Fast Q-learning for Improved Finite Length Performance of Irregular Repetition Slotted ALOHA

Eleni Nisioti and Nikolaos Thomos *Senior Member, IEEE*

***Abstract*—In this paper, we study the problem of designing adaptive Medium Access Control (MAC) solutions for wireless sensor networks (WSNs) under the Irregular Repetition Slotted ALOHA (IRSA) protocol. In particular, we optimize the degree distribution employed by IRSA for finite frame sizes. Motivated by characteristics of WSNs, such as the restricted computational resources and partial observability, we model the design of IRSA as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). We have theoretically analyzed our solution in terms of optimality of the learned IRSA design and derived guarantees for finding near-optimal policies. These guarantees are generic and can be applied in resource allocation problems that exhibit the *waterfall effect*, which in our setting manifests itself as a severe degradation in the overall throughput of the network above a particular channel load. Furthermore, we combat the inherent non-stationarity of the learning environment in WSNs by advancing classical Q-learning through the use of virtual experience (VE), a technique that enables the update of multiple state-action pairs per learning iteration and, thus, accelerates convergence. Our simulations confirm the superiority of our learning-based MAC solution compared to traditional IRSA and provide insights into the effect of WSN characteristics on the quality of learned policies.**

***Index Terms*—Medium Access Control, Q-learning, Irregular Repetition Slotted ALOHA, wireless sensor networks, POMDP, Independent learning**

I. Introduction

Wireless sensor networks (WSNs) have drawn the attention of the research community due to their wide applicability and the challenges inherent in their optimization. Characteristics such as uncertain and changing channel conditions render static, pre-designed solutions inefficient, while restrictions such as the limited computational power, battery capacity and transmission range of sensors impose limits to the performance of potential solutions. In order to render WSNs adaptive, reinforcement learning is often employed [1] and preferred to supervised learning, as generating labeled data for WSNs is expensive and a priori simulating all possible conditions the network may operate in is not feasible. The challenges that the aforementioned traits impose to learning, such as partial observability, decentralization and non-stationarity, render traditional learning solutions inappropriate due to their prohibitive time and computational complexity, as well as the lack of formal convergence guarantees.

Manuscript received June 17, 2019; revised Sep. 27, 2019; accepted Nov. 21, 2019. This work was partly funded by the European Union Horizon 2020 research and innovation programme under Marie Sklodowska-Curie Research and Innovation Staff Exchange grant through the project RECENT (grant agreement No. 823903). E. Nisioti and N. Thomos are with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, United Kingdom (e-mail:e.nisioti, nthomos@essex.ac.uk).

Medium Access Control (MAC) orchestrates the access of devices to the common communication channel with the objective of maximizing its throughput. In our work, we study MAC design under the Irregular Repetition Slotted ALOHA (IRSA) protocol [2], a state-of-the-art probabilistic protocol that employs successive interference cancellation (SIC) to resolve collisions. IRSA is ruled by the employed degree distribution, a probability distribution that describes how many replicas of the packets available to each sensor should be transmitted in each frame. The work in [2] proves that IRSA can approach optimal throughput in asymptotic settings. However, the assumption of frames of infinite length is not accurate in practical implementations, where the length of the frame size is directly associated with delays in the transmission of packets and, therefore, is often restricted to small values.

To design adaptive MAC protocols appropriate for WSNs, we first formulate MAC design as a multi-agent system, where agents learn how to transmit their packets by interacting with their environment in the framework of decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) [3] [1]. Specifically, we assume that each sensor is an agent aiming at maximizing the throughput of the sensor network. Agents in our formulation perform actions that correspond to the coefficients of the degree distribution, with the objective of maximizing the common channel throughput, while rewards are associated with the success of transmission. We also consider partial observability in our definition of states, as sensors can observe only information that is local to them and is possibly inaccurate due to the limited capabilities of sensors.

The reinforcement learning algorithm employed in our solution is Q-learning [5], which has been extensively used for rendering WSNs adaptive [1], as it achieves a fine balance between low complexity and satisfactory performance. Q-learning does not require a model of the environment, a particularly advantageous trait for WSNs, where a model of the environment is often absent. Although model-based reinforcement learning [5] can be used to derive such a model, it significantly increases the time and computational complexity of the learning algorithm, while the inherent non-stationarity suggests that re-modeling will often be required. Another advantage of Q-learning is that it can be applied both in an offline and online manner, in contrast to Monte Carlo algorithms [5] that can only be applied offline. Finally, Q-learning was selected as it is an off-policy learning algorithm and, thus, convergence in stationary environments is guaranteed irrespectively of the policy being followed, in contrast to on-policy algorithms, such as SARSA [5], where convergence

[1] Part of this work was presented at PIMRC 2018 [4].

analysis is much more complex.

Ensuring realistic complexity is essential when designing solutions for WSNs. The large number of sensors and partial observability may lead to an explosion in the complexity of learning, which we remedy by employing two techniques: (i) adopting finite histories of observations to approximate the continuous beliefs of Belief MDPs, which significantly reduces the size of the state space and, as we prove in Section VII-A, can still lead to policies with near-optimal performance, and (ii) assuming that each sensor learns independently from other sensors, by updating its local Q-function based on its individual observations and actions. Although naïvely ignoring agents' interactions, this technique has been found to converge when coupled with exploitive exploration strategies in [6] and exhibits low complexity.

Furthermore, we investigate the effect that partial observability and decentralization have on the quality of the solution and prove the existence of near-optimal solutions for a POMDP employing a finite history of past observations. This result is novel and can be employed for characterizing the optimality of learning solutions for resource allocation problems that exhibit the waterfall effect. Our analysis is based on a work that corresponds to a centralized setting [7], we therefore investigate how decentralization and the assumption of independent learning affect the quality of the learned policies based on observations derived in [6] and our analysis of the equilibrium points of our setting.

Apart from the quality of the solution, speed is also an important criterion when evaluating a technique. In Q-learning, convergence to a stationary policy is guaranteed at the end of an episode, which refers to the period of interaction of an agent with its environment until a terminal state is reached, provided that the environment is stationary. However, in WSNs, the time-varying nature of the network and channel conditions renders the learning environment non-stationary, which effectively means that the optimal policy can change during the course of an episode. Thus, Q-learning will exhibit sub-optimal performance, if the environment changes at a rate quicker than its convergence rate. To address this issue, our solution equips Q-learning with the concept of virtual experience (VE) [8], where an agent updates multiple state-actions pairs at each Q-learning iteration by "imagining" state visits. These visits correspond to state-action pairs termed virtual, whose defining property is that they are equivalent in front of the unknown environment dynamics. Our theoretical analysis formulates the effect that virtual experience has on the convergence rate of Q-learning, which we also empirically measure in our simulations.

Our main contributions consist in:

- empowering IRSA with optimized degree distributions for small frame sizes so that throughput is significantly improved for high channel loads. We examine the performance of the proposed scheme in simulations of various settings and compare the derived distributions with those used by classical IRSA;
- the novel definition of IRSA design as a Dec-POMDP that adapts to varying communication conditions;
- the formulation of a fast Q-learning variant that utilizes finite and locally available information. Our learning algorithm comes with an analysis on the optimality of the MAC solution. We also characterize its complexity;
- a novel theoretical study of the effect of virtual experience on the convergence properties of Q-learning, as well as its impact on the convergence rate as empirically evaluated in our setting;
- the investigation of the impact of the waterfall effect on the behavior of agents and the ability of our proposed algorithm to alleviate it.

The rest of the paper is structured as follows. In Section II, we provide a short review on related works. Section III contains a discussion on how our design choices were motivated by properties of Q-learning in WSNs. Section IV models the considered WSN, highlighting underlying assumptions. Section V presents IRSA and formulates the optimization objective of our proposed learning algorithm. In Section VI, we present our self-configuring MAC protocol, henceforth referred to as RL-IRSA. In particular, we model MAC design under the Dec-POMDP framework and describe our Q-learning variant. In Section VII, we analyze the optimality, convergence rate and complexity of our solution. Section VIII exhibits the simulations performed to configure and evaluate our learning-based solution. Concluding remarks are presented in Section IX.

II. Related work

Due to the increasing need for efficient communication over shared channels, contention-based MAC protocols have seen a continuous improvement in their performance. The original Slotted ALOHA [9] is inappropriate for energy-constrained WSNs, as throughput, measured as the probability of successful transmission of a packet in a communication slot, cannot exceed a value of 0.37 [2]. Two modifications significantly improved its performance: (i) transmitting a number of predefined replicas of the original packets, introduced in Diversity Slotted ALOHA [10], and, (ii) employing SIC, a mechanism for resolving collisions, which was introduced in Contention Resolution Slotted ALOHA [11]. By combining SIC with the ability of sensors to transmit different numbers of replicas, decided by sampling the degree distribution, IRSA asymptotically achieves a throughput of 0.97 [2].

The performance of IRSA depends on the optimization scheme used to derive the degree distribution, with differential evolution traditionally preferred [2]. More recently, in [12], the use of Multi-armed Bandits (MABs) was introduced, as a remedy for inaccurate asymptotic analysis in non-asymptotic settings and as an alternative to computationally expensive finite length block analysis. This work has been proposed for an IRSA variant that incorporates users' prioritization [13]. The main disadvantage of MABs is that they are stateless and, thus, cannot take into account sensors' characteristics, such as battery level, memory size, etc., information that can be valuable to the decision-making process.

Learning automata (LA) is an alternative framework to MDPs that has successfully been employed in MAC design [14], [15]. LA have been found to converge in stationary

environments, where other learning algorithms fail [16], but their application in the domain of WSNs faces the same limitations with MABs, as they are also stateless. Feedforward structured networks of generalized learning automata [17] overcome this problem by introducing the notion of states. However, the quality of the offered local optima depends on both the complexity of the problem as well as their architecture [16], which renders them inappropriate for WSNs.

Convergence of Q-learning to optimal policies is guaranteed in problems formulated as a Markov Decision Process [18], but partial observability renders states non-Markovian. In this case, the framework of Belief MDPs can be employed to learn optimal policies [19] and, in practice, finite approximations of beliefs, in the form of tuples of consecutive observations, are preferred to reduce complexity [20]. In [7], the optimality of POMDPs employing histories of observations is studied under the framework of finite state controllers, which are finite state machines that can be employed to convert a POMDP to a finite state Markov chain. The analysis in [7] proves that there exists a near-optimal policy for such a POMDP independently of the initial distribution of beliefs.

The imperfectness of assuming independent learning in a multi-agent setting is investigated in [6] and the conditions sufficient for convergence are derived. Although well-performing solutions are often found [6], [20], the equilibrium points can correspond to sub-optimal solutions [6]. Game-theoretic approaches are often employed [21], [22] to study multi-agent learning problems, but the entailed complexity of Nash equilibrium points in multi-state sequential problems, as well as the assumptions of hyper-rationality and omniscience of players are not appropriate for WSNs [23].

The convergence rate of classical Q-learning was studied in [24] in relation to the learning rate and discount factor. In [25], the effect of VE on the convergence rate of Q-learning was examined. In particular, the work in [25] separated the effect of the environment into "known" and "unknown" dynamics and introduced the concept of VE in their attempt to extrapolate experience of rewards to states that do not affect the unknown dynamics and are, therefore, equivalent in the light of new information. VE was applied to post-decision states, in contrast to our solution applying it on actual states. A related concept is that of experience replay [26], where agents "remember" state visits, instead of imagining them. Experience replay leverages past experience during the application of Q-learning, a technique that has been proven essential for training deep neural networks used as Q-function approximators.

III. Q-LEARNING IN WSNS: MOTIVATING REMARKS

Despite the widespread use of Q-learning in various problems related to WSNs, it exhibits traits that impose challenges to the design of realistic, efficient and well-performing solutions. Next, we explain how we address these challenges during the design of our learning algorithm.

The time complexity of Q-learning is primarily dictated by the size of the state-action space. WSNs' characteristics such as partial observability, continuous quantities, and increasing network sizes often lead to prohibitively large state-action spaces. In tabular Q-learning, experience is stored in the Q-table, which, due to the aforementioned traits, is often impractical to implement. Deep reinforcement learning (DRL) approaches can be used to approximate the Q-function [21], [22], [27], but convergence of DRL to optimal strategies is an open issue and training has to be performed offline and centrally due to the computational complexity of training deep neural networks. This, in combination with the fact that, if the network conditions change significantly retraining of the system is required, suggests that DRL cannot offer computationally feasible solutions that ensure adaptivity for sensor networks of realistic size and for changing channel conditions. To overcome this challenge, we employ techniques that help maintain a reasonable state-action space size without employing function approximation. In particular, we propose a technique to reformulate a continuous action space as a discrete one and employ finite histories of observations to approximate beliefs.

Q-learning is traditionally applied on the global state-action space, which in our setting, would either require sensors to learn jointly or assume a centralized point of control. The former approach imposes extensive communication among sensors, which consumes resources and can potentially affect the operation of the network, while the latter jeopardizes it. Furthermore, the restricted capabilities of sensors limit the transmission range and suggest that interactions among sensors can only be local. To this aim, we propose a decentralized solution, while we further reduce complexity by assuming independence in learning among sensors.

Due to the absence of a model of the environment, Q-learning requires extensive interaction with it. This leads to slow convergence to the optimal solution, thus sensors will need to consume significant resources prior to finding an optimal transmission strategy. Also, low throughput is anticipated at the beginning of the episode, when the learned policy is still far from the optimal one. To avoid these shortcomings in our solution, we accelerate the convergence rate of Q-learning through the use of virtual experience, and thus find well-performing policies within reasonable training times.

TABLE I: System-related variables

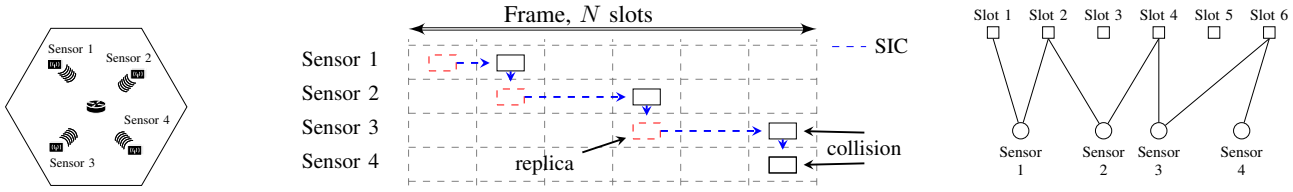
Symbol	System-related
M	number of sensors
N	number of slots in frame
G	channel load
T	packet throughput
K	number of transmitted packets
PLR	probability loss rate
F	size of packet
S^u	uncontrolled state
L_E	number of learning iterations
α	learning rate
γ	discount factor
w	history window
L	coverage time
ϕ	exponent of learning rate
$\mathcal{T}$	virtual experience transformation

TABLE II: Sensor-related variables

Symbol	Sensor-related
C_t	condition
l	number of replicas
F_t	number of arrivals in buffer
B	size of sensor's buffer
d	maximum number of replicas
b_t	current state of buffer
$\Lambda(x)$	degree distribution
$\mathcal{S}$	state-space
$\mathcal{A}$	action-space
Ω	observation-space
$\mathcal{H}$	history-space
$\widetilde{\mathcal{H}}$	history-space using VE
R_t	immediate reward
ρ_t	expected reward
$\pi(s)$	policy
$Q^\pi(\cdot)$	Q-function under π
$V^\pi(\cdot)$	state value function under π

IV. NETWORK AND SENSOR MODELING

This section presents our assumptions made regarding the physical layer. Tables I and II summarize the notation used for system-related and node-related variables, respectively.



(a) (b) (c)

Fig. 1: (a) A sensor network where sensors transmit wirelessly to a common access medium (b) Transmission under the standard Slotted ALOHA and IRSA protocols. (c) Bipartite graph describing IRSA.

A. Physical layer

We consider frequency non-selective channels, characterized at the beginning of each time frame by the traffic G_t, which represents the average number of attempted packet transmissions by all sensors per time slot, with t indicating the time index at the beginning of a frame. We assume that traffic can be estimated perfectly in light of the number of sensors and frame size, and that it remains constant during a frame, similar to the works in [28], [29].

Following the work in [25], we assume that the packet throughput T_t and number of transmitted packets K_t can be expressed as:

$$T_t = T(G_{t-1}, K_{t-1}, PLR_{t-1}) \tag{1}$$

$$K_t = K(G_t, T_t, C_t, PLR_t) \tag{2}$$

where PLR_t is the packet loss rate and C_t denotes a sensor's condition. The latter can include any information that could potentially affect the behavior of sensors, such as the buffer state and battery level. For the sake of simplicity, we consider only the buffer state (number of packets in the buffer) of sensors. However, the proposed framework is generic, and, depending on the application of interest, can incorporate additional characteristics to C_t. From (1), we can also see that T_t is a non-deterministic function of its arguments, as sensors randomly select the slots to transmit in. Note that the proposed framework is oblivious to the underlying modulation, coding schemes and channel noise.

B. Buffer and traffic model

We assume that the transmission buffer of a sensor is modeled as a first-in first-out queue. At the beginning of a frame, a source injects F_t packets into a finite-length buffer of capacity B. Therefore, the buffer state $b_t^i \in \mathcal{B} = \{0, 1, ..., B\}$ of a sensor i evolves recursively as follows:

$$\begin{aligned} b_0^i &= b_{init}^i \\ b_{t+1}^i &= \min\{b_t^i - T_t^i(PLR_t, K_t, G_t) + F_t^i, B\} \end{aligned} \tag{3}$$

where b_{init}^i denotes the initial buffer state and $T_t^i(PLR_t, K_t, G_t)$ is the packet goodput, representing the number of successfully and error-free transmitted packets in a frame for sensor i, i.e., packets that were: (i) successfully transmitted (no collisions occurred or the occurred collisions were resolved through SIC), and (ii) not corrupted by channel noise. This follows the convention in [25] and, in our framework, coincides with throughput. We consider that the packets arriving after the beginning of frame t cannot be transmitted until frame $t+1$. Also, packets whose transmission fails stay in the buffer for future retransmission.

V. Problem formulation

A. IRSA

Let us consider a network of M sensors collecting measurements from their environment and transmitting them to a core network for further process, as shown in Fig. 1a. The main bottleneck of the operation of the network is the transmission of the packets sensors possess through a common communication channel, as it is also used by neighboring sensors. Abiding to Slotted ALOHA and its variants, in our work time is divided into frames of fixed duration, each one consisting of N time slots. At the beginning of each frame each sensor randomly chooses one of the N available slots to transmit its packet. The channel traffic can be calculated as $G = M/N$. IRSA aims at maximizing the normalized throughput T, defined as the probability of successful packet transmission per slot. Note that this definition is transparent to the size of the transmitted packets and slot duration, but can easily be converted to the commonly employed measurement unit of bits/s as Tp_t/s_t, where p_t denotes the size of a packet in bits and s_t the duration of a slot in secs.

In IRSA, a sensor has the capability of transmitting a variable number of replicas of the original message in the available time slots, decided by randomly sampling the node perspective degree distribution. The latter is a polynomial probability distribution describing the probability Λ_l that a sensor transmits l replicas of its message at a particular time frame, which is expressed as [2]:

$$\Lambda(x) \triangleq \sum_{l=1}^{d} \Lambda_l x^l, \quad x \in [0, 1] \tag{4}$$

where d is the maximum number of replicas a sensor node is allowed to send. Packet replicas under IRSA are indicated in Fig. 1b with red, dashed outlines. If one of the replicas is transmitted in a collision-free slot, then the packet is successfully transmitted. If however two replicas collide, as highlighted in Fig. 1b, they might still be recovered by removing the interference of a replica that has previously been successfully

[2]When defining degree distributions, we denote the degree of a coefficient as a subscript, while a superscript corresponds to raising to a power. Also, in the rest of the paper, indexes appearing as subscripts (superscripts) refer to the time (sensor) index.

received. This leads to substantial improvements in the network's throughput. Fig. 1b illustrates SIC by indicating with a blue, dashed line that the interference of a replica can be removed due to a replica of the same packet having been successfully received in another slot.

The packet transmissions depicted in Fig. 1b can be mapped to a bipartite graph consisting of variable nodes, representing sensors, and check nodes representing time slots. Accordingly, SIC is implemented as message-passing on this graph [2]. IRSA can also be characterized by the coefficients λ_l, which denote the probability of an edge being adjacent to a variable node of degree Λ_l. These coefficients can be used to form the following edge perspective degree distribution:

$$\lambda(x) \triangleq \sum_{l=2}^{d} \lambda_l x^{l-1}, \quad x \in [0,1] \tag{5}$$

The two degree distributions presented in (4) and (5) are related with the formula: $\lambda(x) = \Lambda'(x)/\Lambda'(1)$, where $\Lambda'(x)$ represents the derivative of the node perspective degree distribution. As our work is concerned with the optimization of $\Lambda(x)$, we henceforth refer to the node perspective degree distribution simply as degree distribution.

B. Optimization objective

The design of IRSA aims at selecting the values Λ_l in (4) so that the overall network throughput T is maximized. The dependence of throughput on the probability distribution chosen permits us to express T in terms of $\Lambda(x)$. This dependence becomes obvious if one considers the waterfall effect in IRSA [2], that indicates the existence of a threshold value G^* for the channel load G, above which transmission will fail with a probability bounded away from 0. It is observed in [2] that, in asymptotic settings ($N \to \infty$), the value of this threshold depends on the degree distribution, namely:

$$G^* < \frac{1}{\lambda_2 \Lambda'(1)} \tag{6}$$

Formally, our optimization objective can be cast as:

$$\begin{aligned} \text{Find:} \quad & (\Lambda^*(x)) : \arg\max_{\Lambda(x)} T(\Lambda(x)) \\ \text{subject to} \quad & \sum_{l=1}^{d} \Lambda_l = 1. \end{aligned} \tag{7}$$

IRSA is optimized in an asymptotic setting by iteratively alternating between choosing values for Λ_l and evaluating them using (6), in order to acquire a higher threshold G^*. The optimization of $\Lambda(x)$ is complex, and is performed offline. As the frame size N grows, the maximum allowable number of replicas d should also grow, and thus the number of Λ_l coefficients in $\Lambda(x)$ increases, making optimization harder. Differential evolution is usually employed to search through the degree distribution space [2], [13] due to its ability to efficiently optimize in large search spaces. However, the calculated degree distributions are sub-optimal for non-asymptotic cases, which are of interest here. We present our proposed framework to solve (7) in Section VI.

VI. RL-IRSA: AN ADAPTIVE MAC PROTOCOL

The discussion will proceed with the formulation of a solution that employs reinforcement learning for the problem of IRSA design, as described in Section V.

A. Modeling as MDP

Recall from (1) and (2) that there are two parameters affecting the environment's state: the current channel load G and a sensor's condition C_t. We first assume that the sensor network is a single agent that interacts with its environment, which includes the channel and itself. This concept is depicted in Fig. 2a. We model the problem as an MDP with state:

$$S = \times_{1 \le i \le M} S^i \times S^u \tag{8}$$

where $S \in \mathcal{S}$ is the state of the agent, $\mathcal{S}$ is the set of all states, S^i represents the state of sensor i and S^u stands for the part of the environment that is uncontrolled by the sensors and, in our formulation, corresponds to G.

The transition probabilities of this MDP can be cast as:

$$P(S^u_{t+1} \mid S^u_t, \times_{1 \le i \le N} S^i_t, K) = P(S^u_{t+1} \mid S^u_t) \tag{9}$$

$$P(S^i_{t+1} \mid S^u_t, \times_{1 \le i \le N} S^i_t, K_t, F^i_t) \propto -T^i_t(K, G) + F^i_t \tag{10}$$

where $T^i_t(\cdot)$ is the individual packet goodput of sensor i, i.e., the number of successfully transmitted packets for agent i, that depends on the current values K_t and G_t. Note that we did not provide a strict definition for the transition probabilitity of the state, but defined it to be proportional to the number of messages that will be added to the buffer before the next frame. This is because the state transition can vary for different applications (e.g. sensors dropping packets to avoid congestion or packets being stochastically corrupted by noise at the receiver). In our simulations, we assume that all packets are successfully added to the buffer, unless there is an overflow, in which case they are dropped.

From (9) we observe that the transitions of the uncontrollable state S^u are independent of the transmission strategy and the states of individual sensors. In particular, we assume that the channel probabilistically switches states based on the arrival and departure of sensors in the network. Further, from (10), we observe that individual transitions in the state of sensor depend on the states and actions of other sensors, channel load, noise conditions and packet throughput. Therefore, state transition independence for sensors does not hold.

The action $A \in \mathcal{A}$ of the agent, with $\mathcal{A}$ being the action space, consists of the joint actions of all sensors in the network. These actions represent the values of the coefficients Λ_l of the degree distribution in (4), that is:

$$A = \mathbf{A}^i \times \cdots \times \mathbf{A}^M, \text{ with } \mathbf{A}^i = \{\Lambda^i_1, \cdots, \Lambda^i_d\} \tag{11}$$

where Λ^i_l denotes the coefficient Λ_l of the degree distribution, as were presented in (4), of sensor i. Recall that d is the maximum number of replicas a sensor is allowed to send.

The above MDP formulation, although genuinely modeling the IRSA optimization problem, leads to a continuous action space, that scales exponentially with the number of sensors. This renders learning of the optimal action intractable for

large-sized problems. To circumvent this drawback, we redefine the actions as the number of replicas to send:

$$A = A^i \times \cdots \times A^M, \text{ with } A^i = l \text{ and } l \in \{1, \cdots, d\} \quad (12)$$

During the learning phase the agent finds a deterministic policy $\pi(a|s)$, with $s \in \mathcal{S}$ and $a \in \mathcal{A}$, by choosing the optimal A^i for each sensor, except for exploratory moves where a random action is performed. After learning has completed, the probability distribution $\Lambda(x)$ is computed using the information of visited state-action pairs. Therefore, upon implementation of our protocol the policy is probabilistic with $\pi(a|s) = \Lambda_a$, where Λ_a is a coefficient in $\Lambda(x)$. This technique allows us to leverage the benefits of maintaining a small action space, while using a stochastic policy.

The choice of the reward function is guided by our aim to design self-interested agents, attempting to improve the overall packet throughput while lacking access to a global performance measure, i.e., the channel throughput. We, thus, define the immediate reward R_t as the negation of the number of packets in the buffer of the sensor in the current time index. This reward makes sensors eager to transmit when their buffers are full, instead of making the decisions purely based on the outcome of the current transmission.

Clearly, the information included in the definition of states cannot be available to all sensors, as this would impose huge communication load. To alleviate this drawback of MDPs, in the next subsection we advance our modeling, based on partially-observable MDPs.

B. Dealing with partial observability

POMDPs [19] remedy the inability of an MDP to observe its state by introducing observations, which contain information that is relevant but insufficient to describe the actual state on their own. In our case, the network and sensors cannot observe $S^u = G$, as this requires a global view of the environment. We, therefore, constrain observability to information only locally available to sensors. Following our description in Section IV regarding a sensor's condition C, we assume that the only state-related information a sensor has access to is the number of messages stored in its buffer, that is:

$$\Omega = \Omega^1 \times \cdots \times \Omega^M, \quad \text{with} \quad \Omega^i = b^i \quad (13)$$

POMDPs can be optimally solved using the framework of Belief MDPs [19], but this renders learning intractable, as it is performed in continuous state spaces. Instead of beliefs, we adopt a fixed history window w and approximate beliefs with a finite-history of observations, which we define as:

$$H_t = \{\Omega_{t-w+1}, \cdots, \Omega_{t-1}, \Omega_t\} \quad (14)$$

We have neglected the distributed nature of the problem in order to focus on the decision process formulation. Next, we proceed by reformulating it under the Dec-POMDP framework. [3].

C. Dec-POMDP Formulation

Decentralized Partially-observable MDPs offer a powerful framework for designing solutions that take into account partial observability and operate in a distributive way. The uncontrolled state of the environment includes information about the number of agents and the number of slots per time frame, both expressed through G. Fig. 2b depicts the sensor network as a Dec-POMDP.

Definition 1. (Dec-POMDP) A decentralized partially observable Markov decision process is defined as a tuple $\langle \mathcal{M}, \mathcal{S}, \mathcal{A}, T, R, \Omega, O, w, I \rangle$, where $\mathcal{M}$ is the set of agents, $\mathcal{S}$ is the finite set of states, $\mathcal{A}$ is the finite set of joint actions, T is the transition probability function, R is the immediate reward function, Ω is the finite set of joint observations, O is the observation probability function, w is the history window and I is the initial state distribution of beliefs at time $t = 0$.

Definition 1 extends the single-agent POMDP model by considering joint actions and observations. In our case $A^i \in \{1, 2, \cdots, d\}$, $\Omega^i \in \{0, 1, ..., B\}$ and R^i is the individual reward agent i observes, as described in Section VI-A. Thus, our algorithm does not need a common reward function R, but agents individually measure their rewards based on their observations. Note that the state space of the Dec-POMDP coincides with the state space of the POMDP, defined in (8). In regard to beliefs, we initialize them, without loss of generality, by sampling a uniform distribution taking values in the range $[0, 1, \cdots, B]$. As we prove in Section VII-A, the existence of an ϵ-optimal solution is independent of this initialization.

D. Learning in a Dec-POMDP framework

In reinforcement learning we quantify how good a particular state is by estimating a value function. We define the value function under a policy $\pi(a|s)$ as the expected discounted reward starting from state s and then following that policy:

$$V_\pi(s) = E_\pi[\rho_t | S_t = s] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^t R_{t+k+1} | S_t = s \right] \quad (15)$$

where ρ_t is the expected reward, and R_t is the immediate reward at time slot t. The parameter $0 \leq \gamma < 1$ is the discount factor that controls the effect of future rewards in the current state; a value of γ closer to zero makes the agent myopic, while when γ is close to 1 the agent is farsighted. Specifically in Q-learning, instead of the value function $V_\pi(s)$, the $Q(s, a)$ function, often termed as Q-function, is employed. The Q-function is defined as the expected discounted reward starting from s, taking the action a, and thereafter following policy π.

The aim is to find the optimal policy, i.e. the policy that maximizes the expected reward for all states, defined as:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \big(Q^*(s, a)\big), \quad \text{with } s \in \mathcal{S} \quad (16)$$

where $Q^*(s, a)$ denotes the optimal Q-function. Q-learning in a Dec-POMDP can be described by the following update mechanism:

$$Q(H_t, A_t) = (1 - \alpha) Q(H_t, A_t) + \alpha [R_t + \gamma \max_a Q(H_{t+1}, a)] \quad (17)$$

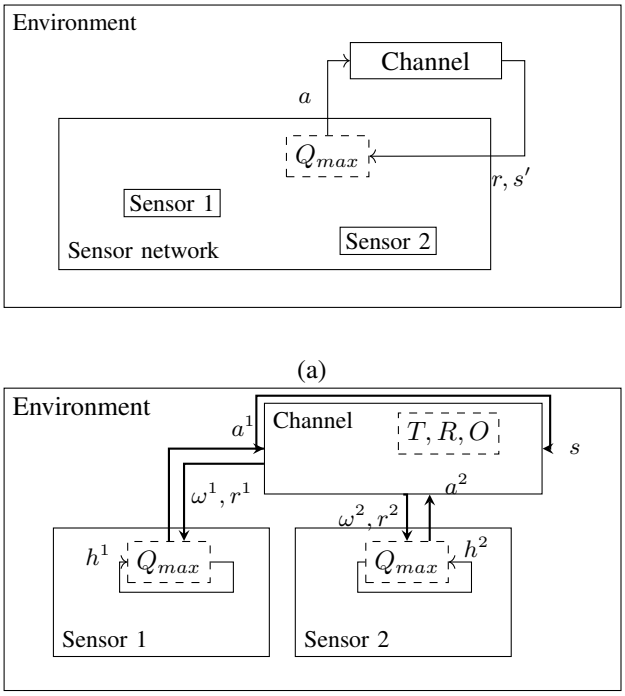


Fig. 2: Flow diagram of a WSN consisting of two sensors that transmit their chosen number of replicas a^1, a^2 to the common channel, which responds with a common reward r when the problem is formulated as an (a) MDP and (b) Dec-POMDP.

This update mechanism expresses the independence in learning, as it is employed by each agent for its individual actions and histories of observations.

E. Virtual experience

Q-learning proves to be inefficient for real-time applications, as extensive interaction with the environment is required in order to converge to $Q^*(s,a)$. To improve the convergence rate of standard Q-learning, we employ virtual experience [25], [30], where an agent updates multiple state-action pairs at each Q-learning iteration by "imagining" state visits.

The intuition behind VE is that an agent can update not only the state-action pairs that it has visited, but also pairs that are equivalent in terms of the unknown environment dynamics. In our case, the unknown environment dynamics include the arrival and collision model, take place after the selection of the number of replicas, and determine the reward the agent experiences, as well as the next observation $\omega \in \Omega$. As defined in (14), an agent's history of observations is a tuple of past buffer states. Based on this information, an agent chooses the preferred number of replicas to send. Although agent's i observation vector h_t^i is essential for determining the optimal action, we should point out that the unknown dynamics do not directly depend on h_t^i. In particular, if the observation tuple is $h_t^i = \{b_{t-w+1}^i, \cdots, b_{t-1}^i, b_t^i\}$, then the unknown dynamics

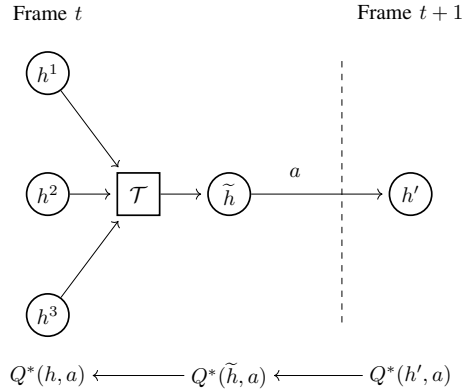


Fig. 3: Virtual experience: h^1, h^2 and h^3 are states of the Dec-POMDP that are mapped to the same virtual state $\widetilde{h}$ through the transformation $\mathcal{T}$. After action a is performed and the next h' is observed, the Q-table is updated for all states.

view states of the following form as equivalent:

$$\begin{aligned} H_t^{i'} &= \{b_{t-w+1}^{i'}, \cdots, b_{t-1}^{i'}, b_t^{i'}\} \\ &= \{b_{t-w+1}^{i'}, \cdots, b_{t-2}^{i'} - \delta b_{t-1}^i, b_{t-1}^{i'} - \delta b_t^i\}, \\ \text{with} \quad \delta b_t^i &= b_{t-1}^i - b_t^i \end{aligned} \tag{18}$$

Note that we have substituted observations with buffer states from the original definition of the observation tuple in (14), as these are equivalent in our current formulation.

The reason for the above formulation is that collisions should intuitively depend on the number of transmitted packets, as they determine the channel congestion. The value of the buffer state is useful in shaping the eagerness of agents to transmit packets. Formally and according to [8], a pair $(\tilde{s}, \tilde{a})$ is equivalent to a pair (s,a) if $p(s'|s,a) = p(s'|\tilde{s},\tilde{a}),\ \forall s' \in S$ and the reward $R(\tilde{s},\tilde{a})$ can be derived from $R(s,a)$. VE can be viewed as applying the following transformation on visited states, and then updating all states that have the same representation:

$$H_t = \{b_{t-w+1}, \cdots, b_{t-2} - \delta b_{t-1}, b_{t-1} - \delta b_t\} \tag{19}$$

$$\xrightarrow{\mathcal{T}} \widetilde{H}_t = \{\delta b_{t-w+2}, \cdots, \delta b_t\} \tag{20}$$

We term $\widetilde{h} \in \widetilde{\mathcal{H}}$ a virtual state, as it is neither visited, nor directly used in the Q-learning update, but serves as an intermediate state in order to acknowledge states equivalent towards the unknown environment dynamics. We illustrate this concept in Fig. 3.

Following the above observation for each move of an agent a batch update on all pairs (s^j, a^j) with $\mathcal{T}(s^j) = \widetilde{h}$ and $a^j = a$ will be performed. Note that we cannot extrapolate experience to states with different actions, as the collision dynamics depend on the performed action.

VII. RL-IRSA: THEORETICAL ANALYSIS

In this section, we theoretically analyze three important properties of the proposed learning algorithm: (i) the quality of the solution (ii) the rate of convergence and (iii) the computational complexity.

A. Optimality analysis

Our analysis begins by studying the waterfall effect and its relation to the optimal performance of IRSA. First, we associate this performance with the cost of a POMDP employed to optimize the degree distribution of IRSA. We then leverage our observations to justify that, a known result about the near-optimality of POMDPs with finite history approximations [7], is applicable in our setting. As this result is for a single agent, and thus corresponds to a centralized approach, we investigate the impact that replacing a centralized POMDP agent with independent learners has on the Q-learning solution, using observations derived in [6] and our own analysis of the equilibrium points of the problem under study.

In [2], the theoretical analysis of SIC in asymptotic settings reveals that the performance of IRSA is governed by the following waterfall effect: there is a sub-space in the space of all valid degree distributions, called a stability region, where SIC can successfully resolve all collisions with a probability close to 1. This leads to near-optimal channel throughput in the stability region. Outside of this region, this probability is bounded away from 0. The degree distribution can be used to calculate an upper bound on the channel load G^*, which signifies the limits of the stability region. The aforementioned observations can be consolidated as:

Observation 1. ([2]) In asymptotic settings ($N \to \infty$), the probability of IRSA having optimal throughput is close to 1 if $G^* \leq \frac{1}{\lambda_2 \Lambda'(1)}$. Otherwise, this probability is bounded away from 0.

Furthermore, we know that the optimal solution of IRSA in our setting corresponds to the case where all sensors successfully transmit one packet in each frame, i.e. $T^* = G$. If we, thus, define the cost as $J^\pi(\beta) = G - T^\pi(\beta)$, where π denotes the learned policy, which corresponds to the optimized degree distribution and, β is the underlying belief state of the finite-history POMDP, we can easily conclude that the optimal cost for IRSA is constant and does not depend on properties of the Dec-POMDP. We thus derive the following observation:

Observation 2. The optimal cost, $J^*(\beta)$, that can be achieved for IRSA is constant and equal to 0.

In [7], the optimality of POMDPs with finite histories of observations was studied and the following theorem was derived:

Theorem 3. (Proposition 2.2 in [7]) If the optimal cost, $J^*(\cdot)$, is a constant function, then, for any $\epsilon > 0$, there exists an integer w, which corresponds to the history window of the POMDP, and an ϵ-optimal policy $\pi^* \in \Pi$ such that π^* at each state depends only on the history of the most recent w stages.

We exploit observations 1 and 2 and Theorem 3 to derive the main result of our optimality analysis:

Theorem 4. When the degree distribution of IRSA is optimized using a POMDP with a finite history window w, then, for any ϵ, there exists an ϵ-optimal solution. Furthermore, this solution lies in the stability region of IRSA and, therefore, throughput is optimal with a probability close to 1.

Proof. The first part of Theorem 4 is a direct result of Theorem 3, which in its turn is valid for the optimization of IRSA due to Observation 2. We will prove the second part of the theorem using a simple argument ad absurdum. Assume that there exists no solution with a probability of optimal performance close to 1. This would mean that all solutions found by the POMDP are bounded away from the optimal solution. But, according to the first part of the theorem, there exists an ϵ-optimal solution for any $\epsilon > 0$, where ϵ is a finite constant that can be arbitrarily close to 0. Thus, we conclude that there exist solutions with a probability of optimal performance close to 1, with closeness to the optimal solution indicated by the parameter ϵ. □

Fig. 4 offers a visual representation of how the solution of the IRSA optimization problem can be mapped into the space of POMDP policies for a simplified scenario where the degree distribution consists of two coefficients. As indicated, the sub-space of valid degree distributions that corresponds to the stability region of IRSA is mapped to a sphere of finite radius ϵ.

Note that, in contrast to finite approximations for POMDPs that require an explicit transition probability model to generate beliefs [31], Theorem 3 can be used in model-free approaches. Therefore, Q-learning can be the method of finding the near-optimal policies, the existence of which is guaranteed by Theorem 4.

Our analysis has so far assumed that the optimization is performed by a single POMDP, which corresponds to a centralized solution. However, as we explained in Section VI-D, to avoid the complexity associated with centralized learning approaches, we allow agents to act independently from each other and learn the Q-values only for their own individual actions. As was observed in [6], the existence of multiple agents renders the environment non-stationary. This suggests that Q-learning is not guaranteed to converge to the optimal solution, as stationarity of the learning environment is one of the traditional requirements for convergence [18]. In order, therefore, for our solution to be optimal, it is first essential to guarantee that all sensors have converged in their attempt to find an optimal policy, as this ensures that the environment becomes stationary.

In [6], sufficient conditions for independent agents to converge were derived and it was observed that a decreasing learning rate and exploitive exploration play an important part. However, guaranteed convergence is not adequate to prove the optimality of independent learning, as the reached equilibrium may correspond to sub-optimal solutions [6].

If we observe the definition of rewards in Section VI-A, we can draw some useful conclusions related to the equilibrium points of our problem. While sensors are learning, and before the optimal transmission strategy has been reached, the rewards are constantly decreasing. This is because, as we assumed in Section IV-A, packets arrive at a steady rate, equal to the optimal rate of transmission. After having learned the optimal policy and acting on it for a few iterations, a sensor reaches an equilibrium point where its rewards are constantly 0, as any packet is successfully transmitted upon arrival. If

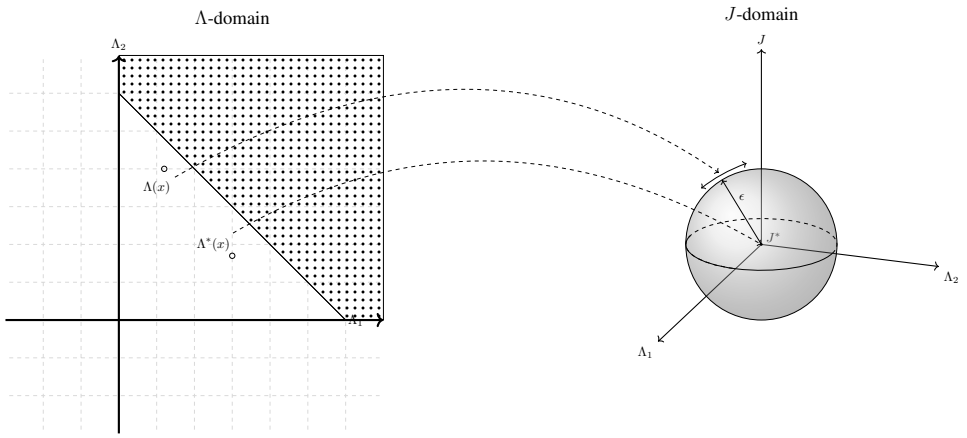


Fig. 4: Visualization of the quality of the solution achieved by a POMDP for a sensor network where d, the maximum number of replicas, equals 2. On the left, the optimal solution and the one our algorithm converges, both of which lie in the stability region of IRSA, are depicted. On the right, we indicate the optimal cost of the POMDP as a point at 0, while the cost of our solution is at most ϵ, and therefore lies in a sphere with a radius equal to ϵ.

learning fails to find the optimal policy, then the sensors quickly converge to a "bad" equilibrium point, where their buffers over-flow and the rewards converge to the negation of the buffer capacity. Due to this structure of the problem, we know that, if sensors converge to the first equilibrium point, then the found solution is optimal. Otherwise, we discard the sub-optimal solution and restart the learning episode, as we explain in our discussion regarding Fig 8 in Section VIII.

B. Rate of convergence analysis

Although convergence of Q-learning is well-understood [18], employing virtual experience, as described in Section VI-E, alters the number of visited states per learning iteration. Under the VE framework the conditions under which Q-learning is guaranteed to converge [18], are still applicable, bu the convergence rate of Q-learning changes. Inspired by the work in [24], where the convergence rate of classical Q-learning is analyzed, we study how VE affects convergence time and derive a lower bound for it. We limit our analysis to asynchronous learning using an exponentially decreasing learning rate $\alpha = 1/t^{\phi}$, where ϕ is a parameter that determines how fast the learning rate converges to zero, as this is the learning scheme implemented in our simulations, and extend the results in [24] by considering multiple updates in each learning iteration.

We first investigate how VE affects coverage time L, i.e., the learning iterations necessary to visit all state action pairs at least once, and then proceed with bounding convergence time. Our remarks are based on Lemma 33 in [24].

Lemma 5. Assume that P is the probability of visiting all state-action pairs in a time interval k, which corresponds to a time period of L iterations. Then, using virtual experience, the probability of visiting all state-action pairs P in an interval k is $|\widetilde{\mathcal{H}}|P$, where $|\widetilde{\mathcal{H}}|$ denotes the size of the virtual state space.

Proof. The probability P is calculated as the percentage of unique state-action pairs visited, i.e., $P = L_u/(|\mathcal{S}||\mathcal{A}|)$, where L_u is the number of iterations where a pair was visited for the first time and the denominator represents the size of the state-action space. We assume that states are sampled with replacement from an i.i.d. probability distribution. As we note in Section VII-C, virtual experience increases the number of states updated in a learning iteration by $|\widetilde{\mathcal{H}}|$. It follows then that $L_u^v = |\widetilde{\mathcal{H}}|L_u$, where L_u^v is the number of iterations where the visited pair was unique using virtual experience. Thus, $P^v = |\widetilde{\mathcal{H}}|P$. □

Lemma 6. Assume that from any initial state we visit all state-action pairs with probability $|\widetilde{\mathcal{H}}|P$ in L steps. Then, from any initial state, we visit all state-action pairs in $L\frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}$ steps for a learning period of length $\left\lceil \frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)} \right\rceil$ with probability $1-\delta$.

Proof. The probability of not visiting all state-action pairs in k consecutive intervals is $(1 - |\widetilde{H}|P)^k$. If we define k as $\log_{1-|\widetilde{\mathcal{H}}|P}(\delta)$, then this probability equals δ and $L\log_{1-|\widetilde{\mathcal{H}}|P}(\delta) = L\frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}$ steps will be necessary to visit all state-action pairs. □

Based on Lemma 6 we can derive the following property of VE:

Corollary 6.1. Virtual experience alters coverage time L by a factor of $\frac{\log_2(1-P)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}$.

Similar to the work in [24], we thus express how the convergence time of virtual experience depends on the covering time in the following theorem:

Theorem 7. Let Q_t be the Q-value computed by the asynchronous Q-learning algorithm using exponentially decreasing learning rate at time t. Then, with probability at least $1-\delta$,

we have $\|Q_t - Q^*\| \leq \epsilon$, given that

$$t = \Omega\Big(\big(L \frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}\big)^{3+1/\phi} + \big(L \frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}\big)^{1/(1-\phi)} \Big)$$

Proof. This theorem is a direct result of Corollary 6.1 of our work and Theorem 4 in [24], where the corresponding bound for classical Q-learning is found to be $\Omega(L^{3+1/\phi} + L^{1/(1-\phi)})$. □

C. Computational complexity

The proposed protocol is a computationally attractive alternative to transmission strategies that are based on finite length analysis [29]. In our framework, at each learning iteration an agent has to choose its transmission strategy and then update its local Q-table. In contrast to the work in [12], in the proposed scheme the action space is discrete and increases linearly with d, i.e., the maximum number of allowed replicas. The size of the observation space, which coincides with the size of the Q-table, is $(B+1)^w$. Recall that B is the capacity of sensors' buffer and w is the history window. The observation space scales exponentially with w and linearly with B. Finally, the complexity associated with the number of sensors is $\mathcal{O}(1)$, as sensors learn independently of each other.

Equipping Q-learning with virtual experience increases computational complexity, as instead of updating one entry of the Q-table in each learning iteration, all (h, a) pairs with the same virtual state $\widetilde{h}$ are updated. This complexity increase is equal to the number of those pairs, which we denote by $|\widetilde{H}|$ and can be bounded as:

$$0 \leq |\widetilde{\mathcal{H}}| \leq \min\{B + 1 - B_{\min}, B_{\max}\} \quad \text{where} \tag{21}$$

$$B_{\min} = \arg\min_b \{b - \sum_{t=w-1}^{\tau} \delta b_t \geq 0\} \quad \text{and} \tag{22}$$

$$B_{\max} = \arg\max_b \{b - \sum_{t=w-1}^{\tau} \delta b_t \leq B\} \qquad \forall \tau \in [0, w-2) \tag{23}$$

where $B_{\min}$ and $B_{\max}$ are used to avoid considering virtual states with numbers of packets in their buffers that are either negative or exceed the maximum capacity B.

VIII. Simulations

This section begins with a performance comparison of the proposed RL-IRSA protocol and classical IRSA, which does not employ learning and is optimized in [2] using differential evolution. It subsequently studies the effect of different learning schemes on the performance of independent learning with the two-fold goal of drawing conclusions about the behavior of agents and providing a guideline for configuring system parameters to determine the optimal transmission strategy. Finally, we evaluate the proposed scheme advanced with virtual experience to show the reduced convergence time. Unless stated otherwise, the simulation parameters are as indicated in Table III. Note that a simulation includes learning of a degree distribution for L_E number of learning iterations and its evaluation for L_T transmission trials.

TABLE III: Simulation setup

Simulation parameters	value
N, frame size	10
G, channel load	$[0.1, 0.2, \cdots, 1.0]$
L_T number of transmission trials	1000
L_E, number of learning iterations	1500
N_E, number of independent simulations	20
c_{level}, confidence level	97.5%
ϵ, exploration	0.05
α, learning rate	$\frac{1}{(1+i)^{0.9}}$
γ, discount factor	0.98

TABLE IV: Degree distributions of classical IRSA and our proposed solution for different frame sizes

Method	$\Lambda(x)$
IRSA	$0.25x^2 + 0.60x^3 + 0.15x^8$
RL-IRSA$_{10}$	$0.4354x^2 + 0.2445x^3 + 0.173x^4 + 0.0855x^5 + 0.0437x^6 + 0.001x^7 + 0.008x^8$
RL-IRSA$_{50}$	$0.0556x^1 + 0.0278x^2 + 0.2732x^3 + 0.1654x^4 + 0.1027x^5 + 0.1178x^6 + 0.0878x^7 + 0.0902x^8$
RL-IRSA$_{200}$	$0.0402x^1 + 0.0754x^2 + 0.3036x^3 + 0.1607x^4 + 0.1131x^5 + 0.1548x^6 + 0.0952x^7 + 0.0476x^8$

A. Protocol Comparison

In Fig. 5, a statistical analysis on the performance of the two protocols under consideration is performed. From this figure, it is obvious that RL-IRSA is superior to classical IRSA for the whole channel load range, with the difference gap becoming wider for channel loads above 0.6. We also observe that performance has higher variations in high channel loads. Fig. 6 illustrates convergence time for independent learning in different channel loads. From this figure, we can see that convergence is guaranteed and is fast for low channel loads. For $G = 0.2$ only four learning iterations are necessary, while for $G = 0.4$ seven iterations are needed. In the case of high channel loads RL-IRSA fails to transmit messages faster than their arrival rate, the sensors' buffers thus quickly overflow and the values of the rewards saturate to $-B$. For maximum channel load ($G = 1$), the saturation occurs very soon at the course of the episode (7 iterations), while for $G = 0.8$ 17 learning iterations take place before all buffers overflow and, even after this time point, some sensors manage to empty their buffers. The highest channel load for which convergence is achieved is $G = 0.6$, which is expected if we consider the fact that packets are arriving randomly with probability 0.5 at each iteration. Thus, a throughput above 0.5 is required on average to avoid saturation. Based on these observations, we design a mechanism for agents to avoid sub-optimal solutions: if the rewards deteriorate for three consecutive iterations, we classify the episode as "*bad*", discard the learned policy, and reset the POMDP to an arbitrary state.

To draw further insights into how our solution differs from an asymptotic optimization, we present in Table IV the IRSA degree distributions, as optimized in [2] and our learned degree distributions for different frame sizes. In contrast to the work in [2], where the majority of the coefficients was a priori set to 0 to reduce the complexity of optimization, our learned

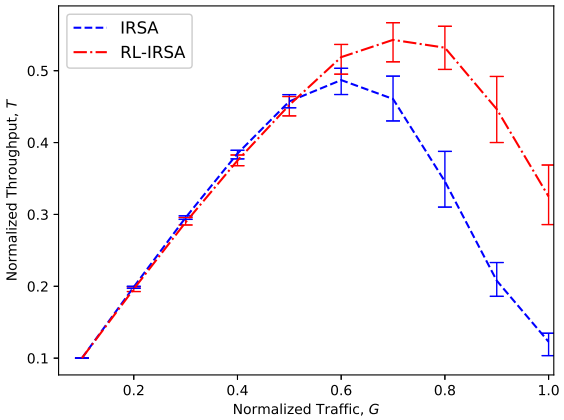


Fig. 5: Achieved throughput comparison of IRSA and RL-IRSA on a toy network for varying channel loads.

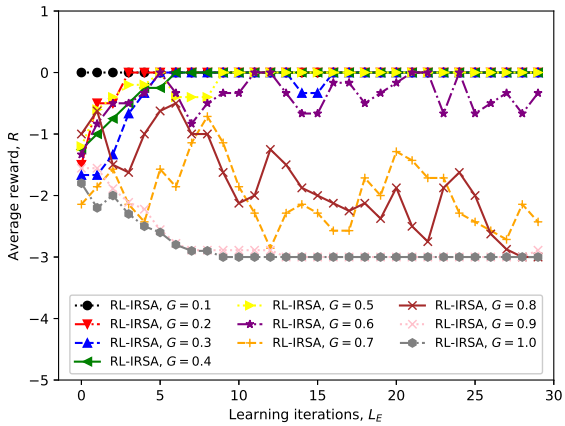


Fig. 6: Average rewards of RL-IRSA IRSA for different channel loads G and the 25^{th} episode of its learning time in a run of 50 episodes.

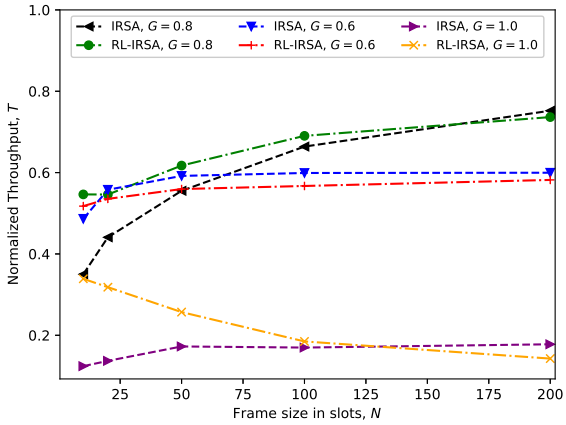


Fig. 7: Achieved throughput comparison of IRSA and Dec-IRSA for varying frame sizes N and number of sensors M
with channel traffic $G \in \{0.6, 0.8, 1\}$.

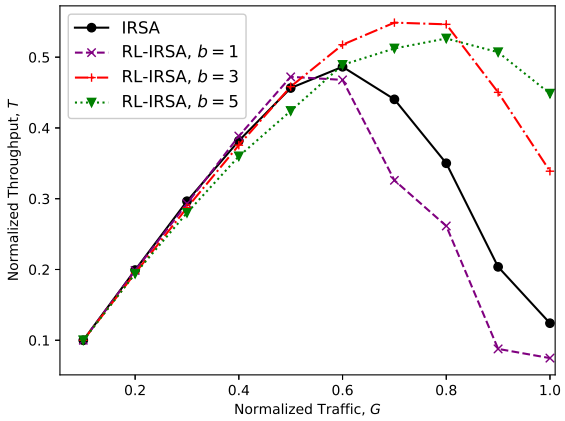


Fig. 8: Comparison of achieved throughput for different buffer sizes of sensors.

$\Lambda(x)$ are more dense. We also observe that, as the frame size N increases, higher degrees become more prevalent, while degrees are left-concentrated for small frame sizes. To explain this tendency, we can borrowing theoretical and empirical insights from the design of random error-correction channel codes [32]. This is due to the similarity between SIC and decoding on graphs, first observed in [2]. In particular, the analysis in [32] proves that higher degrees lead to better probability of successful decoding, while observing that the simulated performance of the codes in finite settings deviates from the asymptotic analysis, with the gap becoming more evident as the maximum degree increases. Note that the maximum simulated frame size in IRSA does not usually exceed 200, while simulations in [32] were done for codewords of length orders of magnitude larger. It is thus natural to restrict the maximum degree to 8 in our simulations.

Fig. 7 illustrates how the throughput achieved by RL-IRSA and classical IRSA changes with frame size and $G \in \{0.6, 0.8, 1\}$. Note that these results can be easily translated into throughput variation with respect to the number of sensors, as, for a constant channel load, the number of sensors increases with the frame size according to the formula $M = G \cdot N$. Thus, the results in Fig. 7 correspond to $M \in \{6, 12, 30, 60, 120\}$ sensors ($G = 0.6$), $M \in \{8, 16, 40, 80, 160\}$ sensors ($G = 0.8$) and $M \in \{10, 20, 50, 100, 200\}$ sensors ($G = 1$). As regards scalability of RL-IRSA, it appears robust and its performance increases with larger frame sizes. This can be attributed to the fact that learning is more meaningful in more complex networks, where collisions occur more often, thus, learning to avoid other agents has a more profound impact on the overall throughput. Classical IRSA also improves its performance for increased frame sizes, as it provingly works better in asymptotic settings. This is attributed to the fact that the probability distribution $\Lambda(x)$ is computed using asymptotic analysis and is therefore closer to optimal for frames whose size exceeds 200 slots. Nevertheless, the performance gap between these two methods remains high in heavy channel loads ($G = 1$), due to the waterfall effect of classical IRSA. To conclude scalability

analysis, the slight superiority of standard IRSA manifested for low G in asymptotic settings is irrelevant to practical scenarios, as the assumption of very large frame size N leads to inefficient implementations, as was discussed in Section I.

B. Effect of state space size

The size of the state space depends on the history window, as well as the maximum value of observations, which equals $B+1$. Increasing B has a two-fold effect. Firstly, it increases the size of the state space, thus urging for longer exploration. Secondly, it dilates the range of rewards, which makes agents more eager to transmit. Assuming buffer sizes of constant size, constrained by sensors' characteristics, one anticipates to improve performance of learning by increasing the history window, as that will lead to better approximation of the underlying beliefs. Nevertheless, letting memory constraints aside, this will result in an exponential increase of the state-action space, leading either to intractable problems or high time complexity. Thus, it is crucial to determine the minimum amount of information necessary for agents to derive efficient policies. Note that, for the sake of a fair comparison, learning iterations were also increased to 3000 for increased history window and buffer size. Fig. 8 demonstrates that using a value of $B=1$, i.e., only one packet is kept in the buffer, leads to lower throughput for channel loads above 0.6, as agents are not made eager enough to transmit. On the other hand, increased buffer size improves the perceived throughput for loads above 0.8, but it slightly degrades it for the rest.

Regarding the history window w, Fig. 9 reveals that the effect of increased world size is more profound. This is due to size scaling exponentially with w, in contrast to its linear scaling with B. We observe that by decreasing the window to $w=2$, a severe degradation in performance is observed, suggesting that the information provided to the agents through the observation tuples is not substantial. Increasing the learning iterations for $w=8$ has a counterintuitive effect, as performance is degraded, whereas we would expect that an increased world size would benefit from larger training times. This expectation is based on the fact that increasing the dimensionality of the state space will require more state-action pairs to be visited, and, thus, a higher number of learning iterations to explore the state-action space and find a well performing solution will be required, a problem often termed as curse of dimensionality. In this case however, 800 learning iterations perform optimally for $w=4$, so we can assume that equipping agents with larger memory leads to learning better policies, thus less iterations are required. If we continue to train after this optimal point, performance degrades due to over-training, a phenomenon that generally refers to learning a behavior that performs well during training, but not during the evaluation of the policy. In particular, after a good policy has been learned, the distribution of visited states is no longer representative of the original problem. We conclude that, considering the current parameterization, $w=4$ is the best performing choice. Note that the optimal value for w is dictated by the learning dynamics (α, γ and ϵ) and is the one that defines a state-space of size appropriate for a satisfactory exploration/exploitation balance.

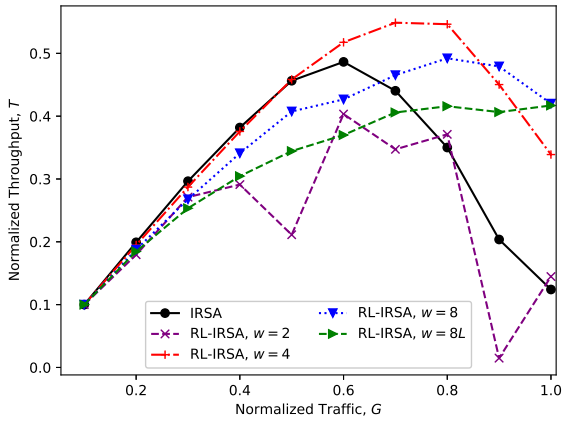


Fig. 9: Comparison of achieved throughput for different history windows.

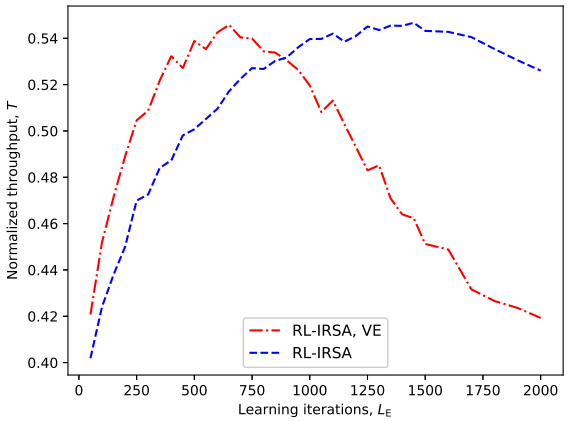


Fig. 10: Comparison of throughput of RL-IRSA with and without VE for different number of learning iterations and $G=0.7$.

C. Virtual experience

Virtual experience was introduced in our solution to reduce convergence time, which we, similarly to the work in [25], measure here using the ϵ-convergence time, i.e., the number of iterations required to learn an ϵ-optimal policy. Fig. 10 shows how throughput changes with learning iterations and reveals that, using VE, the optimal number of iterations was reduced from 1500 to 500. Note that the degradation in performance with increasing learning iterations, manifested at around 1500 iterations for RL-IRSA and 500 using VE, is due to over-training. Fig. 10 is also useful for understanding the complexity of our learning solution. Although finding the optimal policy requires around 700 iterations (using VE), satisfactory performance is achieved very early during the course of the episode. From Fig. 10, it is clear that even at 100 iterations, the algorithm is not very far from the optimum, while performance improves rapidly. Therefore, we can conclude that this solution will give good results even if the problem changes often, i.e. every 100-200 iterations. To gain some intuition on how simulation time maps to real time, we note that an iteration

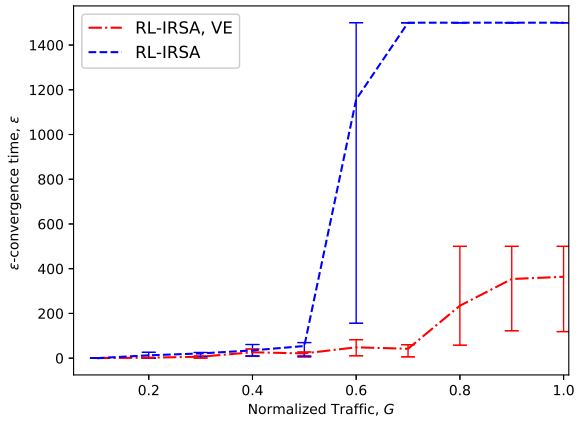


Fig. 11: Statistical comparison of ϵ-convergence times for simple RL-IRSA and RL-IRSA using virtual experience, with $\epsilon = 0.5$.

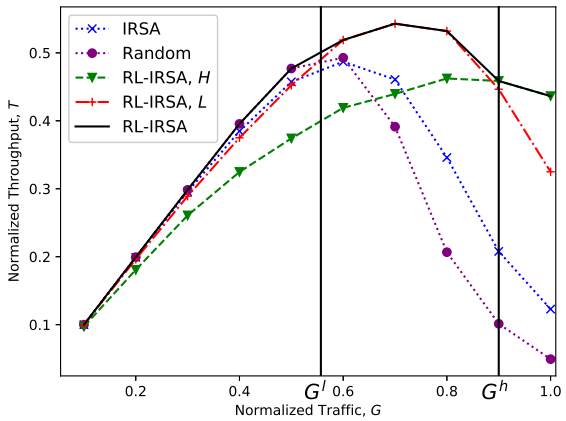


Fig. 12: Performance comparison of classical IRSA, a random strategy, RL-IRSA optimized for low G and RL-IRSA optimized for high values of G.

in our simulations required 3 milliseconds using an Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz processor. As the code was not optimized in terms of time complexity, we expect even smaller time requirements from software solutions specialized for a specific hardware architecture.

Fig. 11 performs a statistical analysis on ϵ-convergence time for different channel loads using a 95% confidence level on 40 independent simulations and $\epsilon = 0.5$. We observe that convergence is fast for low loads regardless of the use of VE. For $G \geq 0.6$, however, we observe that VE exhibits an improvement of around 80%, which can be attributed to increasing the number of batch updates by a factor of $|\widetilde{\mathcal{H}}|$. Also, RL-IRSA without VE usually fails to converge for high channel loads (i.e. convergence has not been achieved within the budget of 15000 iterations), although throughput remains close to optimal. This observation suggests that, in this case, there are different policies that lead to optimal behavior, so RL-IRSA without VE is less biased to the optimal one. From Fig. 11 we can also observe that the confidence interval observed for RL-IRSA without VE for channel load $G = 0.6$ is large. This load value is a transitional case, as for some simulations Q-learning converged, similarly to $G < 0.6$, while for others convergence was not achieved by the end of the episode, similarly to $G > 0.6$. This is because, as we can see in Fig. 5, the learned transmission strategies up to channel load $G = 0.5$ are optimal, i.e. $T = G$, while $T < G$ for $G >= 0.6$, which suggests that a unique optimal policy has not been found.

D. Waterfall effect

As explained in Section VII-A, the performance of IRSA has been proven to be governed by a stability condition [2], which leads to a waterfall effect in the performance of SIC. From a learning perspective, this profoundly changes the nature of the problem and, thus, the learning objective. As described in Section VI-C, the problem is one of agents competing for a pool of common resources. In the realm of low channel loads ($G < G^*$), where resources are abundant, agents must learn to coordinate their actions, as there is a number of replicas to transmit that optimizes packet throughput. Observe that, for low channel loads ($G \leq 0.5$), even a random strategy, implemented by sampling the number of replicas l uniformly from $\{1, \cdots, d\}$, is appropriate, as illustrated in Fig. 12, so learning is of no practical interest. In the realm of high channel loads ($G \geq G^*$), however, we can acknowledge the task as a Dispersion game [33], where agents need to cooperate in order to avoid congesting the channel by exploiting it in different time frames. Different problem nature urges for different learning behavior, thus we expect that parameterization of learning should vary with G. Fig. 12 illustrates the performance of three different parameterizations, each one optimal for a different range of values for G. Note that G^* and G^{low} stand for the threshold load value below which the probability of unsuccessful transmission is negligible and a random strategy is optimal, respectively. We observe that by optimizing the parameters for a particular range of G values, we obtain significant gains in the region of interest ($G > 0.6$).

IX. Conclusions

We have examined the problem of IRSA design for finite frame sizes from a reinforcement learning perspective and proved that learning degree distributions can be beneficial even under the assumption of sensors' independence in learning. The theoretical analysis proves that our learning algorithm finds a near-optimal solution for the problem of IRSA design. Our simulations suggest that the *waterfall effect* of the problem, common in optimization problems where agents compete for common resources, leads to different learning dynamics, and thus, demands adaptive solutions. Our method's superiority is particularly manifested in high channel loads and small frame sizes, where the degree distributions of classical IRSA are inappropriate. Our simulations indicate that making learning tractable for online application scenarios requires achieving fast convergence. We observed that even when maintaining a small observation space, by restricting the history window to 2, the performance remains satisfactory. Finally, the

results show that we significantly reduced convergence time by introducing virtual experience into Q-learning. We believe that, in addition, employing transfer learning techniques to quickly adapt to changing conditions within an episode is essential for addressing the non-stationarity inherent in WSNs.

REFERENCES

[1] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *arXiv:1809.08707 [cs]*, Sept. 2018. [Online]. Available: http://arxiv.org/abs/1809.08707

[2] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted aloha," *IEEE Trans. on Communications*, vol. 59, no. 2, pp. 477–487, Feb. 2011.

[3] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The Complexity of Decentralized Control of Markov Decision Processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, Nov. 2002.

[4] E. Nisioti and N. Thomos, "Decentralized reinforcement learning based mac optimization," in *Proc. of IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications, PIMRC'18*, Bologna, Italy, Sep. 2018.

[5] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[6] C. Claus and G. Boutilier, "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems," in *Proc. of the 15th National/10th Int. Conf. on Artificial Intelligence/Innovative Applications of Artificial Intelligence, AAAI '98/IAAI '98*, Madison, WI, USA, Jul. 1998.

[7] H. Yu and D. P. Bertsekas, "On near optimality of the set of finite-state controllers for average cost POMDP," *Mathematics of Operations Research*, vol. 33, no. 1, pp. 1–11, Feb. 2008.

[8] N. H. Mastronarde, "Online learning for energy-efficient multimedia systems," Ph.D. dissertation, University of California, 2011.

[9] N. M. Abramson, "THE ALOHA SYSTEM: Another Alternative for Computer Communications," in *Proc. of joint Computing Conf. AFIPS'70*, Honolulu, HI, USA, Nov. 1970.

[10] G. L. Choudhury and S. S. Rappaport, "Diversity ALOHA - A Random Access Scheme for Satellite Communications," *IEEE Trans. on Communications*, vol. 31, no. 3, pp. 450–457, Mar. 1983.

[11] E. Casini, R. D. Gaudenzi, and O. D. R. Herrero, "Contention resolution diversity slotted aloha (crdsa): An enhanced random access schemefor satellite access packet networks," *IEEE Trans. on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.

[12] L. Toni and P. Frossard, "IRSA Transmission Optimization via Online Learning," *arXiv:1801.09060 [cs, math, IT]*, 2018. [Online]. Available: http://arxiv.org/abs/1801.09060

[13] ——, "Prioritized random mac optimization via graph-based analysis," *IEEE Trans. on Communications*, vol. 63, no. 12, pp. 5002–5013, Dec. 2015.

[14] P. Nicopolitidis, G. Papadimitriou, A. Pomportsis, P. Sarigiannidis, and M. Obaidat, "Adaptive wireless networks using learning automata," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 75–81, Apr. 2011.

[15] I. Kakalou, G. Papadimitriou, P. Nicopolitidis, P. Sarigiannidis, and M. Obaidat, "A reinforcement learning-based cognitive MAC protocol," in *Proc. of IEEE Int. Conf. on Communications, ICC'15*, London, UK, Jun. 2015, pp. 5608–5613.

[16] M. Thathachar and P. Sastry, "Varieties of learning automata: an overview," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 711–722, Dec. 2002.

[17] Y.-M. D. Hauwere, P. Vrancx, and A. Now, "Generalized learning automata for multi-agent reinforcement learning," *IOS Press Journal of AI Communications*, vol. 23, no. 4, pp. 311–324, Apr. 2010.

[18] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[19] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1-2, pp. 99–134, May 1998.

[20] C. Zhang and V. Lesser, "Coordinated multi-agent reinforcement learning in networked distributed pomdps," in *Proc. of the 25th Conf. on Artificial Intelligence, AAAI'11*, San Francisco, CA, USA, Aug. 2011.

[21] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *arXiv:1704.02613 [cs]*, April 2017. [Online]. Available: http://arxiv.org/abs/1704.02613

[22] U. Challita, L. Dong, and W. Saad, "Proactive resource management in LTE-u systems: A deep learning perspective," *arXiv:1702.07031 [cs, math]*, Feb. 2017. [Online]. Available: http://arxiv.org/abs/1702.07031

[23] K. J. Arrow, "Rationality of Self and Others in an Economic System," *The Journal of Business*, vol. 59, no. 4, pp. 385–399, Oct. 1986.

[24] E. Even-Dar and Y. Mansour, "Learning rates for q-learning," *J. Mach. Learn. Res.*, vol. 5, pp. 1–25, Dec. 2004.

[25] N. Mastronarde and M. van der Schaar, "Fast Reinforcement Learning for Energy-Efficient Wireless Communication," *IEEE Trans. on Signal Processing*, vol. 59, no. 12, pp. 6262–6266, Dec. 2011.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," in *Proc. of NIPS Deep Learning Workshop, NIPS'13*, Lake Tahoe, CA, USA, Dec. 2013.

[27] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *arXiv:1802.06958 [cs]*, Aug. 2018. [Online]. Available: http://arxiv.org/abs/1802.06958

[28] E. Paolini, G. Liva, and M. Chiani, "Coded slotted aloha: A graph-based method for uncoordinated multiple access," *IEEE Trans. on Information Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.

[29] E. Paolini, "Finite length analysis of irregular repetition slotted aloha (irsa) access protocols," in *Proc. of IEEE Int. Conf. on Communication Workshop, ICCW'15*, London, UK, Jun. 2015.

[30] N. Thomos, E. Kurdoglu, P. Frossard, and M. van der Schaar, "Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers," *IEEE Trans. on Multimedia*, vol. 17, no. 6, pp. 893–906, June 2015.

[31] H. Yu and D. P. Bertsekas, "Discretized approximations for pomdp with average cost," in *Proc. of the 20th Conference on Uncertainty in Artificial Intelligence*, Arlington, VI, USA, Jul. 2004.

[32] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb 2001.

[33] T. Grenager, R. Powers, and Y. Shoham, "Dispersion Games: General Definitions and Some Specific Learning Results," in *Proc. of 18th National/14th Conf. on Artificial Intelligence/Innovative Applications of Artificial Intelligence, AAAI '02*, Edmonton, AL, Canada, Jul. 2002.

Eleni Nisioti is a PhD candidate at the University of Essex, UK at the Department of Computer Science and Electronics Engineering. She is part of the Communications and Networks group and her research interests revolve around the application of machine learning for communications with a focus on multi-agent reinforcement learning for wireless sensor networks. She received the Diploma in Electrical and Computer Engineering degree from Aristotle University of Thessaloniki in 2017.

Nikolaos Thomos (S'02-M'06-SM'16) is an Associate Professor at the University of Essex, UK and the group leader of the Communications and Networks group. Previously, he was senior researcher at the Ecole Polytechnique Fédérale de Lausanne (EPFL), and the University of Bern, Switzerland. He received the highly esteemed Ambizione career award from Swiss National Science Foundation (SNSF). He received the Diploma and Ph.D. degrees from Aristotle University of Thessaloniki, Greece in 2000 and 2005 respectively. He is an elected member of IEEE MMSP Technical Committee (MMSP - TC) for the period 2019 - 2022. His research interests include machine learning for communications, multimedia communications, network coding, information-centric networking, source and channel coding, device-to-device communication, and signal processing.