Cooperation for Spreading Factor Assignment in a Multi-Operator LoRaWAN Deployment

Hassan Fawaz[1], Kinda Khawam[1], Samer Lahoud[2], Steven Martin[3], and Melhem El Helou[2]

[1]Université Paris-Saclay, UVSQ, 78035, Versailles, France
[2]Université Saint-Joseph de Beyrouth, ESIB, CIMTI, Beirut, Lebanon
[3]Université Paris-Saclay, LRI, 91190, Gif-sur-Yvette, France

Abstract—Faced with the limitations of the Aloha random access scheme and spread spectrum techniques, LoRaWAN is yet to realize its potential as the flagship technology for large-scale Internet of Things applications. LoRaWAN allows for low power and long range communications. Nonetheless, concurrent transmissions on the same spreading factors, increased with the inevitable densification of device deployment, will lead to collisions and degradation in performance. The problem is further amplified due to the shortage in radio resources, with multiple operators utilizing the same unlicensed frequency bands. In this paper, we investigate different inter-operator cooperation schemes and devise multiple algorithms for spreading factor assignment in a multi-operator LoRaWAN deployment scenario. We start by proposing a proportional fair optimal formulation for the assignment with the objective of maximizing the logarithmic sum of the normalized throughput per spreading factor. Under the assumption of partial operator cooperation, we propose a gradient ascent based iterative algorithm for solving the spreading factor assignment problem, and a game theory based approach, wherein each network operator seeks to maximize its own normalized throughput. Finally, and with cooperation between different operators bound to be limited, we use recurrent neural networks to enable the prediction of the success rate per spreading factor. This prediction allows the different operators to assign spreading factors with minimum cooperation. We simulate our proposals and compare them to the legacy LoRaWAN approach as well as others in the state-of-the-art, highlighting the gains they produce in terms of total normalized throughput and packet delivery ratios.

Index Terms—LoRa, LoRaWAN, Recurrent Neural Networks, LSTM, Optimization

I. INTRODUCTION

With over 30 billion currently connected Internet of Things (IoT) devices, and over 75 billion ones expected in the market by 2025 [1], it is essential that IoT networks move past their legacy configurations in order to support this increasing demand within their limited bandwidth. Developed by Semtech, Long-Range (LoRa) is a physical layer technology that uses chirp spread spectrum (CSS) techniques to spread a narrow-band signal over a certain channel bandwidth. This increases the distance the signal could travel while reducing the power costs, enabling thus the connectivity of thousands of devices with battery lifespans in the years.

Among the solutions for the MAC layer is LoRaWAN [2], an upper layer protocol developed by the LoRa Alliance. LoRaWAN uses Aloha [3] for channel access. Devices transmit on a randomly chosen channel without any coordination. They use a spread-spectrum modulation technique derived from CSS technology. The spreading code applied to the original data signal is called the spreading factor (SF). LoRa supports a total of six spreading factors (SF7 to SF12). Only packets transmitted at the same time, on the same SF, and on the same channel, will collide. Despite their soaring popularity, LoRa and LoRaWAN still face multiple challenges pertaining to duty cycle limitations, inter-network interferences, and scalability issues.

First, LoRaWAN transmissions are restricted to a 1% duty cycle. This is to ensure fair access within the industrial, scientific, and medical (ISM) bands. Such a restriction limits the available radio resources, especially on the downlink, making traditional scheduling approaches, which require heavy signaling on the downlink, infeasible. As such, it was necessary to have SF assignment algorithms that require as little signaling as possible.

Second, the ISM band could be heavily utilized. Different operators and different technologies could easily be co-located with a certain LoRa network. This means that inter-network interferences are to be expected as LoRaWAN networks must be able to coexist. In current LoRaWAN deployments, the duty cycle limitation, intended to provide fair resource sharing between different networks in the ISM bands, is the only aspect connecting different co-located networks.

Thirdly, LoRaWAN suffers from a scalability problem. A LoRa network is supposed to be able to handle hundreds to thousands of devices within a large area. Nonetheless, as LoRaWAN uses random access techniques, such as Aloha, an increase in the number of devices would lead to an exponential increase in the number of collisions. Devices further away from the gateways would suffer the most, a significant inconvenience as LoRa is supposed to provide access at the longest ranges.

Legacy LoRaWAN supports an adaptive data rate (ADR) scheme where each node increases its SF in order to reach a gateway. Thus in its default state, each device would be transmitting using the SF which gives it the highest data rate while maintaining connectivity. We aim to go past the legacy approach as we propose multiple algorithms for SF assignment in multi-operator LoRaWAN deployments. While centrally allocating SFs on all devices is desirable and optimal, it would require all the different operators to relay the entirety

of their network information to a central entity *i.e.,* complete cooperation. This is not feasible or expected in practical implementations. In this paper, we examine three other levels of operator cooperation. The first assumes the complete lack of such cooperation, wherein each network uses legacy LoRaWAN approaches and ADR for SF selection. The second, partial cooperation, leverages secure aggregation to have operators safely share bulked updates. And the third, which we label as minimal cooperation, only requires established turns between the different operators seeking to optimally assign SFs to their devices. In our work, we propose SF assignment algorithms to tackle all the aforementioned challenges and others resulting from co-located network operators using the same ISM band.

We start by proposing an optimal formulation for SF assignment in LoRaWAN deployments with multiple network operators present. We propose a gradient ascent based iterative algorithm to solve this problem with partial cooperation among the different operators. Using the same level of cooperation, we put forward a game theory based approach, wherein every network seeks to maximize its own normalized throughput. Finally, under the pretense of minimal cooperation between operators, we propose a learning based approach capable of solving the SF assignment problem for all the co-located operators relying only on established turns. We simulate our proposals under different scenarios studying the impact of the number of devices, packet generation rate, deployment area size, and the number of gateways on the performance. We compare our algorithms to the legacy ADR LoRaWAN approach as well as to algorithms from state-of-the-art, highlighting thus the advantages of our proposals.

The rest of this paper is structured as follows: section II has the related works from the state-of-the-art. In section III, we introduce our system model. We discuss our optimal formulation for SF assignment in section IV and our iterative gradient ascent approach to solving it among different network operators in section V. We propose a game theoretic based algorithm for SF assignment, present the player utilities, and discuss attaining the Nash equilibrium in section VI. In section VII, we introduce recurrent neural networks and long short term memory networks and discuss how we can model the success rate as a time series to be predicted. We then introduce our learning-based proposal for SF assignment in LoRaWAN. Simulations and results are presented in section VIII. Finally, the paper is concluded with section IX.

II. RELATED WORKS AND CONTRIBUTIONS

We aim to highlight the different proposals in the state-of-the-art seeking to provide solutions for SF assignment in LoRaWAN.

A plethora of articles in the state-of-the-art discussed the potentials as well as challenges facing LoRa networks. In [4], the authors study the scalability of legacy LoRaWAN using an ns3 simulation environment. They state that downstream traffic has a negative impact on the packet delivery ratio and that gateway densification can improve performance. They conclude that duty cycle limitations will always limit the

scalability of LoRaWAN. In [5], the authors verify that while ADR SF selection reduces device energy consumption, it remains detrimental to the scalability of LoRaWAN. In [6], the authors propose a monitoring system for LoRa architecture by using a smart gateway. They then study the performance of LoRa under different scenarios. The authors in [7] study the impact of ISM band interferences on LoRa bit error rate performance. They show that there exists a signal-to-interference ratio threshold beyond which the impact of narrow-band interference becomes negligible.

More specifically in the context of our work, many in the state-of-the-art propose SF assignment algorithms which go beyond the legacy LoRaWAN approach. In [8], the authors propose what they described as a "*smart spreading factor assignment*" algorithm for LoRaWAN. They use support vector machines and decision tree classifier machine learning to optimize the SF assignment. They show via simulations that their proposal improves the packet delivery ratio for LoRaWAN.

The authors in [9] propose an interference-aware SF assignment algorithm. They take multiple factors into consideration including gateway sensitivity, interfering SFs, and the interfering energy in order to assign SFs in a manner that reduces the significance of the interference. They produce an assignment that decreases the time-on-air for every device. They show that their algorithm improves the packet delivery ratio with respect to legacy distance-based LoRaWAN SF selection.

With a traffic oriented approach in mind, the authors in [10] propose an SF assignment algorithm based on equalizing the traffic load on the SF channels. They also propose another algorithm which uses K-means to relieve critical regions which suffer from a significant number of collisions. The authors claim that their proposals, which have incremental complexity with respect to the state-of-the-art, open the door for enhancing the scalability of LoRaWAN in heterogeneous IoT scenarios.

With the aim of improving the probability of data delivery in LoRaWAN, the authors in [11] and [12] highlight the shortcomings of legacy SF assignment techniques and propose algorithms to address them. While the authors in the former claim that they can improve the scalability of LoRaWAN with a small increase in the end devices' power consumption, the authors in the latter illustrate that they can improve data delivery in dense LoRaWAN deployments.

In [13], the authors present an extension to the classic ADR strategy which they called ExpLoRa-SF. This approach seeks to equally distribute the SFs among the devices in the network subject to the radio conditions (and subsequently SNR) constraints experienced by the devices. They show that their proposal outperforms the legacy approaches to SF selection.

In [14], the authors propose a lightweight scheduling algorithm for SF selection and transmit power assignment. In their proposal, RS-LoRa, devices choose their own spreading factors based on probabilities assigned depending on the rate produced due to using each of the available SFs. 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 approach by about 20%.

Finally, recurrent neural networks [15], game theory for channel access [16], and the dynamics of Nash equilibriums [17], have been studied in the state-of-the-art before. Nonetheless, to the best of our knowledge, we are the first to use these approaches for spreading factor assignment in LoRaWAN.

In our work, we consider multiple co-located network operators unlike the vast majority of the state-of-the-art which works on single operators [8], [9], [11], [12], [14]. This allows us to study scenarios where the competing networks are not cooperative rendering many of the proposals in the state-of-the-art moot. We additionally enlist recurrent neural networks to help predict the success rate at the network gateways, enabling thus a distributed SF assignment process.

We summarize our contributions as follows:

(a) We propose an optimal problem for proportional fair SF assignment in a multi-operator scenario. Using this formulation, we propose multiple algorithms for SF assignment under different scenarios of inter-operator cooperation. We simulate our algorithms and compare them to the legacy LoRaWAN SF selection approach as well as to others in the state-of-the-art.

(b) Because of the existence of multiple operators, centrally solving such a problem is unfeasible. As such we propose a gradient ascent algorithm, wherein the different networks would solve the problem iteratively with partial inter-network cooperation maintained via secure aggregation.

(c) We propose a second algorithm for SF assignment based on game theory. The different operators would greedily seek to maximize their own normalized throughput. We show that the game converges to an efficient Nash equilibrium.

(d) We consider a realistic scenario wherein different operators are unlikely to cooperate. We enlist recurrent neural networks, specifically long short term memory networks, to help predict the success rate per spreading factor and enable the different networks to choose the device SFs while retroactively accounting for the different network SF selections.

III. SYSTEM MODEL AND SPECIFICATIONS

We consider multiple LoRaWAN deployments belonging to different operators. Each network $i \in \mathcal{N}$ has N^i nodes and r^i gateways. The gateways for the different operators are coincident *i.e.,* they are present at the same location sites. The devices are scattered across a square shaped area as shown in Fig. 1, with every color marking a device belonging to a different operator.

All nodes generate packets with a rate λ (considered the same for all devices presently). Transmit attempts are done according to a Poisson process with the packet size l being constant. Let T_s be the time needed to transmit a packet using spreading factor s. T_s is dependent on the packet size. LoRa

supports SFs ranging from 7 to 12. The latter represent a trade-off between coverage and data rate. For instance, SF7 produces the highest data rate, but covers the shortest distance. The SFs are orthogonal, this means that packets transmitted during the same time frame with different SFs will 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 the variation of data rate, sensitivity and SNR thresholds as a function of the SFs utilized in the 868 MHz band (C=1).
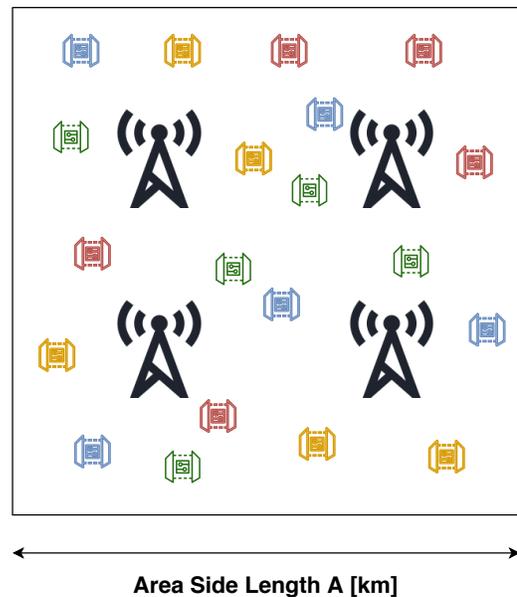

Area Side Length A [km]

Fig. 1. A 4 gateway multi-operator LoRaWAN deployment

TABLE I
LORA RATES, RECEIVER SENSITIVITY, AND SNR THRESHOLDS AS A
FUNCTION OF THE SFS

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

The upper layer protocol LoRaWAN uses a star topology. Each network connects to its own network server with one hop communications, where all the gateways forward the packets to the server. A typical architecture of such a network can be seen in Fig. 2.

As LoRaWAN uses pure Aloha as a channel access scheme, the duty cycle in the LoRa band is limited to d=1%. As such, the packet generation rate must verify $\lambda T_s {\leq} d$. LoRaWAN supports multiple frequency bands within the unlicensed bands of 433, 868, and 915 MHz. In Europe, the 868 MHz band is used with the 125, 250, and 500 kHz bandwidth channels.
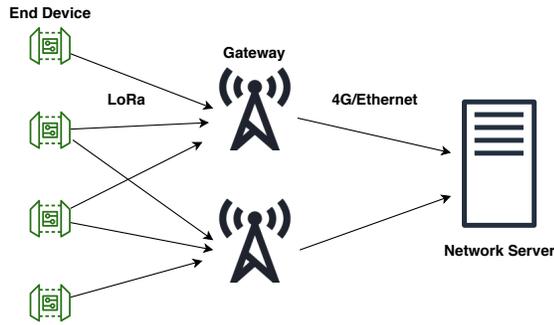
Fig. 2. LoRaWAN architecture

Fig. 3 shows a typical network stack of a LoRa node. Based on the application, LoRa devices can be divided into three classes:

- Class A: The device wakes up to send a certain packet. After sending the packet, the device listens for a short period of time. This class of LoRa devices consumes the least amount of energy.
- Class B: The device has regularly scheduled listening periods.
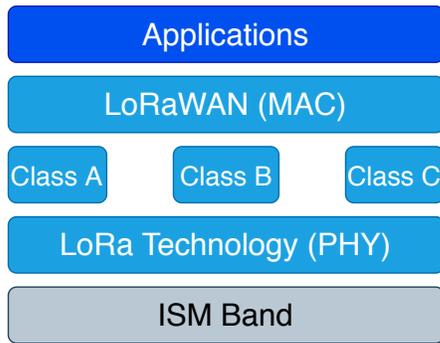- Class C: Under this configuration, a device is always listening. This leads to maximum energy consumption.

Fig. 3. LoRa node network stack

An additional advantage for our proposals in this paper is that, contrary to many in the state-of-the-art, they do not require additional synchronization with respect to current LoRaWAN specifications for class A. The network server can notify the nodes about the SF assignments within their receiving windows. Finally, Table II has a summary of the notations concerning our system model.

IV. OPTIMAL PROBLEM FOR SF SELECTION

The total normalized channel traffic load per spreading factor s can be written as:

$$G_s = \sum_{i=1}^{\mathcal{N}} (\lambda \cdot p_s^i \cdot N_c^i) T_s \qquad (1)$$

Where N_c^i is the number of covered nodes belonging to operator i. The normalized Aloha throughput on each SF s

TABLE II
NOTATION SUMMARY

Notation	Definition
λ	Packet generation rate
T_s	Time to transmit a packet on SF s
l	Packet length in bytes
d	Duty cycle
N_c	Total number of covered nodes
N_s	Total number of nodes that can use SF s and higher
N_c^i	Number of covered nodes for operator i
N_s^i	Number of nodes that can use SF s and higher for operator i
p_s	Ratio of devices using SF s
p_s^i	Ratio of devices using SF s and belonging to operator i
G_s	Total normalized traffic load on SF s
$\mathcal{N}$	Set of operators
$\mathcal{T}$	Total normalized throughput

can be expressed as $G_s \exp(-2G_s)$, and as such the total normalized throughput in the network becomes:

$$\mathcal{T} = \sum_{s=1}^{S} G_s \exp(-2G_s) \qquad (2)$$

The problem of finding the optimal SF assignment ratios p_s can be formulated as follows:

$$\underset{p_s}{\text{Maximize}} \sum_{s=1}^{S} \log(G_s \exp(-2G_s)) \qquad (3a)$$

Subject to

$$\sum_{s=1}^{S} p_s \leq 1, \qquad (3b)$$

$$\sum_{k=1}^{s} p_k \leq \sum_{k=1}^{s} \frac{N_k}{N_c}, \quad \forall s = 1, ..., S. \qquad (3c)$$

The logarithmic function in the objective in (3a) enforces proportional fairness in the throughput among the different spreading factors. The constraint in (3b) ensures that the sum of spreading factor percentages does not exceed one, while the constraints in (3c) indicate that the percentage of devices utilizing a certain SF s does not exceed the number of devices capable of using s and higher. Note that $s = 1$ represent SF7 and so on.

This problem in its current state can only be solved at a centralized server which has all the information on all the devices. This would be possible in the presence of one operator. However, with devices and gateways belonging to different operators, it is unrealistic to expect operators to share their data in this manner.

V. SF SELECTION WITH AN ITERATIVE GRADIENT ASCENT ALGORITHM

In our first proposal, we use a gradient ascent [18] based approach to solve our SF assignment problem iteratively for the different networks. We propose the usage of the federated technology of secure aggregation [19] to have the networks transmit to each other what we label as λ_s^e, the traffic generated by other co-located operators. The different networks agree

on masks, that sum to zero, which they add to their updates. The individual updates appear to be randomized but their sum results in meaningful data. This ensues that even the aggregator does not have access to individual updates. In succession, each network i will seek to maximize its own utility until convergence. This process is illustrated in Algorithm 1.

Algorithm 1 Iterative SF Selection Algorithm

1: **Requires:** Set of networks $\mathcal{N}$ and device arrival rate λ
2: **Input:** Maximum tolerance $\epsilon \leftarrow 10^{-5}$
3: **Initialize:** $t = 0$
4: **Do**
5:     $t \leftarrow t+1$
6:     **For** $i=1,\dots,\mathcal{N}$
7:         Calculate $p_s^i(t)$ by solving the problem in (3a) as
8:         shown in Algorithm 2
9:         Update $\lambda_s^i = \lambda \cdot p_s^i \cdot N_c^i$ and send it for aggregation
10:        Compute $\delta^i = ||p_s^i(t) - p_s^i(t-1)||$
11:    **End For**
12: **While** $\exists\, i \in \mathcal{N}$ such that $\delta^i \geq \epsilon$

In order to solve the problem in (3a), we use gradient ascent. We calculate the gradient for (3a) and network i as:

$$\nabla f(p_s^i) = \frac{T_s \cdot \lambda \cdot N_c^i}{T_s \cdot \lambda \cdot N_c^i \cdot p_s^i + \lambda_s^e \cdot T_s} - 2 \cdot \lambda \cdot N_c^i, \quad (4)$$

where the external traffic per spreading factor λ_s^e represents the traffic from the other co-located networks and can be expressed as:

$$\lambda_s^e = \sum_{j \neq i} \lambda \cdot p_s^j \cdot N_c^j. \quad (5)$$

Successively, and for each iteration t_g, the value of the gradient is multiplied by a learning rate β and added to the previous value of p_s^i. The result is projected onto a feasible solution.

$$p_s^i(t_g+1) = p_s^i(t_g) + \beta\left(\frac{T_s \cdot \lambda \cdot N_c^i}{T_s \cdot \lambda \cdot N_c^i \cdot p_s^i + \lambda_s^e \cdot T_s} - 2 \cdot \lambda \cdot N_c^i\right) \quad (6)$$

This is done until the value of p_s^i no longer changes within a certain margin of error. This process is illustrated in Algorithm 2.

Algorithm 2 Optimization Using Gradient Ascent

1: **Requires:** Learning rate β small enough
2: **Input:** Maximum tolerance $\epsilon \leftarrow 10^{-3}$
3: **Initialize:** $t_g = 0$
4: **Do**
5:     $t_g \leftarrow t_g+1$
6:     $p_s^i(t_g+1) \leftarrow p_s^i(t_g) + \beta(\frac{T_s \cdot \lambda \cdot N_c^i}{T_s \cdot \lambda \cdot N_c^i \cdot p_s^i + \lambda_s^e \cdot T_s} - 2 \cdot \lambda \cdot N_c^i)$
7:     Project p_s^i onto the problem constraints as shown in (7)
8:     Compute $\delta = ||p_s^i(t_g) - p_s^i(t_g-1)||$
9: **While** $\delta \geq \epsilon$

As the gradient ascent is usually used for unconstrained problems, it is important to project the result of each iteration onto the feasible subset. We do so following [20] where the projection of an n-dimensional point p_s^{i0} onto a simplex can be described as the solution of the following problem:

$$\underset{p_s^i}{\text{Minimize}} \quad ||p_s^i - p_s^{i0}||^2 \quad (7a)$$

Subject to

$$p_s^i \geq 0, \forall s \in S \quad (7b)$$

$$\sum_{s=1}^{S} p_s^i \leq 1, \quad (7c)$$

$$\sum_{k=1}^{s} p_k^i \leq \sum_{k=1}^{s} \frac{N_k^i}{N_c^i}, \quad \forall s = 1, \dots, S. \quad (7d)$$

Finally, the iterative SF algorithm is guaranteed to converge with an average of 3 to 4 iterations.

VI. MULTI-OPERATOR GAME FOR SF ASSIGNMENT

Under the same level of operator cooperation, we aim to use non-cooperative game theory to propose an algorithm for SF assignment. Each network seeks to select the set of spreading factors which maximizes its own normalized throughput. Nonetheless, if all networks do so greedily and independently, they would entirely default to the lowest SF selections the radio conditions allow, increasing thus the risk of collisions. The different operators, *i.e.*, the players of our game proposal, are competing for contradicting objectives. This makes game theory well adapted to SF assignment in multi-operator LoRaWAN deployments.

A. Game Formulation

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

- A finite set of players $i \in \mathcal{N}$, the set of operators.
- The action of a given player is the percentage of devices allocated any spreading factor s, the strategy chosen by an operator i is then $\boldsymbol{p}^i = (p_1^i, \dots, p_6^i)$, the percentage of its devices on each spreading factor, where for example, p_1^i is the percentage of devices using SF7.
- For each player i, the space of pure strategies is S_i given by what follows: $S_i =$

$$\{\boldsymbol{p}^i \in [0,1]^S|\ h_i(\boldsymbol{p}^i) \geq 0 \text{ and } g_s^i(\boldsymbol{p}^i) \geq 0, \forall s = 1, \dots, S\} \quad (8)$$

where

$$h_i(\boldsymbol{p}^i) = 1 - \sum_{s=1}^{S} p_s^i \quad (9)$$

$$g_s^i(\boldsymbol{p}^i) = \sum_{k=1}^{s} \frac{N_k^i}{N_c^i} - \sum_{k=1}^{s} p_k^i \quad (10)$$

$S = S_1 \times \dots \times S_{|\mathcal{N}|}$ is the set of all strategies.
- A set of utility functions $(U_{i \in \mathcal{N}})$ that quantify players' profit for a given strategy profile.

B. Player Utilities

Each player, *i.e.*, network operator, will seek to greedily maximize the logarithmic sum of its own normalized throughput. The utility per network operator can be expressed as:

$$U_i = \sum_{s=1}^{S} \log(G_s^i \exp(-2G_s)), \quad (11)$$

where $\exp(-2G_s)$ represents the success ratio per spreading factor in the entire system (and coincides with the success ratio of any network i), and G_s^i represents the traffic load of network i.

$$G_s^i = (\lambda \cdot p_s^i \cdot N_c^i) T_s. \quad (12)$$

As such, each player will seek to maximize its own delivered traffic, regardless of the others.

C. Existence of Nash Equilibrium

In game theory, a rational solution is one where all competing players adhere to a Nash equilibrium (NE) [21]. An NE is a profile of strategies in which no player will take advantage of the others by deviating its strategy unilaterally. In our game $\mathcal{G}$, for every network i, the strategy space S_i is a compact and convex set as the functions $h_i(\boldsymbol{p}^i)$ and $g_s^i(\boldsymbol{p}^i), \forall s = 1, ..., S$ are concave in $\boldsymbol{p}^i$ (linear in $\boldsymbol{p}^i$). Furthermore, U_i is concave and continuously differentiable in $\boldsymbol{p}^i$ and continuous in $\boldsymbol{p}^{-i}$ (the strategy of all other players). Hence, pure NE exist as proven by Rosen [22]. Given these strategy sets, $p^* \in S$ is a pure strategy Nash equilibrium if and only if $\forall i \in \mathcal{N}$, $\boldsymbol{p}^{i*}$ is an optimal solution of the following problem:

$$\underset{p_s^i}{\text{Maximize}} \ U_i(\boldsymbol{p}^i, \boldsymbol{p}^{-i}) \quad (13a)$$

Subject to

$$h_i(\boldsymbol{p}^i) \geq 0 \quad (13b)$$

$$g_s^i(\boldsymbol{p}^i) \geq 0, \quad \forall s = 1, ..., S. \quad (13c)$$

The optimum $\boldsymbol{p}^{i*}$ must satisfy the Karush-Kuhn-Tucker (KKT) conditions. The latter are sufficient because we are in the presence of convex optimization. There exists a unique Lagrange multiplier α such that:

$$p_s^{i*} = \frac{1}{\alpha + 2\lambda N_c^i T_s}, \quad (14)$$

$$\alpha \cdot h_i(\boldsymbol{p}^{i*}) = 0. \quad (15)$$

This produces two cases:

- if $\alpha = 0$, given that $\sum_{s=1}^{S} p_s^{i*} \leq 1$, then $p_s^{i*} = \frac{1}{2\lambda N_c^i T_s}$.
- if $\alpha \neq 0$, and given that $\sum_{s=1}^{S} p_s^{i*} \leq 1$, then $\sum_{s=1}^{S} \frac{1}{\alpha + 2\lambda N_c^i T_s} = 1$.

D. Uniqueness of Nash Equilibrium

The uniqueness of the Nash Equilibrium is obtained if the utility functions $U_i, \forall i \in \mathcal{N}$ are diagonally strictly concave (DSC). We denote by $\nabla u(p) = [\nabla_i U_i(p)] \in \mathbb{R}^S$ where $\nabla_i U_i(p) = [\frac{\partial U_i}{\partial p_s^i}, s = 1, ..., S]$ and by $JU(p)$ the Jacobian

of $\nabla u(p)$. Hence, the $(i,j)^{th}$ element of $JU(p)$ is given by what follows:

$$ju(i,j) = \nabla_j \nabla_i U_i(p) = [\frac{\partial^2 U_i(p)}{\partial p_s^i \partial p_s^j}, s = 1, ..., S] = 0 \quad (16a)$$

$$ju(i,i) = [-(\frac{1}{p_s^i})^2, s = 1, ..., S] \quad (16b)$$

A sufficient condition for the DSC property (theorem 6 in [22]) is that the matrix $(JU(p) + JU^T(p))$ is negative definite $\forall p \in S$. The latter is verified as the matrix $(JU(p) + JU^T(p))$ is diagonal with elements strictly negative owing to (16).

E. Reaching Nash Equilibrium

The primary challenge in game theory is to propose algorithms capable of reaching an NE. The simplest of these algorithms are the repeated best response dynamics. Following these dynamics, each player selects the best, and locally optimal, response to other players' strategies, until the algorithm converges. To benefit from such a property, we show that our game $\mathcal{G}$ is an exact potential game with the potential function $V(p)$ given by:

$$V(p) = \sum_{i=1}^{\mathcal{N}} \sum_{s=1}^{S} \log(G_s^i) - 2\sum_{s=1}^{S} G_s \quad (17)$$

According to [23], the game $\mathcal{G}$ is a continuous exact potential game as we have what follows:

$$\frac{\partial U_i}{\partial p_s^i} = \frac{\partial V(p)}{\partial p_s^i}, \ \forall i \in \mathcal{N}, \forall s \in \mathcal{S},$$

Exact potential games have a distinct computational advantage in that reaching the pure NE is done by implementing greedy best response dynamics, obviating the need for computational fixed point theory. Therefore, the computation of an equilibrium is reduced to solving the optimization problem in (13) at each iteration t_b of the Best Response dynamics, where each network i seeks, in turn, to find its optimal SF distribution as a response to $\boldsymbol{p}^{-i}(t_b - 1)$.

The best response algorithm we implemented is sketched in Algorithm 3. In order for the best response algorithm to progress, each operator needs to assess the impact of all the other co-located networks. It does not however need access to individual network information. As such, we proposed using secure aggregation, wherein all the networks would send the value of $\lambda_s^i = \lambda \cdot p_s^i \cdot N_c^i$ for aggregation after locally solving the SF assignment problem. This allows for a non-invasive level of cooperation between the different operators.

VII. LEARNING BASED ALGORITHM FOR SF ASSIGNMENT

In our two previous approaches, we considered that some level of cooperation, guaranteed via secure aggregation, remains between the different network operators. In the following learning-based approach, we assume that no such cooperation is necessary. In order to optimize the spreading factor assignment for LoRaWAN, each network operator must then be able to estimate the total packet delivery ratio (success rate). If a network operator has the success rate per spreading

Algorithm 3 Game Theory Based Algorithm for SF Selection

1: **Requires:** Maximum tolerance $\epsilon \leftarrow 10^{-5}$
2: **Input:** Initial SF assignment limits
3: **Initialize:** $t_b = 0$
4: **Do**
5:     $t_b \leftarrow t_b + 1$
6:     **For** $i=1,\ldots,\mathcal{N}$
7:         Solve the problem in (13) for i
8:         Update p_s^i
9:         Update $\lambda_s^i = \lambda \cdot p_s^i \cdot N_c^i$ and send it for aggregation
10:        Compute $\delta^i = ||p_s^i(t_b) - p_s^i(t_b - 1)||$
11:     **End For**
12: **While** $\exists\, i$ such that $\delta^i \geq \epsilon$

Fig. 4. An LSTM unit

factor $(\exp(-2G_s))$, and given that it knows its own traffic, it can calculate the impact of the other networks λ_s^e. We use recurrent neural networks (RNN), specifically long short term memory (LSTM) networks to predict the success rate per SF in the network. In our work, we treat the latter as a time series to be predicted.

A. Recurrent neural networks for time series prediction

Recurrent Neural Networks have been adopted in a wide range of machine learning tasks, particularly, when they require handling time-related data in either input or output, such as image captioning, music generation, speech recognition, handwriting recognition, machine translation and times-series prediction. RNNs are suitable for capturing the relationship between sequential data points owing to their recurrent structures. In detail, recurrent hidden states depend on both the current input and the network states at the previous time steps, instead of only the current input as in a conventional feed-forward neural network.

Nonetheless, RNNs suffer from many problems which prohibit them, in their traditional form, from effectively dealing with time series, among which are the vanishing and exploding gradient issues. As such, alternative units have been proposed such as LSTMs. LSTMs [24] introduce multiple gates to deal with memory problems such as the vanishing gradient. An LSTM unit is illustrated in Fig. 4.

Let c_{t-1} be the previous cell state and c_t the new cell state. Denote by f_t the output of the forget state and i_t the output of the input gate. $\hat{c}_t$ is called the candidate and h_t is the hidden state. Finally, b_f, b_c, b_i, and b_o are the bias parameters associated with aforementioned outputs.
The unit works as follows:

1) First, the previous hidden state h_{t-1} and the current input x_t get concatenated. This is called the combine.
2) This combine is fed into the forget layer which removes non-relevant data. The output after the first Sigmoid function is expressed as:

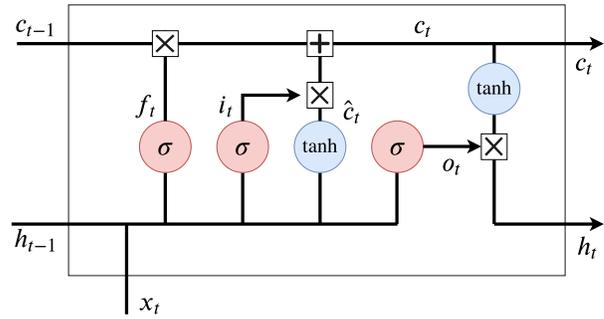$$f_t = \sigma\left(x_t W_{xf} + h_{t-1} W_{hf} + b_f\right). \quad (18)$$

3) Using the combine, a candidate layer is created. The candidate $\hat{c}_t$ holds values to possibly add to the cell state.

$$\hat{c}_t = \tanh\left(x_t W_{xc} + h_{t-1} W_{hc} + b_c\right). \quad (19)$$

4) The combine is additionally fed into the input layer which decides what data from the candidate should be added to the new cell state. The result after the second Sigmoid function i_t is expressed as:

$$i_t = \sigma\left(x_t W_{xi} + h_{t-1} W_{hi} + b_i\right). \quad (20)$$

5) After computing the forget layer, candidate layer, and the input layer, the cell state c_t is calculated as:

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t. \quad (21)$$

6) Finally, the output o_t is computed:

$$o_t = \sigma\left(x_t W_{xo} + h_{t-1} W_{hi} + b_o\right). \quad (22)$$

7) The multiplication of the output and the new cell state yields the new hidden state h_t:

$$h_t = o_t * \tanh\left(c_t\right). \quad (23)$$

B. Predicting the Success Rate

We aim to predict the success rate for packet arrivals *i.e.*, the percentage of packets which were correctly received at the gateways. Using packet frame numbering we assume that the network can record the success ratio for a certain number of time frames. We train an LSTM network to predict the success rate for successive time frames. The neural network parameters are shown in Table III.

TABLE III
LSTM PARAMETERS FOR SUCCESS RATE PREDICTION

Parameter	Value
Number of hidden units	140
Percentage of training data	90%
Gradient threshold	1
Maximum number of training epochs	1000
Initial learning rate	0.015
Learning rate drop period (in epochs)	160
Learning rate drop factor	0.25

We track the learning progress for the LSTM network and

plot the loss and the root mean squared error (RMSE) as a function of the number of epochs. The results can be seen in Fig. 6. We further use the trained model to predict the next 500 time frames and measure the difference between the simulated values and the predicted ones. The results can be seen in the box plot of Fig. 5. The median error in prediction sits at about 1.5% with the error being less than 3% more than 75% of the time. Such small errors might not always have an impact on the performance of the algorithm, with respect to the gradient ascent proposal, but resulting discrepancies can be seen in the difference in the total normalized throughput in the simulations.
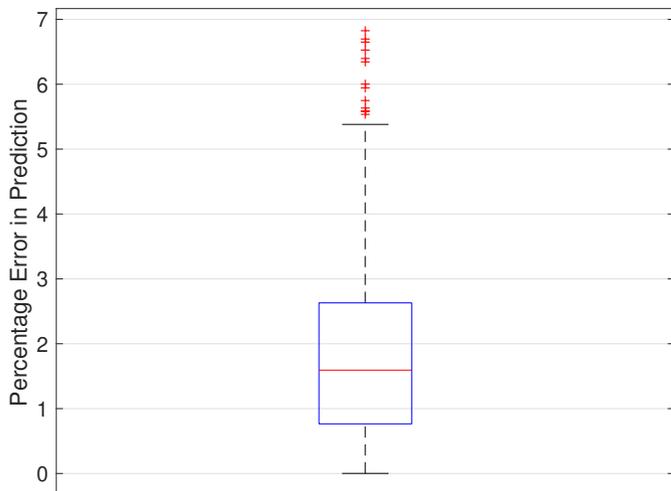
Fig. 5. Error in success rate prediction

C. Algorithm for SF Assignment

Using LSTMs we propose an algorithm to assign the SFs in the different networks following the optimal problem in (3). This keeps cooperation between the different network operators to a minimum. All what needs to exist is established turns between the different network operators. While this is not necessary for the algorithm to converge, it does insure that a few number of iterations (≤ 5) would be sufficient for convergence.

VIII. SIMULATIONS AND RESULTS

We seek via simulations to compare our proposals and test their effectiveness in different scenarios. Unless changed by the simulated scenario, the simulation parameters are as stated in Table IV.

A. Impact of Packet Size

We first aim to study the impact ot the LoRa packet size on the performance of our algorithms. To this end, we simulate our three proposals alongside the legacy LoRaWAN ADR SF assignment algorithm for the packet lengths l of: 10, 20, 30, 40, and 50 bytes. Figure 7 has a plot showing the resulting total normalized network throughput as a function of the latter.

For all the different packet lengths, our algorithms vastly

Algorithm 4 Learning Based Algorithm for SF Selection

1: **Requires:** Maximum tolerance $\epsilon \leftarrow 10^{-3}$
2: **Input:** Initial SF assignment limits
3: **Initialize:** $t_l = 0$
4: **Do**
5:     **For** i=1,...,$\mathcal{N}$
6:         Solve the problem in (3) for i
7:         Estimate own success rate
8:         Send values to the LSTM for training
9:     **End For**
10: **While** LSTM did not converge
11: **Do**
12:     $t_l \leftarrow t_l + 1$
13:     **For** i=1,...,$\mathcal{N}$
14:         Predict the success rate using the LSTM
15:         Calculate λ_s^e for network i
16:         Solve the problem in (3) for i
17:         Compute $\delta^i = ||p_s^i(t_l) - p_s^i(t_l - 1)||$
18:     **End For**
19: **While** $\exists\ i$ such that $\delta^i \geq \epsilon$

TABLE IV
SIMULATION PARAMETERS

Parameter	Value
Number of different network operators	4
Number of devices	750 per network (3000 total)
Number of gateways per network	4
Deployment layout	Square with side S = 8 km
Path loss model	Okumura-Hata Model
Spreading factors	7-12
Tx power	14 dBm
Carrier frequency	868 MHz
Gateway/Device height	30/1.5 m
Packet generation rate λ	5 packets/hour
Packet length	50 Bytes

outperform the legacy LoRaWAN SF selection. For a packet length of 10 bytes, our proposals achieve a total normalized throughput of about 0.65, compared to 0.2 for the legacy LoRaWAN approach. This gap increases as the packet length increases. At l=50 bytes, the gradient ascent algorithm produces a total normalized throughput close to 0.95, compared to 0.33 for the legacy algorithm. Finally, our learning proposal achieves near identical results to the gradient ascent algorithm, and they both slightly outperform the game theoretic approach for larger packet lengths. We further compare between the different algorithms in terms of the total packet delivery (success) ratio. Figure 8 has a plot with the results.

Similar to before, our algorithms outperform the legacy LoRaWAN approach, with the exception of the game theory proposal in the case of small packet length. As the packet length increases, the total packet delivery ratio degrades for all algorithms as the performance gap between our proposals and legacy LoRaWAN approach increases. At l=50 bytes, the packet delivery ratio for the gradient ascent algorithm is about 0.61 compared to 0.44 for the legacy LoRaWAN algorithm.
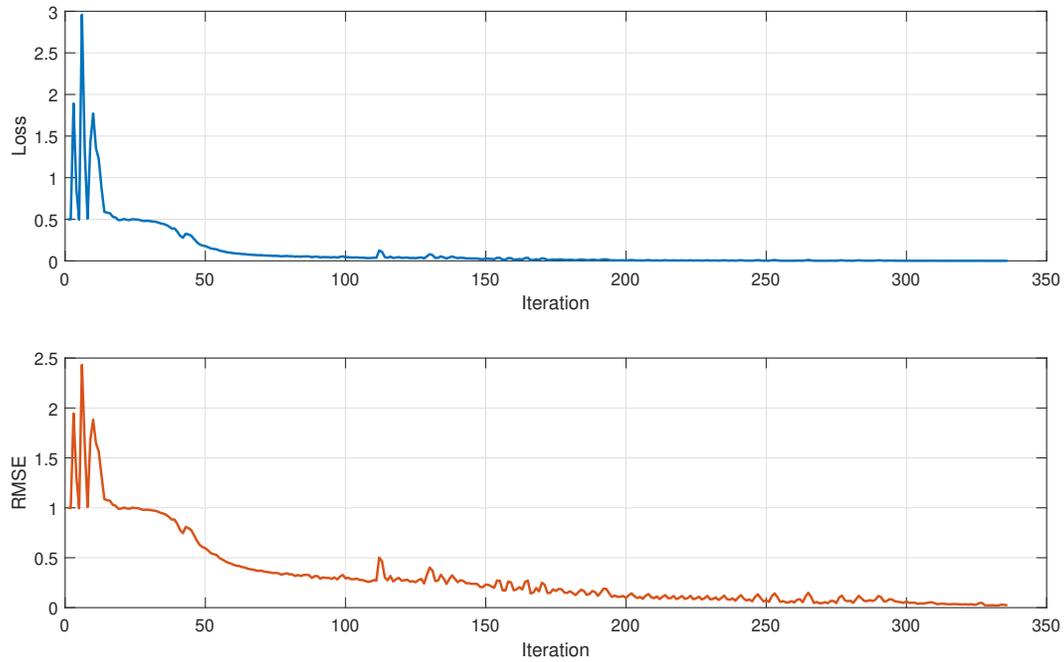
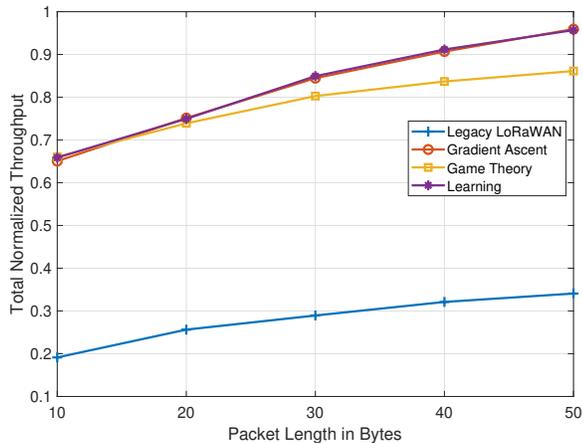Fig. 6. Training progress for success rate prediction

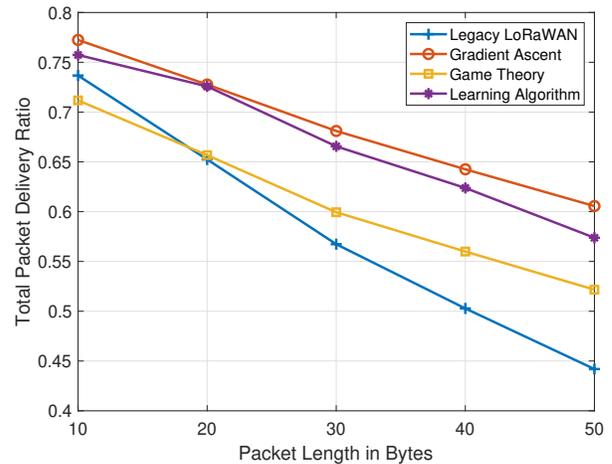Fig. 7. Impact of packet length on the total normalized throughput

Fig. 8. Impact of packet length on the total packet delivery ratio

Finally, the errors resulting from the learning process are more visible here, with the gap between the learning algorithm and the gradient ascent algorithm widening.

B. Individual Network Normalized Throughput

As noted in the previous section, the gradient ascent algorithm outperforms the game theoretic approach in terms of the total normalized throughput $(\sum_S G_s \exp(-2G_s))$. In this section we look at the results in terms of the individual network normalized throughput values $(\sum_S G_s^i \exp(-2G_s))$. Figure 9 has a bar graph with the results.

While the logarithmic function in the objective of the gradient ascent algorithm forces fairness among the different SFs, this does not mean the normalized throughput per network would be the same. While it does come at the cost of the total normalized throughput, the greediness of each network operator in the game theoretic approach ensures that they all end up with the same normalized throughput values when the algorithm converges. Figure 9 shows that for the game theory based algorithm, each of the networks achieves a normalized throughput value close to 0.22. This is much more lucrative
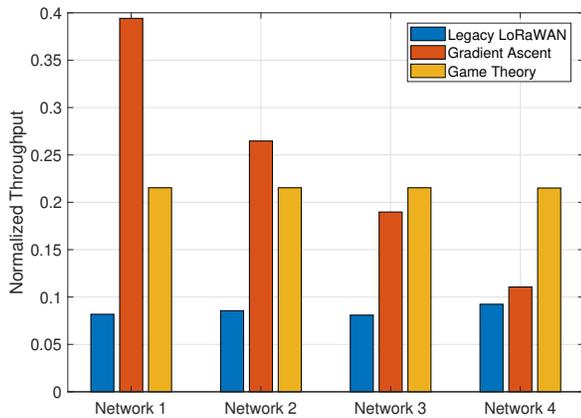
Fig. 9. Performance in terms of individual network normalized throughput

than what the legacy algorithm provides and much more balanced than what the gradient ascent algorithm produces.

C. Impact of Variation in the Packet Generation Rate

In this section, we study the impact of varying the LoRa devices' packet generation rate on the performance of our algorithms. To this end, we vary the generation rate between 1 and 5 packets per hour and plot the effect this has on the total normalized throughput in Fig. 10
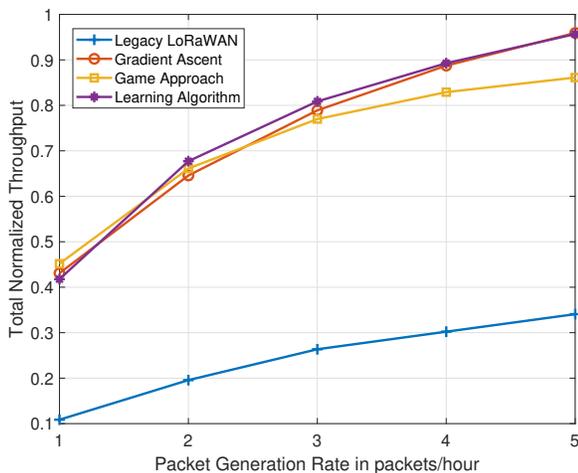

Fig. 10. Impact of packet generation rate on total normalized throughput

Similar to before all our proposals outperform the legacy approach. The latter produces a normalized throughput of about 0.12 at a packet generation rate of 1 packet/hour, which increases to 0.34 at a packet generation rate of 5 packets/hour. In comparison, our gradient ascent and learning algorithms produce normalized throughput values between 0.42 and 0.96 for the same generation rates. Again, the game theoretic approach lags our other proposals when it comes to a congested network. This is the cost of added fairness.

Inversely, the total packet delivery ratio degrades as the packet generation rate increases. This is because the possibility of collisions increases as the devices transmit more. The results are plotted in Fig. 11.
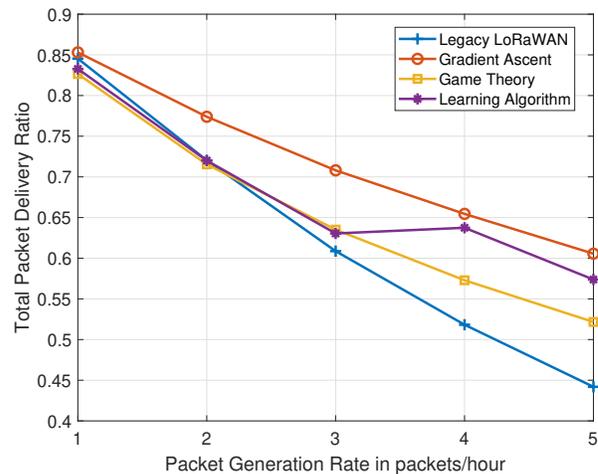

Fig. 11. Impact of packet generation rate on total packet delivery ratio

We notice that for a small packet generation rate, the legacy LoRaWAN algorithm performs well in terms of packet delivery ratio. This significantly changes as the generation rate increases. For $\lambda = 5$ packets/hour, the gradient ascent algorithm has a delivery ratio of about 0.6, compared to 0.53 for the game theoretic approach and just 0.44 for the legacy algorithm. Furthermore, for lower packet generation rates, the error in prediction for the learning algorithm is more visible on its performance with the gap between it and the gradient ascent algorithm growing.

D. Impact of the Number of Devices

We aim to assess the performance of our algorithms as a function of the number of devices present in each network. We vary the number of devices per network from 100 (400 total) to 1250 (5000 total). Figure 12 has the resulting total normalized throughput for each algorithm.

As the number of devices increases, the total normalized throughput increases as well. At 100 devices per network, our algorithms produce a total normalized throughput value close to 0.32 compared to 0.08 for the legacy algorithm. At 1250 devices per network, the gradient ascent approach results in a normalized throughput close to 1.08 compared to 0.42 for the legacy algorithm.

An inverse effect can be seen with the total packet delivery ratio as more devices means more collisions. Figure 13 has a plot showing the impact of the number of devices on the total packet delivery ratio. For a small number of 100 devices per network only, all the algorithms show a similar packet delivery ratio close to 0.9. The differences between the algorithms widen as the number of devices increases. At 1250 devices per network *i.e.,* 5000 total in the simulation area, the packet delivery ratio for our gradient ascent algorithm is 0.46,

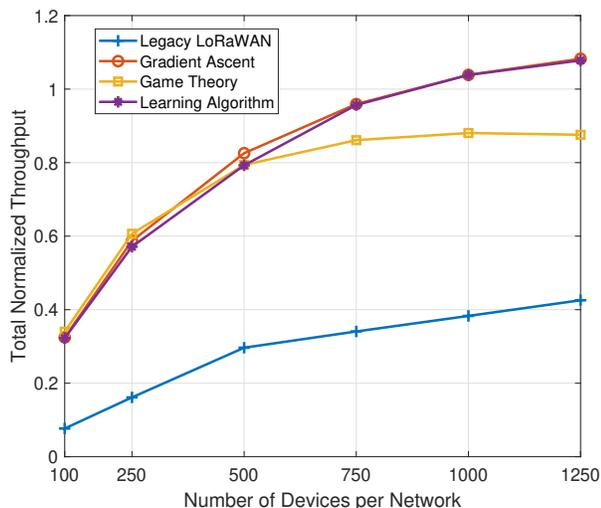compared to approximately 0.28 for the legacy approach.

Fig. 12. Impact of the number of devices on the total normalized throughput
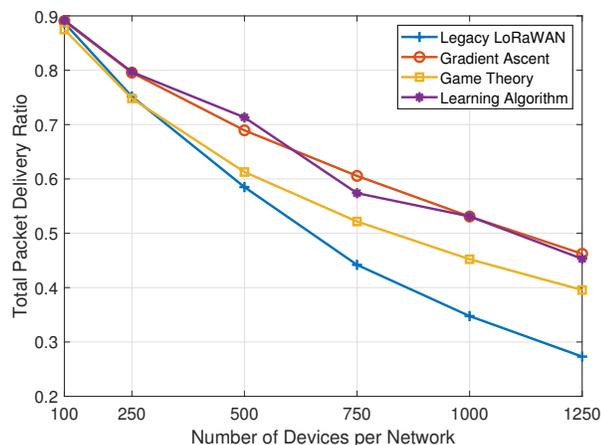

Fig. 13. Impact of the number of devices on the total packet delivery ratio

Finally, we study the fairness our proposal enforces between the different SFs, and how it differs with the number of devices present. We calculate Jain's fairness index [25] following the Raj-Jain equation as:

$$\mathcal{J}(x_1, x_2, ... x_S) = \frac{(\sum_{s=1}^{S} x_s)^2}{S \cdot \sum_{s=1}^{S} x_s^2} \quad (24)$$

Where x_s represents the throughput per spreading factor, and S is the total number of available spreading factors. For 2000 total devices in the networks, the ADR approach produces a Jain index value of only 0.469, compared to 0.893 for the game approach and 0.9312 for the gradient ascent algorithm. In the case of 5000 total devices in the simulation, the legacy approach produces a Jain index value of 0.735, with 0.958 for the game approach and 0.999 for the gradient ascent algorithm. The increase in the fairness among different SFs

with the increase in the number of devices might seem counter-intuitive. Nonetheless, it is a direct effect of previously less populated SFs being crowded with additional devices. Even with the resulting increase in collisions, the throughput would still increase as illustrated in the simulation above.

E. Effect of the Area Size

We vary the simulation area size and study the effects it has on the performance of our algorithms. We consider seven cases with the square area side A = 1, 2, 4, 8, 12, 16, and 20 km. The impact on the total normalized throughput can be seen in Fig. 14

Up until A = 20 km, our proposed algorithms are unaffected by the change in the size of the area where the devices are spread. The gradient ascent and learning algorithms produce values near 0.95 and the game theoretic algorithm values close to 0.86. On the other hand, the legacy LoRaWAN algorithm suffers when the area size is small. This is because the algorithm will have most of the devices transmitting on the same SF of 7. The performance of the algorithm gradually improves as A increases until it starts degrading with the rest of the algorithms when A is greater than 16 km.
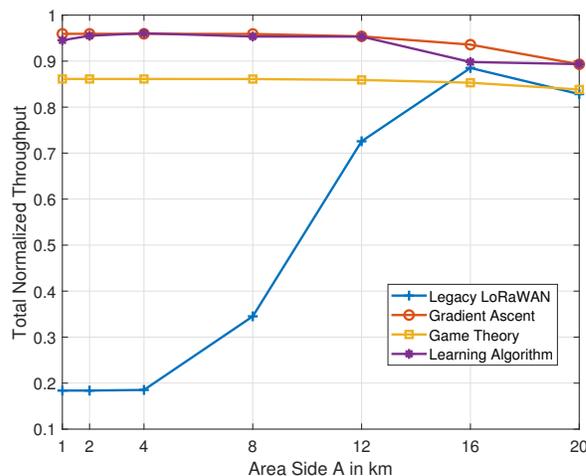

Fig. 14. Impact of the area size on the total normalized throughput

A similar impact can be seen on the packet delivery ratio. Figure 15 shows the packet delivery ratio for each of the algorithms as a function of the area size. For the gradient ascent and learning algorithms, the value stagnates at about 0.6 and for the game theory based algorithm at about 0.52 before it ticks up slightly at the end. For the legacy LoRaWAN algorithm, a low packet delivery ratio of 0.35 is recorded for small area sizes before it gradually increases to reach a maximum of about 0.6 at A = 16 km. Finally, the impact of an increase in the area size is seen primarily in the number of covered devices N_c which decreases as A increases. This is illustrated in Table V.
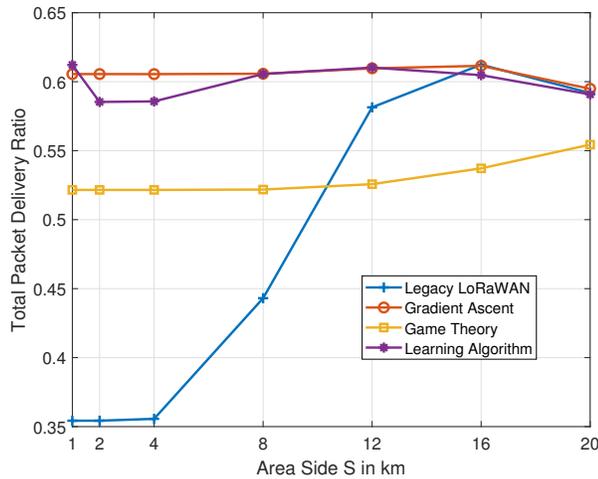
Fig. 15. Impact of the number area size on the total packet delivery ratio

TABLE V
N_c AS A FUNCTION OF S

Area Size S [km]	Total Number of Covered Devices N_c	%
≤ 4	3000	100%
8	2997	99.9%
16	2806	93.5%
20	2590	86.3%

F. Impact of Traffic From a Non-Cooperative Operator

A classic LoRaWAN problem is the existence of an external interfering source of traffic. While the multiple operators are interfering upon each other in our scenario, they still maintain a level of cooperation. In this simulation, we consider that an additional external interference source from a non-cooperative co-located network exists. The network has the same number of devices as the others and uses ADR for SF selection. We vary the packet generation rate per device for this external source and plot the impact it has on the total normalized throughput and the total packet delivery ratio.

Figure 16 has the results in terms of total normalized throughput for the four cooperative networks. All the algorithms show a degradation in terms of throughput as the intensity of the added traffic source increases. The gradient ascent proposal varies between 0.95 and 0.92 while the legacy approach results varies between 0.326 and 0.313. Similar results can be noted in terms of the total packet delivery ratio. Figure 17 shows that the performance of all the algorithms degrades as the intensity of the external source increases. Nonetheless, in terms of both the packet delivery ratio and the normalized throughput, the degradation in performance for the operators remains minimal as our algorithms show resilience to external interferers.

G. Energy Consumed per Delivered Byte

We aim to assess the energy expenditure per successful delivery. For this simulation, we vary the packet generation
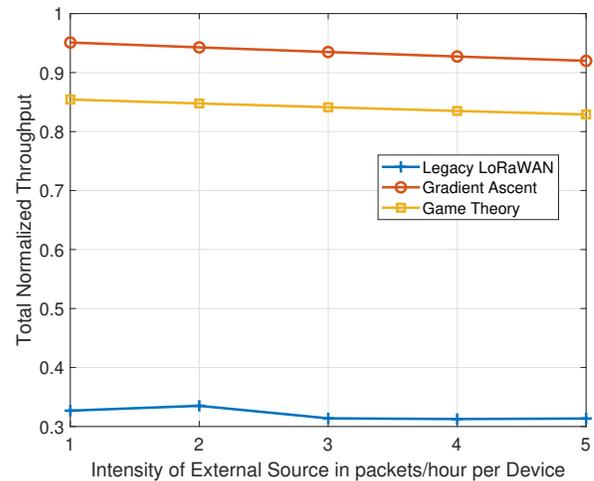

Fig. 16. Impact of external traffic from non-cooperative source on the total normalized throughput
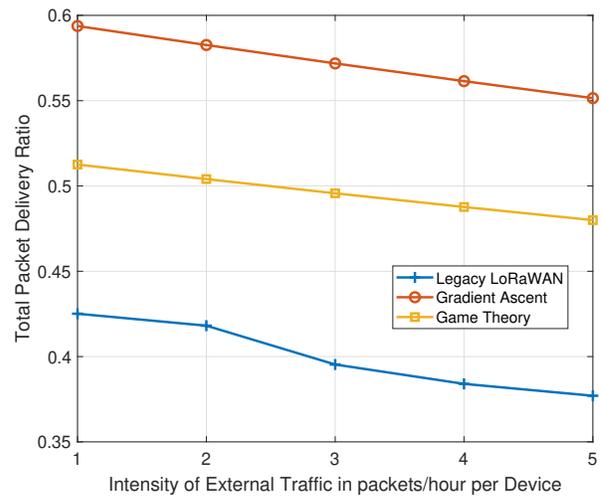

Fig. 17. Impact of external traffic from non-cooperative source on the total packet delivery ratio

rate and compare between our gradient ascent proposal and the legacy LoRaWAN approach. Following [26], the energy cost of data delivery can be expressed as:

$$EC_{delivery} \approx \frac{a \cdot \exp(2G_s)}{l_{pay}}, \qquad (25)$$

where a is the energy consumed per transmission attempt, G_s is the normalized traffic load per spreading factor and l_{pay} is the payload size. The results are plotted in Fig. 18.

The results show that our gradient ascent proposal always leads to a lower expenditure per delivered byte. For $\lambda = 1$ packet/hour, our algorithm saves about 1% in expenditure per byte delivered. For $\lambda = 5$ packets/hour, our algorithm consumes 30% less energy per byte delivered.
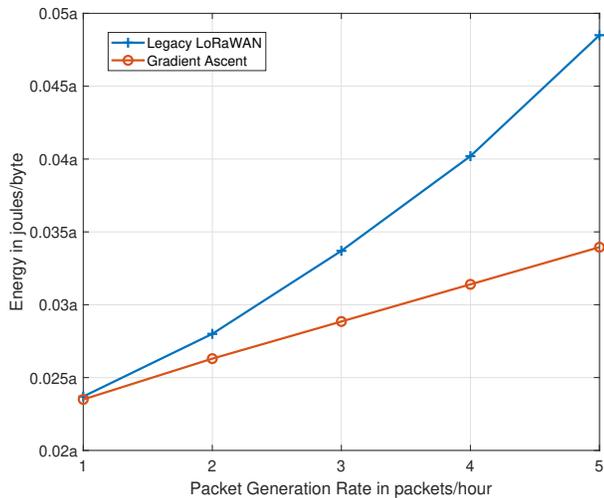
Fig. 18. Energy consumed per delivered byte as a function of the packet generation rate

H. Impact of the Number of Gateways and Comparison to the State-of-the-Art

In this section, we aim to compare our proposals to the state-of-the-art, as well as study the impact of the number of gateways in the simulation on the performance of the algorithms. To this end, we consider three gateway deployment scenarios as illustrated in Fig. 19. The rest of the parameters remain as in Table IV. In addition, we simulate two algorithms from the state-of-the-art. The first, ExpLoRa-SF, seeks to balance between the utilization of SFs with constraints relating only to the radio conditions of devices. The second, RS-LoRa, assigns probabilities for each device selecting an SF depending on the rate capabilities on each one.
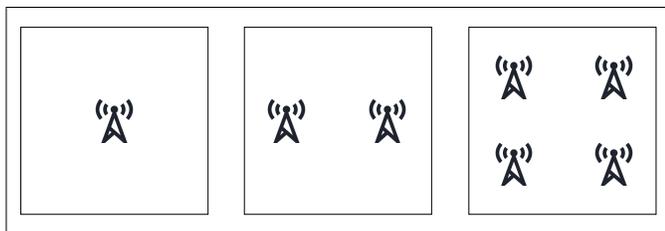

Fig. 19. Our different gateway scenarios

We study the impact of the number of gateways on the coverage. Table VI shows the total number of covered devices for each scenario.

TABLE VI
N_c AS A FUNCTION OF THE NUMBER OF GATEWAYS

Gateways per Network	Number of Covered Devices N_c	%
1	2597	86.57%
2	2902	96.7%
4	2999	99.9%

We start with evaluating the total normalized throughput. Figure 20 shows the results for our gradient ascent algorithm alongside the two proposals from the state-of-the-art and the legacy LoRaWAN algorithm as well.
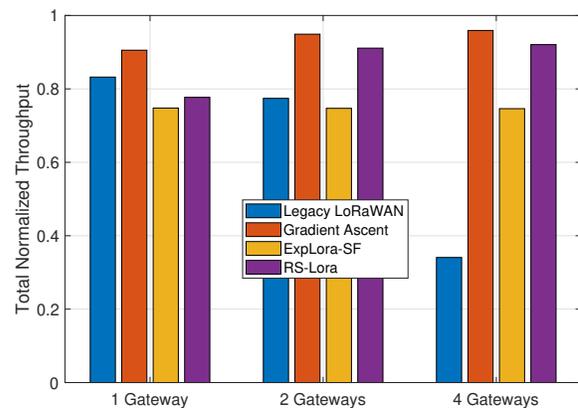

Fig. 20. Total normalized throughput as a function of the number of gateways

Our gradient ascent algorithm outperforms both proposals in terms of throughput regardless of the number of access points. For the gradient ascent algorithm and RS-LoRa, the throughput increases with the number of access points. For ExpLoRa-SF it stagnates, and for legacy LoRaWAN approach it actually degrades.
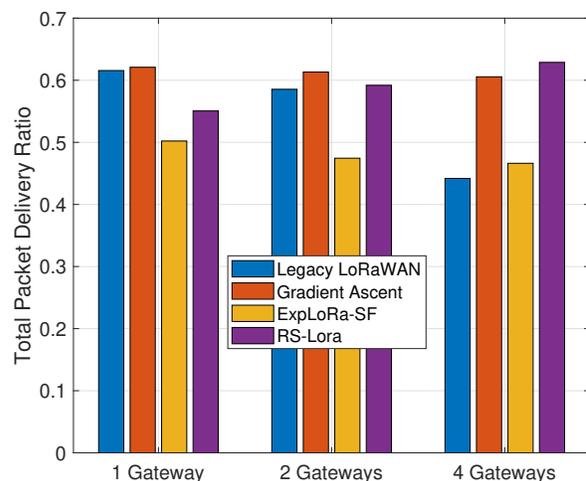

Fig. 21. Total packet delivery ratio as a function of the number of gateways

We additionally look at the results in terms of the total packet delivery ratio, with the results seen in Fig. 21. In case of one or two gateway scenarios, the legacy LoRaWAN approach performs well in terms of packet delivery ratio, with our proposal producing the better values in the vicinity of 0.62. In the case where four gateways are present, the RS-LoRa algorithm slightly outperforms our proposal in terms of the total packet delivery ratio. Nonetheless, implementing

RS-LoRa in this multi-operator scenario requires devices with computational power not available for all classes of LoRa devices.

I. Comments and Analysis

In this paper, we proposed three different approaches for SF assignment under different levels of inter-operator cooperation. We classified this cooperation as illustrated in Table VII and compared it to legacy LoRaWAN SF selection process wherein no such cooperation exists.

TABLE VII
INTER-OPERATOR COOPERATION LEVELS FOR OUR PROPOSALS

Proposal	Inter-Operator Cooperation Level
Gradient Ascent	Partial
Game Theory	Partial
Learning Algorithm	Minimal/Absent

We further highlight the main points that can be drawn from our different simulation scenarios:

(a) Our gradient ascent based proposal generally produces the best results in terms of total normalized throughput and total packet delivery ratio as well.

(b) Our game proposal achieves total equity between the operators in terms of individual normalized throughput, but at a cost in the total normalized throughput.

(c) For less congested scenarios, resulting from small packet sizes, low packet generation rates, or a small amount of devices, the game theoretic proposal becomes the most suitable.

(d) The simulations in general show the weaknesses of the legacy ADR approach. Whenever a majority of the devices are in good radio conditions, usually a requisite for reliable wireless communications, the algorithm would stack the lower SFs causing more collisions.

(e) Thus, the scalability of LoRaWAN depends on smarter SF assignment algorithms. This however cannot be the cause of any added signaling in the network, especially not on the downlink as duty cycle limitations would erase any gains as attested for in [4].

(f) Gateway densification is necessary to improve device coverage even in moderate area sizes. Nonetheless, this does add to the drawbacks of the legacy LoRaWAN SF selection algorithm with lower SFs expected to be crowded as a result.

IX. CONCLUSION

In this paper, we proposed multiple spreading factor assignment algorithms for LoRaWAN in the case of multi-operator co-location. We started with introducing an optimal formulation for spreading factor assignment and proceeded to propose an iterative gradient ascent algorithm which can solve the problem for different operators with a small amount of cooperation guaranteed via secure aggregation. Under the same operator cooperation assumption, we proposed a game theory based SF assignment algorithm, wherein the utility that each network seeks to maximize is the logarithmic sum of its own normalized throughput per spreading factor. Finally, we introduced an LSTM network to enable different network operators to predict the success rate per spreading factor. Using the latter, we proposed a spreading factor assignment algorithm for the different networks that works without cooperation and with minimum synchronization. We simulate our algorithms under multiple scenarios including: different packet sizes, packet generation rates, device numbers, area sizes, and gateways deployed. We show that with respect to legacy LoRaWAN, our proposals significantly improve the total normalized throughput as well as the total packet delivery ratio. And finally, we show that our gradient ascent algorithm provides better normalized throughput results with respect to proposals in the state-of-the-art.

X. ACKNOWLEDGMENT

REFERENCES

[1] L. Horwitzi, "The future of iot miniguide: The burgeoning iot market continues," July 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html

[2] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "Lorawan specification," *LoRa alliance*, 2015.

[3] N. Abramson, "The aloha system: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, 1970, pp. 281–285.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] F. Cuomo, J. C. C. Gámez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "Towards traffic-oriented spreading factor allocations in lorawan systems," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2018, pp. 1–8.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[16] F. Zhang, M. Zhou, L. Qi, Y. Du, and H. Sun, "A game theoretic approach for distributed and coordinated channel access control in cooperative vehicle safety systems," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[17] D. Cheng, T. Xu, F. He, and H. Qi, "On dynamics and nash equilibriums of networked games," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 1, pp. 10–18, 2014.

[18] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th international conference on machine learning (icml-03)*, 2003, pp. 928–936.

[19] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

[20] D. P. Palomar, "Convex primal decomposition for multicarrier linear mimo transceivers," *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4661–4674, 2005.

[21] 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.

[22] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica: Journal of the Econometric Society*, pp. 520–534, 1965.

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

[24] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[25] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," in *ATM Forum contribution*, vol. 99, no. 45, 1999.

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

Kinda Khawam got her engineering degree from Ecole Superieure des Ingenieurs de Beyrouth (ESIB) in 2002, the Master's degree in computer networks from Telecom ParisTech (ENST), Paris, France, in 2003, and the Ph.D. from the same school in 2006. She was a post-doctoral fellow researcher in France Telecom, Issy-Les-Moulineau, France in 2007. Actually, she is an associate professor and researcher at the University of Versailles in France. Her research interests include radio resource management, modeling and performance evaluation of mobile networks.

Samer Lahoud received the Ph.D. degree in communication networks from IMT Atlantique, Rennes, in 2006. After his Ph.D. degree, he spent one year at Nokia Bell Labs Europe. From 2007 to 2016, he was with the University of Rennes 1 and with IRISA Rennes as an Associate Professor. He is currently an Associate Professor with the Saint Joseph University of Beirut, where he lectures computer networking courses with the Faculty of Engineering, Ecole Supérieure d'Ingénieurs de Beyrouth (ESIB). His research activities focus on routing and resource allocation algorithms for wired and wireless communication networks.

Steven Martin received his Ph.D. degree from IN-RIA, France, in 2004. Since 2005, Steven MARTIN is working at Paris-Saclay University. Leading the research group "Networking and Optimization" at LRI (the Laboratory for Computer Science at Paris-Saclay University, joint with CNRS), his research interests include quality of service, wireless networks, network coding, ad hoc networks and real-time scheduling. He is the author of a large number of papers published in leading conference proceedings and journals.

Hassan Fawaz received his diploma in Telecommunication Engineering from the Lebanese University in 2015. He received his masters degree in Telecom Networks and Security and his Ph.D. in Wireless Communications from Saint Joseph University of Beirut in 2016 and 2019, respectively. He is currently a PostDoc researcher at the University of Versailles, Paris-Saclay, with his work revolving around resource allocation in LoRaWAN. His research interests include full-duplex communications, IoT, and machine learning solutions for wireless networks.

Melhem El Helou (S'08–M'15–SM'19) received the engineer's and master's degrees in Telecommunications and Networking Engineering from Ecole Supérieure d'Ingénieurs de Beyrouth (ESIB), Saint Joseph University of Beirut, Beirut, Lebanon, in 2009 and 2010, respectively and the Ph.D. degree in Computer and Telecommunications Engineering from IRISA Research Institute, University of Rennes 1, France and Saint Joseph University of Beirut, in 2014. He joined ESIB in September 2013 where he is currently an Assistant Professor (fr: Maître de conférences). His research interests include wireless networking, Internet of Things, and quality of service.