Al-Tous, Hanan; Barhumi, Imad

Reinforcement Learning Framework for Delay Sensitive Energy Harvesting Wireless Sensor Networks

Published in:
IEEE Sensors Journal

DOI:
10.1109/JSEN.2020.3044049

Published: 01/03/2021

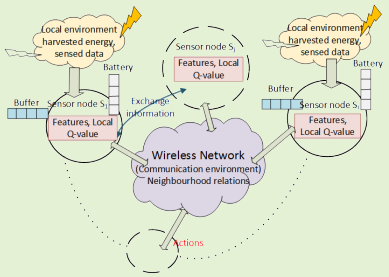
Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Al-Tous, H., & Barhumi, I. (2021). Reinforcement Learning Framework for Delay Sensitive Energy Harvesting Wireless Sensor Networks. *IEEE Sensors Journal*, *21*(5), 7103-7113. Article 9292079. https://doi.org/10.1109/JSEN.2020.3044049

Reinforcement Learning Framework for Delay Sensitive Energy Harvesting Wireless Sensor Networks

Hanan Al-Tous, *Senior Member, IEEE*, and Imad Barhumi, *Senior Member, IEEE*

***Abstract*— A multi-hop energy harvesting wireless sensor network (EH-WSNs) is a key enabler for future communication systems such as the internet-of-things. Optimal power management and routing selection are important for the operation and successful deployment of EH-WSNs. Characterizing the optimal policies increases significantly with the number of nodes in the network. In this paper, optimal control policy is devised based on minimum-delay transmission in a multi-hop EH-WSN using reinforcement learning (RL). The WSN consists of M EH sensor nodes aiming to transmit their data to a sink node with a minimum delay. Each sensor node is equipped with a battery of limited capacity to save the harvested energy and a data buffer of limited size to store both the sensed and relayed data from neighboring nodes. Centralized and distributed RL algorithms are considered for EH-WSNs. In the centralized RL algorithm the control action is taken at a central unit using the state information of all sensor nodes. In the distributed RL algorithm the control action is taken locally at each sensor node using its state of information and the state information of neighboring nodes. The proposed RL algorithms are based on the state-action-reward-state-action (SARSA) algorithm. Simulation results demonstrate the merits of the proposed algorithms.**



***Index Terms*— Wireless sensor network, energy harvesting, reinforcement learning, SARSA, action-value-function approximation.**

I. Introduction

Wireless sensor networks (WSNs) are autonomous networks of distributed sensor nodes that are communicating with each other wirelessly in a multi-hop fashion. WSN has been identified as one of the major technologies for future wireless communication systems such as the internet-of-things (IoT) and green fifth generation (5G) and beyond 5G (B5G) networks. A crucial characteristic of WSNs is to have a very long network lifetime span, since human intervention for battery replenishment may not be possible in inaccessible locations [1]–[3].

Energy harvesting wireless sensor nodes are usually equipped with limited-capacity energy storage, and limited buffer size. To overcome these constraints in EH-WSNs, optimal resource allocation techniques are crucial for the deployment and operation of EH-WSNs [4]–[7]. Several resource allocation problems for EH sources were studied aiming to achieve different objectives such as throughput and reward maximization. Different solution approaches are proposed using offline, online optimizations and game theory as in [8]–[17].

EH-WSNs can be deployed to monitor a physical space (field) to improve the ecosystem and human life. EH-WSNs gained increasing popularity for a range of applications for environmental monitoring, including air quality monitoring, water quality monitoring, and disaster monitoring such as earthquakes, hurricanes, and floods, as well as health monitoring of civil structures such as bridges, buildings, transportation infrastructures and smart cities monitoring applications [18], [19].

Resource allocation in EH-WSNs can be modeled as a Markov decision process (MDP), where the future state is only dependent on the current state. Classical Dynamic programming (DP) can be used to obtain optimal policies for MDP systems. However, DP suffers from the curse of dimensionality. In addition, DP requires exact knowledge of the transition probabilities, which are often hard to obtain in practical systems. Reinforcement learning (RL) algorithms can, to a great extent, alleviate the dimensionality problem and obtain near optimal solutions without knowing the precise values of the transition probabilities [20], [21].

In general, the complexity in characterizing the optimal policies increases significantly with the number of nodes in the network. The transmission policy of a sensor node affects the data arrivals at the next-hop node, hence, couples the optimal transmission scheme across the network. Dynamic resource allocation in EH-WSNs is considered in this paper. Each sensor node is equipped with an energy harvesting device, a finite energy storage and a finite data buffer. Based on the

H. Al-Tous is with the Department of Communications and Networking, Aalto University, Espoo, Finland (e-mail: hanan.al-tous@aalto.fi).

I. Barhumi is with the Department of Electrical Engineering, United Arab Emirates University, Al Ain, UAE (e-mail: imad.barhumi@uaeu.ac.ae).

harvested energy, the channel state information, the buffer state, the energy level of the sensor node and neighboring nodes, an adaptive policy is proposed. The objective of the proposed policy is to minimize the overall delay in the EH-WSN. Minimum delay is important for many future IoT applications, such as: detection, estimation and video streaming [22]. In a multi-hop WSN, the control policy consists of power allocation and route (next-hop) selection. Two RL algorithms are considered to find the optimal control policy based on the action value-function approach. A centralized RL algorithm, where the state information of all sensor nodes are known at a central unit, and a distributed RL algorithm, where the state information of the sensor node and only neighboring nodes are used to learn the optimal policy locally.

A. Related Work and Contribution

In this subsection we survey the literature related to EH-WSNs and present the main contributions of this paper.

In [22], asymptotically optimal low-complexity power control is proposed for delay-aware resource allocation in point-to-point EH wireless system. Energy harvesting for delay-limited point-to-point wireless communication is considered in [23]. The transmitter is equipped with a finite-capacity rechargeable battery. Q-learning framework is used to determine the transmission policy assuming finite state and action spaces. In [24], optimal selective transmission policy for single link EH-WSN is proposed using monotone neural network. Based on the channel state information, the battery status, and the packet priority, the node adapts its selective transmission policy. RL for EH decode-and-forward (DF) two-hop communication is considered in [25]. The transmission policy aims to maximize the network throughput where the source and relay nodes are assumed to have only local information. The two-hop joint power allocation problem is separated into two point-to-point power allocation problems. In [26], the optimal transmission policy is formulated to minimize the symbol-error rate in EH DF relay network. Optimal transmission policy in EH cooperative two-hop multi-relay communication is proposed in [27]. The problem is formulated as a partially observable stochastic game, where the power control is obtained locally at each relay node using RL. In [28], the authors proposed a novel energy management algorithm to maximize the packet rate for point-to-point communication based on the actor-critic RL framework. In [29], a distributed multi-agent RL algorithm is proposed based on an identical reward function for all nodes.

In the aforementioned work, the resource allocation and the corresponding power control are proposed for point-to-point EH communication. The formulated resource allocation problems may not be directly applied to EH-WSNs, since each sensor node has his own data in addition to the relayed data from its neighbors. In addition, the action of one sensor node may affect the actions of the other nodes. The main contributions of this paper are summarized as follows:

- A multi-hop connected EH-WSN is considered. The routing is done based on graph structure, and all sensor nodes are equipped with limited buffer capacity and limited battery storage. The sensor nodes relay their own sensed data and the relayed data from neighboring nodes.

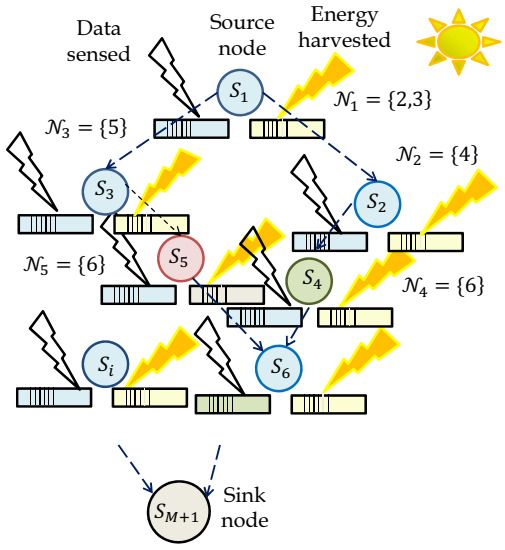


Fig. 1. EH-WSN System model.

- The optimal policy consists of optimal power control and route (next-hop) selection. The cost function is formulated aiming to minimize the average delay.
- Two RL algorithms are proposed: Centralized and distributed. Linear function approximation with binary features is used to approximate the Q-value function. The binary features are devised to capture the system's dynamics and constraints.
- In the distributed RL algorithm, the sensor nodes exchange with their neighbors the difference in their Q-value, and their buffer states.
- The findings are confirmed with numerical results. The performance of the proposed RL algorithms are compared with the offline resource allocation.
- To the authors best of knowledge, this is the first time that RL framework is proposed for the online operation of EH-WSNs[1].

The remainder of this paper is organized as follows. In Section II, the system model of EH-WSN is introduced. In Section III, the problem formulation, SARSA and the D-SARSA algorithms are presented. Numerical results are presented and discussed in Section IV. Finally, conclusions are drawn in Section V.

II. System Model

The system under consideration is shown in Fig. 1. The sender (source) node S_i for $i = 1, \ldots, M$ aims to transmit data to a sink node S_{M+1} using multi-hop communication. The routing protocol is based on a connected graph structure. The graph structure captures the organization of sensor nodes based on their distance from each other. It can be constructed in the setup phase based on the average received power, if location information is not available. Sensor node S_i communicates with its neighboring set denoted as $\mathcal{N}_i$ using a single-hop transmission. The set $\mathcal{N}_i$ consists of the one-hop neighbors

[1]Preliminary results of this work have been accepted as work in progress in IEEE BlackSeaCom 2019 conference [30].

of node S_i that serve as the next-hop towards the sink node. Transmission is organized in time-slots of fixed duration T over K time-slots with $K \to \infty$. The data rate and power are assumed to be fixed at each time-slot. Orthogonal multiple access to the medium is assumed, where only non-interfering links can transmit simultaneously. A control channel can be used to coordinate nodes' transmission. Orthogonal multiple access to the medium is nearly optimal when interference is strong [31]. Each sensor node is assumed to transmit to only one of its neighbors at each time-slot, this is to simplify the action selection of the proposed RL algorithm as discussed in Section III.

No energy loss is assumed during the harvesting time-slots, and the harvested energy at time-slot t can be used for transmission at time-slot $t+1$. Sensor node S_i is equipped with a limited battery storage $E_{\max}^{(i)}$ units of energy, and limited buffer capacity $C_{\max}^{(i)}$ in bits. We have assumed that the control messages will occupy a fixed amount of memory that is different from the data buffer.

At each time-slot t, the transmitter has to guarantee that the energy spent is not greater than the available energy in the battery. Hence, the state dynamics of the energy level at sensor node S_i for $i = 1, \ldots, M$, can expressed as:

$$E_{t+1}^{(i)} = \min\{E_t^{(i)} - T(\sum_{j\in\mathcal{N}_i} Y_t^{(i,j)} P_t^{(i,j)} + P_c^{(i)}) + H_t^{(i)},\ E_{\max}^{(i)}\}, \tag{1}$$

where $E_t^{(i)}$ is the energy level, and $Y_t^{(i,j)} \in \{0,1\}$ is the relay selection indicator of sensor node S_i with $\sum_{j\in\mathcal{N}_i} Y_t^{(i,j)} \leq 1$. $P_t^{(i,j)} \geq 0$ denotes the amount of power used to transmit $r_t^{(i,j)}$ bits from sensor node S_i to sensor node S_j for $j \in \mathcal{N}_i$, which is kept constant during time-slot t. The processing cost $P_c^{(i)}$ represents the sum of all other power consumption of sensor node S_i [32], and $H_t^{(i)}$ is the harvested energy. Since the harvested energy $H_t^{(i)}$ needs to be stored in the battery before being used during time-slot t, the transmit power feasibility constraint can be stated as:

$$\sum_{j\in\mathcal{N}_i} Y_t^{(i,j)} P_t^{(i,j)} + P_c^{(i)} \leq E_t^{(i)}/T. \tag{2}$$

Similarly, at each time-slot t the sensor nodes must guarantee that there is enough buffer space before receiving data from neighboring nodes. Hence, the state dynamic of the data buffer of sensor node S_i can be expressed as:

$$C_{t+1}^{(i)} = \min\{C_t^{(i)} - \sum_{j\in\mathcal{N}_i} Y_t^{(i,j)} r_t^{(i,j)} + \sum_{j|_{i\in\mathcal{N}_j}} Y_t^{(j,i)} r_t^{(j,i)} + d_t^{(i)}, C_{\max}^{(i)}\}, \tag{3}$$

where $C_t^{(i)}$ is the data buffer level in bits, and $d_t^{(i)}$ is the data sensed by source node S_i during time-slot t. The sink buffer is assumed to have a very large capacity compared to the buffer of source nodes S_i for $i = 1, \ldots, M$, i.e., $C_{\max}^{(M+1)} >> C_{\max}^{(i)}$.

The data transmitted from sensor node S_i to a neighboring node S_j for $j \in \mathcal{N}_i$ are constrained by the channel capacity of the link as:

$$r_t^{(i,j)} \leq TW \log_2\left(1 + \gamma_t^{(i,j)} P_t^{(i,j)}\right), \tag{4}$$

where $\gamma_t^{(i,j)} = \frac{|h_t^{(i,j)}|^2}{\sigma^2}$ with $h_t^{(i,j)}$ is the fading channel coefficient between nodes S_i and S_j at time-slot t, σ^2 is the noise power at the receiver node S_j, and W is the channel bandwidth. Since the received data need to be stored in the buffer before being transmitted, the data transmission feasibility constraint at sensor node S_i can be expressed as:

$$\sum_{j\in\mathcal{N}_i} Y_t^{(i,j)} r_t^{(i,j)} \leq C_t^{(i)}. \tag{5}$$

III. Problem Formulation

The main objective of the considered dynamic resource allocation problem is to minimize the delay subject to limited data buffer capacity and battery storage constraints. Several scenarios have been considered in the paper: First, a centralized offline minimum delay problem is formulated assuming non-causal knowledge of the harvested energy, sensed data, and channel gains. Second, a centralized online minimum delay problem assuming causal information. The optimal power allocation, and next-hop selection policy is obtained using RL based SARSA algorithm. The optimal policy is computed at a central unit and then disseminated to all sensor nodes. Third, a distributed online minimum delay problem is formulated assuming causal information. The optimal power allocation and next-hop selection policy is obtained locally at each sensor node using RL based distributed SARSA (D-SARSA) algorithm.

In the centralized offline problem, the minimum delay transmission policy is obtained for a finite number of time-slots over a realization of the stochastic processes (i.e., data/energy arrivals and channel gains). Whereas in the online scenarios, the minimum delay transmission policy is obtained over an infinite horizon. The average performance of the offline centralized scenario can be considered as a benchmark for the performance of the proposed centralized and distributed online scenarios as in [23].

In the following, we present the offline resource allocation problem, the basic concepts of MDPs, action-value-function approximation, SARSA and D-SARSA algorithms.

A. Offline Resource Allocation

In the offline resource allocation, all future data/energy arrivals of all sensor nodes and the channel gains of all links are known non-causally at a central unit before transmission. Offline optimization is relevant in applications for which the stochastic processes can be estimated accurately in advance at a central unit [7].

The offline resource allocation problem aiming to minimize the delay can be formulated based on Little's law. Since the delay is directly related to the amount of data stored in the sensor nodes' buffers [33], the objective function is formulated as the sum of the empty spaces of all buffers in bits. The

optimization problem can then be expressed as:

$$\max_{\boldsymbol{P},\boldsymbol{Y},\boldsymbol{r}} \sum_{i=1}^{M}\sum_{t=0}^{K-1}\big(C_{\max}^{(i)}-C_{t+1}^{(i)}\big), \tag{6a}$$

subject to:

$$\text{given } E_0^{(i)}\&C_0^{(i)}, \text{ for } i=1,\dots,M, \tag{6b}$$

$$(1)\text{-}(5), \text{ for } i=1,\dots,M,\ t=0,\dots,K-1, \tag{6c}$$

$$P_t^{(i,j)}\geq 0,\ r_t^{(i,j)}\geq 0,\ i=1,\dots,M,\ j\in\mathcal{N}_i,\ t=0,\dots,K-1, \tag{6d}$$

$$C_t^{(i)}\geq 0,\ E_t^{(i)}\geq 0,\ i=1,\dots,M,\ t=1,\dots,K, \tag{6e}$$

$$\sum_{j\in\mathcal{N}_i} Y_t^{(i,j)}\leq 1,\ i=1,\dots,M,\ t=0,\dots,K-1, \tag{6f}$$

$$Y_t^{(i,j)}\in\{0,1\},\ i=1,\dots,M,\ j\in\mathcal{N}_i,\ t=0,\dots,K-1, \tag{6g}$$

where $\boldsymbol{P}$, $\boldsymbol{Y}$ and $\boldsymbol{r}$ denote the power allocation, relay selection indicator and data transmitted vectors, respectively. The power allocation vector $\boldsymbol{P}=[\boldsymbol{P}_0^{(1)},\dots,\boldsymbol{P}_{K-1}^{(M)}]$ where $\boldsymbol{P}_t^{(i)}=[P_t^{(i,j)}|_{\forall j\in\mathcal{N}_i}]$ represents the power allocation vector of sensor node S_i at time-slot t. The relay selection vector $\boldsymbol{Y}=[\boldsymbol{Y}_0^{(1)},\dots,\boldsymbol{Y}_{K-1}^{(M)}]$ with $\boldsymbol{Y}_t^{(i)}=[Y_t^{(i,j)}|_{\forall j\in\mathcal{N}_i}]$ represents the relay selection indicator vector of sensor node S_i at time-slot t. The vector $\boldsymbol{r}=[\boldsymbol{r}_0^{(1)},\dots,\boldsymbol{r}_{K-1}^{(M)}]$ with $\boldsymbol{r}_t^{(i)}=[r_t^{(i,j)}|_{\forall j\in\mathcal{N}_i}]$ represents the data transmitted vector of sensor node S_i at time-slot t. Constraint (6c), represents the buffer and battery dynamics, causality and feasibility constraints as in [34]. The initial energy and buffer states of sensor node S_i denoted as $E_0^{(i)}$ and $C_0^{(i)}$, respectively, are assumed given.

Problem (6) is a mixed integer non-convex optimization problem that is difficult to solve. An upper bound of the optimal value can be obtained by relaxing the integer variables, such that $Y_t^{(i,j)}\in[0,1]$, for $i=1,\dots,M$, $j\in\mathcal{N}_i$, and $t=0,\dots,K-1$. The variable transformation technique can then be used to convert (1), (2), (3) and (5) to linear constraints as in [35]–[37]. Therefore, the relaxed problem is transformed to a convex optimization problem, that can be easily solved using different convex optimization techniques such as the interior point method [35].

B. MDP and RL Basic Concepts

A MDP is defined by the 4-tuple $\mathcal{M}=(\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{A}},\mathcal{R},\mathcal{P})$, compromised of the state space $\boldsymbol{\mathcal{X}}$, action space $\boldsymbol{\mathcal{A}}$, reward function $\mathcal{R}$ and the transition probabilities $\mathcal{P}$. A deterministic policy π is a rule to select actions given a current state, $\pi:\boldsymbol{\mathcal{X}}\rightarrow\boldsymbol{\mathcal{A}}$ [20].

Typical objective functions of MDPs are the expected discounted reward and the average reward per stage [20]. In this paper, the objective function is selected as the average reward. The average reward per stage fits the delay performance much better than the discounted reward as explained in [27]. The per stage average reward is expressed as:

$$\eta^{\pi}(\boldsymbol{x})=\lim_{K\rightarrow\infty}\frac{1}{K}\mathbb{E}\big[\sum_{t=0}^{K-1}R_{t+1}|\boldsymbol{x}_0=\boldsymbol{x},\pi\big], \tag{7}$$

where $\boldsymbol{x}_0$ is the starting state, R_t is the reward function at time-slot t, and $\mathbb{E}$ is the expectation operator. When the Markov chain resulting from applying every stationary policy is recurrent or ergodic, the optimal average reward per stage is independent of the initial state $\boldsymbol{x}_0$ [38], i.e., $\eta^{\pi}(\boldsymbol{x})=\eta^{\pi}$.

The (action)-value-function is used with discounted reward objective functions, whereas, differential (action)-value-function is used with the average reward per stage objective functions [20]. Hence, the differential value function $V^{\pi}(\boldsymbol{x})$ is defined as the sum of the differential rewards starting at state $\boldsymbol{x}$, and following the policy π, thereafter, i.e., $V^{\pi}(\boldsymbol{x})=\mathbb{E}\big[\sum_{t=0}^{\infty}R_{t+1}-\eta^{\pi}|\boldsymbol{x}_0=\boldsymbol{x},\pi\big]$. Similarly, the differential-action-value-function $Q^{\pi}(\boldsymbol{x},\boldsymbol{a})$ is the expected sum of the differential rewards starting at state $\boldsymbol{x}$, taking action $\boldsymbol{a}$ and then following the policy π thereafter, i.e., $Q^{\pi}(\boldsymbol{x},\boldsymbol{a})=\mathbb{E}\big[\sum_{t=0}^{\infty}R_{t+1}-\eta^{\pi}|\boldsymbol{x}_0=\boldsymbol{x},\boldsymbol{a}_0=\boldsymbol{a},\pi\big]$ [20], [38]. The solution of a MDP is obtained by finding the optimal policy π^* that maximizes the objective/value function, i.e., $\pi^*=\arg\max_{\pi} V^{\pi}(\boldsymbol{x})$, $\forall\boldsymbol{x}\in\boldsymbol{\mathcal{X}}$. For finite state and action spaces and assuming full knowledge of transition probabilities, the optimal policy π^* can be obtained using value iteration or policy iteration [20].

Definition An EH-WSN consisting of M sensor nodes is defined at time-slot t by: the state vector of all sensor nodes $\boldsymbol{x}_t=[\boldsymbol{x}_t^{(1)T},\dots,\boldsymbol{x}_t^{(i)T},\dots,\boldsymbol{x}_t^{(M)T}]^T$ with $\boldsymbol{x}_t^{(i)}=[E_t^{(i)},H_t^{(i)},C_t^{(i)},d_t^{(i)},h_t^{(i,j)}|_{\forall j\in\mathcal{N}_i}]^T$, and the action vector of all sensor nodes $\boldsymbol{a}_t=[\boldsymbol{a}_t^{(1)T},\dots,\boldsymbol{a}_t^{(i)T},\dots,\boldsymbol{a}_t^{(M)T}]^T$, the action $\boldsymbol{a}_t^{(i)}$ of sensor node S_i is defined by the transmit power and the relay selection indicator that is used for the next-hop transmission, i.e., $\boldsymbol{a}_t^{(i)}=[P_t^{(i)},Y_t^{(i,j)}|_{\forall j\in\mathcal{N}_i}]^T$ with $P_t^{(i,j)}=Y_t^{(i,j)}P_t^{(i)}$, $Y_t^{(i,j)}\in\{0,1\}$, and $\sum_{j\in\mathcal{N}_i}Y_t^{(i,j)}\leq 1$. A finite set of power actions is considered at sensor node S_i for $i=1,\dots,M$ by discretizing the transmit power, i.e., $P_t^{(i)}\in\{0,\delta,2\delta,\dots,E_{\max}^{(i)}/T\}$, where δ is the step size. The transmitted data $r_t^{(i,j)}$ from sensor node S_i to a neighboring node S_j is computed as $r_t^{(i,j)}=TW\log_2(1+\gamma_t^{(i,j)}P_t^{(i)})$. The reward function R_{t+1} is defined as: $R_{t+1}=\sum_{i=1}^{M}C_{\max}^{(i)}-C_{t+1}^{(i)}$.

The harvested energy and sensed data arrivals, and the channel gains are assumed to change according to independent probabilistic distributions. The sensed data and harvested energy arrivals $d_t^{(i)}$ and $H_t^{(i)}$, respectively, are assumed to be known at time-slot t, but could not be used before time-slot $t+1$.

In EH-WSNs, the state space is infinite and transition probabilities may not be easy to obtain. In this sense, model-free RL algorithm is proposed to devise the optimal transmission policy without the need to know the transition probabilities. Furthermore, action-value-function approximations can be used to deal efficiently with the infinite state space.

C. Q-value Linear Approximation

To deal with continuous state spaces, Q-value approximation techniques are used. Linear-function-approximation is widely used because of their simplicity and mathematical tractability [20]. Since the battery and buffer states of sensor node S_i

evolve based on its own state and action and the actions of neighboring nodes, the Q-value linear-function-approximation of the EH-WSN is defined as:

$$\hat{Q}^{\pi}(\boldsymbol{x}_t, \boldsymbol{a}_t; \boldsymbol{\theta}) = \sum_{i=1}^{M} \boldsymbol{\Phi}^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t)\, \boldsymbol{\theta}^{(i)}, \tag{8}$$

where $\boldsymbol{\Phi}^{(i)}(\cdot) = [\Phi_1^{(i)}(\cdot), \ldots, \Phi_p^{(i)}(\cdot)]$ is a vector of p features, and $\boldsymbol{\theta}^{(i)} = [\theta_1^{(i)}, \ldots, \theta_p^{(i)}]^T$ is a vector of p parameters (weights). Devising feature functions based on the problem/solution structure is more efficient than using general approximation techniques as shown in [20], [25]. The Q-value linear approximation using the standard fixed-sparse-representation (FSR) technique is presented in the Appendix. In general, the state and action constraints can be handled either using the reward function or the feature functions. In this paper, binary feature functions are used to handle the constraints as in [25]. The proposed feature functions using sensor node S_i state vector $\boldsymbol{x}_t^{(i)}$ are divided into two groups: In one group, the feature is a function of the sensor node action vector $\boldsymbol{a}_t^{(i)}$ and is not affected by the actions of other sensor nodes, whereas in another group, the feature is a function of the action vector $\boldsymbol{a}_t$ of all sensor nodes. Given a state vector $\boldsymbol{x}_i^{(t)}$ for $i = 1, \ldots, M$, the features $\Phi_1^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ to $\Phi_6^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ belong to the first group of features, whereas, the feature $\Phi_7^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t)$ belongs to the second group as explained next.

The first feature function $\Phi_1^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ deals with the transmit power feasibility constraint and the battery capacity constraint of sensor node S_i given as [25]:

$$\Phi_1^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, & \text{if } E_t^{(i)} - T(P_t^{(i)} + P_c^{(i)}) + H_t^{(i)} \le E_{\max}^{(i)} \\ & \quad \cap\, T(P_t^{(i)} + P_c^{(i)}) \le E_t^{(i)}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

This feature is active when the transmit power (action) of sensor node S_i satisfies the feasibility and battery capacity constraints.

The second feature is developed based on the idea of model predictive control, where sensor node S_i plans for a horizon of several time-slots but applies the optimal action at the first time-slot. For simplicity and based on the available state information, the horizon length is chosen to be two time-slots. The average value of the channel gain is used in the second-time slot assuming the same relay link is used. Sensor node S_i maximizes the sum of the data transmitted over two successive time-slots. Therefore, the optimal power allocation $\rho_t^{(i,j)}$ can be computed as:

$$\rho_t^{(i,j)} = \begin{cases} \min\left\{(\frac{1}{\lambda_t^{(i)}} - \frac{1}{\gamma_t^{(i,j)}})^+, \frac{E_t^{(i)}}{T} - P_c^{(i)}\right\} & \text{if } Y_t^{(i,j)} = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

where $(x)^+ = \max(x, 0)$, and the water-level $\lambda_t^{(i)}$ is computed as:

$$\lambda_t^{(i)} = \Big(\frac{1}{2}\Big[\frac{E_t^{(i)} + H_t^{(i)}}{T} - P_c^{(i)} + \frac{1}{\gamma_t^{(i,j)}} + \frac{1}{\overline{\gamma}_t^{(i,j)}}\Big]\Big)^{-1}, \tag{11}$$

where $\overline{\gamma}_t^{(i,j)} = \frac{\sum_{l=0}^{t} \gamma_l^{(i,j)}}{t+1}$ is the mean-value estimated using past channel realizations. The second feature function $\Phi_2^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ can be expressed using optimal power allocation $\rho_t^{(i,j)}$ as:

$$\Phi_2^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, & \text{if } \sum_{j \in \mathcal{N}_i} \delta \left\lfloor \frac{\rho_t^{(i,j)}}{\delta} \right\rfloor = P_i^{(t)}, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

where $\lfloor y \rfloor$ is the integer part of y. A third feature function $\Phi_3^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ can be devised based on battery overflow. Sensor node S_i is encouraged to transmit with maximum available power when the harvested energy at time-slot t is larger than the battery capacity. Hence, $\Phi_3^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ can be expressed as [25]:

$$\Phi_3^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, & \text{if } H_t^{(i)} + E_t^{(i)} - TP_c^{(i)} \ge E_{\max}^{(i)} \\ & \quad \cap\, \delta \left\lfloor \frac{E_t^{(i)} - TP_c^{(i)}}{T\delta} \right\rfloor = P_t^{(i)} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

The communication of sensor node S_i with the sink node S_{M+1} for $M+1 \in \mathcal{N}_i$ using maximum available energy may reduce the delay at the buffer of sensor node S_i. In this regard, feature function $\Phi_4^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ can be defined as:

$$\Phi_4^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, \text{ if } & M+1 \in \mathcal{N}_i \cap Y_t^{(i,M+1)} = 1 \\ & \quad \cap\, P_t^{(i)} = \delta \lfloor \frac{E_t^{(i)} - TP_c^{(i)}}{T\delta} \rfloor, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The next-hop that maximizes the data rate should be selected. This feature is captured by the function $\Phi_5^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ expressed as:

$$\Phi_5^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, & \text{if } \sum_{j \in \mathcal{N}_i} Y_t^{(i,j)} r_t^{(i,j)} \ge \max\limits_{k \in \mathcal{N}_i} r_t^{(i,k)}, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The data transmission feasibility constraint of sensor node S_i is captured by feature function $\Phi_6^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})$ defined as [25]:

$$\Phi_6^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, & \text{if } \sum_{j \in \mathcal{N}_i} Y_t^{(i,j)} r_t^{(i,j)} \le C_t^{(i)}, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The buffer feasibility constraint is handled using feature function $\Phi_7^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t)$ given as:

$$\Phi_7^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t) = \begin{cases} 1, & \text{if } C_t^{(i)} + d_t^{(i)} + \sum_{j|_{i \in \mathcal{N}_j}} Y_t^{(j,i)} r_t^{(j,i)} \\ & \quad - \sum_{j \in \mathcal{N}_i} Y_t^{(i,j)} r_t^{(i,j)} \le C_{\max}^{(i)}. \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

This feature function is affected by the actions of other nodes.

Algorithm 1 SARSA Algorithm

Set $t = 0$.
Given α_t, β_t, and ϵ_t.
Initialize $\overline{h}_t^{(i,j)}$, $\forall j \in \mathcal{N}_i$ and $i = 1, \ldots, M$.
Initialize $\boldsymbol{\theta}_t$ randomly, and $\eta_t^{\pi} = 0$.
Observe the state $\boldsymbol{x}_t$.
Select the action $\boldsymbol{a}_t$ according to the ϵ_t-greedy policy.
Apply the action $\boldsymbol{a}_t$.
while Stopping criterion is not satisfied **do**
  for $t = 0, \ldots, K-1$ **do**
    Measure the reward R_{t+1}.
    Observe the next state $\boldsymbol{x}_{t+1}$.
    Select the action $\boldsymbol{a}_{t+1}$ according to the
      ϵ_{t+1}-greedy policy.
    Update $\boldsymbol{\theta}_t$ and η_t^{π} using (19) and (22), respectively.
    Apply the action $\boldsymbol{a}_{t+1}$.
  end for
end while

D. SARSA Algorithm

SARSA Algorithm is an incremental centralized (single-agent) online-policy learning algorithm for (near) optimal control. It estimates the Q-value-function from the states that are visited and earned rewards [20], [21]. The action vector $\boldsymbol{a}_t$ at time-slot t is selected using an ϵ_t-greedy policy, where at time-slot t, the action $\boldsymbol{a}_t$ is selected randomly from the action space $\boldsymbol{\mathcal{A}}$ with probability ϵ_t and greedy with probability $1-\epsilon_t$ as:

$$\boldsymbol{a}_t = \arg\max_{\boldsymbol{A} \in \boldsymbol{\mathcal{A}}} \hat{Q}^{\pi}(\boldsymbol{x}_t, \boldsymbol{A}; \boldsymbol{\theta}_t). \tag{18}$$

The exploration probability ϵ_t decreases with the learning-time in order to find the optimal (greedy) policy [20]. SARSA algorithm uses gradient-descent to update the parameters vector $\boldsymbol{\theta}$ as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \delta_t \nabla_{\boldsymbol{\theta}} \hat{Q}^{\pi}(\boldsymbol{x}_t, \boldsymbol{a}_t; \boldsymbol{\theta}_t), \tag{19}$$

where α_t is the learning rate that needs to satisfy the Robbins-Monro conditions [21], δ_t is the temporal difference (TD) error computed as [20]:

$$\delta_t = R_{t+1} - \eta_t^{\pi} + \hat{Q}^{\pi}(\boldsymbol{x}_{t+1}, \boldsymbol{a}_{t+1}; \boldsymbol{\theta}_t) - \hat{Q}^{\pi}(\boldsymbol{x}_t, \boldsymbol{a}_t; \boldsymbol{\theta}_t), \tag{20}$$

and $\nabla_{\boldsymbol{\theta}} \hat{Q}^{\pi}(.)$ is the gradient of the approximated Q-value with respect to $\boldsymbol{\theta}$ computed as:

$$\nabla_{\boldsymbol{\theta}} \hat{Q}^{\pi}(.) = [\boldsymbol{\Phi}^{(1)}(\cdot), \ldots, \boldsymbol{\Phi}^{(M)}(\cdot)]^T. \tag{21}$$

The average value η_t^{π} is updated as [20]:

$$\eta_{t+1}^{\pi} = \eta_t^{\pi} + \beta_t \delta_t, \tag{22}$$

where β_t is the learning rate that needs to satisfy the Robbins-Monro conditions [21]. The SARSA algorithm is summarized in Algorithm 1. The SARSA algorithm for EH-WSNs needs to be implemented at the central unit. The inputs to the SARSA algorithm at time-slot t are the initial parameters' vector $\boldsymbol{\theta}_t$, and the state vector $\boldsymbol{x}_t$. The output is the action vector $\boldsymbol{a}_t$. The central unit disseminate the optimal policy to all sensor nodes at each time-slot. The sate vector and the parameters vector are then updated, and the reward is computed. The operation of the SARSA algorithm consists of two phases: the learning phase, where the parameters' vector $\boldsymbol{\theta}$ is learned followed by a testing phase. In both phases the state vector of all senor nodes is acquired by the central unit. The stopping criteria for the learning phase of the algorithm is chosen as the number of episodes N.

E. D-SARSA Algorithm

The centralized SARSA algorithm entails high complexity and may be prohibitive for implementation and for online operation especially for large scale EH-WSNs for the following reasons. First, the communication overhead is high and the delay is unpredictable, since the central unit needs to receive the state vector $\boldsymbol{x}_t$ at each time-slot before taking an action and each sensor node needs to receive its action vector from the central unit at each time slot. Second, the action space grows exponentially with the number of sensor nodes (using discrete power action). Therefore, the action selection at each time-slot is computationally expensive. To overcome the complexity of the centralized SARSA, a multi-agent RL is proposed to devise a distributed transmission policy, where each sensor node adapts its operation over time in response to its own state information, statistical information and exchanged information from its neighboring sensor nodes.

To devise a multi-agent RL algorithm, each of the feature functions of sensor node S_i needs to be a function of its own action and state vectors, statistical information, and information exchanged from its neighboring nodes. In this sense, features $\Phi_1^{(i)}(\cdot)$ to $\Phi_6^{(i)}(\cdot)$ satisfy these requirements. However, feature $\Phi_7^{(i)}(\cdot)$ does not satisfy these requirements, since it depends on the actions of other sensor nodes.

In the proposed distributed RL algorithm, we assume that sensor node S_i uses the average received data from previous time slots instead of the received data (action) at the current time slot, i.e., an estimate of the actions of other sensor nodes are used. Following assumption, allows to decouple the action selection at each sensor node. Therefore, feature function $\Phi_7^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t)$ is replaced by the feature function:

$$\tilde{\Phi}_7^{(i)}(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \begin{cases} 1, \text{ if } & C_t^{(i)} + d_t^{(i)} + \overline{D}_t^{(i)} \\ & - \sum_{j \in \mathcal{N}_i} Y_t^{(i,j)} r_t^{(i,j)} \leq C_{\max}^{(i)}, \\ 0, & \text{otherwise}, \end{cases} \tag{23}$$

where $\overline{D}_t^{(i)}$ is the mean-value of the received data from the other sensor nodes estimated using previous time-slots as $\overline{D}_t^{(i)} = \frac{\sum_{l=0}^{t-1} \sum_{j \mid i \in \mathcal{N}_j} Y_l^{(j,i)} r_l^{(j,i)}}{t}$.

The buffer state $C_t^{(j)}$ for $j \in \mathcal{N}_i$ is assumed to be known at sensor node S_i by information exchange, which allows sensor node S_i to estimate the average remaining data in the buffer of sensor node S_j. An additional feature function related to

the buffer state of neighboring node S_j can be expressed as:

$$\Phi_8^{(i)}\big(\boldsymbol{x}_t^{(i)}, C_t^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{a}_t^{(i)}\big) = \begin{cases} 1, & \text{if } C_t^{(j)} + \overline{D}_t^{(j,i)} + r_t^{(i,j)} \leq C_{\max}^{(j)}, \\ & \text{where } j:\ Y_t^{(i,j)} = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (24)$$

where $\overline{D}_t^{(j,i)}$ is the average data remaining at the buffer of sensor node S_j excluding the arrived data from sensor node S_i, which is computed as: $\overline{D}_t^{(j,i)} = \frac{\sum_{l=0}^{t-1}(C_{l+1}^{(j)} - C_l^{(j)} - Y_l^{(i,j)} r_l^{(i,j)})}{t}$.

In the proposed multi-agent (distributed) RL algorithm, the action space of each sensor node S_i for $i = 1, \ldots, M$ is affected only by the number of its neighbors $|\mathcal{N}_i|$ and the number of discrete power levels. Sensor node S_i for $i = 1, \ldots, M$ selects its action at time-slot t from its action space $\boldsymbol{\mathcal{A}}^{(i)}$ whereas, in the centralized RL algorithm, the action space scales exponentially with the number of sensor nodes, i.e., $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}}^{(1)} \times \cdots \times \boldsymbol{\mathcal{A}}^{(M)}$. In the proposed distributed RL algorithm, each sensor node determines its action without the need for a central unit. The approximated Q-value $\hat{Q}_i^{\pi}(\cdot)$ of sensor node S_i is obtained as:

$$\hat{Q}_i^{\pi}\big(\boldsymbol{x}_t^{(i)}, C_t^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{a}_t^{(i)}; \boldsymbol{\theta}^{(i)}\big) = \boldsymbol{\Phi}^{(i)}\big(\boldsymbol{x}_t^{(i)}, C_t^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{a}_t^{(i)}\big)\, \boldsymbol{\theta}^{(i)}, \quad (25)$$

where sensor node S_i features' vector is given as: $\boldsymbol{\Phi}^{(i)}(\cdot) = [\Phi_1^{(i)}(\cdot), \ldots, \Phi_8^{(i)}(\cdot)]$ and sensor node S_i parameters' vector is defined as $\boldsymbol{\theta}^{(i)} = [\theta_1^{(i)}, \ldots, \theta_8^{(i)}]^T$.

A distributed-value-function for multi-agent systems is proposed in [39], based on the Q-learning algorithm using discounted reward. The Q-learning requires additional computational cost at each parameter update step since the Q-value of the greedy policy needs to be computed. In this paper, the distributed RL algorithm is based on the SARSA algorithm using average reward. At time-slot t, sensor node S_i observes its own state $\boldsymbol{x}_t^{(i)}$ and the buffer states of its neighboring nodes $C_t^{(j)}$, $\forall j \in \mathcal{N}_i$. The action $\boldsymbol{a}_t^{(i)}$ is then selected using ϵ_t greedy policy, where the action $\boldsymbol{a}_t^{(i)}$ is selected randomly with probability ϵ_t from the action space $\boldsymbol{\mathcal{A}}^{(i)}$ and greedy with probability $1 - \epsilon_t$ as:

$$\boldsymbol{a}_t^{(i)} = \arg\max_{\boldsymbol{A}\in\boldsymbol{\mathcal{A}}^{(i)}} \hat{Q}_i^{\pi}\big(\boldsymbol{x}_t^{(i)}, C_t^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{A}\big). \quad (26)$$

Sensor nodes S_i for $i = 1, \cdots, M$ apply their selected actions, and then sensor node S_i observes/computes its reward $R_{t+1}^{(i)}$ which is designed to have a cooperative behavior. The reward function of sensor node S_i is aligned with the EH-WSN wide objective of minimum delay transmission of all sensor nodes. The reward function $R_{t+1}^{(i)}$ of sensor node S_i is defined based on information exchange with neighboring nodes as:

$$R_{t+1}^{(i)} = C_{\max}^{(i)} - C_{t+1}^{(i)} + \sum_{j\in\mathcal{N}_i \,\&\, j\neq M+1} (C_{\max}^{(j)} - C_{t+1}^{(j)}). \quad (27)$$

Sensor node S_i assigns equal weights for the data remaining in its own buffer and the data remaining in the buffers of its neighbors. Without loss of generality, different weights can be assigned as in [39], [40]. It can be shown that the proposed reward (utility) $R_{t+1}^{(i)}$ has an associated potential function R_{t+1} (the centralized RL algorithm reward function) based on the potential games framework [41], [42]. Therefore, the optimization of the individual reward functions leads to a local optimal of the potential function under certain conditions.

Algorithm 2 D-SARSA Algorithm

Set $t = 0$.
Given α_t, β_t and ϵ_t.
Initialize $\overline{D}_t^{(i)}$, $\overline{h}_t^{(i,j)}$, $\overline{D}_t^{(j,i)}$, $\forall j \in \mathcal{N}_i$.
Initialize $\boldsymbol{\theta}_t^{(i)}$ randomly, and $\eta_t^{(i)} = 0$.
Observe the state $\boldsymbol{x}_t^{(i)}$.
Communicate with S_j, and get $C_t^{(j)}$, $\forall j \in \mathcal{N}_i$.
Select the action $\boldsymbol{a}_t^{(i)}$ according to the ϵ_t-greedy policy.
Apply the action $\boldsymbol{a}_t^{(i)}$.
while Stopping criterion is not satisfied **do**
  for $t = 0, \ldots, K-1$ **do**
    Observe the next state $\boldsymbol{x}_{t+1}^{(i)}$.
    Communicate with S_j, and get $C_{t+1}^{(j)}$, $\forall j \in \mathcal{N}_i$.
    Measure the reward $R_{t+1}^{(i)}$.
    Update $\overline{D}_t^{(i)}$, $\overline{h}_t^{(i,j)}$ and $\overline{D}_t^{(j,i)}$, $\forall j \in \mathcal{N}_i$.
    Select the action $\boldsymbol{a}_{t+1}^{(i)}$ according to the ϵ_{t+1}-greedy policy.
    Communicate with S_j, $\forall j \in \mathcal{N}_i$ and get $\Delta_t^{\hat{Q}_j^{\pi}}$.
    Compute $\delta_t^{(i)}$.
    Update $\boldsymbol{\theta}_t^{(i)}$ and $\eta_t^{(i)}$ using (28) and (29).
    Apply the action $\boldsymbol{a}_{t+1}^{(i)}$.
  end for
end while

D-SARSA uses gradient-descent to update the parameters' vector $\boldsymbol{\theta}^{(i)}$ as:

$$\boldsymbol{\theta}_{t+1}^{(i)} = \boldsymbol{\theta}_t^{(i)} + \alpha_t \delta_t^{(i)} \nabla_{\boldsymbol{\theta}^{(i)}} \hat{Q}_i^{\pi}(\boldsymbol{x}_t^{(i)},\ C_t^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{a}_t^{(i)}; \boldsymbol{\theta}_t^{(i)}), \quad (28)$$

where $\nabla_{\boldsymbol{\theta}^{(i)}} \hat{Q}_i^{\pi}(\cdot)$ is the gradient of the approximated Q-value with respect to $\boldsymbol{\theta}^{(i)}$ and $\delta_t^{(i)}$ is the TD error computed as:

$$\delta_t^{(i)} = R_{t+1}^{(i)} - \eta_t^{(i)} + \Delta_t^{\hat{Q}_i^{\pi}} + \sum_{j\in\mathcal{N}_i \,\&\, j\neq M+1} \Delta_t^{\hat{Q}_j^{\pi}}, \quad (29)$$

Sensor node S_i computes the difference in the Q-value $\Delta_t^{\hat{Q}_i^{\pi}}$ as:

$$\Delta_t^{\hat{Q}_i^{\pi}} = \hat{Q}_i^{\pi}\big(\boldsymbol{x}_{t+1}^{(i)},\ C_{t+1}^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{a}_{t+1}^{(i)}; \boldsymbol{\theta}_t^{(i)}\big) - \hat{Q}_i^{\pi}\big(\boldsymbol{x}_t^{(i)},\ C_t^{(j)}|_{\forall j\in\mathcal{N}_i}, \boldsymbol{a}_t^{(i)}; \boldsymbol{\theta}_t^{(i)}\big). \quad (30)$$

Sensor node S_i communicates with its neighboring nodes at each time-slot to acquire $\Delta_t^{\hat{Q}_j^{\pi}}$ for $j \in \mathcal{N}_i$. After computing the difference, sensor node S_i computes the TD error $\delta_t^{(i)}$ and updates the average value $\eta_t^{(i)}$ as:

$$\eta_{t+1}^{(i)} = \eta_t^{(i)} + \beta_t \delta_t^{(i)}. \quad (31)$$

The D-SARSA algorithm is summarized in Algorithm 2. The inputs to the D-SARSA algorithm at time-slot t are the initial parameters' vector $\boldsymbol{\theta}_t^{(i)}$, and the state vector $\boldsymbol{x}_t^{(i)}$.

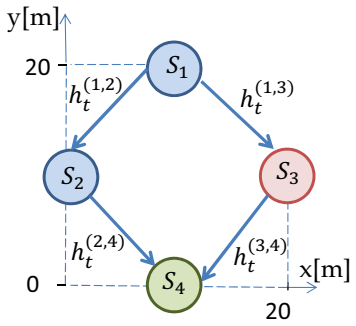


Fig. 2. Small EH-WSN. The data of sensor node S_1 at each time slot can be routed either to sensor node S_2 or S_3.

The output is the action vector $\boldsymbol{a}_t^{(i)}$. Then the sate vector is updated, the reward is computed and the parameters vector is updated. It's worth mentioning that, the operation of the D-SARSA algorithm consists of two phases: the learning phase, where the parameters' vector $\boldsymbol{\theta}^{(i)}$ is learned followed by a testing phase. In the testing phase, only the buffer state needs to be exchanged between neighboring sensor nodes at each time-slot. Usually, the duration of the testing phase is longer compared to the duration of the learning phase. The network will return to the learning phase only if the environment has changed. In addition, for the proposed D-SARSA algorithm, the information exchange is only between neighboring nodes and not between each sensor node and the sink node.

IV. Simulation Results and Discussion

In this section, numerical results for the performance of the proposed SARSA and D-SARSA algorithms are presented. The comparison is made with the offline algorithm. For each sensor node the harvested energy is modeled as a uniform random variable over $[0, H_{\max}]$. The maximum harvested energy $H_{\max}$ depends on the size and the efficiency of the solar panel, the environmental condition (incidence solar radiation and the month of the year), $H_{\max} = 2.5$ μJoules as in [23]. As the processing power is assumed to be constant over the transmission duration, the processing power doesn't affect the convergence of the RL algorithms and the convexity of the optimization problem. Without loss of generality, the processing power is assumed $P_c = 0$. The channel coefficients $h_t^{(i,j)}$ are modeled as zero mean complex Gaussian with variance $\frac{\nu}{d_{ij}^{\kappa}}$, with d_{ij} is the separating distance between sensor nodes S_i and S_j, κ=4 is the propagation loss factor and $\nu = 0.01$ is the propagation gain. The data arrival at each sensor node is modeled as a Poisson random variable with arrival packet rate $\lambda = 0.6$ and packet size of 200 bits. Transmission is organized in time-slots with duration $T = 5$ ms, the transmission bandwidth $W = 2$ MHz, and the noise power $\sigma^2 = -70$ dBm. For the RL algorithms, we consider 2000 independent realizations, in each realization, the episode length $K = 100$. The learning rates $\alpha_t = 5 \times 10^{-3}$ and $\beta_t = 1 \times 10^{-3}$ and the ϵ_t-greedy probability $\epsilon_t = 1 \times 10^{-3}$.

First we consider a small EH-WSN that consists of three sensor nodes and a sink node as depicted in Fig. 2. This allows to compare the centralized SARSA, the D-SARSA, with the offline algorithm obtained using CVX MATLAB toolbox [43]. Each sensor node is equipped with a finite battery storage $E_{\max}^{(i)} = 2H_{\max}$ μJoules and a finite buffer size $C_{\max}^{(i)} = 100$ Kbits. Fig. 3 shows the average reward as a function of episodes for the offline, centralized SARSA, the D-SARSA, with the offline algorithm. As clear in this figure, the gap of the average reward of the proposed centralized and distributed RL algorithms using power-step values of $\delta = 0.05E_{\max}/T$ and $\delta = 0.01E_{\max}/T$ is negligible compared to the average reward of the offline solution.

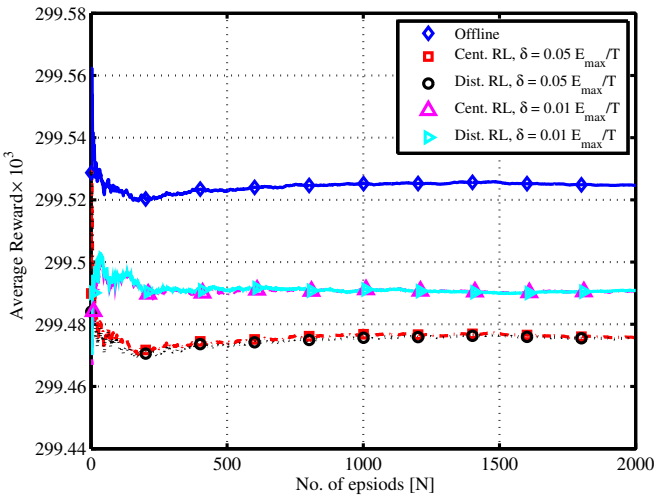


Fig. 3. The average reward for different power-step values.

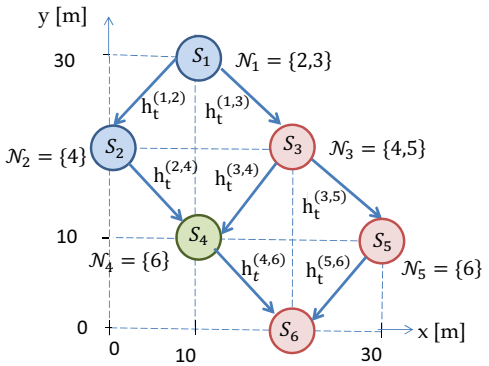


Fig. 4. EH-WSN with $M = 5$. The solid line represents a possible route between two sensor nodes.

To study the effect of the maximum buffer and maximum battery capacities on the reward function, a slightly larger EH-WSN consists of $M = 5$ sensor nodes is considered as shown in Fig. 4. Sensor nodes $\{S_1, \ldots, S_5\}$ are aiming to transmit their data to the sink node S_6. Figs. 5 & 6 show the average reward of the distributed RL algorithm and the offline approach as a function of the number of episodes for different maximum buffer and maximum battery capacities. The average reward increases by increasing the maximum buffer capacity. Similarly, the average reward increases by increasing the battery maximum capacity.

The average reward of the distributed RL algorithm using the proposed binary features is compared with average reward using a hasty approach, where each sensor node uses the maximum available transmission power and selects randomly

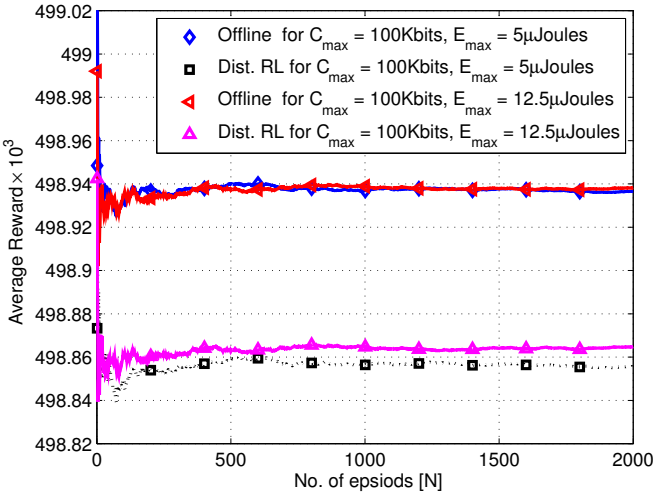


Fig. 5. The average reward for different maximum buffer capacities.

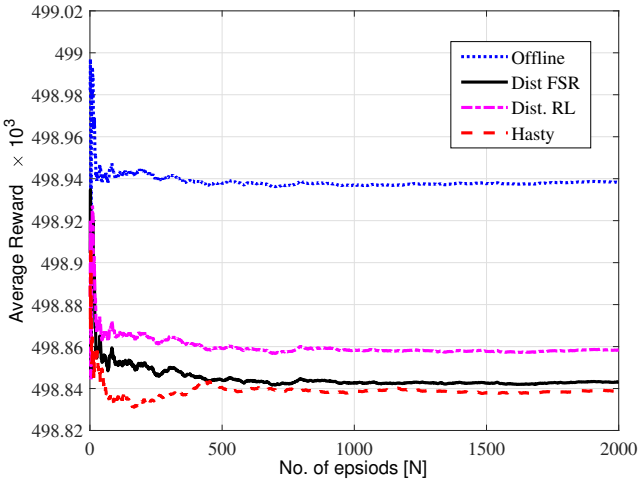


Fig. 7. The average reward based on distributed proposed, distributed FSR and hasty algorithms.

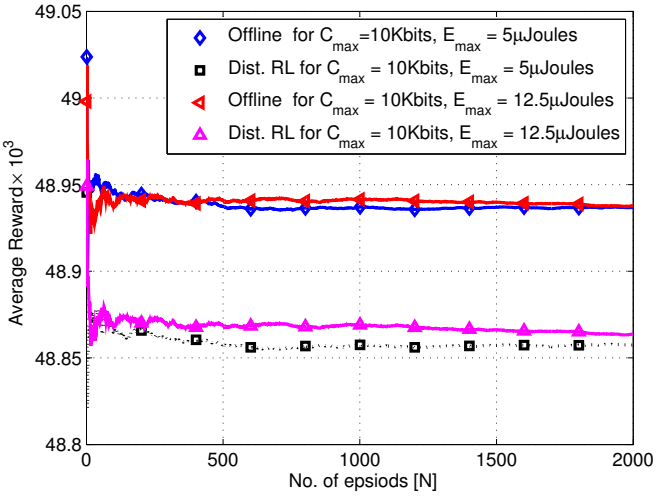


Fig. 6. The average reward for different maximum battery capacities.

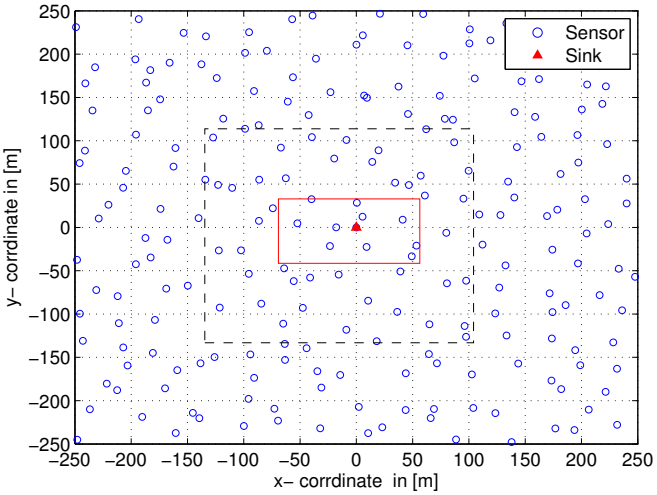


Fig. 8. System layout for $\boldsymbol{M = 225, 49, \& 9}$ senor nodes.

the next-hop. As shown in Fig. 7, the average reward of the distributed RL outperforms the hasty approach. Similarly, the average reward of the distributed RL algorithm using the proposed binary features is compared with the average reward using standard FSR technique. The average reward of the proposed binary features slightly outperforms the average reward of the FSR. The number of parameters for the FSR depends on the dimension of the state space, the step size of each dimension and the dimension of the action space of each sensor node, whereas, the number of parameters of the proposed feature functions is fixed. In general, the number of tiles/reference points of FSR is problem dependent [20]. In this paper, the number of tiles of the FSR is selected with a step size $\delta_m = 0.05x_{\max}(m)$, where $x_{\max}(m)$ is the maximum value of the state in the mth dimension. The number of parameters of the standard FSR technique at sensor node S_i is $|\boldsymbol{\mathcal{A}}_i| \sum_{m=1}^{5+|\mathcal{N}_i|}$. For example, the number of parameters for the standard FSR technique for sensor node S_1 is $40 \times 7 \times 20$. The computation complexity of the proposed binary features is much lower than the computational complexity of the FSR.

To show the scalability of the proposed distributed RL algorithm, M EH-sensor nodes are randomly distributed is considered as shown in Fig. 8. The number of nodes are chosen as $M = 9, 49\&225$ sensor nodes as shown in the marked rectangular areas. The network is constructed in away to make sure that each sensor node has at least one path to the sink node. For $M = 225$ the maximum number of neighboring nodes is 7. The power step size $\delta = 0.05E_{\max}/T$ and $E_{\max} = 2H_{\max}$ Joules. The buffer capacity $C_{\max}^{(i)} = 100$ Kbits, and the packet arrival rate $\lambda = 0.4$ with packet size of 200 bits. The average reward normalized by the number of sensor nodes is shown in Fig. 9. It is clear that the proposed distributed RL is able to empty the buffers in the network. A maximum average reward of 100 Kbits per sensor will be obtained if all buffers are fully empty.

V. Conclusion

In this paper, RL framework for online operation of multi-hop EH-WSNs was proposed aiming to minimize the average delay. The optimal policy comprises of optimal power control and optimal next-hop selection. Centralized and distributed

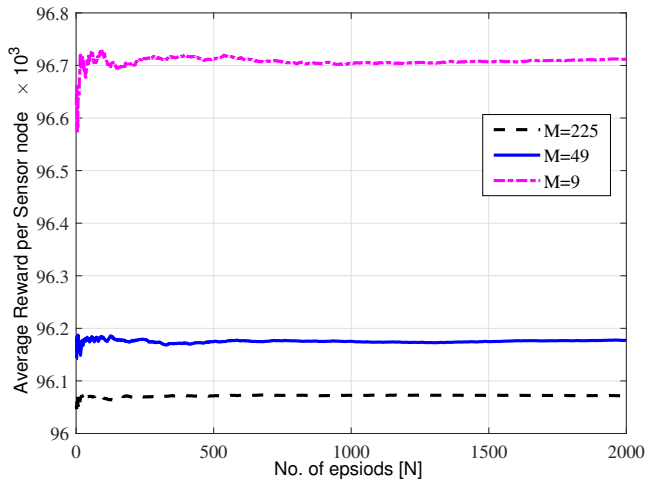


Fig. 9. The normalized average reward for different number of senor nodes $M = \mathbf{225, 49, \& 9}$.

SARSA based RL algorithms were devised. Differential Q-value function approximation using linear function approximation with binary features was proposed for the centralized and distributed algorithms. The reward function is defined as the number of empty spaces in all data buffers. The centralized RL algorithm learns the approximated Q-value function of the network at a central unit using the state information of all sensor nodes at every time-slot. On the other hand, the distributed RL algorithm learns the approximated Q-value function locally using sensor node's state information and state information of neighboring nodes. The distributed RL algorithm can play a key role in the deployment of large scale EH-WSNs, since the computational complexity scales only with the moderate number of neighboring nodes. The average reward of the distributed RL algorithm is comparable with the average reward of the centralized RL algorithm (confirmed for a small scale network).

APPENDIX I
FSR FEATURE FUNCTIONS

Fixed-sparse representation (FSR) is one of the simplest techniques to represent continuous state spaces, where each state is represented using binary codes in each dimension [20], [44]. Let the state vector $\boldsymbol{x}_t^{(i)}$ of sensor node S_i for $i = 1, \ldots, M$ at the tth time-slot be represented by a d dimensional vector, where $x_t^{(i)}(m)$ corresponds to the mth component, hence $\boldsymbol{x}_t^{(i)} = [x_t^{(i)}(1), x_t^{(i)}(2), \ldots, x_t^{(i)}(d)]^T$. Let n_m be the number of tiles that represent the mth dimension of the state space. The tiles in the mth dimension are generated using a step size δ_m. A given feature function is equal to one if the corresponding variable lies in that tile and zero otherwise. The FSR vector is expressed as:

$$\begin{aligned}\boldsymbol{\Phi}^{(i)}(\boldsymbol{x}_t^{(i)}) = \big[&\Phi_{1,1}^{(i)}(\boldsymbol{x}_t^{(i)}), \ldots, \Phi_{1,n_1}^{(i)}(\boldsymbol{x}_t^{(i)}), \ldots, \\ &\Phi_{2,1}^{(i)}(\boldsymbol{x}_t^{(i)}), \ldots, \Phi_{2,n_2}^{(i)}(\boldsymbol{x}_t^{(i)}), \ldots, \\ &\Phi_{d,1}^{(i)}(\boldsymbol{x}_t^{(i)}), \ldots, \Phi_{d,n_d}^{(i)}(\boldsymbol{x}_t^{(i)})\big]^T, \end{aligned} \tag{A1}$$

where,

$$\Phi_{m,n}^{(i)}(\boldsymbol{x}_t^{(i)}) = \begin{cases} 1 & \text{if } x_t^{(i)}(m) \in \mathcal{T}_n^{(i)}(m), \\ 0, & \text{otherwise}, \end{cases} \tag{A2}$$

and $\mathcal{T}_n^{(i)}(m)$ is the nth tile of the mth dimension of the state space of sensor node S_i for $n = 1, \ldots, |n_m|$, and $m = 1, \ldots, d$. The total number of FSR features per action is computed as [44]: $\sum_{m=1}^{d} n_m$.

REFERENCES

[1] C. Mahapatra, Z. Sheng, P. Kamalinejad, V. Leung, and S. Mirabbasi, "Optimal power control in green wireless sensor networks with wireless energy harvesting, wake-up radio and transmission control," *IEEE Access*, vol. 5, pp. 501–518, 2017.

[2] Q. Wu, G. Y. Li, W. Chen, D. W. K. Ng, and R. Schober, "An overview of sustainable green 5G networks," *IEEE Wireless Commun. Mag.*, vol. 24, no. 4, pp. 72–80, Aug. 2017.

[3] D. Zhang, M. Liu, S. Zhang, and Q. Zhang, "Non-myopic energy allocation for target tracking in energy harvesting UWSNs," *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3772–3783, 2019.

[4] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[5] M. Ku, W. Li, Y. Chen, and K. J. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1384–1412, Secondquarter 2016.

[6] Y. He, X. Cheng, W. Peng, and G. Stuber, "A survey of energy harvesting communications: models and offline optimal policies," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 79–85, Jun. 2015.

[7] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Trans. Commun.*, vol. 60, no. 1, pp. 220–230, Jan. 2012.

[8] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1326–1336, Apr. 2010.

[9] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732–1743, Sep. 2011.

[10] K. F. Trillingsgaard and P. Popovski, "Communication strategies for two models of discrete energy harvesting," in *Proc. IEEE International Conference on Communications, ICC, Sydney, Australia*, Jun. 2014, pp. 2081–2086.

[11] C. Ho and R. Zhang, "Optimal energy allocation for wireless communications with energy harvesting constraints," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4808–4818, Sep. 2012.

[12] I. Ahmed, A. Ikhlef, R. Schober, and R. Mallik, "Power allocation for conventional and buffer-aided link adaptive relaying systems with energy harvesting nodes," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1182–1195, 2014.

[13] C. Huang, R. Zhang, and S. Cui, "Optimal power allocation for outage probability minimization in fading channels with energy harvesting constraints," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 1074–1087, Feb. 2014.

[14] M. Rezaee, M. Mirmohseni, and M. Aref, "Energy harvesting systems with continuous energy and data arrivals: The optimal offline and heuristic online algorithms," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3739–3753, Dec. 2016.

[15] H. Al-Tous and I. Barhumi, "Differential game for resource allocation in energy harvesting wireless sensor networks," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 4, pp. 1165–1173, 2020.

[16] T. Ruan, Z. J. Chew, and M. Zhu, "Energy-aware approaches for energy harvesting powered wireless sensor nodes," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2165–2173, 2017.

[17] H. Chen, X. Li, and F. Zhao, "A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2763–2774, 2016.

[18] K. S. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, and W. Heinzelman, "Energy-harvesting wireless sensor networks (eh-wsns): A review," *ACM Trans. Sen. Netw.*, vol. 14, no. 2, Apr. 2018.

[19] H. Sharma, A. Haque, and Z. A. Jaffery, "Solar energy harvesting wireless sensor network nodes: A survey," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 2, Mar. 2018.
[20] R. Sutton and A. Barto, *Reinforcement Learning An Introduction*. The MIT Press, 1998.
[21] C. Szepesvari, *Algorithms for Reinforcement Learning*. Morgan and Claypool, 2010.
[22] F. Zhang and V. K. N. Lau, "Delay-sensitive dynamic resource control for energy harvesting wireless systems with finite energy storage," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 106–113, Aug. 2015.
[23] P. Blasco, D. Gunduz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1882, Apr. 2013.
[24] K. Wu, F. Li, C. Tellambura, and H. Jiang, "Optimal selective transmission policy for energy-harvesting wireless sensors via monotone neural networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9963–9978, 2019.
[25] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting decode-and-forward two-hop communications," *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 3, pp. 309–319, Sep. 2017.
[26] M. Ku, W. Li, Y. Chen, and K. J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2641–2657, Dec. 2015.
[27] V. Hakami and M. Dehghan, "Distributed power control for delay optimization in energy harvesting cooperative relay networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4742–4755, Jun. 2017.
[28] F. Ait Aoudia, M. Gautier, and O. Berder, "RLMan: An energy manager based on reinforcement learning for energy harvesting wireless sensor networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 408–417, Jun. 2018.
[29] M. K. Sharma, A. Zappone, M. Assaad, M. Debbah, and S. Vassilaras, "Distributed power control for large energy harvesting networks: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1140–1154, 2019.
[30] H. Al-Tous and I. Barhumi, "Distributed reinforcement learning algorithm for energy harvesting sensor networks," in *Proc. IEEE International BlackSea Conference on Communications and Networking, (BlackSeaCom)*, 2019, pp. 1–3.
[31] N. G. A. Marques and G. B. Giannakis, "Optimal cross-layer design of wireless fading multi-hop networks," in *Cross Layer Designs in WLAN Systems*. Leicester, UK: Troubador Publishing, 2011, pp. 1–44.
[32] J. Wang, C. Jiang, Z. Han, Y. Ren, and L. Hanzo, "Network association strategies for an energy harvesting aided Super-WiFi network relying on measured solar activity," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3785–3797, Dec. 2016.
[33] Y. Cui, V. Lau, R. Wang, H. Huang, and S. Zhang, "A survey on delay-aware resource control for wireless systems;large deviation theory, stochastic lyapunov drift, and distributed stochastic learning," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1677–1701, Mar. 2012.
[34] K. Tutuncuoglu and A. Yener, "Sum-rate optimal power policies for energy harvesting transmitters in an interference channel," *Journal of Communications and Networks*, vol. 14, no. 2, pp. 151–161, April 2012.
[35] S. Boyd and P. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[36] A. Arafa and S. Ulukus, "Optimal policies for wireless networks with energy harvesting transmitters and receivers: Effects of decoding costs," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2611–2625, Dec. 2015.
[37] Q. Ni and C. C. Zarakovitis, "Nash bargaining game theoretic scheduling for joint channel and power allocation in cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 70–81, Jan. 2012.
[42] S. Zazo, S. V. Macua, M. Sanchez-Fernndez, and J. Zazo, "Dynamic potential games with constraints: Fundamentals and applications in
[38] X.-R. Cao, *Stochastic Learning and Optimization: A Sensitivity Based Approach*. Springer, 2007.
[39] J. Schneider, W. Wong, A. Moore, and M. Riedmiller, "Distributed value functions," in *Proc. 16th International Conference on Machine Learning*, ser. ICML 99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 371–378.
[40] G. Shirazi, P. Kong, and C. Tham, "Distributed reinforcement learning frameworks for cooperative retransmission in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 8, pp. 4157–4162, Oct. 2010.
[41] J. R. Marden, G. Arslan, and J. S. Shamma, "Cooperative control and potential games," *IEEE Trans. Syst., Man, Cybern. B*, vol. 39, no. 6, pp. 1393–1407, Dec. 2009.
communications," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3806–3821, Jul. 2016.
[43] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.
[44] A. Geramifard, T. Walsh, S. Tellex, G. Chowdhary, N. Roy, and J. How, "A tutorial on linear function approximators for dynamic programming and reinforcement learning," *Found. Trends Mach. Learn.*, vol. 6, no. 4, pp. 375–451, Dec. 2013.

Hanan Al-Tous (M'2000, SM' 2017) received the B.Sc. degree in Electrical Engineering in 1998 and the M.Sc. in Communication Engineering in 2000 from the University of Jordan, Amman, Jordan, and the Ph.D. degree in electrical engineering in 2014 from United Arab Emirates University, Al Ain, United Arab Emirates. She worked as a Lecturer with the Electrical Engineering Department, Al-Ahliyya Amman University, Amman, Jordan from 2000-2010. She worked as a Postdoctoral Research Fellow at the United Arab Emirates University, Al Ain, United Arab Emirates from 2014-2018. She is currently a Postdoctoral Research Fellow at Aalto University, Finland. Her research interests include CDMA, cooperative communications, energy harvesting sensor networks, resource allocation for wireless communications, game theory, compressive sensing and machine learning.

Imad Barhumi (M'2005, SM'2013) received the B.Sc. degree in electrical engineering from Birzeit University, Birzeit, Palestine, in 1996, the M.Sc. degree in telecommunications from the University of Jordan, Amman, Jordan, in 1999, and the Ph.D. degree in electrical engineering from the Katholieke Universiteit Leuven (KUL), Leuven, Belgium, in 2005. From 1999 to 2000, he was with the Department of Electrical Engineering, Birzeit University, as a Lecturer. After his Ph.D. graduation, he was a Postdoctoral Research Fellow for one year with the Department of Electrical Engineering, KUL. He is currently an Associate Professor with the Department of Electrical Engineering, United Arab Emirates University, Al Ain, United Arab Emirates. His research interests include signal processing for mobile and wireless communications, cooperative communications, resource allocation and management in wireless communications and networking, game theory and compressive sensing.