Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 7902579, 13 pages
https://doi.org/10.1155/2017/7902579

WILEY | Hindawi

Research Article

Optimal Channel Selection Based on Online Decision and Offline Learning in Multichannel Wireless Sensor Networks

Mu Qiao, Haitao Zhao, Shengchun Huang, Li Zhou, and Shan Wang

College of Electronic Science, National University of Defense Technology, Changsha, China

Correspondence should be addressed to Mu Qiao; qiaomu16@nudt.edu.cn

Received 31 August 2017; Accepted 15 November 2017; Published 13 December 2017

Academic Editor: Kun Bai

We propose a channel selection strategy with hybrid architecture, which combines the centralized method and the distributed method to alleviate the overhead of access point and at the same time provide more flexibility in network deployment. By this architecture, we make use of game theory and reinforcement learning to fulfill the optimal channel selection under different communication scenarios. Particularly, when the network can satisfy the requirements of energy and computational costs, the online decision algorithm based on noncooperative game can help each individual sensor node immediately select the optimal channel. Alternatively, when the network cannot satisfy the requirements of energy and computational costs, the offline learning algorithm based on reinforcement learning can help each individual sensor node to learn from its experience and iteratively adjust its behavior toward the expected target. Extensive simulation results validate the effectiveness of our proposal and also prove that higher system throughput can be achieved by our channel selection strategy over the conventional off-policy channel selection approaches.

1. Introduction

Multichannel communication enables terminals to transmit on different channels simultaneously without mutual interferences. It has been widely used in Wireless Sensor Networks (WSNs) or Internet of Things (IoT) to support large and dense networks [1]. Since different channels may result in different transmission qualities, the channel selection plays crucial role in multichannel WSNs.

Owing to the constraints of energy budget and memory size of WSN nodes, centralized approaches are usually considered to conduct channel selection. In these approaches, a central node, for example, access point (AP) or sink node, performs all the necessary computations and informs reasonable channel selection decision to other sensor nodes. Wu et al. [2] adopt a static tree-based channel selection approach where the sink node can operate on attribute sensor node to switch to a channel with minimum interference. Li et al. [3] extend the typical two-level architecture by using aggregator coordinate associated with sensor nodes to avoid the transmission of huge collected data. However, centralized approaches have limited performances in large-scale networks. Therefore, the distributed approaches have attracted more interests, since they allow better flexibility and scalability in node deployment. Tang et al. [4] design a counter-based approach in which nodes select channels based on the channel quality. Nevertheless, information exchange and negotiation among nodes in this approach require tight synchronization.

In order to implement self-decision and self-learning, the approaches based on game theory and reinforcement learning have been introduced to improve channel selection or other resource allocation problems, for example, [5–11]. Game theory has been a powerful tool to model decentralized networks to obtain an equilibrium state. Its common drawbacks lie in the huge instant computational costs. Félegyházi et al. [5] define two-tier noncooperative medium access game which composes of a channel allocation and a multiple access subgames. Han and Kawanishi [6] provide two types of game strategies to adapt to different collision probabilities. Canzian et al. [7] design an equilibrium game between the pricing and intervention to achieve the maximum efficiency

in the perfect monitoring scenario. On the other hand, reinforcement learning can be used to help each individual node learn from a sequence of their individual feedback history and adjust their behaviors towards expected state, gradually. Nie and Haykin [8] provide a classical reinforcement learning framework to solve the channel assignment problem. Naddafzadeh-Shirazi et al. [9] and Zhou et al. [10] investigate reinforcement learning schemes to help secondary users capture the state of the primary user and learn the satisfactory feedback to improve its own utility. Zame et al. [11] design a statistical count learning scheme to make secondary stations learn from and coordinate their own histories, while simultaneously teaching other stations about these histories of counter. Unfortunately, reinforcement learning approaches have a common disadvantage that they usually require lots of learning iterations to converge to an acceptable solution. Furthermore, most existing work is based on information exchange and negotiation among users, which may cause computational complexity and communication overhead.

In this paper, we propose an intelligent channel selection strategy with hybrid architecture which benefits from the combination of centralized methods and distributed method. Our work requires neither central control nor any exchange or negotiation messages among sensors. Most importantly, we make use of the intelligent technique, for example, game theory and reinforcement learning, to find a solution to the application limitation problem of optimal channel selection with different communication overhead. To achieve this goal, we formulate two algorithms from the perspective of sensors which, respectively, named online decision algorithm and offline learning algorithm. We consider the proposed online strategy and offline strategy have their own merits and are targeted at different application scenarios.

The online decision algorithm based on noncooperative game is to help each individual sensor immediately select optimal channel when the network can satisfy the requirements of energy and computational costs. In terms of the computational complexity, the online strategy is based on the noncooperative game and is less complex than the cooperative game. The "online" here means real-time computation that sensors can obtain immediate results. This approach is focused on how to find the optimal equilibrium state through local computation by each individual sensor.

The offline learning algorithm based on reinforcement learning is to address the iterative channel selection to decrease energy consumption. Each sensor can learn from a sequence of its individual feedback history and adjust its behavior towards the expected target. This approach emphasizes the learning ability that sensor learns its behavior and picks optimal choices while converging to an acceptable and stable solution. Different from the online decision, the offline learning algorithm cannot affect node's selection behavior immediately, but in an iterative way. Therefore, the main contributions of this paper are twofold:

(i) We propose a hybrid architecture, in which centralized processing and distributed processing are jointly considered in order to alleviate overhead of AP node and allow more flexible deployment.

(ii) In this architecture, we present two types of optimal channel selection algorithm based on intelligent decision and learning. They can be used to adapt to different requirements of the communication overhead. It requires no central control, no information exchange, or negotiation among individual nodes, which allows low computational complexity, communication overhead, and storage requirement.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 introduces the system model. In Sections 4 and 5, the online decision and offline learning method for channel selection are, respectively, explained in detail. Section 6 validates our proposal via simulation. Section 7 concludes this research.

2. Related Work

Intelligent methods have been extensively investigated on the channel selection problem in the past decade. They are different from conventional channel selection strategies by taking advantage of the sense information around external environment to make a decision in the channel selection process. The intelligent methods can be broadly divided into two categories: the game-based category that focuses on applying game theoretical tools to make a real-time decision [5–7, 12–23] and the learning-based category that focuses on introducing reinforcement learning techniques to select optimized resource [8–11, 24–31].

For the game-based category, a host of game strategies are presented to fit the optimal channel selection process, including cooperative game and noncooperative game solutions. The cooperative game can improve the performance of resource allocation protocol, while it needs more information exchange and negotiation among nodes, which incurs high communication overhead and computational complexity. Nuggehalli et al. in [12] exploit AP node to manage the priority of other nodes, which guarantees the fairness of the bargaining process. This thread is enhanced by penalizing and pricing mechanisms [13–16]. In specific, Shrestha et al. [13] and Chatterjee and Wong [14] investigate punishment mechanism to promote cooperation. Wang et al. [15] and Cui et al. [16] propose a pricing model based on Stackelberg game, in which the leader node formulates price list for follower node to access certain desirable channel. Recently, a new incentive scheme, called intervention, has gained utilization in [17–19]. These approaches aim at formulating several incentive mechanisms to achieve higher utility. However, these methods need designers or coordinators to price and formulate reasonable rules during the initial phase. Nodes should perform strictly with the rules during the execution phase, which brings an inevitable problem for dynamic network scenarios that designers need to monitor and adjust rules constantly. All of the aforementioned works depend on a centralized server to solve the resource allocation issue and inform the decision to each individual node. However, in many cases, the synchronization information may not be available for all nodes, and some nodes may deviate from cooperation. The second thread in this category uses

noncooperative game-based algorithms and policies for the distributed scenario, in which each node makes real-time decisions considering only to maximize its individual utility. Cho and Tobagi [20] indicate that the noncooperative game has lower computational complexity than cooperative game. The work suggests that noncooperative system converges the individual benefits with an appropriate selfish strategy that can lead to a global network optimal result. In [21], each node maintains some local counters to collect the states of packet transmission, based on which the conditional collision probability and the transmission probability of the opponents' behavior can be calculated without negotiation. Zheng et al. [23] extend the works in [22] and investigate the problem of channel selection where no information exchange is available among users without the centralized controller. Each user adaptively updates its channel selection strategy relying on the individual experienced action-reward. It can be noted that the prior solutions in [22, 23] are similar to our design as they do not need to exchange information in dynamic and distributed networks. Nevertheless, the prior work needs a mechanism to distinguish active users from inactive ones, which is not required in our design.

For the learning-based category, reinforcement learning approaches are firstly proposed to achieve low energy consumption and low computational complexity in WSN [24–26]. Subsequently, more and more investigations are performed by combining it with the other mechanisms [29–33] in cognitive networks. Teng et al. [27] have discussed a scheme which adopts a Q-learning-based auction game to help nodes compete channel access opportunity. Kakalou et al. [28] and Saleem et al. [29] use cluster-based architectures instead of the central entity, in which cluster head observes the traffic of primary user (PU) to avoid collisions while keeping other member nodes synchronized. In [30], Lin et al. have investigated a novel dynamic spectrum access framework with control information exchange through beacons. In [31], a novel distributed Q-learning algorithm with heuristically accelerated scheme has been shown to be a powerful approach to solve dynamic spectrum access problem. The main insight of these contributions is employing interactive information among nodes; however, an excess of information exchange may oppose the alleviation of communication overhead. Motivated by these observations, in our solution, the information exchange is not necessary in our approach.

3. System Model

There are two types of node in the considered system: AP and sensor node. Assume that there are k orthogonal channels and n selfish sensors in the networks ($k \leq n$). Orthogonal Frequency Division Multiple Access (OFDMA) is applied so that each sensor can access different channels by utilizing the feedback information (e.g., the channel gain) from AP [33, 34]. The sensors in the network are related to diverse channel gains. We further assume that the interference comes only from the sensors that are intended to contend the same channel. Figure 1 shows the considered network model. There are two crucial problems in this system, that is, how the sensors select proper channels and how they compete to access if two or more sensors select the same channel simultaneously. We will discuss these problems in detail in the followed subsections.

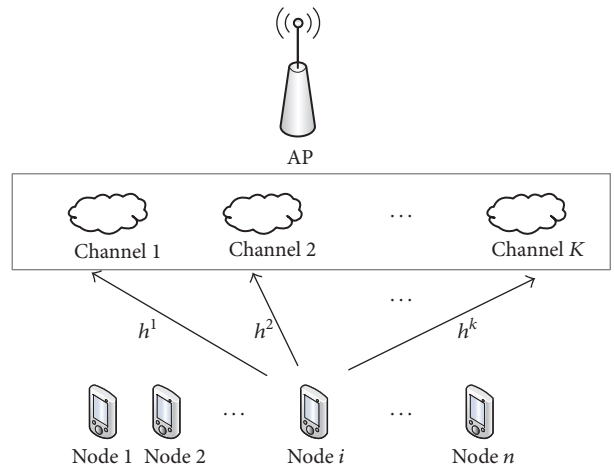


Figure 1: System model.

3.1. Channel Selection. We consider a noncooperative channel selection scenario in Figure 1, where $N = \{1, 2, \ldots, n\}$ is the set of fully distributed sensors, $K = \{1, 2, \ldots, k\}$ is the set of the available channels, and $h = \{h^1, h^2, \ldots, h^k\}$ denotes the set of channel gains. We assume that each sensor is equipped with a single radio transceiver, and it can dynamically access to any channel. It is worth noting that there is a unique policy in the channel selection process that the sensors individually select the channels with maximum channel gain. Due to the selfish behavior among the sensors, there are no negotiation messages exchanged in this process. However, the request and acknowledgement interacting messages between the sensors and the AP are still existed. According to the different strategies, we divide the on-policy channel selection method into two categories.

One category belongs to the real-time strategy, which is named online decision method based on noncooperative game. During the online decision process, the sensor first transmits a random access request to AP and then AP will send the feedback acknowledgement message. After that, the sensor can calculate its own maximum utility via the feedback from AP, which covers the information on channel gains of different channels. The optimal channel, that is, usually the one with the highest channel gain, will be selected.

The other category depends on the history of sensor states, and we name it offline learning method based on reinforcement learning. The offline learning process is actually an iterative exploration and exploitation process, in which each sensor will evaluate its current behavior and then improve it greedily. By this means, the random selection will gradually converge to the optimal one. It is noteworthy that the reinforcement learning algorithm is generally known as a real-time machine learning approach due to the immediate reward back from the environment. However, the immediate reward requires a number of learning iterations and may

bring unaffordable overhead to the network. Therefore, we revised it to gain an "offline" learning approach which will be affordable for the system with limited computational capability and power supply. The detailed procedures will be explained in Sections 4 and 5, and it is validated that this "offline" algorithm can prolong the network lifetime significantly.

3.2. Multiple Access Contention. The multiple access model is to determine the opportunity of channel access when some sensors select the same channel. We consider a CSAMA/CA scheme (e.g., the 802.11 DCF protocol) which is used in this distributed application to resolve the channel contention. In the carrier sensing phase, each sensor detects whether the channel is idle, and then, the sensor takes binary exponential backoff (BEB) algorithm to access the channel in the collision avoidance phase. According to the well-known research [35], in saturation conditions, the conditional transmission probability of sensor can be calculated as

$$\tau = \frac{2 \cdot (1-2p)}{(1-2p) \cdot (\mathrm{CW}_{\min}+1) + p \cdot \mathrm{CW}_{\min} \cdot \left(1-(2p)^m\right)}, \quad (1)$$

where τ, p, $\mathrm{CW}_{\min}$, and m, respectively, denotes transmission probability, collision probability, minimum contention window, and backoff stage.

Based on (1), probability that at least one sensor transmits packets can be expressed as

$$p_{\mathrm{tr}} = 1 - (1-\tau)^n. \quad (2)$$

The transmission success probability of each sensor can be written as

$$p_s = \frac{N \cdot \tau \cdot (1-\tau)^{n-1}}{p_{\mathrm{tr}}}. \quad (3)$$

Accordingly, the achievable rate of sensor i is given as follows

$$r = W \log_2 (1 + \beta \cdot \mathrm{SINR}) = W \log_2 \left(1 + \frac{\beta \cdot P_{\mathrm{tx}} \cdot h_i}{P_{\mathrm{tx}} \cdot h_i \cdot \rho + \sigma^2}\right), \quad (4)$$

where W is the bandwidth. SINR is the signal-to-interference-plus-noise ratio. β denotes the ratio of the current data rate to the Shannon capacity. P_{tx} is the transmission power. h_i denotes the channel gain of sensor node i. ρ is the correlation coefficient between sensor node and AP. σ^2 is the noise power.

Next, we can calculate the time consumption by different events in the transmission process as follows:

(i) The time consumption due to conflict when more than two sensors contend for one transmission opportunity.

$$T_c = \frac{(\mathrm{PHY\_header} + \mathrm{MAC\_header} + \mathrm{packet\_length})}{r} + \mathrm{difs} + \delta, \quad (5)$$

where δ is the propagation delay. PHY_header denotes the length of header for the physical layer. MAC_header denotes the length of header for the MAC layer. packet_length denotes the length of one data packet. r denotes the achievable rate. difs denotes the length of Short Interframe Space.

(ii) The time consumption associated with a successful transmission,

$$T_s = \frac{(\mathrm{PHY\_header} + \mathrm{MAC\_header} + \mathrm{packet\_length})}{r} + \mathrm{sifs} + \mathrm{difs} + \mathrm{ack} + 2 \cdot \delta, \quad (6)$$

where sifs denotes the length of Short Interframe Space. ack denotes the length of acknowledgement message.

(iii) Therefore, the average time consumption can be calculated respectively as

$$T_{\mathrm{fail}} = (1-p_{\mathrm{tr}}) \cdot \mathrm{slot} + p_{\mathrm{tr}} \cdot (1-p_s) \cdot T_c, \quad T_{\mathrm{success}} = p_s \cdot p_{\mathrm{tr}} \cdot T_s, \quad (7)$$

where slot denotes the length of one slot.

And the average throughput can be formulated as

$$T_r = \frac{\mathrm{packet\_length}}{\mathrm{PHY\_header} + \mathrm{MAC\_header} + \mathrm{packet\_length}} \cdot \frac{T_{\mathrm{success}}}{T_{\mathrm{success}} + T_{\mathrm{fail}}} \cdot r, \quad (8)$$

where T_{success} denotes the average time consumption of transmission success. T_{fail} denotes the average time consumption of transmission fail.

From the above expression, we observe that the throughput of sensor is influenced by the achievable data rate r, which is essentially affected by the channel gain according to (4). Due to this fact, the channel with the maximum gain will be selected through the channel selection procedure.

The frequently used notations in this paper are summarized in Notations.

4. Online Decision Algorithm for Channel Selection

Taking account of the selfish behavior of each sensor in the network, we formulate the channel selection problem in multichannel WSN as a noncooperative game. The benefit for noncooperative game is that it requires no coordination control or information exchange among nodes. Based on this model, we propose an online decision algorithm for channel selection.

4.1. Noncooperative Game Model. Let us denote the game as $\Gamma = \{N, \{S_i\}_{i \in N}, \{U_i\}_{i \in N}\}$, where N is the player set (i.e., the set of sensor nodes), $\{S_i\}_{i \in N}$ is the channel selection strategy set for player i, and $\{U_i\}_{i \in N}$ is the utility of player i.

The utility function reflects the throughput of a sensor in the selected channel k, which can be defined as

$$u_i = T_r\left(h_i^k\right), \quad (9)$$

// Initialization
(1) AP evaluates the h^k
// Each sensor obtains the channel gain
(2) Sensor i transmits request to AP
(3) Sensor i obtains channel gain $\{h_i^k\}$, $i \in N$, $k \in K$ from the feedback
// Each sensor select the optimal channel through the Non-cooperative game
(4) **while** $(u_{i,k}(S_i^*, S_{-i}^*) < u_{i,k}(S_i', S_{-i}^*))$ **do**
(5) **for** $i = 1$ to n
(6) **for** $k = 1$ to k
(7) Sensor i calculates the utility function $u_{i,k}$ on each channel according to (9)
(8) **end for**
(9) **end for**
(10) **end while**
(11) $S_i^*(h_i^k) = \arg\max_{S_i} u_{i,k}(S_i^*, S_{-i}^*)$
(12) Take greedy strategy S_i^* to select the kth channel

Algorithm 1: Channel selection based on noncooperative game.

where h_i^k denotes the channel gain obtained by player i in channel k.

Definition 1. A channel selection profile $S^* \in \{S_i\}_{i \in N}$ is a pure strategy Nash Equilibrium (NE) of the game Γ if and only if no player can improve its utility by deviating unilaterally,

$$u_i(S^*) \geq u_i(S') \quad \text{s.t.} \quad \forall i \in N, \forall S \in \{S_i\}_{i \in N}, S^* \neq S'. \tag{10}$$

Note that NE can be generally classified into pure strategy NE and mixed strategy NE. The mixed strategy NE usually seeks for a stable equilibrium state in which sensors select channel with negotiation. In this paper, we employ pure strategy NE in which each sensor selects channel in an on-policy manner. If player i decides to deviate from its individual NE, its utility will be degraded if the system is at such NE. Therefore, this property is particularly desirable. However, the sum-utility optimal channel selection problem is NP-hard. Thus, conventional optimization techniques cannot be applied directly and even centralized algorithms cannot guarantee the globally optimal solution. We propose Theorem 2 to characterize the game.

Theorem 2. *With the maximization of an individual node utility the global benefit of the system is also maximized.*

Proof. Dov Monderer and Shapleyb [32] have proven that the individual or global NE is the maximization of the potential function. According to the concept of NE and (9), we can find that the utility u_i is the best response for node i with strategy S^*, either individually or globally.

In terms of the global optimization, the sum-utility optimal channel selection problem can be formalized as

$$U = \text{maximize} \sum_{i \in N, S^* \in S} u_i, \tag{11}$$

where U denoted the sum of each individual node's utility. According to Theorem 2, we should develop an effective algorithm to obtain the global optimal NE. □

4.2. Algorithm Description. Each sensor is regarded as an online decision automaton agent, which selects the channel according to greed strategy S^* (S^* is the strategy that selects the channel with the highest channel gain). In other words, each sensor will maximize its utility function $u_{i,k}$ in a greedy way. And the algorithm can be described as below.

In the initialization phase, the AP evaluates the channel gain in each channel. Each sensor first transmits the required message to the AP. Next, sensor i obtains channel gains of different channels in feedback acknowledgement message from the AP, based on which the utility function of sensor i is locally calculated. The above operations will be performed iteratively until the expected utility function $u_{i,k}$ converges to the unique NE. Finally, sensor i takes greedy strategy S_i^* to select the corresponding channel which has the optimal channel gain value. Algorithm 1 describes the channel selection process based on noncooperative game.

4.3. Convergence and Complexity Analysis. The convergence of Algorithm 1 is guaranteed since the expected utility function $u_{i,k}$ converges to the unique NE, and the number of iteration is $O(n)$. Within each iteration, the maximum computation of a sensor is $O(n^k)$. Therefore, the total computational complexity of Algorithm 1 is $O(n^k)$. In terms of storage requirement, each sensor needs to cache channel gains of different channels in feedback acknowledgement message; thus k memory units are required to store the immediate feedback acknowledgement messages in the case of the online decision algorithm based on noncooperative game. Obviously, the computational complexity and storage requirement of this immediate algorithm will be increased with the number of sensors and channels. In the following section, we will present an alternative channel selection algorithm with reduced computational complexity.

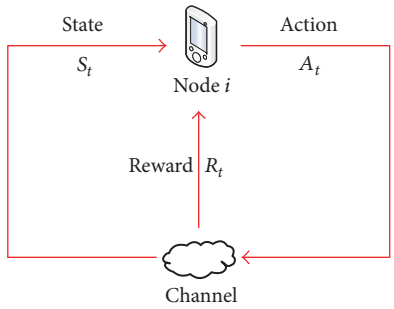


Figure 2: A simplified RL model.

5. Offline Learning Algorithm for Channel Selection

In this section, we present a decentralized channel selection algorithm by using the reinforcement learning framework. This framework selects the channel "offline" in a simpler way than the online algorithm.

5.1. Decentralized Reinforcement Learning Framework. Reinforcement learning (RL) is usually adopted to solve the problem that a learning agent is interacting with its environment to achieve goals related to the state of the environment [36]. Such an agent should be able to observe the state of the environment and take actions according to the feedback of the observation that affects the state in next time. As illustrated in Figure 2, at time t, sensor i observes the channel state and obtains the current channel state S_t, and then sensor i takes action A_t and obtains the reward R_t.

In a general reinforcement learning framework, there are two interacting objects which are agent and environment. And three types of exchange information are included in the learning process: state S_t, action A_t, and reward R_t. For each sensor, the learning process is called the exploration-exploitation tradeoff with two important characteristics: trial-and-error search and delayed reward. The current reward may not affect the next time state immediately, and the expected learning target will be obtained after a period of time. Therefore, we called this learning algorithm "offline" to distinguish it from the online decision algorithm in Section 4.

We propose a Q-learning algorithm for sensors to optimally select the channels according to the histories of observed states and the rewards accumulated into the current choice of action. Some definitions of the algorithm are given as follows.

State. We define that state S_t is observed by node i at time t. We use $S = \{S_1, \ldots, S_t\}$ to denote the finite set of state space, $S_t \in S$. The state transition from S_t to S_{t+1} depends on the action, and accordingly the next state S_{t+1} can be observed when the next action occurs.

Reward. We use $R = \{R_1, \ldots, R_t\}, \forall R_t = \{-1, +1\}$ denotes the finite set of reward space. If node i at time t selects optimal channel, the reward value will be equal to "1." Otherwise, the reward value equals "–1."

Action. We define that the action $A_{i,t}$ is implemented by node i at time t. We use $A = \{A_{i,1}, \ldots, A_{i,t}\}, \forall A_{i_t} = \{A_{t,R_t=1}, A_{t,R_t=-1}\}$ to denote the finite set of action space, where $A_{t,R_t=+1}$ and $A_{t,R_t=-1}$, respectively, denote that the node obtained positive reward and negative reward after action A_t are taken.

Action-Value Function. The action-value function $Q(S, A)$ is associated with the action A and state S at time t. In Section 4, it is equivalent to the utility function.

5.2. Channel Selection with Reinforcement Learning. In the above reinforcement learning framework, each sensor interacts with the channel environment. At each discrete time step t, sensor observes current state S_t, takes action A_t, and obtains feedback reward R_t. As we assume that there are k orthogonal channels allocated by the AP in the system model, each sensor selects the optimal channel with the probability of $1/k$ and other channels with the probability of $(k-1)/k$. The channel selection process is a memory less random process and obtains a sequence of random states with the Markov property. Therefore, it can be modeled as the Markov Decision Process (MDP) with a sequence of state information. The "state information" includes actions, states, and rewards. According to policy $\pi(A \mid S)$, sensor observes state S which is affected by action A. The task for a sensor is to learn the policy $\pi(A \mid S)$ to maximize the expectation of action-value function $Q_\pi(S, A)$.

Since S_t is a Markov process, that is, the information related to past states $[S_1, \ldots, S_{t-1}]$ is covered by state S_t. We only need to store the current state S_t, which allows a considerable reduction of storage requirement.

Definition 3. A history H_t is a sequence of actions, states, and rewards

$$H_t = A_0, S_1, R_1, \ldots, A_{t-1}, S_t, R_t. \tag{12}$$

Definition 4. A channel selection process is a tuple $\langle S, A, R, P, \gamma \rangle$, where S, A, R, and P, respectively, denote a finite set of states, actions, R rewards, and the transition probability matrix. γ ($\gamma \in [0, 1]$) is the discount factor in order to avoid infinite returns in cyclic Markov processes.

As illustrated in Figure 3, sensor i fulfills the channel selection process through two discrete time steps. White point is the initial state, red point implies sensor i selected optimal channel with the probability $1/k$, and black point implies sensor i selected other channel with the probability $(k-1)/k$.

Let us consider the operation at time $t+1$ as an example, and the optimal state transition probability is defined as

$$P_{S_{t+1},\mathrm{RED}} = P\left[S_{t+1} = \mathrm{RED} \mid S_t = \mathrm{WHITE}\right], \tag{13}$$

and the other state transition probability is defined as

$$P_{S_{t+1},\mathrm{BLACK}} = P\left[S_{t+1} = \mathrm{BLACK} \mid S_t = \mathrm{WHITE}\right]. \tag{14}$$

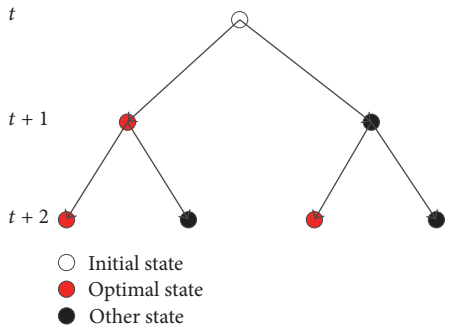


Figure 3: Two steps state transition model.

Definition 5. The optimal action-value function $Q_*(S, A)$ is the maximum action-value function over all policies

$$Q_*(S, A) = \max_{\pi} Q_\pi(S, A). \tag{15}$$

$Q_*(S, A)$ can be recursively computed by adopting the Bellman optimality equation:

$$Q_*(S, A) = R_S^A + \gamma \sum_{S_{\mathrm{RED}} \in S} P_{S,\mathrm{RED}}^A \max_{A_{\mathrm{RED}}} Q_*(\mathrm{RED}, A_{\mathrm{RED}}). \tag{16}$$

$Q_*(S, A)$ always specifies the best possible performance in the MDP. For the MDP of channel selection, this property is hold according to the following theorems.

Theorem 6. *There exists an optimal policy π_* that is better than or equal to all other policies, $\pi_* \geq \pi, \forall \pi$.*

Theorem 7. *All optimal policies achieve the optimal action-value function, $Q_{\pi_*}(S, A) = Q_*(S, A)$.*

Now, we use the Q-learning algorithm to solve the MDP problem of channel selection. Each sensor decides its next action A_{t+1} based on the trend of a sequence of actions, states, and rewards, and then it observes current reward R_t and state S_t to update the Q-value of the action-value function $Q(S, A)$. The updated Q-value will affect the next-round channel selection. The Q-value can be expressed as

$$\begin{aligned} &Q(S_t, A_t) \\ &\longleftarrow (1 - \alpha) Q(S_t, A_t) \\ &\quad + \alpha \left(R_t + \gamma \max_{a_t} Q(S_{t+1}, A_{t+1}) \right) \\ &\text{s.t.} \quad 0 \leq \alpha \leq 1, \ 0 \leq \gamma \leq 1, \end{aligned} \tag{17}$$

where α is the learning rate, which specifies the updating speed of the Q-value. γ is the discount factor that determines the present value of future rewards. $Q(S_t, A_t)$ denotes current Q-value and $Q(S_{t+1}, A_{t+1})$ denotes the expected Q-value.

As shown in Figure 4, we improve the learning policy π by acting greedily with respect to optimal policy $\pi_{*_{\mathrm{greedy}}}$ in each turn of iteration. Figure 4(a) represents the learning policy π that will converge to the optimal policy $\pi_{*_{\mathrm{greedy}}}$ after t times iteration. In Figure 4(b), we use contracting mapping theorem to represent this channel selection process. The red solid line denotes optimal policy $\pi_{*_{\mathrm{greedy}}}$, and the blue solid line denotes learning policy π. The green dash arrow denotes stochastic action, and this action will be improved gradually by a series of feedback information according to the optimal policy $\pi_{*_{\mathrm{greedy}}}$. The rose dash arrow and purple dash arrow, respectively, denote reward information and state information. They are used to evaluate the action according to the learning policy π after n'-step returns ($n' = 1, 2, \ldots, \infty$).

Using the offline algorithm to select channel is an exploration-exploitation process that requires no prior knowledge for each sensor. Through the learning process, the stochastic behavior of sensors can gradually converge to the optimal channel selection.

5.3. Algorithm Description. Each sensor is regarded as an offline learning automaton agent, and the relevant task is to learn the policy π maximizing the expectation of Q-value $Q(S_t, A_t)$. The algorithm is described as follows.

In the initialization phase, each sensor initializes its action space and Q-value. Then, sensor i first takes an random channel selection action A_t and the AP feedback reward R_t to sensor i. Next, sensor i observes the current reward value and the next time state. If sensor i selects optimal channel at time t, the corresponding reward value will be equal to "1"; otherwise the reward value equals "−1." Subsequently, sensor i updates Q-value. This process is repeated until the learning policy π that converges to optimal policy $\pi_{*_{\mathrm{greedy}}}$. Finally, sensor i chooses channel selection action A_{t+1} from S_{t+1} by using policy π which derives from $Q(S_t, A_t)$. Algorithm 2 describes the channel selection process based on Q-learning.

5.4. Complexity and Convergence Analysis. The number of iterations for Algorithm 2 is $O(n)$. Within each iteration, the maximum computation of a sensor is $O(1)$. Therefore, the total computational complexity of Algorithm 2 is $O(n)$. In terms of storage requirement, each sensor only needs to use one memory unit to store the current state information. In this way, the computational complexity and storage requirement can experience signification reduction compared to Algorithm 1.

The convergence of this channel selection algorithm is characterized by the following theorem.

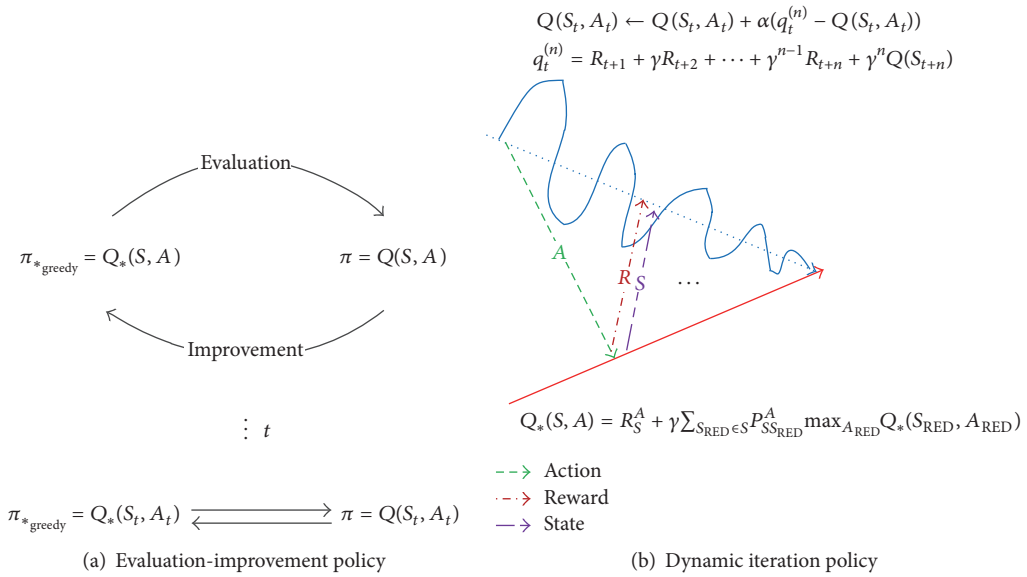
Theorem 8. *The learning policy π converges to optimal policy $\pi_{*_{greedy}}$ if the following conditions are met:*

(i) *The optimal policy $\pi_{*_{greedy}}$ has unique point.*

(ii) *Robbins-Monro sequence [37] of step-sizes α satisfies $\sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty$.*

(iii) *The state S and action A spaces are finite.*

From Theorem 8, we know that the learning policy π converges to the optimal policy $\pi_{*_{\mathrm{greedy}}}$. Since a finite channel



(a) Evaluation-improvement policy (b) Dynamic iteration policy

Figure 4: Policy iteration process model.

// Initialization
(1) Each sensor initializes its action space A
(2) Each sensor initializes its Q-value $Q(S_t, A_t) = 0,\ \forall S_t \in S,\ \forall A_t \in A$
//Learning
(3) Sensor i takes random action A_t
(4) Sensor i observes reward R_t and state S_{t+1}
(5) **while** $(Q(S_t, A_t) = Q_*(S_t, A_t))$ **do**
(6) **for** $i = 1$ to n
(7) **for** $t = 1$ to t
(8) **if** S_{t+1} is good
(9) $R_t = +1$
(10) **else**
(11) $R_t = -1$
(12) **end if**
(13) Sensor i updates its Q-value according to (17)
(13) $Q(S_t, A_t) \leftarrow (1 - \alpha_t) Q(S_t, A_t) + \alpha_t (R_t + \gamma \max_{\alpha_{t+1}} Q(S_{t+1}, A_{t+1}))$
(14) $\pi(S_t, A_t) = \arg\max(Q(S_t, A_t))$
(15) **end for**
(16) **end for**
(17) **end while**
(18) Choose A_{t+1} from S_{t+1} using policy π derived from $Q(S_t, A_t)$

Algorithm 2: Channel selection based on Q-learning.

selection process has only a finite number of policies, this process must converge to an optimal policy and optimal action-value function within a finite number of iterations. The proof is given as follows.

Proof. We can rewrite the updating rule of the Q-value with learning policy in (18) as follows:

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + \alpha\left(q_t^{(n')} - Q(S_t, A_t)\right), \quad (18)$$

where $q_t^{(n')}$ denotes the n'-step Q-value return value and it can be expanded as

$$q_t^{(n')} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n'-1} R_{t+n'} + \gamma^{n'} Q(S_{t+n'}). \quad (19)$$

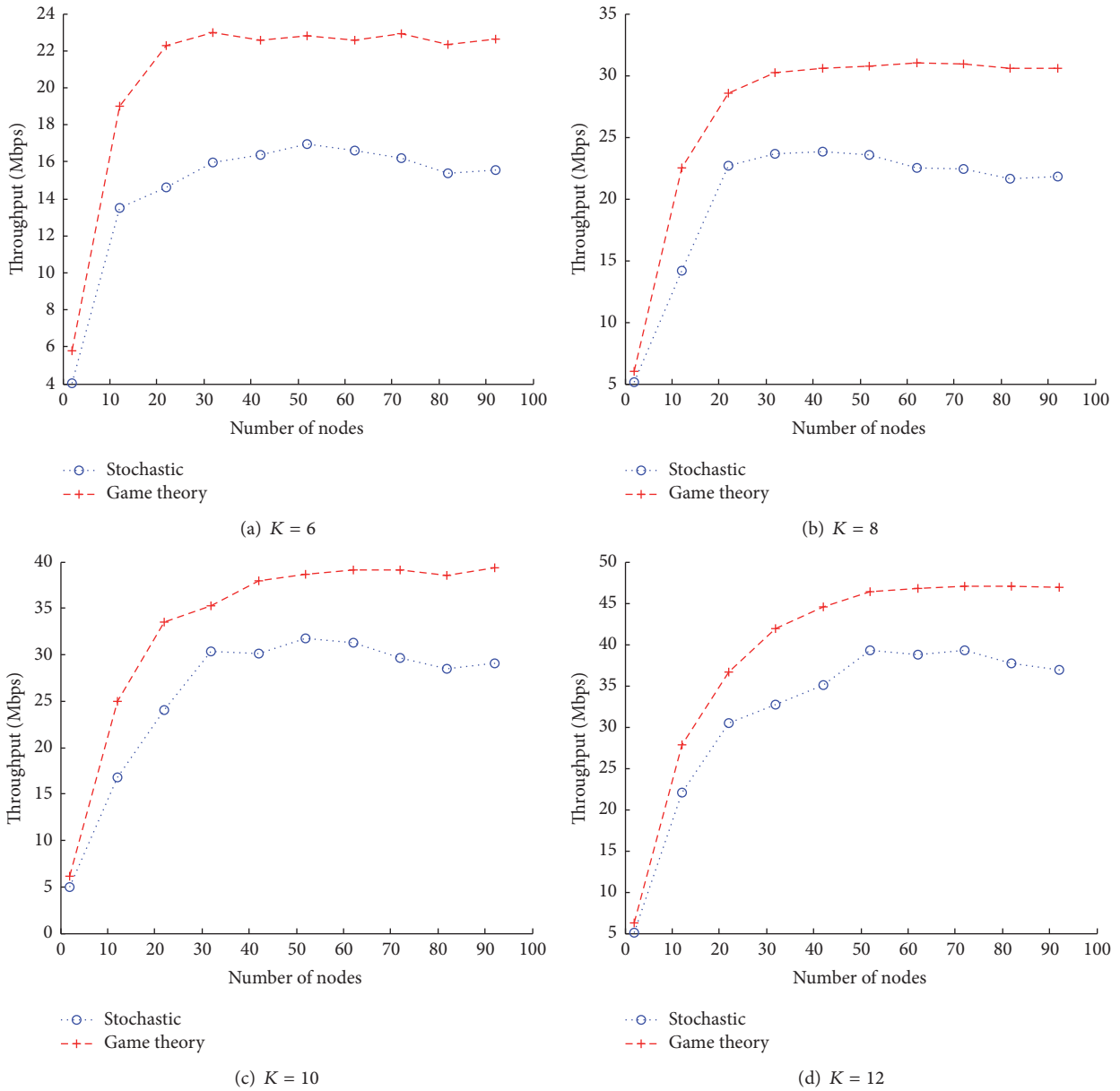


FIGURE 5: Throughput performance of online noncooperative game algorithm.

The n'-step ($n' = 1, 2, \ldots, \infty$) return values can be considered as follows:

$$\begin{aligned} n' &= 1 \quad q_t^{(1)} = R_{t+1} + \gamma Q(S_{t+1}) \\ n' &= 2 \quad q_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(S_{t+2}) \\ &\vdots \qquad\qquad \vdots \\ n' &= \infty \quad q_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T. \end{aligned} \tag{20}$$

In (18), the Q-value at time t is a determinate value. For the return value $q_t^{(n')}$ of the n'-step Q-value, we can see that it is finite and based on the number of evaluation T, which suggests the convergence property of $q_t^{(n')}$ is irrelevant to the number of step.

Therefore, the Q-value in (18) must be a finite value. Theorem 8 is proved. □

6. Evaluation

6.1. Simulation Setup. When multiple sensors select one same channel, we assume that all sensor nodes resolve contention based on IEEE 802.11 standard, that is, the BEB backoff algorithm. The parameters are set as $\mathrm{CW}_{\min} = 32$, $m = 5$, $P_{\mathrm{tx}} = 12\,\mathrm{dBm}$, $\rho = 0.1$, $\sigma^2 = -80\,\mathrm{dBm}$, $W = 1\,\mathrm{Hz}$, and $\beta = 0.5$. The AP randomly assigns the channel gain in each turn of iteration according to $h = \mathrm{rand}(n, k) \times 0.3 + 0.6$. OFDMA technique is used in the physical layer, where the packet length is 512 bits, the acknowledgement message length is 304 bits, and the headers for the physical layer and

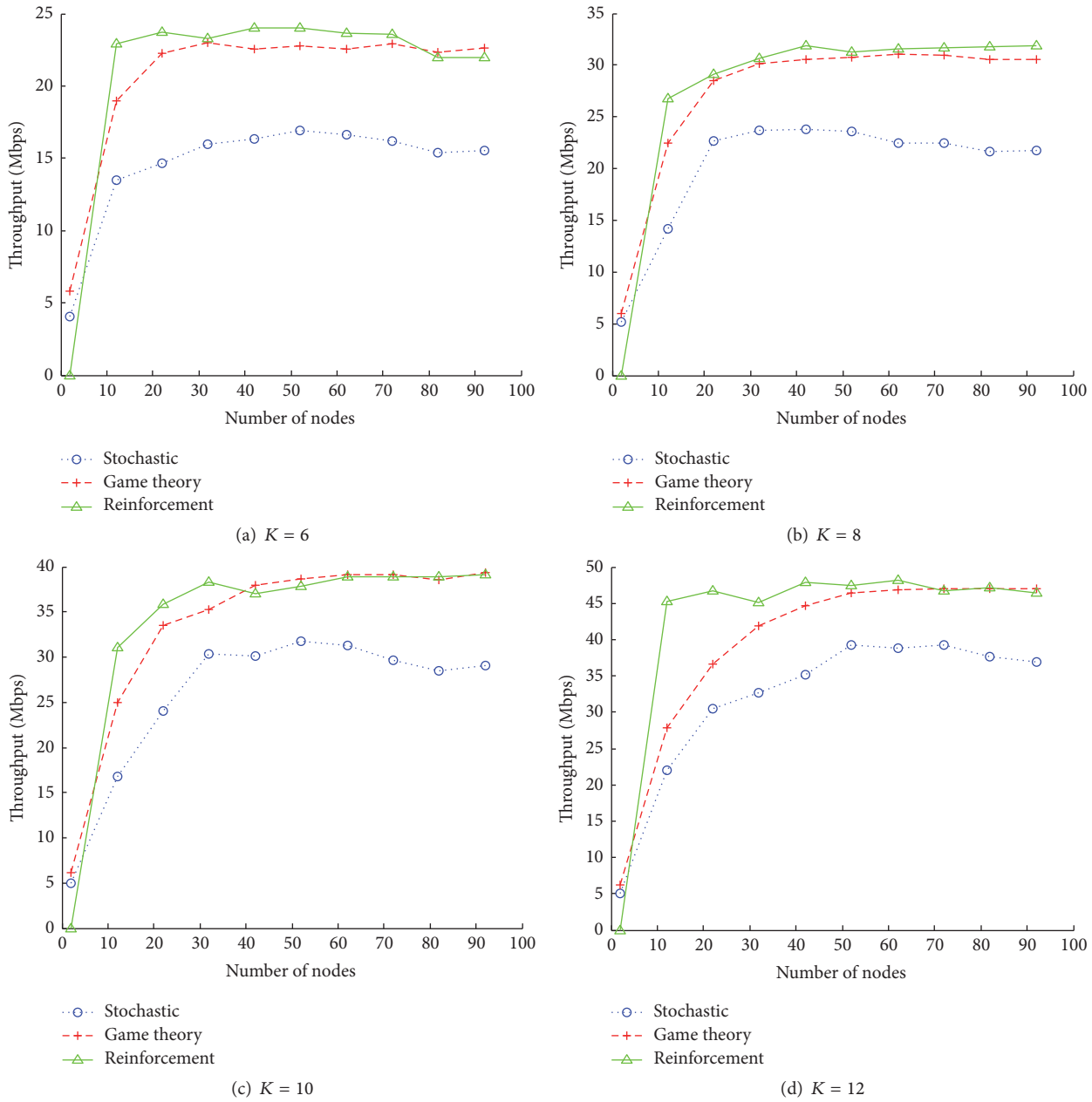


FIGURE 6: Throughput performance of offline reinforcement learning algorithm.

MAC are, respectively, 192 bits and 272 bits. And the slot time is 20 μs, the Short Interframe Space (SIFS) and the DCF Interframe Space (DIFS) are, respectively, 10 μs and 50 μs, and the propagation delay is 1 μs. The discount factor γ was chosen to be 0.9 and the learning rate α was designed to be 0.6. To simulate the dynamic network environment, we set the initial number of sensor to 2, and then 10 sensors are added in each turn of iteration until the total number of sensors reaches 92. In the simulation, we vary the number of channels from 6 to 12. The presented results are obtained by 5000 independent Monte Carlo simulations.

6.2. Evaluation of Online Decision Based on Game Theory. Figure 5 shows the performance comparison in terms of the global throughput, which is the sum of individual throughput achieved by each sensor. The stochastic algorithm acts as the competing scheme as it is a typical off-policy channel selection algorithm. As shown in Figure 5, the noncooperative game algorithm performs better in throughput under various numbers of channels. In particular, the throughput is higher and more stable than the stochastic algorithm. We can also observe that the increasing tendency of the curves becomes slower when the number of sensors becomes larger. It can

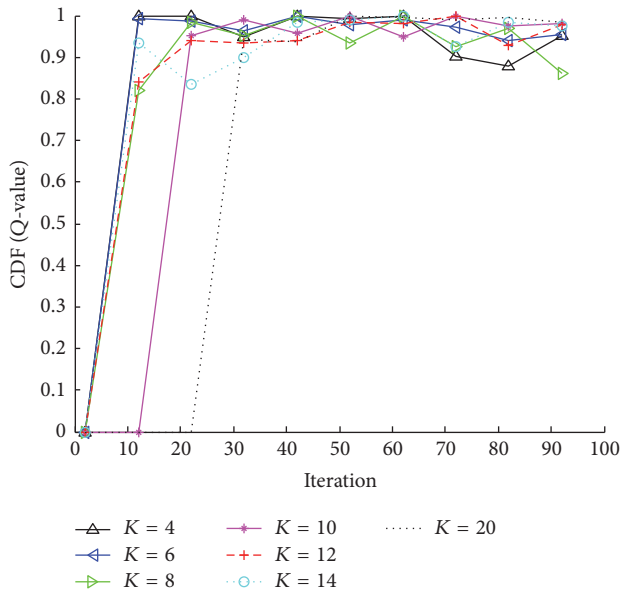


FIGURE 7: Convergence performance of offline reinforcement learning algorithm under different numbers of channels.

be concluded that the throughput gets saturation in each channel, and more sensors bring more severe contention.

6.3. Evaluation of Offline Learning Based on Reinforcement Learning. When the offline reinforcement learning algorithm is adopted, as shown in Figure 6, a better performance in the global throughput is attained compared to the stochastic channel selection algorithm. Similar to the simulation results given by Figure 5, the throughput will get saturation with the increasing number of sensors, which also stems from the severer contention. As shown in Figure 6, we can see that the initial process of reinforcement learning algorithm performance curve is unstable. This is because the learning algorithm is used to help sensors to reduce stochastic behavior. Therefore, at the beginning the sensor behaves stochastically, while it eventually converges to the stable and optimal behavior. The performance gaps between stochastic algorithm and our proposed algorithm increase with the number of sensors. However, the gaps between our proposed algorithms are minor. The reasons can be as follows: (1) more sensors incur severe contention, and (2) our learning algorithm gradually converges toward a pure NE employed in Section 4.

Figure 7 plots the cumulative distribution function (CDF) of Q-value versus the number of sensors. It can be seen that the convergence performance becomes stable with the increasing number of sensors. In addition, the convergence can be gradually achieved when the number of sensors is larger than the number of channels. Moreover, the increasing tendency of the curves becomes stable when the number of sensors added in each turn of iteration is larger than the number of channel. The reasons can be as follows: (1) the samples space becomes larger on each channel when the total number of sensor increases, and (2) sensors have been accumulated enough historical observations and decision experiences.

7. Conclusions

In this paper, two channel selection algorithms based on online self-decision and offline self-learning, respectively, have been investigated in a multichannel wireless sensor networks. Sensor nodes in both algorithms behave selfishly and do not mutually negotiate information among other sensors. The online self-decision is made based on noncooperative game, and the offline self-learning is done based on the reinforcement learning. The online self-decision can be made immediately and is suitable for the real-time application. By contrast, the offline self-learning algorithm can iteratively converge to the optimal channel selection with lower occupation of computational and storage resources; thus it is available for the applications with low computational complexity, communication overhead, and storage requirement. Theoretical analysis and simulation results demonstrated that the proposed channel selection methods can improve the throughput performance compared to the existing off-policy strategies.

Notations

K: Set of the available channels
N: Set of fully distributed sensor nodes
n: Number of sensor nodes
k: Number of channels
h: Set of channel gains
h^k: Channel gain in the kth channel
h_i^k: Channel gain of sensor i in the kth channel
S_i^*: The optimal strategy for sensor i
u_i: The utility function of sensor i
S_t: The channel state at t time
R_t: The feedback reward at t time
A_t: The sensor node take action at t time
t: The discrete time steps
n': Number of steps
$Q_\pi(S, A)$: The action-value function with policy π
$Q_*(S, A)$: The optimal action-value function
$P_{S_{t+1},\mathrm{RED}}$: State transition probability of sensor selected optimal state from t to $t + 1$
$P_{S_{t+1},\mathrm{BLACK}}$: State transition probability of sensor selected other state from t to $t + 1$
$\pi_{*_{\mathrm{greedy}}}$: The optimal policy with greedy algorithm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (Grants no. 61471376, no. 61601482, and no. 61501482) and by the foundation from

Science and Technology on Information Transmission and Dissemination in Communication Networks Lab.

References

[1] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data Collection and Wireless Communication in Internet of Things (IoT) Using Economic Analysis and Pricing Models: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2546–2590, 2016.

[2] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proceedings of the 27th Conference on Computer Communications (INFOCOM '08)*, pp. 1193–1201, IEEE, Phoenix, Ariz, USA, April 2008.

[3] C. Li, P. Wang, H.-H. Chen, and M. Guizani, "A cluster based on-demand multi-channel MAC protocol for wireless multimedia sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 2371–2376, Beijing, China, May 2008.

[4] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '11)*, ACM, Las Vegas, Nev, USA, May 2011.

[5] M. Félegyházi, M. Čagalj, and J.-P. Hubaux, "Efficient MAC in cognitive radio systems: A game-theoretic approach," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1984–1995, 2009.

[6] B. Han and K. Kawanishi, "Using game theory to investigate stochastic channel selection for multi-channel MAC protocol," in *Proceedings of the 2012 IEEE International Conference on Communication Systems, ICCS 2012*, pp. 172–176, Singapore, November 2012.

[7] L. Canzian, M. Zorzi, and M. van der Schaar, "Game theoretic design of MAC protocols: Pricing and intervention in slotted-Aloha," in *Proceedings of the 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 707–714, Monticello, Ill, USA, October 2013.

[8] J. Nie and S. Haykin, "A Q-learning-based dynamic channel assignment technique for mobile communication systems," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1676–1687, 1999.

[9] G. Naddafzadeh-Shirazi, P.-Y. Kong, and C.-K. Tham, "Distributed reinforcement learning frameworks for cooperative retransmission in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 4157–4162, 2010.

[10] P. Zhou, Y. Chang, and J. A. Copeland, "Learning through reinforcement for repeated power control game in cognitive radio networks," in *Proceedings of the 53rd IEEE Global Communications Conference, GLOBECOM 2010*, December 2010.

[11] W. Zame, J. Xu, and M. Van Der Schaar, "Cooperative multi-agent learning and coordination for cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 3, pp. 464–477, 2014.

[12] P. Nuggehalli, M. Sarkar, K. Kulkarni, and R. R. Rao, "A Game-theoretic analysis of QoS in wireless MAC," in *Proceedings of the INFOCOM 2008: 27th IEEE Communications Society Conference on Computer Communications*, pp. 46–50, April 2008.

[13] B. Shrestha, D. Niyato, Z. Han, and E. Hossain, "Wireless access in vehicular environments using BitTorrent and bargaining," in *Proceedings of the 2008 IEEE Global Telecommunications Conference, GLOBECOM 2008*, pp. 5254–5258, December 2008.

[14] D. Chatterjee and T. F. Wong, "Resource allocation and cooperative behavior in fading multiple-access channels under uncertainty," in *Proceedings of the 2009 IEEE Military Communications Conference, MILCOM 2009*, October 2009.

[15] B. Wang, Z. Han, and K. J. R. Liu, "Distributed relay selection and power control for multiuser cooperative communication networks using stackelberg game," *IEEE Transactions on Mobile Computing*, vol. 8, no. 7, pp. 975–990, 2009.

[16] H. Cui, Y. Wang, Q. Guan, and H. Zhang, "Distributed Interference-Aware Cooperative MAC Based on Stackelberg Pricing Game," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4124–4134, 2015.

[17] Y. Chen and S. Kishore, "A game-theoretic analysis of decode-and-forward user cooperation," *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1941–1951, 2008.

[18] M. Janzamin, M. Pakravan, and H. Sedghi, "A game-theoretic approach for power allocation in bidirectional cooperative communication," in *Proceedings of the IEEE Wireless Communications and Networking Conference 2010, WCNC 2010*, April 2010.

[19] P. Ju and W. Song, "Repeated Game Analysis for Cooperative MAC With Incentive Design for Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5045–5059, 2016.

[20] Y. Cho and F. A. Tobagi, "Cooperative and non-cooperative aloha games with channel capture," in *Proceedings of the 2008 IEEE Global Telecommunications Conference, GLOBECOM 2008*, pp. 4533–4538, December 2008.

[21] L. Zhao and H. Zhang, "Using incompletely cooperative game theory in wireless LANs," in *Proceedings of the IET Conference on Wireless, Mobile and Sensor Networks 2007, CCWMSN'07*, pp. 129–132, December 2007.

[22] L. Zheng, Y. Cai, and Y. Xu, "MAC-layer interference mitigation in dynamic and distributed environment: Dynamic graphic game with stochastic learning," in *Proceedings of the 2014 5th International Conference on Game Theory for Networks, GameNets 2014*, November 2014.

[23] J. Zheng, Y. Cai, N. Lu, Y. Xu, and X. Shen, "Stochastic Game-Theoretic Spectrum Access in Distributed and Dynamic Environment," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4807–4820, 2015.

[24] J. Niu, "Self-learning scheduling approach for wireless sensor network," in *Proceedings of the 2010 2nd International Conference on Future Computer and Communication*, pp. V3-253–V3-257, Wuhan, China, May 2010.

[25] Z. Liu and I. Elhanany, "RL-MAC: A reinforcement learning based MAC protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 117–124, 2006.

[26] M. Mihaylov, Y.-A. Le Borgne, K. Tuyls, and A. Nowé, "Reinforcement Learning for Self-organizing Wake-Up Scheduling in Wireless Sensor Networks," *Communications in Computer and Information Science*, vol. 271, pp. 382–396, 2013.

[27] Y. Teng, Y. Zhang, F. Niu, C. Dai, and M. Song, "Reinforcement learning based auction algorithm for dynamic spectrum access in cognitive radio networks," in *Proceedings of the 2010 IEEE 72nd Vehicular Technology Conference Fall, VTC2010-Fall*, September 2010.

[28] I. Kakalou, G. I. Papadimitriou, P. Nicopolitidis, P. G. Sarigiannidis, and M. S. Obaidat, "A Reinforcement learning-based cognitive MAC protocol," in *Proceedings of the IEEE International Conference on Communications, ICC 2015*, pp. 5608–5613, June 2015.

[29] Y. Saleem, K.-L. A. Yau, H. Mohamad, N. Ramli, and M. H. Rehmani, "Joint channel selection and cluster-based routing scheme based on reinforcement learning for cognitive radio networks," in *Proceedings of the 2nd International Conference on Computer, Communications, and Control Technology, I4CT 2015*, pp. 21–25, April 2015.

[30] Y. Lin, C. Wang, J. Wang, and Z. Dou, "A novel dynamic spectrum access framework based on reinforcement learning for cognitive radio sensor networks," *Sensors*, vol. 16, no. 10, article no. 1675, 2016.

[31] N. Morozs, T. Clarke, and D. Grace, "Distributed Heuristically Accelerated Q-Learning for Robust Cognitive Spectrum Management in LTE Cellular Systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 4, pp. 817–825, 2016.

[32] D. Monderer and L. S. Shapleyb, "Potential Games," *Games & Economic Behavior*, vol. 14, no. 44, pp. 124–143, 1996.

[33] D. Kivanc, G. Li, and H. Liu, "Computationally efficient bandwidth allocation and power control for OFDMA," *IEEE Transactions on Wireless Communications*, vol. 2, no. 6, pp. 1150–1158, 2003.

[34] G. Li and H. Liu, "Resource allocation for OFDMA relay networks with fairness constraints," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2061–2069, 2006.

[35] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, New York, NY, USA, 1998.

[37] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.