Decentralized Reinforcement Learning Based MAC Optimization

Eleni Nisioti and Nikolaos Thomos
School of Computer Science and Electronic Engineering, University of Essex, UK
Email:{e.nisioti, nthomos}@essex.ac.uk

***Abstract*—In this paper, we propose a novel decentralized framework for optimizing the transmission strategy of Irregular Repetition Slotted ALOHA (IRSA) protocol in sensor networks. The proposed method is inspired by reinforcement learning algorithms. To deal with sensor nodes limited lifetime and communication range, we allow sensor nodes to decide how many packet replicas to transmit and when to transmit them considering only their own buffer state. We show that this information is sufficient and can help avoiding packets' collisions and improving the throughput significantly. We solve the problem using the decentralized partially observable Markov Decision Process (POMDP) framework, where we allow each node to decide independently of the others how many packet replicas to transmit and when. The performance of the proposed method is compared with the native IRSA protocol. The results make clear that our method leads to large throughput gains without imposing nodes coordination, in particular, when network traffic is heavy.**

I. Introduction

The Internet of Things (IoT) is gradually assimilating into everyday life. IoT networks consist of devices with constrained capabilities, such as limited battery lifetime and communication range. Efficient communication strategies that target energy efficiency and adaptivity to altering network conditions have been studied in the context of wireless sensor networks (WSNs), which share common aspects with IoT networks. For example, medium access control (MAC) has been widely used to deal with the case where multiple sensor nodes attempt to transmit packets over a shared bandwidth constrained communication channel. MAC protocol design can be studied as a distributed resource allocation problem. In applications where energy consumption is important, contention-based approaches are preferred, as they offer adaptivity by allowing sensor nodes to randomly decide when to transmit their packets. However, these approaches often lead to packet collisions and hence need for packets' retransmissions, so if performed naïvely can lead to channel congestion.

Slotted ALOHA [1], which belongs to the family of contention-based protocols, offers a simple, uncoordinated random access mechanism, but is inefficient when networks are large, and the channel traffic is heavy. Diversity Slotted ALOHA (DSA) [2] significantly improves upon it by introducing a burst repetition rate, so that network nodes transmit a pre-defined number of replicas of the original messages. The burst repetition rate enables Contention Resolution Slotted ALOHA (CRDSA) [3] to exploit successive interference cancellation (SIC) for the retrieval of collided packets.

IRSA was introduced in [4] and allows for a variable number of replicas for each user. This number is decided by randomly sampling from a probability distribution, which is designed such as to decrease packet loss. The employment of IRSA can lead to improved network throughput. IRSA performance depends on the optimization scheme used to derive the sampling probability distribution. The work in [4] relates SIC to the process of iterative belief-propagation (BP) erasure-decoding of codes-on-graphs.

Traditionally, differential evolution, which is an efficient, global optimization scheme, has been used for asymptotically analyzing the transmission strategies [4], [5]. Recently, the use of Multi-armed Bandits (MAB) has been proposed in [6] as a remedy for inaccurate asymptotic analysis in non-asymptotic settings as well as an alternative to computationally expensive finite length block analysis. The work in [6] is based on a variant of IRSA algorithm originally proposed in [5] that incorporates users' prioritization. However, the use of MABs implies a continuous action space. To address this intractability, the space was discretized, which has been proven to significantly degrade the performance [7]. Another disadvantage of MABs is that their framework is not expressive enough as they are stateless. This renders MABs inappropriate for sensor networks, where operations are constrained by sensor nodes' characteristics, such as battery level, memory size, *etc.*, valuable information that MABs fail to incorporate in the decision-making process.

In this paper, we investigate the optimization of the transmission strategy in sensor networks following the Markov Decision Process (MDP) [8] framework. In our framework, sensor nodes are capable of independently and distributively learning the optimal number of replicas to transmit in an IRSA protocol. Guided by the nature of the problem under consideration, we design a distributed, model-free, offline learning algorithm that deals with partial observability. Partial observability refers to sensor nodes' inability to observe information

that requires global access to the network. Therefore, nodes should make decisions based on information only locally available to them, for example, as we assume here, nodes are only aware of the state of their buffer. To solve the optimization problem, we express it as a decentralized POMDP, which is known to be NEXP-complete [9]. Hence, to render learning feasible, we employ independent learning. Our solution leverages techniques from the multi-agent reinforcement learning literature to design agents that optimally manage the available time slots by learning how to behave optimally in terms of packet throughput. The results show that the proposed method can lead to significantly improved throughput compared to IRSA without imposing nodes' coordination, in particular, when the network traffic is heavy.

II. IRSA

Let us consider a network of M sensor nodes collecting measurements from their environment and transmitting their data to the core network. Hereafter, we will use the terms sensor node, node, sensor and user interchangeably when referring to devices in sensor networks. A bottleneck in the operation of such a network is the transmission of packets through a common communication medium, used also by neighboring sensor nodes to transmit their packets. Similar to the original Slotted ALOHA framework and its variants, time is divided in frames of fixed duration, each one consisting of N time slots. At the beginning of each frame, each sensor attempts transmission by randomly choosing one of the N slots to transmit a packet. Since a contention-based approach is followed, collisions may occur due to the fact that multiple sensors may choose to transmit simultaneously in the same time slot.

In vanilla Slotted ALOHA protocol, a transmission is successful only if no other node chooses to transmit in the same slot. The resulting throughput is a function of the normalized load $T(G) = Ge^{-G}$, where $G = M/N$. In an IRSA protocol, however, a user can transmit a variable number of replicas of a packet, which leads to increased throughput due to SIC [3]. The behavior of users is governed by the degree distribution $\Lambda(x)$, a polynomial probability distribution describing the probability Λ_l that each user transmits l replicas of its packet at a particular time frame. This probability distribution is given by

$$\Lambda(x) \triangleq \sum_{l=1}^{l=d} \Lambda_l x^l, \tag{1}$$

where d is the maximum number of replicas a sensor node is allowed to transmit.

The objective of a MAC optimization algorithm is to select the values of the coefficients Λ_l in (1) so that channel throughput is maximized. In Section IV, we introduce our protocol that learns the optimal Λ_l values in non-asymptotic settings, *i.e.*, when N is finite.

III. Preliminaries

The proposed framework is transparent to the underlying modulation and coding schemes. Similar to the work of [10], the packet throughput and the number of transmitted packets can be expressed, respectively, as

$$T^t = T(G^{t-1}, K^{t-1}), \tag{2}$$

$$K^t = K(G^t, T^t, C^t), \tag{3}$$

where T^t is the packet throughput, representing the number of successfully transmitted packets, K^t represents the packets transmitted by all nodes during the particular frame, and C^t is the node's condition at time slot t. A node's condition can, in general, depend on its buffer state, its battery level and any information that potentially affects its behavior. Here, we consider only the buffer state of the network nodes, as incorporating more variables in C^t will increase computational requirements. This does not affect the generality of our framework which can easily incorporate further node-related information. For the sake of simplicity, we also assume that transmission is not affected by noise in the channel, thus successful transmission is guaranteed upon successful SIC. The consideration of noise is straightforward in our framework. It is also worth noting that packet throughput is a non-deterministic function of its arguments due to the random selection of slots.

We assume that the transmission buffer is a first-in first-out queue. The sensor nodes produce f^t packets in each time frame. Each packet is of size D bits and the arrival process $\{f^t : t = 0, 1, \cdots\}$ is assumed to be independent and identically distributed (i.i.d) with respect to the time slot t. The arriving packets are stored in a finite-length buffer, with a maximum capacity of B packets. The buffer state of each sensor node $b_i \in \{0, 1, \cdots, B\}$ during simulations evolves as follows:

$$\begin{aligned} b_i^0 &= b_i^{init}, \\ b_i^{t+1} &= \min\{b_i^t - T_i^t(K^t, G^t) + f^t, B\}, \end{aligned} \tag{4}$$

where b_i^{init} is the initial buffer state, which takes a value in the range $\{0, 1, \cdots, B\}$, and $T_i^t(K^t, G^t)$ is the number of successfully transmitted packets in a frame for sensor node i. Unsuccessfully transmitted packets stay in the transmission buffer for later retransmission.

IV. Proposed DEC-POMDP based IRSA design

In this section, we first express IRSA using MDP, and then we describe how to deal with partial observability and decentralization. As suggested by (2) and (3), there are two factors affecting the state of the environment, namely: the channel load G^t and the node's condition C^t at time t. We can view the sensor network as an agent that interacts with its environment. Hence, the

environment includes the sensor nodes that comprise the network and the shared communication channel. Prior to proceeding with the proposed decentralized design of our protocol, we define the state of the MDP as

$$S = \times_{1 \leq i \leq M} S_i \times S_u, \tag{5}$$

where S is the state of the agent, S_i is the state of a sensor node and S_u stands for the part of the environment that is not controlled by the sensor nodes and corresponds to G^t.

The transition probabilities $P(s, s')$ from state s to state s' of the MDP can be formulated as

$$P(S'_u \mid S_u, \times_{1 \leq i \leq M} S_i, K) = P(S'_u \mid S_u), \tag{6}$$

$$P(S'_i \mid S_u, \times_{1 \leq i \leq M} S_i, K, T) \propto T_i(G, K) + f_i, \tag{7}$$

From (6), we can see that the transitions of the non controllable state S_u are independent of the transmission strategy and the states of sensor nodes. We assume that the reward function $R(s, a) = \times_{1 \leq i \leq M} R_i(s_i, a)$ also follows $R(S_i \mid S_u, \times_{1 \leq i \leq M} S_i, K, T, a) \propto T_i(G, K)$.

Solving the formulated MDP requires finding the optimal transmission policy π, a function that specifies the action $\pi(s)$ that the agent will choose when in state $s \in S$. The goal of the agent is to maximize the expected discounted reward, which is represented as

$$V^{\pi(s)} = E_\pi \Big\{ \sum_{t=0}^{\infty} \gamma^t R^t(s^t, \pi(s^t)) | s_0 = s \Big\}, \tag{8}$$

where $0 \leq \gamma < 1$ is the discount factor that controls whether an agent is myopic or farsighted (the closer γ is to zero, the more myopic an agent is). $V^{\pi(s)}$ denotes the expected return when starting in state s and following policy π thereafter. Finally, s_0 corresponds to the state of the agent at the beginning of an episode.

A drawback of the MDP is that in many practical scenarios, as in our case, the transition probability $P(s, s')$ and the reward function $R(s, a)$ are unknown, which makes evaluation of policy π using (8) impossible. To this aim, we adopt Q-learning [11], which is one of the most effective and popular algorithms for learning from delayed rewards that enables us to determine the optimal policy, in absence of the transition probability and reward function. In Q-learning, both the policy and the value function are represented by a two-dimensional lookup table indexed by state-action pairs (s, a), which are recursively updated using

$$\begin{aligned} Q(s^t, a^t) \leftarrow (1 - \alpha) Q(s^t, a^t) + \\ + \alpha \Big(R^t + \gamma \max_a Q(s^{t+1}, a) \Big) \end{aligned} \tag{9}$$

where $Q(s, a)$ is the expected discounted reward when we start from s, take action a, and thereafter follow policy π. The parameter α is the learning rate, determining to what extent newly acquired experience overrides the past. Equation (9) is guaranteed to converge to the optimal solution under the Robbins-Monro conditions [8]:

$$\sum_t \alpha^t = \infty \quad \text{and} \quad \sum_t (\alpha^t)^2 < \infty \tag{10}$$

As noted earlier, a state consists of all the information necessary for the network to choose the optimal action $a \in A$, where $A = \times_{1 \leq i \leq M} A_i$ is the overall action space and A_i is the independent action space of sensor node i. This necessity urged us to define a state s as the joint combination of the nodes' condition (*e.g.*, battery level, buffer size, number of packets to transmit, *etc.*) and the network load. Clearly, this information cannot be available as it would impose huge communication load, while a MAC protocol should prevent channel congestion and be unintrusive. To alleviate this drawback of MDPs and Q-learning, we base our framework on partially-observable MDPs (POMDPs).

POMDPs [12] acknowledge the inability of an MDP to observe the overall state, which they remedy by introducing the notion of observations. Observations contain information that is relevant but insufficient to describe the actual state on their own. In our case, the network and the sensor nodes cannot observe $S_u = G$, as this requires global knowledge of the environment, which is hard to be achieved. We, therefore, constrain observability to information only locally available to the sensor nodes. Following our description in Section III regarding a sensor node's condition C^t, we assume that the only state-related information the network has access to is the number of packets stored in the buffer of each sensor node, that is

$$\Omega(s) = \Omega_1 \times \cdots \times \Omega_m, \qquad \text{where} \quad \Omega_i = b_i \tag{11}$$

with b_i indicating the buffer state of sensor node i.

POMDPs can be optimally solved using the framework of Belief MDPs [12], but this renders learning intractable, as it is performed in continuous state spaces. Histories of observations serve as an approximation to beliefs, therefore Q-learning can be applied as in the general MDP case. Through the adoption of a fixed history window w, the observation tuple of each sensor node is defined as

$$h_i = \{\omega_i^{t-w+1}, \cdots, \omega_i^{t-1}, \omega_i^t\} \tag{12}$$

The distributed nature of the problem has so far been purposely neglected in order to focus on the decision process formulation. Next, we proceed by formulating a distributed representation of the problem under the Decentralized Partially-observable MDP (Dec-POMDP) framework, introduced in [9].

Definition 1. Dec-POMDP: A Decentralized Partially Observable Markov Decision process is defined as a tuple $\langle D, S, A, T, R, \Omega, O, w, I \rangle$, where

$\mathcal{D}$ is the set of M agents

S is a finite set of states s in which the environment can be
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 of the problem
I is the initial state distribution at t = 0

Definition 1 extends the single-agent POMDP model by considering joint actions and observations. In our case, the action space for agent i (*i.e.*, node i) is $A_i \in \{1, 2, \cdots, d\}$ where d is the maximum number of replicas an agent can send. The set of observations of node i is $\Omega_i \in \{0, 1, ..., B\}$ where B is the size of the agent's buffer. Finally, the observed individual rewards for node s is $R_i = -b_i$. As regards the initial distribution I, we assume that the buffers of sensor nodes are initialized with a random number of packets $0 \leq b_i^{init} \leq B$.

To learn the optimal policy using a model-free approach one can apply simple single-agent Q-learning in order to learn the optimal joint actions for a joint history of observations using (9), which we can rewrite as

$$Q(h^t, a^t) \leftarrow (1-\alpha)Q(h^t, a^t) + \\ + \alpha\Big(r_t + \gamma \max_a Q(h^{t+1}, a^t)\Big) \quad (13)$$

where $h_t \in H = \times_{1 \leq i \leq M} H_i$ denotes the joint history of observations.

Although this approach leads to an optimal policy, it is inappropriate in the Dec-POMDP framework as adopting a centralized point of control creates a large state space and demands global access to information. The work in [13] proposes independent learning in which each agent learns its own Q-value function ignoring other agents' actions and observations. By ignoring the effect of interaction among agents this approach may converge to local optimal policies or oscillate. Nevertheless, results in [13] are encouraging as agents appear to implicitly coordinate. Thus, in our method the sensor nodes employ (13) where h_t is replaced by h_i^t and a^t by a_i^t. After learning has completed, we derive the probability distribution $\Lambda(x)$ based on the actions chosen during learning.

V. Experimental results

In order to evaluate the performance of the proposed Dec-IRSA protocol we assume toy networks exhibiting frames of size 10 and channel loads $G \in [0.1, ..., 1.0]$. All the results are averaged over 1000 Monte Carlo trials. Based on tuning simulations, we set the number of learning iterations to 1500, w and γ to 4 and 0.98, respectively. We also use an exponentially decreasing with the number of QL iterations α. As a comparison

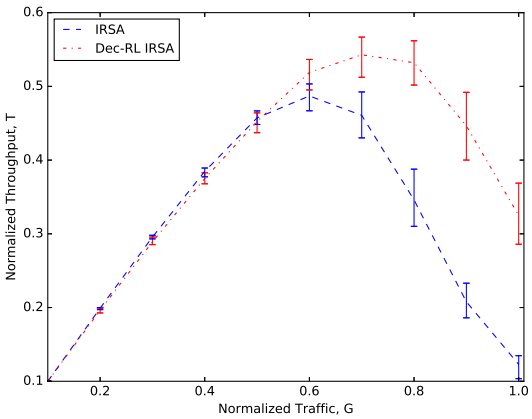

Fig. 1. Performance of IRSA and Dec-RL MAC on a toy network with $N = 10$ and varying channel loads.

scheme we use IRSA with distribution $\Lambda(x) = 0.25x^2 + 0.60x^3 + 0.15x^8$, which was experimentally evaluated in [4] and proved to be superior to other commonly used distributions derived in [14].

Figure 1 performs a statistical analysis on the performance of the Dec-IRSA and the vanilla IRSA protocols by presenting confidence intervals. The confidence intervals were calculated based on 20 independent experiments. From the results, we can see that Dec-IRSA is at least superior to vanilla IRSA for all channel loads with the difference becoming larger for $G > 0.6$. This is attributed to the use of distributions $\Lambda(x)$ derived in asymptotic settings based on characteristics of the network instead of optimizing in asymptotic settings as in [4].

Figure 2 illustrates convergence rate for independent learning for various channel loads. From this figure, we observe that convergence is easily achievable 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 the learning algorithms fails to transmit messages faster than their arrival rate, the node's buffer thus saturates fast to $r_i = -B$ when $G = 1$, while for $G = 0.8$ the buffer tends to saturate at the end of the episode. Based on this observation, we design a mechanism for agents to detect "*bad*" episodes and reset the POMDP to an arbitrary state. We classify an episode as "*bad*" if the rewards deteriorate for 3 consecutive iterations.

Figure 3 investigates how Dec-IRSA scales for different frame sizes. As regards scalability, Dec-IRSA appears to be robust and its performance slightly improves for increased frame sizes. This can be attributed to the fact that learning is more effective in more complex networks.

In Figure 4, we compare the performance of IRSA and Dec-IRSA for various frame sizes N and channel

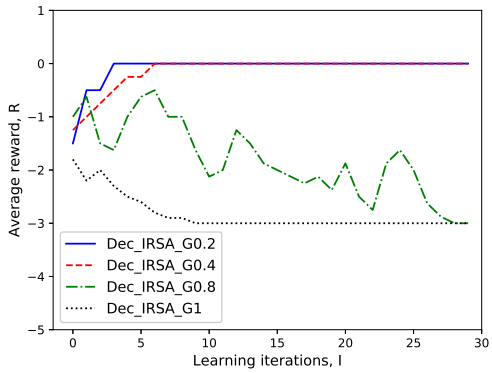

Fig. 2. Average rewards of independent learning for different channel loads G and the 25^{th} episode of its learning time.

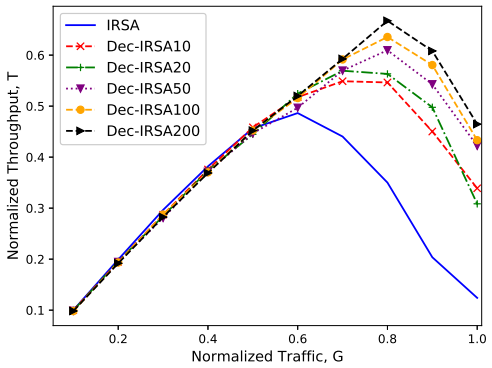

Fig. 3. Performance of Dec-RL IRSA with varying frame sizes.

load $G = 0.8$. Vanilla IRSA significantly improves its performance for increased N, as it has been proven that it 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, optimal for larger frame sizes N. Nevertheless, the Dec-IRSA outperforms significantly IRSA when channel load is heavy and frame size is small. Furthermore, it is expected that the performance of independent learning will improve under different configurations of its parameters, as the learning problem has been altered and agents need to adjust their exploration policy, learning rate and discounting of future rewards. The experiments in this case however would be highly computationally demanding. To conclude scalability analysis, the superiority of vanilla IRSA in asymptotic settings is irrelevant to sensor network scenarios, as the assumption of very large frame size N leads to inefficient implementations that require a complex receiver and introduce delay.

VI. Conclusion

In this paper, we examined the problem of decentralized MAC design through a decentralized reinforcement learning perspective. We proved that learning transmission strategies can be beneficial even under the assumption of independent learning. Our method's superiority is

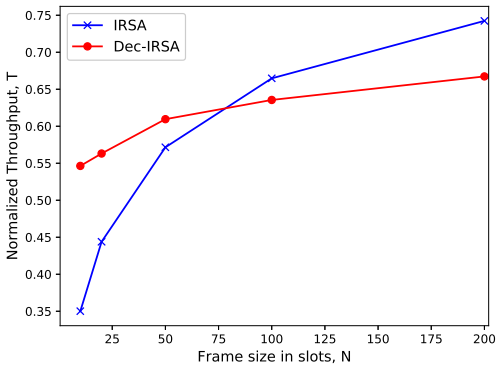

Fig. 4. Comparison of IRSA and Dec-IRSA for varying frame sizes N and channel traffic $G = 0.8$.

manifested especially in heavy channel loads, where the need for adaptivity is more eminent and agents benefit from short-sightedness and increased exploration, which implicitly ensures better coordination.

References

[1] N. M. Abramson, "THE ALOHA SYSTEM: Another Alternative for Computer Communications," in *AFIPS Fall Joint Computing Conference*, 1970.

[2] 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.

[3] E. Casini, R. D. Gaudenzi, and O. D. R. Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks," *IEEE Trans. on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.

[4] 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.

[5] L. Toni and P. Frossard, "Prioritized Random MAC Optimization Via Graph-Based Analysis," *IEEE Trans. on Communications*, vol. 63, no. 12, pp. 5002–5013, Dec. 2015.

[6] ——, "IRSA Transmission Optimization via Online Learning," 2018. [Online]. Available: http://arxiv.org/abs/1801.09060

[7] K. Waugh, D. Schnizlein, M. Bowling, and D. Szafron, "Abstraction Pathologies in Extensive Games," in *Proc. of AAMAS '09*, Budapest, Hungary, May 2009.

[8] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[9] 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.

[10] 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.

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

[12] 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.

[13] C. Claus and G. Boutilier, "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems," in *Proc. of AAAI '98/IAAI '98*, Madison, WI, USA, Jul. 1998.

[14] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.