system in lora network architecture using smart gateway in simple lora protocol,” in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 200–204.

[5] T. Elshabrawy and J. Robert, “The impact of ism interference on lora ber performance,” in *2018 IEEE Global Conference on Internet of Things (GCIoT)*, 2018, pp. 1–5.

[6] T. Yatagan and S. Oktug, “Smart spreading factor assignment for lorawans,” in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–7.

[7] A. Farhad, D. Kim, P. Sthapit, and J. Pyun, “Interference-aware spreading factor assignment scheme for the massive lorawan network,” in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, 2019, pp. 1–2.

[8] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, “Explora: Extending the performance of lora by suitable spreading factor allocations,” in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2017, pp. 1–8.

[9] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, “Improving reliability and scalability of lorawans through lightweight scheduling,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, 2018.

[10] Q. Zhou, J. Xing, L. Hou, R. Xu, and K. Zheng, “A novel rate and channel control scheme based on data extraction rate for lora networks,” in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6.

[11] J. Lee, W. Jeong, and B. Choi, “A scheduling algorithm for improving scalability of lorawan,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 1383–1388.

[12] S. Hasegawa, S. Kim, Y. Shoji, and M. Hasegawa, “Performance evaluation of machine learning based channel selection algorithm implemented on iot sensor devices in coexisting iot networks,” in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–5.

[13] O. Gusev, A. Turlikov, S. Kuzmichev, and N. Stepanov, “Data delivery efficient spreading factor allocation in dense lorawan deployments,” in *2019 XVI International Symposium ”Problems of Redundancy in Information and Control Systems” (REDUNDANCY)*, 2019, pp. 199–204.

[14] S. Lahoud, K. Khawam, S. Martin, G. Feng, Z. Liang, and J. Nasreddine, “Energy-efficient joint scheduling and power control in multi-cell wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3409–3426, Dec 2016.

[15] D. Monderer and L. S. Shapley, “Potential games,” *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[16] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.

[17] T. Börgers and R. Sarin, “Learning through reinforcement and replicator dynamics,” *Journal of economic theory*, vol. 77, no. 1, pp. 1–14, 1997.

[18] O. Bournez and J. Cohen, “Learning equilibria in games by stochastic distributed algorithms,” in *Computer and information sciences III*. Springer, 2013, pp. 31–38.

[19] R. J. Aumann, “Subjectivity and correlation in randomized strategies,” *Journal of mathematical Economics*, vol. 1, no. 1, pp. 67–96, 1974.

[20] C. H. Papadimitriou and T. Roughgarden, “Computing correlated equilibria in multi-player games,” *Journal of the ACM (JACM)*, vol. 55, no. 3, pp. 1–29, 2008.

[21] D. Wu, L. Zhou, Y. Cai, and J. Rodrigues, “Energy-efficient resource allocation for uplink ofdma systems using correlated equilibrium,” in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 4589–4593.

[22] C. Daskalakis and K. Leyton-Brown, “Equilibrium computation: Theory and practice.”

[23] L. Casals, B. Mir, R. Vidal, and C. Gomez, “Modeling the energy performance of lorawan,” *Sensors*, vol. 17, no. 10, p. 2364, 2017.