Learning-Based Spectrum Selection in Cognitive Radio Ad Hoc Networks

Marco Di Felice[1], Kaushik Roy Chowdhury[2], Cheng Wu[2], Luciano Bononi[1], and Waleed Meleis[2]

[1] Department of Computer Science, University of Bologna, Italy
[2] Department of Electrical and Computer Engineering, Northeastern University, Boston, USA

Abstract. Cognitive Radio Ad Hoc Networks (CRAHNs) must identify the best operational characteristics based on the local spectrum availability, reachability with other nodes, choice of spectrum, while maintaining an acceptable end-to-end performance. The distributed nature of the operation forces each node to act autonomously, and yet has a goal of optimizing the overall network performance. These unique characteristics of CRAHNs make reinforcement learning (RL) techniques an attractive choice as a tool for protocol design. In this paper, we survey the state-of-the-art in the existing RL schemes that can be applied to CRAHNs, and propose modifications from the viewpoint of routing, and link layer spectrum-aware operations. We provide a framework of applying RL techniques for joint power and spectrum allocation as an example of Q-learning. Finally, through simulation study, we demonstrate the benefits of using RL schemes in dynamic spectrum conditions.

Keywords: Reinforcement Learning, Cognitive Radio Ad Hoc Networks, Routing, Spectrum Decision, Spectrum Sensing.

1 Introduction

Recent advances in Software Define Radio (SDR) technology have given an impetus towards developing a new generation of highly reconfigurable wireless devices, leading to the novel paradigm of "intelligent" radio systems. Here, the word "intelligence" refers to the ability of radio devices to learn from and adapt to their environment. Cognitive Radio (CR) constitutes the most promising and investigated approach in this research area. A CR device can be formally defined as a radio which changes its transmitter parameters based on interaction with the environment in which it operates [1]. CR technology is envisaged to solve the current problems of inefficiency in spectrum allocation and usage, by implementing dynamic spectrum access (DSA) techniques and often relying on opportunistic transmission in the licensed frequencies. Moreover, it provides reconfigurability at each layer of the protocol stack, in order to support different transmission access technologies and to dynamically meet the Quality of Service (QoS) requirements of end-users. The CR concept is extended in Cognitive Radio Ad Hoc Networks (CRAHNs), in which the network is deployed in an ad-hoc

E. Osipov et al. (Eds.): WWIC 2010, LNCS 6074, pp. 133–145, 2010.

manner with no centralized controllers [1]. However, the benefits in terms of spectrum usage come at the price of higher complexity in the effective deployment of CRAHNs. The problems of how such a network self-organizes and adapts to the dynamic topologies changes and varying spectrum availability are some of the key distinguishing factors. Current research on CRAHNs demonstrates the need for a novel generation of adaptive protocols and algorithms, which should cope with the high fluctuation in the spectrum availability, as well as with diverse QoS requirements [16].

Reinforcement Learning (RL) is a biologically inspired machine learning technique (ML), in which an agent acquires its knowledge through trial-and-error interactions with its environment [2,12]. At each step, the agent performs an action and gets a feedback from the environment, which can be used to optimize its behaviour in the future. The dynamic interaction with the environment and the adaptivity of the learning process are two of the main features which make RL techniques appealing for CRAHNs applications, mainly for routing and spectrum decision tasks [7,9]. In some cases, the RL-based solutions are proved to work better than traditional solutions [3,17,22]. However, a comprehensive analysis of benefits and risks of RL techniques over CRAHNs is still missing.

This paper investigates the application of RL techniques over CRAHNs, as a general framework for the deployment of intelligent and reconfigurable radio networks. We provide three main contributions in this research field. First, we analyze benefits and drawbacks of RL approach over CRAHNs (Section 2), by identifying the RL techniques which are best suitable for protocol design (Section 3). Second, we review existing RL-based proposals, for these CRAHNs issues: routing (Section 4.1), spectrum sensing (Section 4.2) and spectrum decision (Section 4.3). Most of the solutions are single-layer, and try to learn the optimal configuration of a single parameter (e.g. spectrum in the spectrum decision problem, next-hop node in routing). Third, we show in Section 5 how RL techniques can be applied for solving complex problems like interference-control in CRAHNs, where each CR user should learn the optimal combination of multiple parameters (e.g. power and spectrum), in a distributed way. We conclude our work in Section 6.

2 RLs over CRAHNs: Benefits and Challenges

Cognitive Radio Ad Hoc Networks (CRAHNs) are multi-hop wireless networks, composed of two kind of users: cognitive radio (CR) users and primary users (PUs) [1]. PUs have license to access the licensed spectrum. CR users may opportunistically transmit in the spectrum holes. The effective deployment of multi-hop CRAHNs depends on the design of efficient spectrum sensing and selection techniques, and of novel routing and transport layer protocols [16]. RL techniques are suitable for protocol design in CRAHNs, as demonstrated by numerous prior works in the literature (Section 4). The main benefits provided by the RL approach are:

- *Adaptivity.* RL techniques help a node to adapt its behavior to the dynamic spectrum environment, by combining exploration and exploitation actions. This is required, for example, by routing protocols, which must identify the best path between CR source-destination nodes, while the quality of each path may dynamically change over time as a function of PU activity, intra-network CR interference, and so on [9].
- *Network-awareness.* RL allows to implement a spectrum-aware paradigm of communication. Many different factors, such as the radio resources and the channel heterogeneity affect the CRAHNs performance in a complex way. Instead of addressing a single factor at a time, a RL agent can observe all the factors as a state, receive an aggregate feedback (e.g. the cost of each transmission) and optimize a general goal as a whole, e.g. throughput [22].
- *Distributed implementation.* In most cases, RL techniques provide a simple yet effective modeling approach [2]. Moreover, multi-agent RL algorithms [5] can be deployed by each node of the network in a distributed way, introducing a limited network overhead.

At the same time, the main drawbacks of RL techniques over CRAHNs are:

- *Random fluctuations.* RL techniques may force a CR node to perform random actions as it learns about the environment, so that feedbacks about the cost of each state-action pair are collected. The benefits of exploration constitute a trade-off with the increased cost of the learning process, which may select suboptimal actions, and thus lead to temporary performance degradation.
- *Slow convergence.* Many RL techniques (specially Time Discounted methods [2]) guarantee convergence to the optimal policy only if each action is executed in each state an infinite number of times. This is clearly not realistic for wireless applications. However, we also highlight that convergence is not fundamental for CRAHNs protocols, due to the non-stationary characteristics of the network environment.

Many different RL algorithms have been proposed in the literature [2]. In some cases, the difficulty relies in identifying the algorithm which best applies to the CRAHNs problem. To this aim, in Section 4 we review the basic RL model and discuss the RL algorithms which are suitable for CRAHNs issues.

3 RL Techniques

In the RL model [2,12], an agent interacts with its environment over a potential infinite sequence of discrete time steps $t = 1, 2, 3, \ldots$. At each step, it observes the current environment, selects a possible action and receives a reward from the environment for that specific action. The goal of the agent is to decide the sequences of actions maximizing some cumulative measures of the rewards, over time. RL model is defined by a Markov Decision Process (MDP), consisting in:

- A discrete *set of states* S which constitute the environment.
- A discrete *set of actions* A.

– A *reward function* $R : S \times A \longrightarrow \Re$.
– A *state-distribution function* $T : S \times A \times S \longrightarrow [0, 1]$.

The environment is defined by a discrete set of states (i.e. S) and must be observable (or partially observable) by the RL agent. The reward function R specifies the expected instantaneous reward, as a function of the current state and of the action performed. For each tuple $< s, a, s' >$, the state-distribution function $T(s, a, s')$ gives the probability to transit from state s to state s' after executing action a. Additionally, the policy π defines the mapping between the states and actions, for each step t. The goal of the agent is to find the optimal policy, defined according to different reward models [2,12]. In the *infinite-horizon discount model*, the policy π attempts to maximize the long-run expected reward, but discounts the rewards received in the future, i.e.:

$$E(\sum_{t=0}^{\infty} \gamma^t r_t) \tag{1}$$

where $0 \leq \gamma \leq 1$ is a discount factor which determines the weight of future rewards. If $\gamma = 0$, the agent aims only at maximizing its immediate reward.

Most of the RL algorithms are based on the concept of state-value function (V) and state-action function (Q). The state value function $V^\pi(s)$ defines the expected reward when executing the policy π, from state s. Analogously, the state-action function $Q^\pi(a, s)$ gives the expected reward when the agent is in state s, executes action a and then follows the policy π. Several RL techniques proposed in the literature differ in the way $V^\pi(s)$ and $Q^\pi(a, s)$ functions are updated at each step, till the optimal policy π^* is found [2,12].

Most suitable RL techniques for CRAHNs are:

Model-based learning. These algorithms requires a model of the environment, i.e. the reward R and state-distribution T functions. For example, the Dyna architecture [15] uses experience to build a model of the environment, and through the model it adjusts the policy in use. It learns T and R by incrementing statistics after each transition from s to s' and averaging the reward $R(s, a)$. Model-based learning techniques have been applied for routing in wireless ad hoc networks [8], modelling the quality of each link as a stochastic process. Generally speaking, it might be difficult to learn the models of the environment in CRAHNs, due to the large number of parameters affecting the network performance.

Q-learning. Q-learning [23] is an on-line RL algorithm which attempts to estimate the optimal action-state function $Q(s, a)$, without requiring a model of the environment and a representation of the policy in use i.e. π. Let the agent be in state s, execute action a and then observe the reward r and the next state s'. Q-learning updates the Q function in this way:

$$Q(s, a) \rightarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s', a') - Q(s, a)], \tag{2}$$

where γ is the discount factor discussed above and α is a parameter tuning the speed of learning. At each state s, the agent chooses the action a which

maximizes the $Q(s, a)$ values (*exploitation*), or chooses randomly among the available actions (*exploration*). Q-learning can be easily implemented over distributed systems, with low overhead in terms of communication, computation and memory usage. As a result, Q-Learning has been applied to different problems over CRAHNs, e.g. PU sensing detection [3], spectrum selection [24] and routing [6]. The main drawback is the speed of learning, which depends on the accurate tuning of the γ and α parameters.

Dual RL. This algorithm is very similar to the Q-learning scheme, but it updates the $Q(s, a)$ values also considering the value of the previous state, instead of next state only [14]. Dual RL techniques have been applied to routing problems in CRAHNs, so that each time a packet is transmitted on a link the $Q(s, a)$ are updated at both transmitter and receivers nodes [22]. Dual RL enhances Q-learning in terms of speed of convergence, but it introduces some additional overheads for its implementation.

Multi-Agent Learning. These techniques extend classical RL algorithms, in a system of homogeneous agents which have system-wide optimization goals. Cooperative RL techniques attempt to coordinate the behavior of the agents, so that a coherent joint behavior is observed [5]. Many CRAHNs problems can be formulated as cooperative problems. For instance, CR users should distributively learn an optimal strategy to control the interference at the primary receivers, at maintain it under a given threshold [10]. The main drawbacks of this approach are: the additional overhead, required for the agent cooperation, and the complexity in implementation, which may also affect the converge of the learning process. At present, very few applications have been proposed for wireless systems [10].

4 RL Applications for CRAHNs

In this section, we review existing RL-applications from the viewpoint of higher level protocol implementations in CRAHNs: (*i*) routing (Section 4.1), (*ii*) link layer spectrum sensing (Section 4.2) and (*iii*) link layer spectrum decision (Section 4.3). For each domain, we analyze the RL-formulation, we shortly review the most significant RL-based protocols proposed in the literature and we discuss advantages and open issues.

4.1 Routing Protocols

Problem Formulation. In CRAHNs, routing protocols have the twofold goal of (*i*) discovering a path from a CR source to a CR destination node, by avoiding regions characterized by PUs activity, and of (*ii*) accounting the spectrum to be used on each intermediate link. Moreover, they should be able to cope with the dynamics of CRAHNs, e.g. mobility, PU interference, variable link quality. As a result, spectrum awareness and re-configurability are two important requirements for routing protocols over CRAHNs [16]. The routing process can be modeled as a RL task, in which the CR source node must learn the optimal

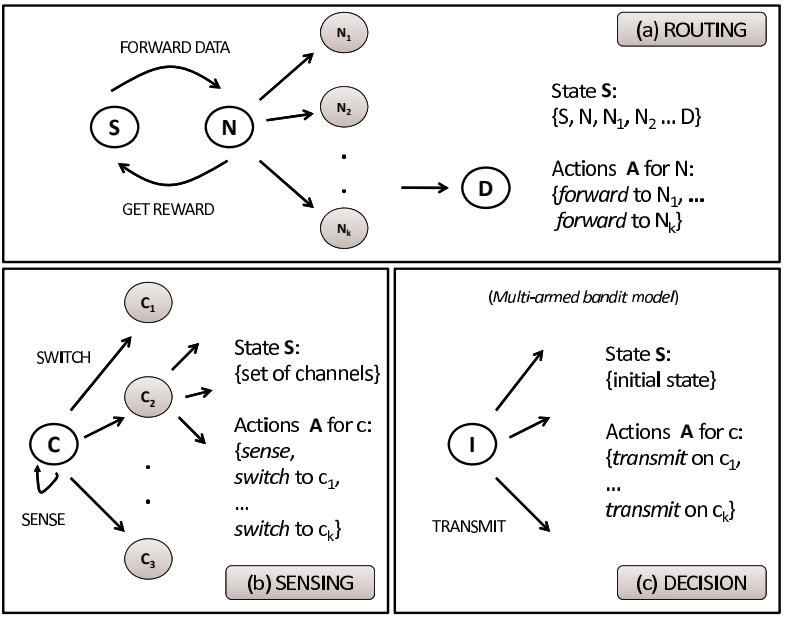


Fig. 1. The MDP process for a generic routing algorithm (a), spectrum sensing problem (b) and spectrum detection problem (c)

path toward the destination by a trial and error interaction [9]. For each data transmission, the CR node receives a reward which is an an estimate of the cost of forwarding, e.g. mean-access delay or amount of energy consumed. The MDP process for a generic routing protocol is depicted in Figure 1(a).

RL-based approaches. Q-routing [18] is a routing protocol for dynamically changing networks, which has been applied in wireless ad hoc networks [6]. It is an application of the Q-Learning scheme, where each node x:

- Has a *set of actions* A, which correspond to the set of neighbors of node x.
- Receives a *reward* r after forwarding a packet to a neighbor y, which is an estimation of the transmission delay between node x and y.
- Maintains a *table* of Q values for each destination d, where the entry $Q_d(y, d)$ is the *expected delivery time* to d using next-hop node y.

After forwarding a packet to node y with destination d, node x updates its Q-table entry in this way:

$$Q_d(x, y) = q_x + \delta + min_z Q_d(y, z) \tag{3}$$

where q_x is the time spent by the packet in the queue of x, δ is the transmission delay and $min_z Q_d(y, z)$ is the best delivery time for node y and for destination d. In [6], Q-routing is evaluated for classical wireless ad hoc networks, and

the Q-table is enlarged to take into account different parameters, e.g. network connectivity and residual energy of each node. In [21], the authors propose the LQ-routing protocol, which uses the conventional Q-routing approach, but also introduces the notion of route lifetime to represent the stability of routes connecting to the destination. SAMPLE [8] protocol models the routing process as a collaborative RL-task among the nodes of the network. The main features of the SAMPLE protocol are: (*i*) it keeps a statistical model of the quality of each link, based on the ratio of successful over attempted unicast transmissions, (*ii*) it uses a decay function, so that routes which are not advertised are gradually degraded and (*iii*) it exploits piggybacking techniques to disseminate Q-values inside the network. The authors of [22] propose two RL-based spectrum-aware routing protocols for CRAHNs. Here, the CR nodes store a table of Q-values that estimate the number of available channels on the routes, as a function of the PUs activity. The Q-values are updated after each successful transmission, using a Dual RL technique. Then, each CR node forwards a packet to the next-hop node which guarantees more available channels toward to the destination.
Discussion. All the Q-routing variants discussed so far are shown to improve the performance of classical ad hoc routing schemes (e.g. AODV, DSR, DSDV), in most of the evaluated scenarios. However, except for [8,22], they do not take into account spectrum considerations, which are fundamental for routing over CRAHNs. Moreover, none of the previous approach integrates channel and next-hop selection into the learning process.

4.2 Spectrum Sensing

Problem Formulation. In CRAHNs, each CR user should perform periodic sensing on the channel in use, to avoid collisions with PUs. Additionally, it must observe the available spectral resources in all the other bands, in order to detect spectrum holes and to switch to a new band, if necessary [1]. While sensing, a CR user can not use the channel for communicating with other nodes. Finding the optimal tradeoff between channel sensing and channel exploitation involves a learning process in which each node should determine the frequency and duration of sensing, on each channel. Spectral resource detection can be modeled as a RL-problem, through an MDP (Figure 1(b)). The available number of licensed bands constitute the set of states S of the MDP. At each step, the CR user can perform one of these actions: (*i*) transmit on the current band, (*ii*) sense the current or another band or (*iii*) switch to another band.

RL-based approaches. In [3], the authors solve the spectral detection problem discussed above by using an actor-critic RL approach [2]. After each action, a CR node receives a reward which is: the number of available subcarriers in case of transmission, 0 in case of sensing and a fixed penalty in case of switching. Using the reward, the CR node updates the state values e.g. $V(s)$, and computes the reliability of each state, which is then used by the actor to update the current policy π. Error-free PU detection is assumed by the authors of [3]. In [20], the Q-learning technique is used to detect the presence of PU on the

spectrum in use by CR users, in order to reduce the occurrence of mis-detection and false detection events. In the implementation proposed in [20], each Q-value represents the signal strength detected during sensing on a given channel.

Discussion. The simulation results in [3] show that their proposed RL-scheme can quickly converge to the optimal solution in scenarios with stationary PU activity, but do not address the environments with dynamically changing availability of the spectrum resources.

4.3 Spectrum Decision

Problem Formulation. In CRAHN, each CR user should determine a channel where to transmit, given the twofold objective of (*i*) maximizing the overall performance for stable QoS provisioning and of (*ii*) mitigating the interference to PUs. The spectrum heterogeneity implies that different channels might be characterized by different properties (in terms of e.g. transmitting range, data-rate, Bit Error Rate) and varying activity patterns by PUs, over time. Spectrum decision can be modeled as a MDP process for each CR user i, as shown in Figure 1(c). The reward can be a local feedback of the node or a network-wide metric (e.g. total interference to PU receivers).

RL-based approaches. Q-learning based spectrum allocation has been extensively investigated for cellular networks [4,19]. In CRAHNs, the authors of [24] consider the RL-formulation discussed above, and assign rewards to the CR users after each data transmission. For each positive data packet transmission, there is a positive constant value of $+RW$, otherwise a negative constant value $-CT$ is incurred. In practice, the value of RW and CT should be the amount of revenue and cost that a network operator earns or incurs for each data successful transmission or failure [24]. In [11], the authors consider a network case composed by a set of transmitting-receiving pairs of nodes. At each step, the Signal-to-Noise Ratio (SNR) is computed at the receivers node. If the SNR value is higher than a given threshold, then the transmitter node increases its Q-value of a fixed weight factor, otherwise it applies a penalty and quits the current spectrum. In [17], an analytical modeling of dynamic spectrum allocation over CRAHNs is proposed. Simulation results show that Q-learning enhances a random spectrum allocation scheme, and its performance are comparable with those of the analytical model. In [10], the authors consider a secondary CR system based on the IEEE 802.22 standard, and apply RL techniques to learn how to control the aggregated interference at the PUs receivers. To this aim, the authors propose a Q-learning based distributed scheme. However, the overhead of the proposed solution is not shown in [10].

Discussion. All the Q-learning solution described so far are shown to provide adaptive and dynamic channel selection for CRAHNs. Performance results show that in some cases RL-based approaches guarantee higher performance than classical distributed channel selection scheme. However, none of the previous

scheme take into account the PUs interference into the MDP process. Moreover, they do not provide an analytical modeling of the reward function, which is usually modeled through a scalar vector [11,24].

5 RL Case Study: Joint Spectrum and Power Allocation

In this section, we consider a case study of RL techniques for wireless ad hoc cognitive sensors networks. They are composed of small, resource constrained nodes, which are suited for monitoring, data gathering, and surveillance operations. In contrast to the classical sensor network, our proposed cognitive sensors can intelligently choose their spectrum for transmission, transmit power, and thereby support high bandwidth applications. This network model poses several challenges. Firstly, the frequency space may be large, composed of several dozens of channels. Moreover, the sensors are typically deployed in large densities, and their individual transmit powers directly decide the level of interference to several nodes in the neighborhood. Here, we show how the spectrum and power allocation problem can be modeled through an multi-agent RL model, so that each CR sensor (also called as agent here) locally adjusts its choice of spectrum, and its transmit power, subject to connectivity and interference constraints.

We assume time to be slotted, and at the start of each slot, the sensor senses the spectrum with perfect accuracy. If the spectrum is available, it goes ahead with the transmission. Now, this transmission may be successful, or it may result in a collision. Through receiver feedback, we allow the sender node to be informed if the collision was a result of intra-CR network interference, or due to simultaneous transmission by a neighboring PU. Based on the result of the transmission, an appropriate reward is assigned to the choice of the state, which in turn, determines future choices of spectrum and power. Considering the RL model described in Section 3, we define the *state* of an agent as the current spectrum and power value of its transmission. We therefore define the state of the system at time t, denoted s_t as:

$$s_t = (\overrightarrow{s_p}; \overrightarrow{p_w})_t \tag{4}$$

where $\overrightarrow{s_p}$ is a vector of spectrums and $\overrightarrow{p_w}$ is a vector of power values across all agents. Analogously, we define the set of action a_t at time t as:

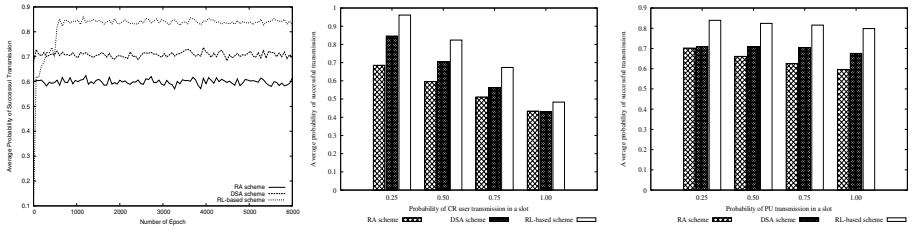
$$a_t = \overrightarrow{k_t}; \tag{5}$$

where $\overrightarrow{k}$ is a vector of actions across all agents. Here k_i is the action of the i-th agent and $k_i = \{jump_{spectrum}; jump_{power}\}$, e.g. each agent can switch from its current spectrum to a new available spectrum, or switch from its current power value to another power value. After each transmission, a CR user receives a reward, which is used to adjust the current policy π. We consider the following different reward r values for different network conditions [13]:

1. *Interference between PU and Sensors*: When the licensed user and the sensor transmit concurrently in the same spectrum, in the same slot, the receiving sensor may experience a collision. Moreover, the PU receiver too may be unable to receive the PU transmission correctly, which is a serious degradation of performance, and must be avoided. Thus, we allocate a comparatively higher penalty P_{PU}, which is equal to -15 in our experiments.
2. *Intra-sensor network Collision*: If the collision is caused by simultaneous transmissions by multiple sensor nodes, then we apply a fixed penalty P_{COL}, equal to -5 in our experiments.
3. *Channel Induced Errors*: The inherent fluctuations in the channel quality, sudden drops in signal strength caused by fading, and noise characteristics result in channel induced bit-flips that may cause the packet to be dropped. We apply the same penalty P_{COL} of the previous discussed case here, thereby forcing the sender to switch to a more robust channel.
4. *Link Disconnection*: If the received power (P^j_{rx}) is less than the threshold of the receiver P_{rth} (here, assumed as -85 dBm), then all the packets are dropped. In such cases, the sender should quickly increase its choice of transmit power so that the link can be re-established. We address this case by applying a fixed penalties P_{LD}, which is equal to -20 in our experiments.
5. *Successful Transmission*: If none of the above conditions are observed to be true in the given transmission slot, then packet is successfully transmitted from the sender to receiver. Since the actual combination of spectrum/power produced a successful outcome, we encourage the agent to stay in the current state through a positive reward (e.g +5 in our experiments).

The learning algorithm is based on a revised version of the Q-learning algorithm described in Section 3. Details of our learning algorithm can be found in [13]. In this paper, we focus on the comparison between the RL-based approach with other classical spectrum and power selection schemes for a random topology CRAHN with 100 CR users. In the Random (RA) scheme, each CR user selects a random combination of spectrum and power in each slot. In the Dynamic Spectrum Assignment (DSA) scheme, each CR user chooses the less interfered spectrum in its 1-hop neighborhood, and switches to a new spectrum band in case of PU detection. Moreover, each CR user chooses the minimum power level, which provides network connectivity with the receiver node. Each CR user can switch among 10 available spectrum bands, where each band has different Bit Error Rate (BER) and transmitting range characteristics. The permissible power values for the CR users are uniformly distributed on 10 discrete levels between $0.5m - 4$ mW. We consider a CRAHN with 25 PUs, where each PU is randomly assigned one default channel in which it stays. We consider two network configurations: (*i*) *stationary* PU activity, i.e. each PU is always active on its band, and (*ii*) *dynamic* PU activity, i.e. each PU is active on its band with a given probability P_{active}. Fig 2(a) shows the average probability of successful transmission over simulation time, in the stationary case. The time scale on the *x-axis* is represented by epochs, each of which is composed of 50 time slots. From Fig. 2(a), we can see that the RL-based approach quickly converges after a learning phase, and

provides higher performance than the non-RL schemes. Fig. 2(b) shows the average probability of successful transmission for the stationary case, as a function of the CR system load. The performance improvement of the RL-based scheme is even more consistent under higher saturation conditions. Fig 2(c) shows the average probability of successful transmission for the non-stationary case, where we vary the P_{active} parameter on the *x-axis*. In this case, the CR users should dynamically adapt their behavior based on the spectrum availability. Again, the RL-approach shows its suitability for operating in dynamically environment.



(a) (b) (c)

Fig. 2. The average probability of successful transmission for the stationary case is shown in Figure 2(a) and 2(b). The average probability of successful transmission as a function of the PU activity is shown in Figure 2(c).

6 Conclusions

In this paper, we addressed the application of Reinforcement Learning (RL) techniques for distributed network protocols in CRAHNs. We surveyed several RL techniques and their applications in three different CRAHNs domains, i.e. routing, spectrum sensing and spectrum decision. We found that in many cases RL-based schemes provide high adaptability to the varying spectrum conditions. However, only few protocols have been evaluated against classical schemes for CRAHNs. Moreover, few of them address joint optimization of cross-layer parameters. To this aim, we proposed our RL-framework for the joint selection of optimal spectrum and transmit power in CRAHNs. The proposed framework have been compared with other non-learning scheme. Simulation results show the performance improvement of the RL-based scheme under both stationary and dynamic spectrum environments.

References

1. Akyildiz, I., Lee, W.Y., Chowdhury, K.R.: CRAHNs: Cognitive Radio Ad Hoc Networks. Ad Hoc Networks Journal 7(5), 810–836 (2009)
2. Barto, A.G., Sutton, R.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

3. Berhold, U., Fu, F., Van Der Schaar, M., Jondral, F.K.: Detection of Spectral Resources in Cognitive Radios Using Reinforcement Learning. In: Proc. of Dyspan'08, Chicago, pp. 1–5 (2008)
4. Bernardo, F., Augusti, R., Perez-Romero, J., Sallent, O.: Distributed Spectrum Management based on Reinforcement Learning. In: Proc. of CROWNCOM'09, Hannover (2009)
5. Busoniu, L., Babuska, R., De Schutter, B.: A Comprehensive Survey of Multi-Agent Reinforcement Learning. IEEE Transactions on Systems, Man, and Cybernetics 38(2), 156–172 (2008)
6. Chetret, D., Tham, C., Wong, L.: Reinforcement Learning and CMAC-based Adaptive Routing for MANETs. In: Proc. of ICON'04, pp. 540–544 (2004)
7. Clancy, C., Hecker, J., Stuntebeck, E., O'Shea, T.: Applications of Machine Learning to Cognitive Radio Networks. Wireless Communications 14(4), 47–52 (2007)
8. Dowling, J., Curran, E., Cunningham, R., Cahill, V.: Using Feedback in Collaborative Reinforcement Learning to Adaptively Optimize MANET Routing. IEEE Transactions on Systems, Man and Cybernetics 35(3), 360–372 (2005)
9. Forster, A.: Machine Learning Techniques Applied for Wireless Ad-Hoc Networks: Guide and Survey. In: Proc. of ISSNIP'07, Melbourne, pp. 367–370 (2007)
10. Galindo-Serrano, A., Giupponi, L.: Aggregated Interference Control for Cognitive Radio Networks Based on Multi-Agent Learning. In: Proc. of CROWNCOM'09, Hannover (2009)
11. Jiang, T., Grace, D., Liu, Y.: Performance of Cognitive Radio Reinforcement Spectrum Sharing Using Different Weighting Factors. In: Proc. of CHINACOM'08, Hangzhou, pp. 1195–1199 (2008)
12. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4(1), 237–285 (1996)
13. Wu, C., Chowdhury, K.R., Di Felice, M., Meleis, W.: Spectrum Management of Cognitive Radio Using Multi-agent Reinforcement Learning. To appear on Proc. of AAMAS'10, Toronto (2010)
14. Kumar, S., Miikkulainen, R.: Dual Reinforcement Q-Routing: An Online Adaptive Routing Algorithm. In: Proc. of Artificial Neural Networks in Engineering Conference, St. Louis, pp. 231–238 (1997)
15. Kuvayev, L., Sutton, R.: Model-based Reinforcement Learning with an approximate, Learned Model. In: Proc. of the Yale Workshop on Adaptive and Learning Systems, Yale, pp. 101–105 (1996)
16. Kyasanur, P., Vaidya, N.H.: Protocol Design Challenges for Multi-hop Dynamic Spectrum Access Networks. In: Proc. of DySPAN'05, Baltimore, pp. 645–648 (2005)
17. Lim, K.A., Komisarczuk, P., Teal, P.D.: Performance Analysis of Reinforcement Learning for Achieving Context-Awareness and Intelligence in Cognitive Radio Networks. In: Proc. of WLN'09, Zurich, pp. 11–35 (1999)
18. Litman, M., Boyan, J.: Packet routing in dynamically changing networks: a reinforcement learning approach. Advances in Neural Information Processing Systems 7(1), 671–678 (1994)
19. Nie, J., Haykin, S.: A Dynamic Channel Assignment Policy Through Q-Learning. IEEE Transactions on Neural Networks 10(1), 1443–1455 (1999)
20. Reddy, Y.B.: Detecting Primary Signals for Efficient Utilization of Spectrum Using Q-Learning. In: Proc. of ITNG'08, Las Vegas, pp. 360–365 (2008)

21. Tao, T., Tagashira, S., Fujita, S.: LQ-Routing Protocol for Mobile Ad Hoc Networks. In: Proc. of ICIS'05, Washington, pp. 441–446 (2005)
22. Wahab, B., Yang, Y., Fan, Z., Sooriyabandara, M.: Reinforcement Learning Based Spectrum-aware Routing in Multi-hop Cognitive Radio Networks. In: Proc. of CROWNCOM'09, Hannover (2009)
23. Watkins, C.: Learning from delayed rewards. PhD thesis, Cambridge, UK (1989)
24. Yau, K.-L.A., Komisarczuk, P., Teal, P.D.: A Context-aware and Intelligent Dynamic Channel Selection Scheme for Cognitive Radio Networks. In: Proc. of CROWNCOM'09, Hannover (2009)