Wireless Personal Communications (2022) 122:3445–3465
https://doi.org/10.1007/s11277-021-09094-8

Schedule-Based Cooperative Multi-agent Reinforcement Learning for Multi-channel Communication in Wireless Sensor Networks

Mohamed Sahraoui[1] · Azeddine Bilami[2] · Abdelmalik Taleb-Ahmed[3]

Accepted: 5 September 2021 / Published online: 16 September 2021

Abstract
Wireless sensor networks (WSNs) have become an important component in the Internet of things (IoT) field. In WSNs, multi-channel protocols have been developed to overcome some limitations related to the throughput and delivery rate which have become necessary for many IoT applications that require sufficient bandwidth to transmit a large amount of data. However, the requirement of frequent negotiation for channel assignment in distributed multi-channel protocols incurs an extra-large communication overhead which results in a reduction of the network lifetime. To deal with this requirement in an energy-efficient way is a challenging task. Hence, the Reinforcement Learning (RL) approach for channel assignment is used to overcome this problem. Nevertheless, the use of the RL approach requires a number of iterations to obtain the best solution which in turn creates a communication overhead and time-wasting. In this paper, a Self-schedule based Cooperative multi-agent Reinforcement Learning for Channel Assignment (SCRL CA) approach is proposed to improve the network lifetime and performance. The proposal addresses both regular traffic scheduling and assignment of the available orthogonal channels in an energy-efficient way. We solve the cooperation between the RL agents problem by using the self-schedule method to accelerate the RL iterations, reduce the communication overhead and balance the energy consumption in the route selection process. Therefore, two algorithms are proposed, the first one is for the Static channel assignment (SSCRL CA) while the second one is for the Dynamic channel assignment (DSCRL CA). The results of extensive simulation experiments show the effectiveness of our approach in improving the network lifetime and performance through the two algorithms.

Keywords Wireless sensor networks · Multi-channel · Reinforcement learning · Self-schedule · IoT

✉ Mohamed Sahraoui
mohamed.sahraoui@univ-msila.dz

Extended author information available on the last page of the article

1 Introduction

The development of multi-channel protocols for wireless sensor networks (WSNs) not only improves the network performance in terms of throughput and delivery rate as it is presented in Fig. 1, but also opens a challenge for the design of distributed resource allocation schemes in an energy-efficient way since the energy consumption is the major challenge that faces WSNs.

In recent years, WSNs have been widely used in the field of the Internet of Things (IoT) in such a way that they represent the interface between the IoT and the real world. Hence, the challenge becomes more important to optimize the network lifetime and performance as much as possible since the sensors are limited by their low-powered sources which are usually small batteries and single antenna radios. Therefore, some multi-channel protocols have been proposed to overcome this challenge and improve the WSN performance. However, in such protocols, it is difficult to fully exploit routing information to achieve a collision-free channel assignment and transmission schedule since nodes must share information dynamically and be involved in the channel assignment and scheduling process, while the requirements of frequent negotiations incur extra-large overhead. In such a case, the problem becomes NP-hard [1, 2], and it becomes even more challenging in large-scale WSNs, where the communication overhead burden should be dispensed across all sensors via distributed protocols.

Consequently, some intelligent protocols have been proposed to overcome the challenge. Some of them are not desirable for WSNs since they need a high level of energy consumption and wasted time in their performance due to their complexity such as the game theories protocols [5]. However, schedule-based reinforcement learning protocols [6, 7, 1, 8] seem more suitable for WSNs from the point of view of their fully distributed nature and their implementation which can take advantage of the routing information with less communication overhead. Implementing such protocols can improve gradually the performances of WSNs based on the principle of action and feedback analysis. Nevertheless, these protocols still require a considerable number of iterations to obtain the best solution since they focus on the single-agent learning approach without cooperation between the agents. Which in turn, costs the network by more energy consumption via the supplementary communication overhead, as well as wasting time. Furthermore, they don't focus on the energy balancing technique although it represents an important solution for improving the network lifetime since it ensures the lifetime optimization of the node and its corresponding nodes as much as possible.

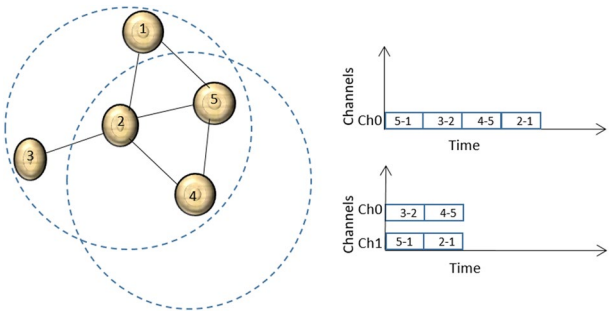

Fig. 1 Single channel Vs multi-channel communication in WSN

In this paper, our main contributions are as follows: First, we propose a Cooperative multi-agent Reinforcement Learning for Channel assignment approach to improve the learning process by accelerating the number of iterations in distributed schedule-based WSNs. Second, to ensure energy efficiency, we strengthen the proposed approach by both self-scheduling and load balance methods. Hence, the RL agents schedule selfishly the cooperation flow to assign optimal channels based on the balance of the load between the communicating nodes. Third, we propose two algorithms to investigate the performance of our approach in static and dynamic modes. The first one is the Static Self-schedule based Cooperative multi-agent for Channel Assignment (SSCRL CA), while the second one is the Dynamic Self-schedule based Cooperative multi-agent for Channel Assignment (DSCRL CA).

The objective is to improve the network performance and lifetime in schedule-based multi-channel WSNs by accelerating the RL iterations and reducing the energy consumption through the reduction of both communication overhead and collisions on one side, and the balance of energy consumption on the other side.

In the remainder of this paper, we discuss in Sect. 2 the related works in schedule-based multi-channel WSNs. In Sect. 3, we present our proposed approach and protocols. The obtained results after evaluation and their discussions are presented in Sect. 4. Finally, a conclusion with future works of this study is presented in Sect. 5.

2 Related Works

Several distributed schedule-based multi-channel protocols have been proposed for WSNs to accommodate parallel transmissions using multiple channels. These protocols can be classified into two categories: intelligent and non-intelligent protocols. The non-intelligent protocols assign the channels in either static or dynamic mode. In the static mode, the nodes keep the same selected channels in the time slots of the repeating frame periods, while in the dynamic one, the channels can be changed in each frame period.

The authors of [9] have proposed a dynamic multi-channel MAC (Y-MAC) protocol. In this TDMA based multi-channel MAC protocol, the time is divided into frames. Each frame is divided into a Broadcast sub-period that takes 3 slots, and Unicast one for the remaining slots. In the Broadcast sub-period, a common channel is used to exchange control messages in order to coordinate the communication that will be in the second sub-period. In [10], a static scheduled-based multi-channel MAC protocol is proposed. In this protocol, the neighboring nodes are splitting into different groups and the time slots in the frame period are divided between these groups. The receiving nodes then, select the channel-slot pairs in the group sub-period that are not chosen in tow-hops neighboring nodes. In [11] a static Multi-Channel MAC (MC-MAC) protocol is proposed. In MC-MAC, the neighboring nodes start by exchanging their frame vectors that indicate the used channel in each time slot then select the time slots to be used on a particular channel in a collision-free manner. In [12], the authors have proposed a dynamic Regret Matching based Channel Assignment algorithm (RMCA). RMCA focus on the fact that each sensor node updates its choice of channels according to the historical record of these channels' performance parameters which is based on the success of the transmissions and delay to reduce interference. The authors of [13] have proposed a static joint time slot and frequency channel allocation algorithm that bases on the loopy belief propagation (BP) approach using a factor graph. The algorithm imprints space, time, frequency and radio hardware constraints into

a loopy factor graph and performs iterative message-passing loopy belief propagation with randomized initial priors.

Nevertheless, the non-intelligent protocols generally require the overall knowledge of the external environment to perform their decisions which results in a high amount of message transfer leads to a high cost of energy consumption.

The intelligent protocols are distinguished from the non-intelligent ones by using only the feedback acquired from the external environment in the channel selection process to obtain approximately the best selection. These protocols can be classified into two categories: Game-based and Learning-based protocols. The Game-based methods involve the application of the game theory rules to study strategic decisions for optimal channel selection.

In [5], the power levels at which a node should transmit are considered and the investigation for the existence of Nash Equilibrium (NE) is done for two different cases, with xed channel conditions, and with varying channel conditions. [3] focuses on optimizing the network performance through using two controlling techniques: nodes transmission power and communication interference. Therefore, after the game period, the node with low residual energy chooses a lower transmission power. the authors of [14] investigated channel allocation to only the receiving nodes by allocating different channels for adjacent nodes. The nodes then, use the receiving channels information to select different channels by giving priority to low energy nodes. The authors of [4] have proposed a centralized scheme for spectrum allocation modeled as a multi-objective optimization problem for maximizing spectrum utilization, proportionally fair allocation, transmission priority among nodes and avoiding unnecessary spectrum handover. Then they have used a modified cooperative game to deal with the multi-objective optimization problem as a single-objective function.

Even though the game-based methods may lead to a better solution in terms of interference-free, the large number of iterations in the game phase to obtain the NE results in high costs in terms of energy consumption and delay, which leads to inappropriate adaptation for large scale WSNs.

Learning-based methods emphasize the application of the Reinforcement Learning (RL) approach that is characterized by the greedy mode, which is more suited to the WSN's nature [15]. The RL approach has already been used to propose solutions to several single-channel problems in WSN, such as routing and task scheduling [16, 15]. However, in the last years, it has been used to deal with the distributed schedule-based channel assignment problem using a single agent RL.

The authors of [8] have proposed a Normal Equation based Channel quality prediction (NEC) algorithm that starts by performing channel rank measurement (CRM) based on the received signal strength indicator (RSSI) and the average of the link quality indicator (LQI) for each channel. The set of accepted channels is then used for training a machine learning operation between the neighboring nodes. In [7], a trade-off between two methods is proposed, an on ligne non-cooperative game and an offline schedule-based machine learning for channel assignment. The first method is used if the network can satisfy the energy requirements, otherwise, the second one is used dynamically. In [6], the authors propose the Coverage and Connectivity Maintenance algorithm based on RL (CCM-RL) to give the node the opportunity to take the best action in order to maximize the coverage rate and maintain network connectivity. In [1], the schedule-based multi-channel communication protocol that performs upon RL MMAC (Reinforcement Learning Multi-channel MAC) algorithm is proposed. It represents an extension work of the one proposed in [2]. In RL-MMAC, the nodes learn to perform their transmission schedule on their parent's channels in a distributed manner using the "*win stay, lost shift*" strategy. After the learning

process, the best action in each slot is chosen if its probability of success exceeds the sleeping threshold.

The cooperative multi-agent RL approach is used to build a cooperative machine learning system between more than one agent which allows multiple agents to learn together utilizing one another's strength for decreasing individual learns weaknesses and enabling learning to be accelerated. Nevertheless, the use of this approach opens up new necessary issues to be tackled that can be resumed in the response of the three following questions: Why, How and when the multiple agents can learn together?

To answer these questions, several solutions have been proposed in the Artificial Intelligence multi-agent RL sub-field for Transfer Learning (TL) depending on the cases for which these solutions are proposed. Hence, the cooperation between the RL agents can be in many kinds such as coordination, competition and advising [17]. Nevertheless, these solutions can not be applied directly to the WSNs without considering their resource-constrained. For this reason, some other solutions have been proposed to deal with the TL between the RL agents in a decentralized WSN focusing on the energy efficiency factor such as in [18–20] for task scheduling. However, these solutions are based on a single channel.

In multi-channel communication, the cooperative multi-agent RL focus especially on Cognitive Radio (CR) networks for spectrum sensing to avoid primary users as in [21, 22]. However, CR networks differ from WSNs in such a way that CR devices have more power and capabilities than those of WSNs.

In [23], the authors propose the Collaborative Multi-agent Anti-jamming Algorithm (CMAA) to avoid unsecured channels on one side and compete for the best channel to be used in each time slot on the other side in wireless networks.

However, these solutions don't take into account the scheduling issue in the transfer learning process, which can result in a high cost of collisions and overhead communication leading to more energy consumptions and negative transfers. Also, most of these protocols don't focus on energy balancing technique that represents an important solution for improving the network lifetime.

As an alternative to these limitations, we propose SCRL CA (Self-schedule based Cooperative multi-agent Reinforcement Learning for Channel Assignment) approach which performs channel selection based on the self-schedule scheme in a cooperative learning manner to improve the performance and lifetime in schedule-based multi-channels WSNs. Also, we investigate the performance of our approach through two protocols, one for the Static mode (SSCRL CA) and the other for the Dynamin mode (DSCRL CA).

3 Self-schedule Based Cooperative Mumti-agent Reinforcement Learning for Channels Assignment (SCRL CA)

3.1 Self-schedule Approach

In WSNs, the use of the routing metrics focuses generally on the receiver selection process. Hence, the nodes have to collect these factors periodically from their neighboring nodes then they use it to select the optimal receiver node which results in a high cost in terms of communication overhead and thus energy-wasting. In contrast, the authors of [24] have used these factors in balance by a self-scheduling manner for data routing in WSNs based single-channel to reduce the communication overhead caused by the exchanged information

between the neighboring nodes. For this purpose, each node that has to send a data message starts by sending an RTS (Request To Send) control message to its neighboring nodes, and then the receiver nodes must wait a time measured by (1) before the response by a CTS (Clear To Send) message. In this way, the node which has waited the smallest time responds first, and the other waiting neighboring nodes learn from this response that the demand is satisfied without sending its CTSs control messages.

$$t = \frac{\left(\frac{1}{d}\right) \times \gamma}{E \times r} \tag{1}$$

where, d is the distance of the node from the sink, E is its self-residual energy, γ represents the number of neighboring nodes and r is the link reliability.

Although the results demonstrate the efficiency of these scheme in term of communication overhead reduction, it has some limitation that can be resumed as follow: first; it doesn't perform the collision-free aspect for the sender nodes since it focuses only on the receiver ones without taking in account that the sender nodes can send RTSs simultaneously. Second; it uses a free waiting time which can be very long even for the smallest one, which in turn increases the wasting time and energy consumption.

3.2 System Model

We dene a WSN as a set of N nodes denoted us N=$\{1, ..., N\}$, randomly dispersed throughout an area. Each node uses a predefined set of orthogonal channels K=$\{f_1, ..., f_k\}$ in a range of communication R, for example the non-overlapping channels proposed by IEEE802.15.4 standard (16 channels in 2.4 GH band and 10 channels in 915 MH band) [25]. All the nodes can communicate with the sink node in a multi-hop fashion through the communication with their M neighbors located within their communication range.

After the initialization phase, all the nodes are synchronized with the global time clock of the sink. Hence, the time is divided into consecutive frame periods which are in turn, divided into a fixed number of slots. Tow types of slots have been considered, Beacon slots for exchanging control information (Beacon messages, new request messages, etc) on the same dedicated channel, and Data slots for data transmission. Furthermore, each node keeps the lowest number D of hops from the sink node as its depth and a list of its parent nodes selected from the list of the neighboring nodes as well as the maximum number of parent lists of the neighboring nodes. The parent nodes are the neighboring nodes that have a smaller D than that of the node.

3.3 Problem Formulation

To perform both channel and time (channel-time) schedules for each time slot, a cooperative multi-agent reinforcement learning approach is applied. Hence, the channel-time schedule problem can be formulated as a Markov Game ([17]) that can be expressed mathematically as MG=$\{n, S, A, T, R_{1...N}, \gamma\}$, where;

- n is the number of agents,
- S is the set of states of all agents,
- A is the joint action space composed of local actions for all the agents,
- T:S×A×S⟶*[0,1]*, is the state transition function,

- R_i:S×A×S⟶ **R** is the reward function of agent i.
- $\gamma \in [0,1)$, is the discount factor, which represents the relative importance of future and present rewards.

Hence, we define the number of agents n by the number of all the nodes N, so each node is considered as an agent. The set of actions for each node is defined as $A_x = \{f_1, ..., Sf_k, Rf_1, ..., Rf_k\}$, where $A_x$is the set of actions for the node x, Sf_i and Rf_i mean send on channel f_i and receive on channel f_i respectively. Therefore, A_1=A_2 =...=A_N. In order to avoid complex calculation that is not desired for the WSN, we use the stateless variant of the Q-learning method that has been demonstrated its efficiency for distributed learning problems [26]. Hence, the state transition function is chosen based on a trade-off between the strategy *"win stay, lost shift"* used in RL MMAC [1] and the stateless Q-learning method. The *"win stay, lost shift"* strategy plays an important role in the learning acceleration process by the fact that if the agent fails in performing action after some successes, it does not lose time in the gradual return. Therefore, T is fined by (2) :

$$Q_{x,t+1}(a_t) = (1-\alpha)Q_{x,t}(a_t) + \alpha r_x(a_t) \tag{2}$$

where, $Q_{x,t+1}(at)$ means the Q value update at time slot t + 1 (the slot in the current frame) by agent x, after executing at time slot t (the same slot in the last frame), the action a_t. $r_x(a_t)$ means the immediate reward calculated by (3) after executing the action a_t at time slot t by agent x. Hence, the reward takes the value (+1) if either the receiver node receives a data message or the sender node receives an Acknowledgement (Ack) message on the chosen channel.

$$r_{x,t}(a_t) = \begin{cases} +1 & \text{if it is a successful communication} \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

α is the learning rate parameter which can be set to a value in [0,1]. In order to formulate the *"win stay, lost shift"* strategy that it is mentioned in 28 without formulation, we use the *"win or learn fast"* for variable learning rate method proposed in [27], where a small value of α is used for successful actions and a higher value is used for unsuccessful ones. Therefore, α is done by (4) as follow:

$$\alpha = \begin{cases} 0.01 & \text{if r} > 0 \\ 0.1 & \text{otherwise} \end{cases} \tag{4}$$

The best joint action to be selected in the next time slot is done by(5);

$$a^* \in Max_{a \in A_{av}} \left[\sum_{x=1}^{NG} Q_{x,t+1}(a) \right] \tag{5}$$

where, a^* is the best joint action to be selected in next time slot by looking at the maximum of Q values sums after performing actions among the available action set A_{av} by the Neighboring nodes Group (NG) since the cooperation between the agents is reduced to that between the neighboring nodes. To perform (5), each node must collect messages from its neighboring nodes according to the cooperation model explained in the next sub-section.

The discount factor γ is not taken since we use the stateless method.

3.4 Cooperation Model and Algorithms

In order to perform the cooperation process in an energy-efficient way, we have used prior coordination based on *"social conventions"* strategy used in [28] to coordinate between Unmanned Aerial Vehicles (UAVs) for field coverage. For that, the UAVs coordinate the action to be selected in advance by the fact that each UAV must not choose the action chosen by the others. To perform this coordination method, specific ranking order is assigned to each UAV. The one with the highest order selects action first and lets the others know its action. The other UAVs then can match their actions with respect to the prior selected ones.

To adapt this method to the WSN in an energy-efficient way, we have used a Self-scheduling mechanism between the neighboring nodes. Hence, before performing actions in the selected channel, the neighboring nodes select actions sequentially by informing one another. To do that, the neighboring nodes must use the same dedicated channel. Therefore, each data time slot in the frame period is divided into two periods: a broadcast period T_B in which all nodes turn back to the dedicated channel, and unicast period T_U in which the communicating nodes use the selected channel for sending data messages. Note that, the dedicated channel can be used in T_U period. The T_B period is divided into two equal and consecutive sub-periods: T_{RTS} and T_{CTS} sub-periods as it is shown in Fig. 2.

T_{RTS} sub-period is used for sending Request To Send (RTS) control messages, while T_{CTS} sub-period is used to send the Clear To Send (CTS) ones by the parent nodes. The sending of control messages is performed in a self-scheduling manner to reduce the communication overhead, balance the remaining energy and avoid collisions. Therefore, in the T_{RTS} sub-period, the sender nodes schedule the sending of their RTSs. However, in T_{CTS} sub-period, the receiver nodes schedule and send the CTSs as follow:

Based on the positive number of its parent nodes (Parents Degree (PD>0)), the maximum of parents degree (Max Parents Degree MaxPD) between the neighboring nodes, the data queue size (λ), and the residual energy E_R, each sender node selects a random waiting time (RT_{WS}) bounded by T_{WS} in T_{RTS} sub-period, before transmitting RTS control message. Note that, the node that has no parent, except the sink, can neither send nor receive since it is excluded. The T_{WS} is calculated by (6).

$$T_{WS} = T_{RTS}^{(P_{WS}/(S_N+1))} \tag{6}$$

Where;

$$P_{WS} = \left(\frac{1}{\lambda}\right)\left(\frac{PD}{MaxPD}\right)\left(\frac{E_R}{E_{Init}}\right) \tag{7}$$

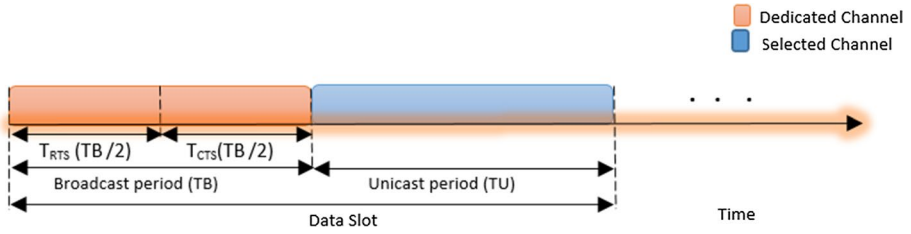

Fig. 2 SCRL CA time slot structure

S_N is the number of successful communications started by a value 0, and E_{Init} is the initial energy of the node.

$$T_{RTS} = T_{CTS} = \frac{T_B}{2} \tag{8}$$

Hence, by (6) we can ensure a collision-free self-scheduling between the sender nodes within the first half of T_B (T_{RTS}) sub-period. Thus, (7) gives priority for the node with less residual energy and parents degree in one hand, and with more data messages in its queue in the other hand, to wait less time in order to preserve energy and prefer the sender with more data messages and fewer parents degree to send first, since nodes with more parents degree have more chance to relay its data messages. In addition, the T_{WS} is decreased gradually based on the number of successful communications S_N to give priority to the sender nodes that have succeeded to remain winners. Furthermore, the sender nodes choose their actions by performing the formula (5) in a sequential manner. Therefore, the first sender node chooses its action, then it broadcast its choice, the other sender nodes, as well as the other neighboring nodes, learn from this choice by adding the sent Q value to their Q tables. Then the following sender nodes will be forced to choose other actions, among the available actions, sequentially. The different steps in the T_B period are presented in the flow chart of Fig. 3.

On the other side, the parent nodes receive in T_{RTS} sub-period, the RTS control messages sequentially depending on the priority of the sender nodes. The parent nodes then select a random waiting time (RT_{WR}) between T_{RTS} and T_{WR} times in T_{CTS} sub-period, before the sending of their responses (CTSs). The T_{WR} is calculated by (9) as follow:

$$T_{WR} = T_{RTS} + T_{CTS}^{(P_{WR}/(S_N+1))} \tag{9}$$

Where;

$$P_{WR} = \left(\frac{(MaxPD + 1) - PD}{MaxPD + 1}\right)\left(1 - \frac{E_r}{2E_{Init}}\right) \tag{10}$$

Hence, by (9) we ensure a self-schedule with collision-free between the receiver nodes based on residual energy and parents degree. The aim is to balance the remaining energy between the neighboring nodes on one hand and select the best path to optimize delivery delay on the other hand. Also, the T_{WR} is decreased gradually based on the number of successful communications S_N to give priority to the sender nodes that have succeeded to remain winners. Thus, (10) prefers the parent with more energy and parent degree to respond first. Therefore, the other parent nodes, as well as the other neighboring nodes, learn from the previous responses and respond to the other RTS control messages sequentially. To do this, each receiver node updates its RTS queue after the waiting of RT_{WR} during which it listened to the other CTSs (the case is mentioned by (*) in Fig. 3). The updating process focuses on delating of the RTSs that belong to the following cases:

- The RTSs for which the node is a parent, and are not answered by the previous parent nodes, but they select a channel that is used by the previous responses.
- The RTSs for which the node is a parent, but they are answered by the previous parent nodes,
- The RTSs for which the node is not a parent,

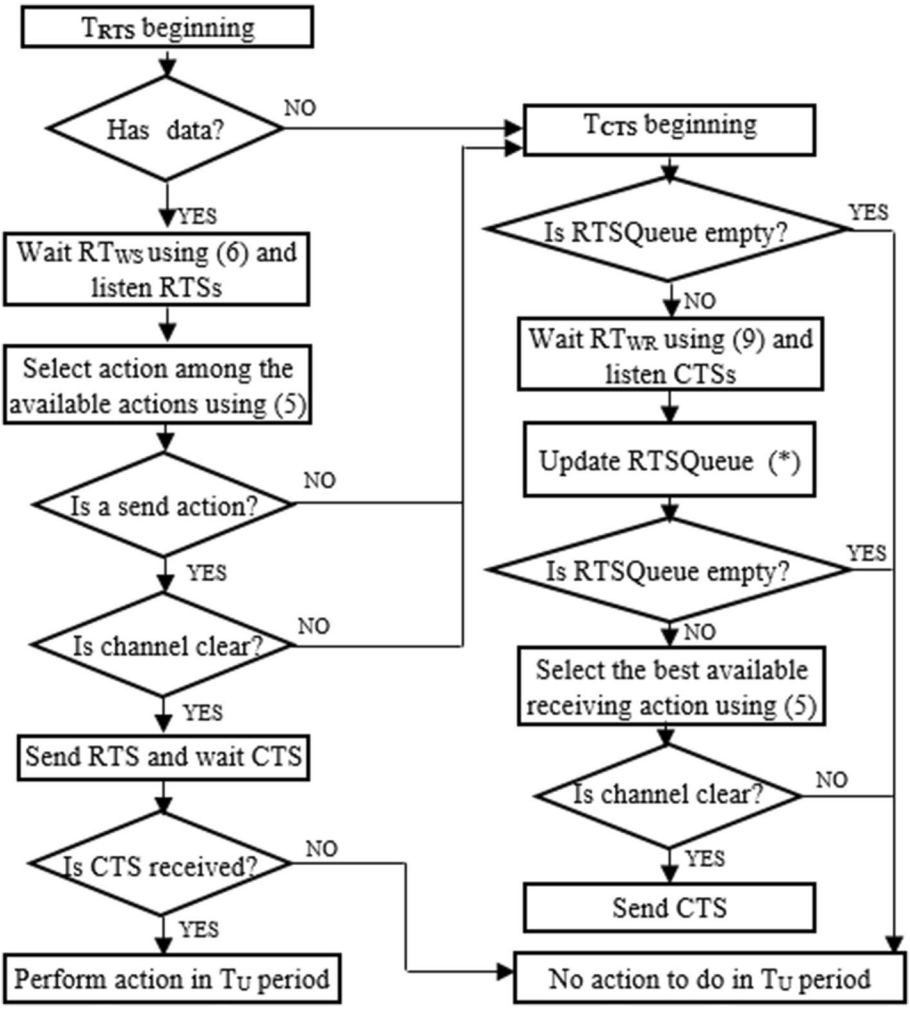

Fig. 3 Cooperation model and scheduling process in T_B period

Then, the parent selects the RTS which corresponds to the best available response action. In this way, we ensure collision-free communication that avoids both collision types: direct and indirect collisions. Furthermore, by (8) we ensure a self-schedule with collision-free between the receiver and the sender nodes.

Algorithm 1: DSCRL CA

```
1. Initialize Q table;
2.  while not end
3.   for each frame
4.    for each data time slot
5.      do LEARNING scheme depicted in Figure (3)
6.      if ther is an action to do in T_U period  then
7.       performe action in T_U period
8.       observe Reward
9.       update Q-table using (2)
10.      update Data queue
11.     else
12.      Sleep in T_U period
13.     end if
14.    end for
15.  end for
16. end wile
```

As mentioned before, we have used two algorithms: Dynamic Self-schedule based Cooperative multi-agent Reinforcement Learning for Channel Assignment (DSCRL CA) that is presented in Algorithm1, and Static Self-schedule based Cooperative multi-agent Reinforcement Learning for Channel Assignment (SSCRL CA) that is presented in Algorithm2.

Algorithm 2: SSCRL CA

```
1. Initialize Q table,Δ,P_Sleep
# LEARNING
2.  while not end of learning frames
3.    for each frame
4.      for each data time slot
5.        do LEARNING scheme depicted in Figure (3)
6.        if ther is an action to do in T_U period  then
7.          performe action in T_U period
8.          observe Reward
9.          update Q-table using (2)
10.         update Data queue
11.       else
12.         Incrimente P_Sleep
13.         Sleep in T_U period
14.       end if
15.     end for
16.   end for
17.  end wile
# SELECTION
18.    for each data time slot in the frame
19.      if P_Sleep > Δ then
20.         Slot-action <– sleep
21.      else
22.         select a* with Q(a*) > Q(a) for all actions
23.         Slot-action <– a*
24.      end if
25.    end for
```

In SSCRL CA, we have used sleeping probability (P_{SLEEP}) for each slot that is initialized by the value 0 and increments if there is no action to do in T_U period (line 13). After the learning period, each node looks for each slot at whether its sleeping probability (P_{SLEEP}) is greater than a threshold value Δ. If it is the case, it takes the sleeping action during this slot as the slot action (lines 19, 20), otherwise, it takes the best-performed action in the learning period as the slot action (lines 22, 23). Furthermore, the data message size is increased to satisfy the slot size in the absence of T_B period after the learning phase since the data message size can achieve 128 bytes in IEEE802.15.4 for example [25]. In addition, The new nodes can discover their parents through the beacon messages that are transmitted periodically in Beacon slots, and start the learning process with the sleeping slots of their parents. In order to accelerate the learning process of the new nodes, each parent keeps a blacklist of channels (Bl_{Chs}) for each slot, created during the learning phase, which includes the different used channels by the neighboring nodes in the concerned slot. During the negotiation between the new node and its parents, each parent sends the list of sleeping data time slots as well as the Bl_{Chs} for each data time slot to be excluded from the available actions of the concerned slot.

4 Simulation and Results

In this section, we present the simulation results of our protocols in comparison with both RL MMAC [1] which uses the same distributed tree topology in static mode and CMAA [23] that is characterized by its dynamicity. The different protocols are implemented using Java language since it is one of the most popular and attractive programming languages especially for building flexible, portable and high-performance network applications. Several simulation tools and libraries based on Java have been developed for the simulation of wireless networks.

We have used JSensor simulator [29] since it uses a parallel simulation based on the available processor cores number. In order to implement the different protocols, we modify JSensor by integrating the following aspects:

- Time Division Multiple Access (TDMA) scheduling mechanism
- Multi-channel mechanism: for which, the nodes can communicate only using the same channel.
- Energy consumption mechanism: For which we have used the same model for calculating the communication energy dissipation that is used in [30]. The energy spent (E_{Tx}) for the transmission of k-bit packet over a range R is given by (11):

$$E_{Tx} = E_{Elect} \times K + E_{amp} \times K \times R^2 \tag{11}$$

Where, E_{Elect} is the required energy for activating the electronic circuits. E_{amp} is the required energy for amplification of transmitted signals to transmit one bit in open space. E_{Elect} and E_{amp} are fixed at $5 * 10^{-8}$ and $3 * 10^{-8}$ respectively.

Energy consumption to receive a packet of K bits (E_{Rx}) is calculated according to (12):

$$E_{Rx} = E_{Elect} \times K \tag{12}$$

Energy consumption in idle time T (ms) (E_{Ix}) is calculated according to (13):

$$E_{Ix} = E_{Elect} \times T \tag{13}$$

An example of the implemented schedule-based process for the SSCRL CA learning period is shown in Fig. 4. For the reason of the presentation, we have used 5 frames with 4 time slots only.

We run simulation experiments with 100 nodes placed randomly in an area of 500m * 500m. The sink is placed in the middle and the transmission range is fixed at 50 meters. The leaf nodes generate data messages every 10 seconds. The different results are averaged over 10 simulations.

Two scenarios are token, the comparison of learning period between RL MMAC and SSCRL CA, then the comparison of network lifetime and performance for a long time measured by 300 minutes between the three protocols CMAA, SSCRL CA and DSCRL CA.

For the first comparison shown in Fig. 5, six metrics are token over the variation of the number of channels:

- End to end packets delivery ratio,
- Total energy consumed ratio,

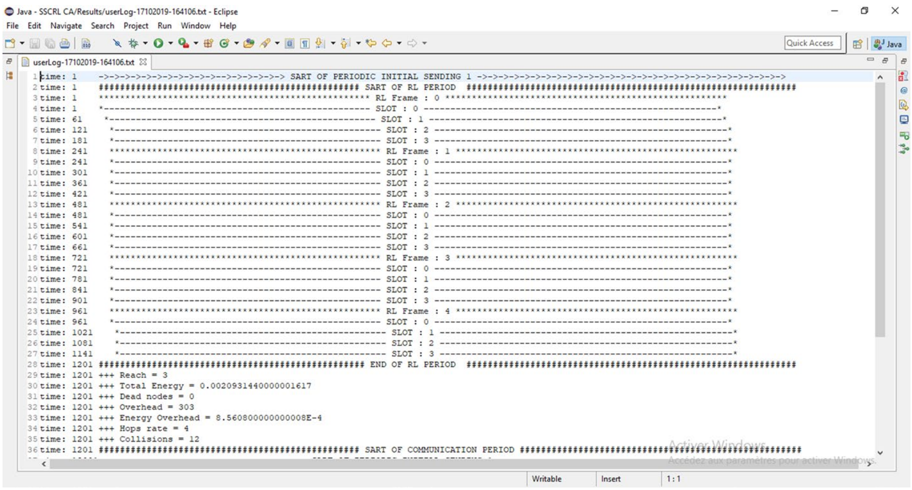

Fig. 4 Run result showing schedule-based process in SSCRL CA learning period

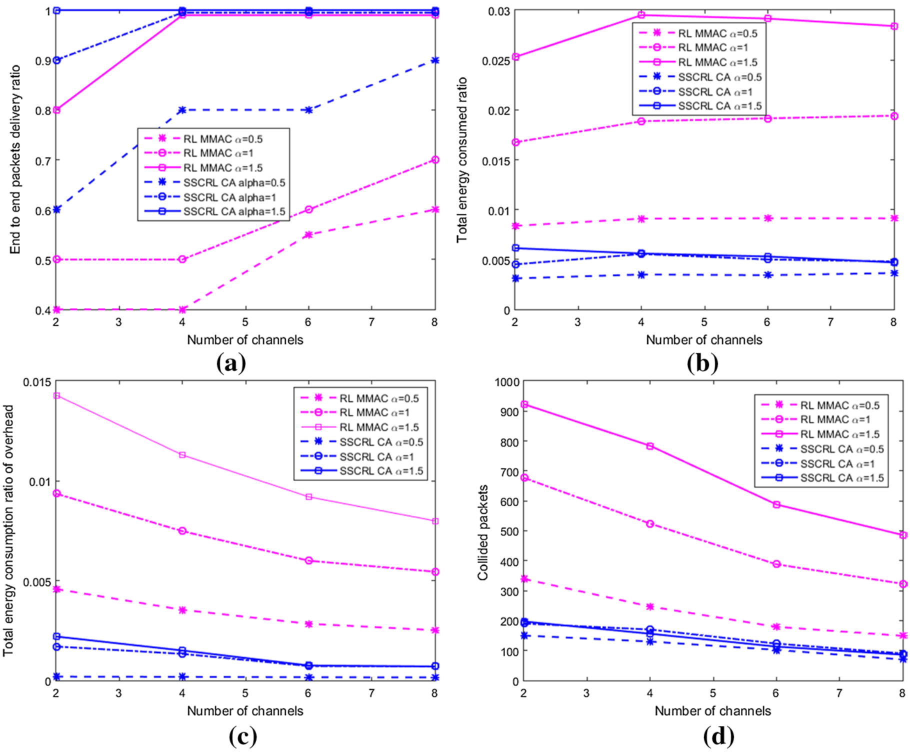

Fig. 5 Comparison of **a** End to end packet delivery ratio, **b** Total energy consumption ratio, **c** Total overhead energy consumption ratio and (d) The collided packets per number of channels in Learning period

– Total energy consumed ratio of overhead,
– Number of colliding packets,
– End to end latency,
– End to end hops rate.

To evaluate the impact of the number of time slots per frame, we have used the parameter α which is defined as the ratio of the number of time slots per frame to the number of periodic generated data messages by the leaf nodes. The length of the slot is changed from RL MMAC to SSCRL CA in such a way that in RL MMAC, the token is 40 ms, while in SSCRL CA, it is 60 ms. The aim is to use the same packet length considered by 64 bytes in 40ms and exploit the first 20 ms in SSCRL CA for the T_B period. As it is mentioned in [1] for the length of the learning period that depends on the maximum path to the sink node, we have used 15 frames which is more than enough.

Figure 5 shows that SSCRL CA can reach the total end to end packets delivery for all the 10 experiences in α=1.5, while RL MMAC succeeded for more than 2 channels in all the experiences and only for two experiences in 2 channels (Fig. 5a). Furthermore, SSCRL CA can optimize the delivery packets by more than 50% in the other values of α. The energy consumed in the learning phase is greatly reduced by SSCRL CA in different situations (Fig. 5b). This reduction can achieve 80% of the energy rate consumed by RL MMAC. The reason can be explained by the fact of the high communication overhead generated by RL MMAC since it broadcasts the data packets instead of the control packets used in SSCRL CA (Fig. 5c), as well as the significant reduction of the collisions performed by SSCRL CA that can be reached 78% due to the self-scheduling mechanism that is used (Fig. 5d).

In Fig. 6, we have taken the values of α for which there is a total end to end packets delivery to investigate the latency and the average of the hops taken by all the data messages from the source nodes to the sink in the learning phase of the two protocols. Note that in Fig. 6a, we have taken only α=1.5 for RL MMAC since it performs the total reach at this value. Therefore, the latency of SSCRL CA is better than that of RL MMAC if the frame periods are identical since the frame period of SSCRL CA in α=1 is identical to that of RL MMAC in α=1.5. However, if the α values are the same, RL MMAC performs well in latency than SSCRL CA since the time slot in the frame period of RL MMAC is lower than that of SSCRL CA by 33 %. Figure 6b shows the path length rate taken by all

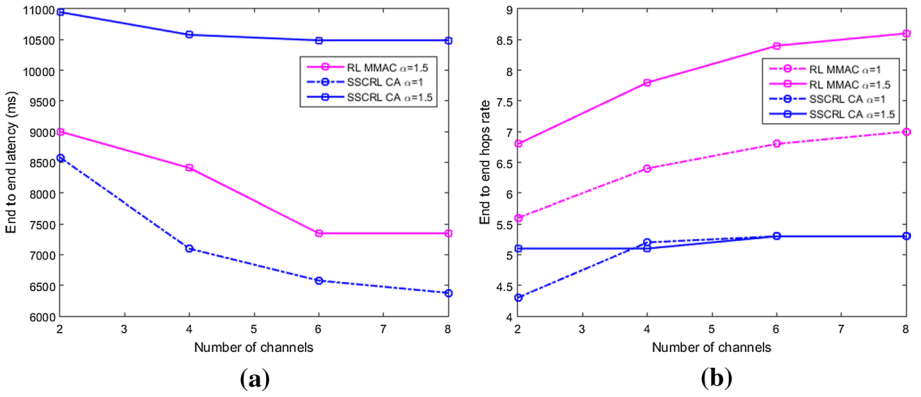

Fig. 6 Comparison of **a** Total end to end delivery and **b** Hops rate per number of channels in Learning period

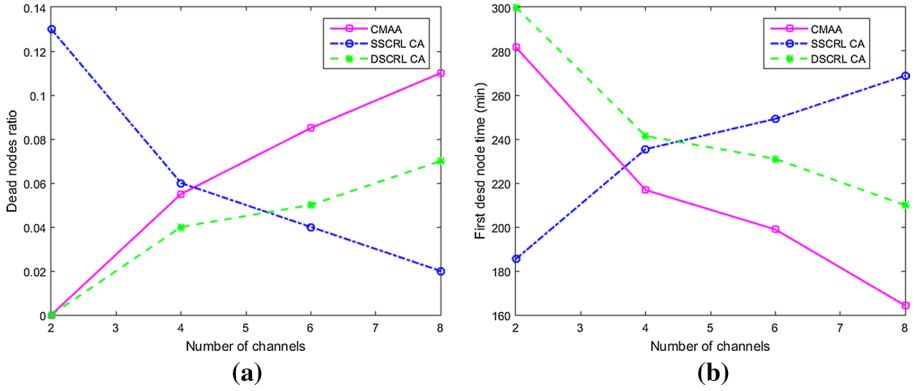

Fig. 7 Comparison of **a** Dead nodes ratio and **b** First dead node per number of channels in 300 minutes of communication

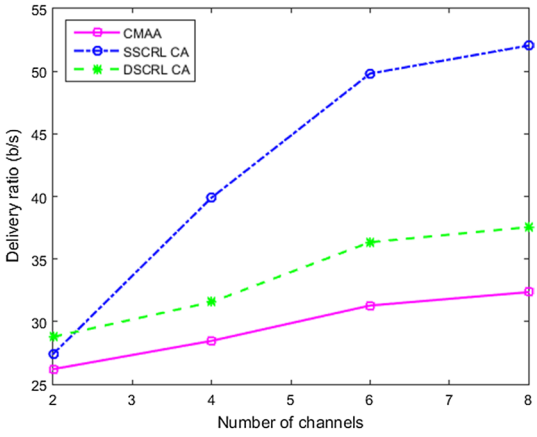

Fig. 8 Comparison of delivery ratio in 300 minutes of communication

the data messages to the sink node by the two protocols in the two values of α (1,1.5). Hence, SSCRL CA continually reduces the length of the path according to the number of used channels by an increasing rate that can reach 36.48 % on 8 channels at α=1.5. This can be explained by the effect of the default channel mechanism used by RL MMAC since the tracking of the default channels can result in very long paths, while SSCRL CA gives priority to build the smallest paths to the sink as much as possible based on the dynamic channel selection mechanism that is used.

For the second comparison shown in Figures (Figs.7, 8, 9), we set the value of α at 1.5 to investigate the network energy consumption and delivery using the three protocols CMAA, SSCRL CA and DSCRL CA for a long time considered by 300 minutes. Therefore, four metrics are token over the variation of the number of channels:

- Dead nodes,
- The first dead node,
- Delivery ratio in bit per second,
- Total energy consumption ratio of overhead.

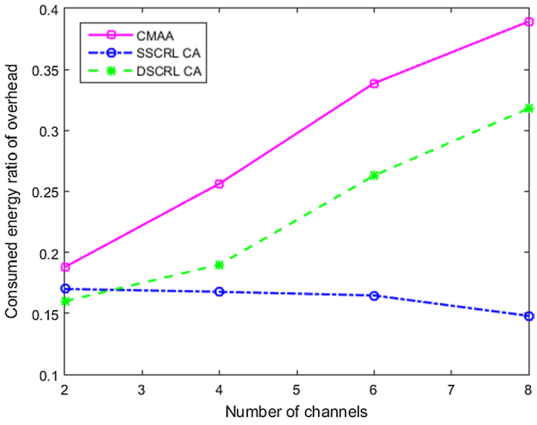

Fig. 9 Comparison of total overhead energy consumption in 300 minutes of communication

Figure 7 shows the dead nodes in 300 minutes of performance using the three protocols. Hence, DSCRL CA is the best one at using of few channels (2,4) in such a way that it does not suffer from any dead node using 2 channels and suffers from the lowest dead nodes ratio using 4 channels. However, unlike the other two protocols, the number of dead nodes in DSCRL CA increases relatively with the increase in the number of the used channels, which makes SSCRL CA the best one at using more than 4 channels. This is due to the increasing level of the communication overhead in T_B period relatively with the increase in the number of the used channels, which leads to an increase of supplementary energy consumption that is shown in Fig. 9, and which has overcome the existed energy balance mechanism in 6 and 8 channels use cases. The delivery ratio in Fig. 8 is optimized especially in SSCRL CA in an increasing manner according to the number of used channels. Hence, The optimization is increased from 6.07% using 2 channels to 60.42% compared to CMAA delivery ratio at 8 channels. The reason returns to the avoidance of the communication overhead as well as the increasing of the packet length after the learning period, which is performed in a successful manner compared to the CMAA as it is explained above. However, the delivery ratio increases slowly in DSCRL CA. It goes from the rate of 10.18 % at 2 channels compared to CMAA to the rate of 15.02 % at 8 channels, and this is due to the costs of supplementary overhead as it is shown in Fig. 9 that's mention an improvement of DSCRL CA compared to CMAA due to the self-scheduling mechanism.

5 Conclusion

In this paper, we propose a distributed cooperative multi-agent reinforcement learning approach which bases on the self-schedule scheme for channel assignment in schedule-based wireless sensor networks. The main challenge of such networks is energy consumption due to the communication overhead, collisions and unbalanced energy consumption that drastically reduce the network lifetime. For this reason, special consideration is given to reduce both the communication overhead and collisions and balance the energy consumption as much as possible based on the self-schedule scheme that plays an important role in accelerating the RL iterations. We investigate the use of our approach through two protocols, static (SSCRL CA) and dynamic (DSCRL CA) protocol. The proposed protocols

enable nodes to implicitly schedule and adapt their energy consumption in an efficient way based on the availability of the channels. They involve a distributed channel selection in a self-schedule manner based on the routing metrics and they take the energy as the base metric by enabling nodes to reduce communication overhead and collisions, but also to balance the energy consumption between them in order to improve the network lifetime in addition of the throughput and latency in static and dynamic fashions. Through simulations and experiments, we demonstrate the effectiveness of our approach through the two protocols. As a result, our approach signicantly reduces the energy consumption and improves the network lifetime with good amounts of packet delivery ratio in both protocols in the cases of using of a small number of channels, however, the performance of the dynamic protocol (DSCRL CA) is deteriorated compared to that of the static one (SSCRL CA) in using of a high number of channels. As future work, we propose a hybrid method that uses the dynamic method in using a small number of channels and it substitutes into the static method by using a high number of channels. Furthermore, it is suitable to implement our protocols in real-life applications to perform real evaluations [8].

Funding This research did not receive any specific funding and is being conducted as part of employment and higher degree of the authors.

Data availability This study did not use data or datasets. This study is based on using randomly generated data by the simulator as input.

Code availability Custom code is available upon request due to privacy or other restrictions.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Phung, K. H., Lemmens, B., Goossens, M., Nowe, K., Tran, L., & Steenhaut, K. (2015). Schedule-based multi-channel communication in wireless sensor networks: A complete design and performance Evaluation. *Ad-hoc Networks, 26,* 88–102.
2. Phung, K. H., Lemmens, B., Mihaylov, M., Tran, L., & Steenhaut, K. (2013). Adaptive learning based scheduling in multichannel protocol for energy-efficient data-gathering wireless sensor networks. *International Journal of Distributed Sensor Networks, 9,* 1–11.
3. Hao, X. C., Gong, Q. Q., Hou, S., & Liu, B. (2014). Joint channel allocation and power control optimal algorithm based on non-cooperative game inwireless sensor networks. *Wireless Personal Communications, 78*(2), 1–15.
4. Hao, X. C., Zhang, Y. X., & Liu, B. (2017). Fair dynamic spectrum allocation using modied game theory for resource-constrained cognitive wireless sensor networks. *Symmetry, 9*(5), 73–87.
5. Sengupta, S., Chatterjee, M., & Kwiat, K. A. (2010). A game theoretic framework for power control in wireless sensor networks. *IEEE Transactions on Computers, 59*(2), 231–42.
6. Anamika, S., & Siddhartha, C. (2020). A distributed reinforcement learning based sensor node scheduling algorithm for coverage and connectivity maintenance in wireless sensor network. *Wireless Networks, 26,* 4411–4429.
7. Mu, Q., Haitao, Z., Shengchun, H., Li, Z., & Shan, W. (2017). Optimal channel selection based on online decision and offline learning in multichannel wireless sensor networks. *Wireless Communications and Mobile Computing*. https://doi.org/10.1155/2017/7902579.
8. Rehan, W., Fischer, S., & Rehan, M. (2016). Machine-learning based channel quality and stability estimation for stream-based multichannel wireless sensor networks. *Journal of Sensors (Basel)*. https://doi.org/10.3390/s16091476.

9. Kim, Y., Shin, H., & Cha, H.(2008). Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In: Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN08), St. Louis, MO, USA.
10. Abdul Hamid, Md., Abdullah-Al-Wadud, M., & Ilyoung, C. (2010). A schedule-based multi-channel MAC protocol for wireless sensor networks. *Sensors (Basel), 10*(10), 9466–9480.
11. Ozlem, D. I., Lodewijk, V. H., Pierre, J., & Paul, H. (2011). MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Networks, 9*(1), 73–94.
12. Chen, J., Yu, Q., Chai, B., Sun, Y., Fan, Y., & Shen, X. (2015). Dynamic channel assignment for wireless sensor networks: A regret matching based approach. *IEEE Transactions on Parallel and Distributed Systems, 26*(1), 95–1068.
13. Panos, NA., Efthymios, AV., & Aggelos, B. (2017). Inference-based distributed channel allocation in wireless sensor Networks. Journal of Information Theory (cs.IT), CoRR abs/1703.06652.
14. Hao, X. C., Gong, Q. Q., Hou, S., & Liu, B. (2015). Energy efficient based channel assignment game algorithm for wireless sensor networks. *Wireless Personal Communications, 85*(4), 2749–2771.
15. Mihaylov, M., Borgne, Y. A., Tuyls, K., & Nowé, A. (2012). Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks. *International Journal of Communication Networks and Distributed Systems, 9,* 207–224.
16. Khan, M. I., Kia, K., Ali, A., & Aslam, N. (2017). Energy-aware task scheduling by a true online reinforcement learning in wireless sensor networks. *International Journal of Sensor Networks, 25*(4), 244–258.
17. Selvia, F. L. D., & Costa, A. H. R. (2019). A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artifitial Intelligence Research, 64,* 645–703.
18. Khan, M., & Rinner, B. (2014). Energy-aware task scheduling in wireless sensor networks based on cooperative reinforcement learning. Proceedings of the IEEE International Conference on Communications Workshops, Sydney, Australia, pp. 871–877.
19. Yujia, G., Yurong, N., & Xiahai, G. (2021). Maximizing network throughput by cooperative reinforcement learning in clustered solar-powered wireless sensor networks. *International Journal of Distributed*. https://doi.org/10.1177/15501477211007411.
20. Pal, A., & Nasipuri, . (2017). A distributed routing and channel selection for multi-channel wireless sensor networks. *Journal of Sensor and Actuator Networks, 6*(3), 1–13.
21. Ibrahim, M., Borhanuddin, M. A., Sali, A., Rasid, M. F. A., & Mohamad, H. (2017). An energy efficient reinforcement learning based cooperative channel sensing for cognitive radio sensor networks. *Pervasive and Mobile Computing, 35,* 165–184.
22. Xuan, T., Li, Z., Haijun, W., Yuli, S., Haitao, Z., Boon-Chong, S., Jibo, W., & Vector, C.M L. (2021). Cooperative Multi-Agent Reinforcement Learning Based Distributed Dynamic Spectrum Access in Cognitive Radio Networks. arXiv:2106.09274.
23. Yao, F., & Jia, L. (2019). A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks. *IEEE Wireless Communications Letters, 8*(4), 1024–1027.
24. Kulshrestha, J., & Mishra, M. K. (2017). An adaptive energy balanced and energy eficient approach for data gathering in wireless sensor networks. *Ad Hoc Networks, 54,* 130–146.
25. IEEE 802.15 WPAN Task Group, IEEE Std 802.15.3TM (2006) Wireless Medium Acces Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN). https://profsite.um.ac.ir/hyaghmae/ACN/WSNMAC1.pdf
26. Morozs, N., Clarke, T., & Grace, D. (2013). A novel adaptive call admission control scheme for distributed reinforcement learning based dynamic spectrum access in cellular networks. Proc. Int. Symp. Wireless Commun. Syst. (ISWCS), pp. 1–5.
27. Bowlingand, M., & Veloso, M. (2002). Multi agent learning using a variable learning rate. *Artificial Intelligence, 136*(2), 215–250.
28. Xuan, P., Hung, ML., David, FS., & Ara, N. (2018). Cooperative and distributed reinforcement learning of drones for field coverage. arXiv:1803.07250.
29. HPC Lab (2016). JSensor: high performance computing, wireless sensor simulator. https://joubertlimadotcomdotbr.wordpress.com/jsensor-a-high-performance-java-simulator-for-sensor-networks/
30. Gharbi, C., Aliouat, Z., & Benmohammed, M. (2018). A novel load balancing scheduling algorithm for wireless sensor networks. *Journal of Network and Systems Management, 27*(2), 430–462.

Mohamed Sahraoui is currently an Assistant Professor at the University of Msila, Algeria and a researcher at the laboratory LIAM. He received his Master degree in 2004 in computer science from High School of Computer science (ESI : Ecole Superieure d'informatique), Algeria, and the magister dugree in Informatics systems from Batna university in 2011. He is a member of the research group: "Emerging networks and distributed systems". His research interests are wireless communication, WSN and IoT applications.

Azeddine Bilami is currently serving as a Professor (full position) at the department of computer science, and the Head of LaSTIC laboratory, at University of Batna2, Prof. Bilami authored publications in many international journals including IJCA (Actapress), IJSNet (Inderscience), IEEE Communications Letters, Springer Verlag, IGI Global, Elsevier. He acts also as a reviewer for many journals (COMNET, COMCOM , TIIS , Journal of King Saud University, etc.). His research interests include IOT applications, WSN, frameworks, middleware, M2M, communication protocols, cloud computing, security, QOS, artificial intelligence, big data, etc.

Abdelmalik Taleb-Ahmed received the Maitrise and DEA degrees in electronics in 1987 and 1989, respectively, and the Doctorate degree in signal processing and image processing from the Universityof Sciences and Technologie of Lille, France, in 1992.From 1992 to 2004, he was a Professor at the University of Sciences and Technologie of Lille. He is currently a full Professor at IEMN DOAE Laboratory, Hauts de France Polytechnic University,Valenciennes, France. His research interests include digital signal processing and biomedical image processing.

Authors and Affiliations

Mohamed Sahraoui[1] · Azeddine Bilami[2] · Abdelmalik Taleb-Ahmed[3]

Azeddine Bilami
az.bilami@univ-batna2.dz

Abdelmalik Taleb-Ahmed
taleb@uphf.fr

[1] Mohamed Khider University, LIAM Laboratory, M'sila, Algeria

[2] LaSTIC Laboratory of Batna2 University, Batna, Algeria

[3] IEMN DOAE Laboratory, Hauts de France Polytechnic University, Valenciennes, France