Spectrum Management of Cognitive Radio Using Multi-agent Reinforcement Learning

Cheng Wu
Northeastern University
360 Huntington Avenue
Boston, MA, U.S.A.
cwu@ece.neu.edu

Kaushik Chowdhury
Northeastern University
360 Huntington Avenue
Boston, MA, U.S.A.
krc@ece.neu.edu

Marco Di Felice
University of Bologna
Via M. Anteo Zamboni, 7
Bologna, Italy
difelice@cs.unibo.it

Waleed Meleis
Northeastern University
360 Huntington Avenue
Boston, MA, U.S.A.
meleis@ece.neu.edu

ABSTRACT

Wireless cognitive radio (CR) is a newly emerging paradigm that attempts to opportunistically transmit in licensed frequencies, without affecting the pre-assigned users of these bands. To enable this functionality, such a radio must predict its operational parameters, such as transmit power and spectrum. These tasks, collectively called *spectrum management*, is difficult to achieve in a dynamic distributed environment, in which CR users may only take local decisions, and react to the environmental changes. In this paper, we introduce a multi-agent reinforcement learning approach based spectrum management. Our approach uses value functions to evaluate the desirability of choosing different transmission parameters, and enables efficient assignment of spectrums and transmit powers by maximizing long-term reward. We then investigate various real-world scenarios, and compare the communication performance using different sets of learning parameters. We also apply Kanerva-based function approximation to improve our approach's ability to handle large cognitive radio networks and evaluate its effect on communication performance. We conclude that our reinforcement learning based spectrum management can significantly reduce the interference to the licensed users, while maintaining a high probability of successful transmissions in a cognitive radio ad hoc network.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Performance, Experimentation

Cite as: Spectrum Management of Cognitive Radio Using Multi-agent Reinforcement Learning, C. Wu, K. Chowdhury, M. D. Felice and W. Meleis, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. . 1705-1712

Keywords

Reinforcement learning, Cognitive Radio, Multi-agent System, Function Approximation, Spectrum Management

1. INTRODUCTION

Wireless spectrum is a costly resource which is licensed by governmental agencies to operators for long periods of use, often spanning several decades. However, a large portion of the assigned spectrum is used sporadically leading to under-utilization of valuable frequency resources. To address this critical problem, Federal Communications Commission (FCC) has recently approved the use of unlicensed devices in licensed bands [6]. Consequently, dynamic spectrum access techniques are proposed to solve these current spectrum inefficiency problems [3, 8, 12]. This new area of research foresees the development of cognitive radio (CR) networks to further improve spectrum efficiency.

The basic idea of CR networks is that the unlicensed devices (also called cognitive radio users) need to vacate the band once the licensed devices (also known as primary users) are detected. CR networks, however, impose a great challenge due to the high fluctuation in the available spectrum as well as diverse quality-of-service (QoS) requirements. Specifically in cognitive radio ad-hoc networks, the distributed multi-hop architecture, the dynamic network topology, and the time and location varying spectrum availability are some of the key distinguishing factors. As the CR network must appropriately choose its transmission parameters based on limited environmental information, it must be able to learn from its experience, and adapt its functioning. The challenge necessitates novel design techniques that simultaneously integrate theoretical research on reinforcement learning and multi-agent interaction with systems level network design.

Reinforcement learning [17] is a learning paradigm, which was inspired by psychological learning theory from biology [18]. Within an environment, a learning agent attempts to perform optimal actions to maximize long-term rewards achieved by interacting with the environment. The long-term reward is the expected accumulated reward that the agent expects to receive in the future under the policy, which can be specified by a value function. The value function is often a look-up table that directly stores values of states.

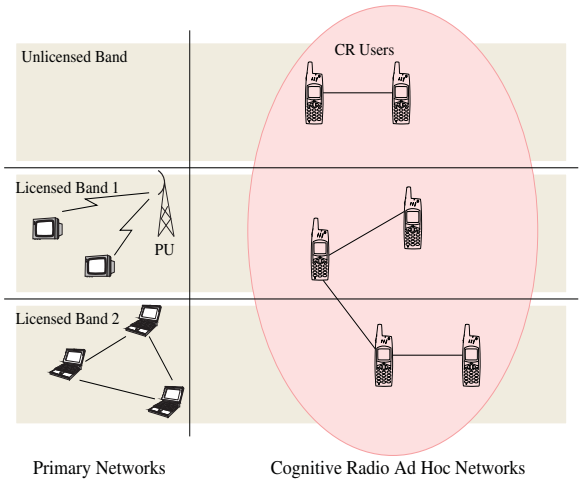

Figure 1: The CR ad hoc architecture

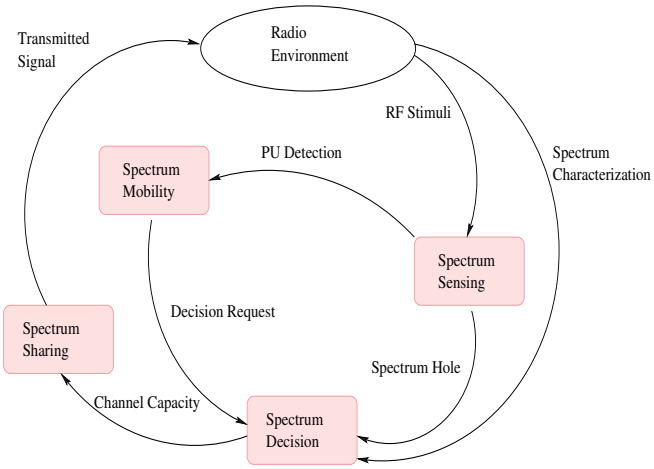

Figure 2: The cognitive radio cycle for the CR ad hoc architecture

The study of multi-agent systems [20] allows us to build complex systems composed of multiple interacting intelligent agents. Each agent in the system can sense the environment and achieve its own local knowledge and experience. The agent can then take select behaviors based on local information and attempt to maximize the global performance of the system. A typical multi-agent system is decentralized, without a designated controlling agent.

A challenge that CR networks face is that the performance of CR networks decreases sharply as the size of network increases. Cognitive radio ad-hoc networks typically have a very large number of unlicensed and licensed users, and the range of possible transmission parameters is wide. There is therefore a need for algorithms to apply function approximation techniques to scale up reinforcement learning for large-scale CR problems.

Function approximation [16] is a generalization technique which has been widely used to solve large-scale learning problems with huge state and action spaces [9, 7, 14]. In general, function approximation uses examples of the desired value function to reconstruct an approximation of this function and compute an estimate of the desired value from the approximation function. Within the context of reinforcement learning, instead of using a look-up table, function approximation generalizes the function values of the states that have not been previously visited from known function values of its neighboring states.

In this paper, we focus on CR ad hoc networks with decentralized control [1] [2]. The architecture of a CR ad hoc network, shown in Figure 1, can be partitioned into two groupsof users: the *primary network* and the *CR network* components. The primary network is composed of *primary users* (PUs) that have a license to operate in a certain spectrum band. The CR network is composed of *cognitive radio users* (CR users) that share wireless channels with licensed users that already have an assigned spectrum.

Under this architecture, the CR users need to continuously monitor spectrum for the presence of the primary users and reconfigure the radio front-end according to the demands and requirements of the higher layers. This capability can be realized, as shown in Figure 2, by the cognitive cycle composed of the following spectrum functions: (1) determining the portions of the spectrum currently available (*Spectrum sensing*), (2) selecting the best available channel (*Spectrum decision*), (3) coordinating access to this channel with other users (*Spectrum sharing*), and (4) effectively vacating the channel when a licensed user is detected (*Spectrum mobility*).

By leveraging reinforcement learning, the tasks of spectrum sensing and sharing could be undertaken effectively, wherein specific interferer transmission patterns may be distinguished from each other. Similarly, by considering each CR user as an agent and exploiting multi-agent interaction algorithms, the network can converge to the fair sharing of the spectrum and fast recovery if the spectrum is reclaimed by the PU.

This paper is organized as follows. In Section 2, we formulate the cognitive radio problem. In Section 3, we describe our reinforcement learning-based cognitive radio. The experimental results of our simulation are given in Section 4. In Section 5, we discuss a Kanerva-based function approximation technique for our reinforcement learning based cognitive radio. The effect of function approximation is evaluated in Section 5.1. We conclude the paper in Section 6.

2. PROBLEM FORMULATION

In this paper, we assume that our network consists of a collection of PUs and CR users, each of which is paired with another user to form transmitter-receiver pairs. We also assume perfect sensing that allows each CR user to correctly infer the presence of the PU if the former lies within the PU's transmission range. Each CR user must now undertake decisions about spectrum and transmission power independently of the others users in the neighborhood.

A choice of spectrum by CR user i is essentially the choice of the frequency represented by $f^i \in F$, the set of available frequencies. The CR users continuously monitor the spectrum that they choose in each time slot. The channels chosen are discrete, and a jump from any channel to another is possible in consecutive time slots.

The transmit power chosen by the CR user i is given by P^i_{tx}. The transmission range and interference range are respectively represented by R_t and R_i. Our simulator uses

the free-space path loss equation to calculate the attenuated power incident at the receiver, denoted P_{rx}^{j}. Thus,

$$P_{rx}^{j} = \alpha_c \cdot P_{tx}^{i} \left\{ D^{i} \right\}^{-\beta},$$

where the path loss exponent $\beta = 2$. The constant

$$\alpha_c = \frac{c^2}{(4\pi f_c)^2},$$

where the the speed of light $c = 3 \times 10^8$m/s, and f_c is the frequency at which the transmission occurs. The transmit power values are real numbers in a pre-decided range, and a jump from any given value to another is possible in consecutive time slots.

3. REINFORCEMENT LEARNING BASED COGNITIVE RADIO

Reinforcement learning enables learning from feedback received through interactions with an external environment. The classic reinforcement learning algorithm is implemented as follows. At each time t, the agent perceives its current state $s_t \in S$ and the set of possible actions A_{s_t}. The agent chooses an action $a \in A_{s_t}$ and receives from the environment a new state s_{t+1} and a reward r_{t+1}. Based on these interactions, the reinforcement learning agent must develop a policy $\pi : S \rightarrow A$ which maximizes the long-term reward $R = \sum_t \gamma r_t$ for Markov Decision Processes (MDPs), where $0 \leq \gamma \leq 1$ is a discounting factor for subsequent rewards.

One of the most successful reinforcement learning algorithm is Q-learning [19]. This approach uses a simple value iteration update process. At time t, for each state s_t and each action a_t, the algorithm calculates an update to its expected discounted reward, $Q(s_t, a_t)$ as follows:

$$\begin{aligned} Q(s_t, a_t) \quad \leftarrow \quad & Q(s_t, a_t) + \\ & \alpha[r_t + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \end{aligned}$$

where r_t is an immediate reward at time t, $\alpha_t(s, a)$ is the learning rate such that $0 \leq \alpha_t(s, a) \leq 1$, and γ is the discount factor such that $0 \leq \gamma \leq 1$. Q-learning stores the state-action values in a table.

3.1 Application to cognitive radio

In cognitive radio network, if we consider each cognitive user to be an agent and the wireless network to be the external environment, cognitive radio can be formulated as a system in which communicating agents sense their environment, learn, and adjust their transmission parameters to maximize their communication performance. This formulation fits well within the context of reinforcement learning.

Figure 3 gives an overview of how we apply reinforcement learning to cognitive radio. Each cognitive user acts as an agent using reinforcement learning. These agents do spectrum sensing and perceive their current states, i.e., spectrums and transmission powers. They then make spectrum decisions and use spectrum mobility to choose actions, i.e. switch channels or change their power value. Finally, the agents use spectrum sharing to transmit signals. Through interaction with the radio environment, these agents receive transmission rewards which are used as the inputs for the next sensing and transmission cycle.

A *state* in reinforcement learning is some information that an agent can perceive within the environment. In RL-based

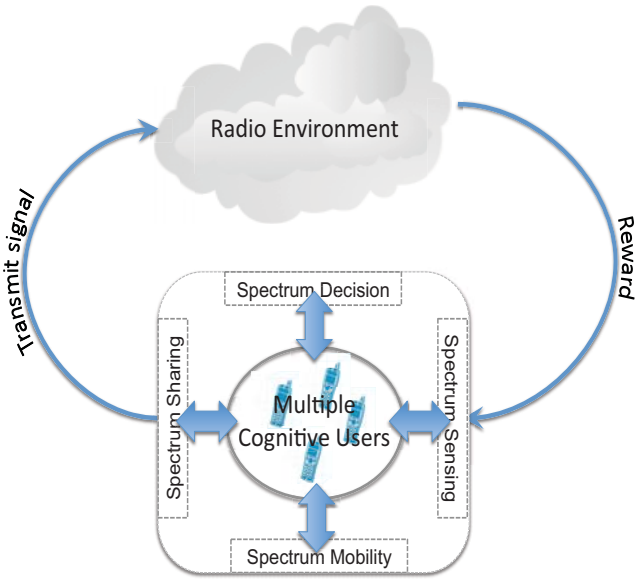

Figure 3: Multi-agent reinforcement learning based cognitive radio.

cognitive radio, the state of an agent is the current spectrum and power value of its transmission. The state of the multi-agent system includes the state of every agent. We therefore define the state of the system at time t, denoted s_t, as

$$s_t = (\vec{sp}, \vec{pw})_t,$$

where $\vec{sp}$ is a vector of spectrums and $\vec{pw}$ is a vector of power values across all agents. Here sp_i and pw_i are the spectrum and power value of the ith agent and $sp_i \in SP$ and $pw_j \in PW$. Normally, if there are M spectrums and N power values, we can using the index to specify these spectrums and power values. In this way, we have $SP = \{1, 2, ..., m\}$ and $PW = \{1, 2, ..., n\}$.

An *action* in reinforcement learning is the behavior of an agent at a specific time at a specific state. In RL-based cognitive radio, an action a allows an agent to either switch from its current spectrum to a new available spectrum in SP, or switch from its current power value to a new available power value in PW. Here we define action a_t at time t as

$$a_t = (\vec{k})_t,$$

where $\vec{k}$ is a vector of actions across all agents. Here k_i is the action of the ith agent and $k_i \in \{jump\_spectrum, jump\_power\}$.

A *reward* in reinforcement learning is a measure of the desirability of an agent's action at a specific state within the environment. In RL-based cognitive radio, the reward r is closely related to the performance of the network. The rewards for the different network conditions are as follows:

- *CR-PU interference:* If primary user (PU) activity occurs in the spectrum shared by the CR user, and in the slot same selected for transmission, then a high penalty of -15 is assigned. The intuitive meaning of this is as follows: We permit collision among the CR users, though that lowers the link throughput. However, the concurrent use of the spectrum with a PU

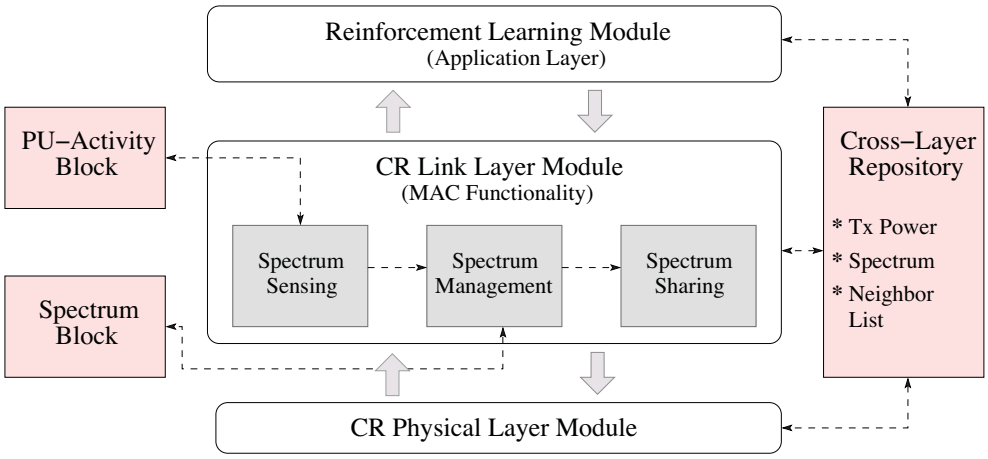

Figure 4: Block diagram of the implemented simulator tool for reinforcement learning based cognitive radio.

goes against the principle of protection of the licensed devices, and hence, must be strictly avoided.

- *Intra-CR network Collision:* If a packet suffers a collision with another concurrent CR user transmission, then a penalty of -5 is imposed. Collisions among the CR users lowers the link throughput, which must be avoided. The comparatively low penalty to the CR users arising from intra-network collisions is aims to force fair sharing of the available spectrum by encouraging the users to choose distinct spectrum bands, if available.

- *Channel Induced Errors:* The inherent unreliability in the wireless channel results in signal variations called as fading caused by multipath reflections (several different copies of the same signal arriving within short periods of each other), relative motion between sender-receiver pair (based on the Doppler effect), among others. This phenomenon results in occasional bit-flips in the received data stream, and in this work, we use standard expressions of bit error rate derived under a Rayleigh fading environment [15]. Moreover, higher received signal strength reduces errors arising out of background noise. Interestingly, certain spectrum bands are more robust to channel errors owing to the lower signal attenuation with distance.

  By preferring the spectrum bands with the lowest packet bit error rate (and hence, packet error rate PER), the CR users reduce re-transmissions and associated network delays. The carrier to noise power ratio at the receiver, say j, can be calculated by $\frac{P^j_{rx}}{N}$, where the noise power N is assumed as 10^{-10} mW. Consequently, the energy per bit E_b/N_0 is given by $\frac{E_b}{N_0} = \frac{P^j_{rx}}{N} \times \frac{Bw}{R}$. From this, the probability of bit error P^j_b at the receiver j can be derived for QPSK modulation as $Q\left(\sqrt{2 \cdot \frac{E_b}{N_0}}\right)$ [13], where the channel bandwidth Bw is 22 MHz, and the bit rate R is 2 Mbps considering a channel structure similar to the one defined in the IEEE 802.11b specification [10]. Q is the Q-function that can be expressed in terms of the error function erf as $Q = \frac{1}{2}\left(1 - erf(\frac{x}{\sqrt{2}})\right)$. Furthermore, assuming each packet transmitted by the sender is ψ bits in length, we can calculate the probability of unsuccessful transmission by the packet error rate or $PER = 1-(1-P^j_b)^{\psi}$.

- *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, and we impose a steep penalty of -20. Thus, the sender should quickly increase its choice of transmit power so that the link can be re-established.

- *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, and a reward of $+5$ is assigned.

In this way, we can apply multi-agent reinforcement learning to solve cognitive radio problem.

4. EXPERIMENTAL SIMULATION

In this section, we describe preliminary results from applying multi-agent reinforcement learning to our cognitive radio model. The overall aim of our proposed learning based approach is to allow the CR users (hence, agents) to decide on an optimal choice of transmission power and spectrum so that (i) PUs are not affected, and (ii) CR users share the spectrum in a fair manner.

4.1 Experimental Design

A novel CR network simulator described in Section 4.1 has been designed to investigate the effect of the proposed reinforcement learning technique on the network operation. As shown in Figure 4, our implemented *ns*-2 model is composed of several modifications to the physical, link and network layers in the form of stand-alone C++ modules. The PU Activity Block describes the activity of PUs based on the on-off model, including their transmission range, location, and spectrum band of use. The Channel Block contains a channel table with the background noise, capacity, and occupancy status. The Spectrum Sensing Block implements the energy-based sensing functionalities, and if a PU is detected, the Spectrum Management Block is notified. This, in turn causes the device to switch to the next available channel, and also alert the upper layers of the change of frequency. The

Spectrum Sharing Block coordinates the distributed channel access, and calculates the interference at any given node due to the ongoing transmissions in the network. The Cross Layer Repository facilitates the information sharing between the different protocol stack layers.

We have conducted a simulation study on two topologies: a 3×3 grid network with a total of 18 CR users (the small topology), and a random deployment of 2000 CR users distributed in a square area of $2000m$ side (the large topology). Half of the total deployed nodes are senders, and the nearest neighbor to each of them becomes their respective receiver. We consider the time to be slotted, and the link layer at each sender node attempts to transmit with a probability $p = 0.2$ in every slot. In the small topology, we assume 4 spectrum bands, given by the set $F = \{50\,\text{MHz}, 500\,\text{MHz}, 2\,\text{GHz}, \text{and } 5\,\text{GHz}\}$, and 4 transmit power values. There are a total of 2 PUs. In the large topology, we assume 400 spectrum bands, chosen as default in the range from 50 MHz to 5 GHz, and 20 transmit power values. There are a total of 100 PUs. In both topologies, the permissible power values are $20m$ uniformly distributed between 0.5 mW to 4 mW.

Each PU is randomly assigned one default channel in which it stays with probability 0.5. It can also switch to three other pre-chosen successively placed channels with the decreasing probability $\{0.4, 0.3, 0.3\}$, respectively. Thus, the PU has an underlying distribution with which it is active on a given channel, but this is unknown to the CR user. The transmission in the CR network occurs on multiple sets of pre-decided node pairs, each such pair forming a link represented as $\{i, j\}$. The terms in the parenthesis denote the directional transmission from the sender i to the receiver j. The choice of spectrum is made by the sender node, and is communicated to the receiver over the common control channel or CCC. This CCC is also used to return feedback to the sender regarding possible collisions that may be experienced by the receiver. However, data transmission occurs exclusively in the spectrum chosen by the node pair forming the link.

We compare the performance of our reinforcement learning based (RL-based) scheme with the other three schemes: (i) random assignment, that selects a random combination of spectrum and power in each round; (ii) greedy assignment with history 1 (G-1), and (iii) greedy assignment with history 30 (G-30). The G-1 algorithm stores for every possible spectrum and power combination the reward received the last time that combination was selected (if any). The algorithm selects the combination with the highest previous reward with probability η and explores a randomly chosen combination with probability $(1 - \eta)$. The G-30 algorithm maintains a repository of the reward obtained in the 30 past slots for every combination of power and spectrum, and selects the best combination in the past 30 slots. Similar to G-1, G-30 selects the best known combination from the history with $\eta = 0.8$, and explores a randomly chosen one with probability $(1 - \eta) = 0.8$. In our RL-based scheme, the exploration rate ϵ is set to 0.2, which we found experimentally to give the best results. Action exploration stops after 25000 epoches. The initial learning rate α is set to 0.8, and it is decreased by a factor of 0.995 after each time slot. Note that G-1 uses the same amount of memory as the RL-based scheme, but the G-30 uses 30 times more memory.

4.2 Results

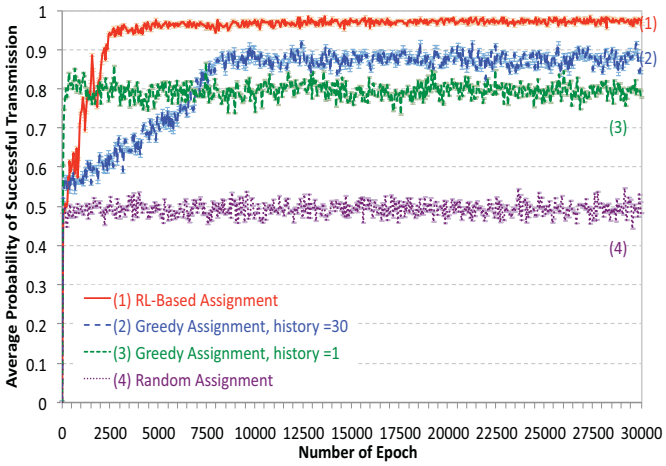

Figure 5: Average probability of successful transmission for the small topology.

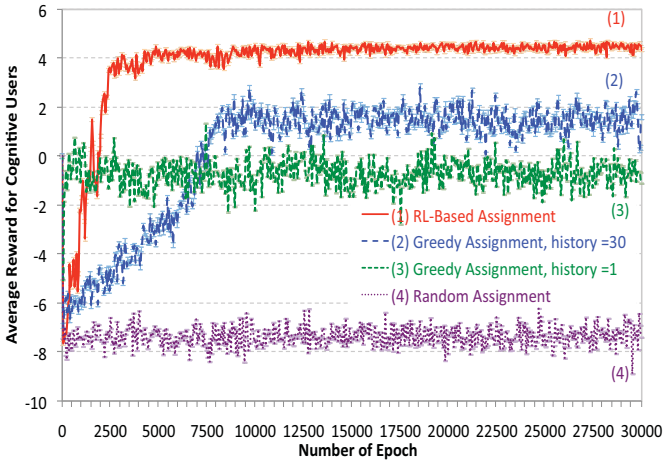

Figure 6: Average reward of CR users for the small topology.

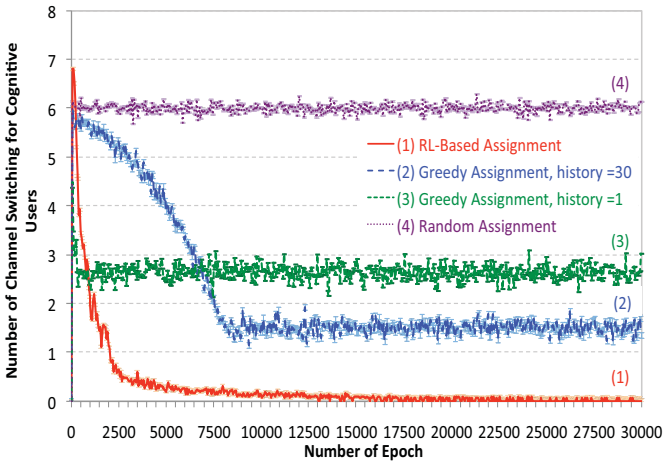

Figure 7: Average number of channel switches for the small topology.

We applied the four schemes, i.e. random, *G-1*, *G-30*, RL-based, to the small and large topologies. We collect the results over 30000 time slots, and record the average probabilities of successful transmission, the average rewards of CR users, and the average number of channel switches by CR users. We then plot these values over time. Each experiment is performed 5 times and we report the means and standard deviations of the recorded values. In our experiments, all runs were found to converge within 30,000 epochs.

Figure 5 shows the average probability of successful transmission when applying the four schemes to the small topology. The results show that the RL-based scheme transmits successful packets with an average probability of approximately 97.5%, while the G-30, G-1 and random schemes transmit successful packets with average probabilities of approximately 88.2%, 79.4%, and 48.7%, respectively. The results indicate that after learning, the RL-based approach can effectively guarantee successful transmissions, and its performance is much better than the others, including the G-30 scheme which uses more than an order of magnitude more memory.

Figure 6 shows the corresponding average rewards received by CR users when applying the four schemes to the small topology. The results show that after learning, the RL-based scheme receives the largest positive reward of approximately $+4.3$, while the G-30 gets a reward of approximately $+1.7$, G-1 gets an negative average reward of approximately -0.8 and the random scheme gets a negative average reward of approximately -7.5. The results indicate that the RL-based approach pushes CR users to gradually achieve higher positive rewards and choose more suitable spectrum and power values for their transmission. The results also indicate that the reward tends to be proportional to the probability of successful transmission.

Figures 7 shows the corresponding average number of channel switches by CR users when applying the four schemes to solve the small topology. The results show that after learning, the RL-based scheme tends to decrease channel switching to 0, while G-30 keeps the channel switches to approximately 1.5, G-1 keeps the channel switches to approximately 2.6, and the random scheme keeps the channel switches to approximately 6.0. The results indicate that our RL-based approach is able to keep the channel switches very low. The results also indicate that our approach can converge to an optimal solution for successful transmission after learning.

We further observe in the graphs of Figures 5, Figures 6 and Figures 7 that the behavior of the RL-based scheme is smoother and more predicatable than the behavior of the other approaches. These results suggest that our approach is more stable than the G-30, G-1, and random approaches.

Figures 8 shows the average probabilities of successful transmissions when applying the RL-based, G-30, and random schemes to the large topology. The results show that the RL-based scheme transmits successful packets with the average probability of approximately 78.2%, while the G-30 and random scheme transmit successful packets with the average probabilities of approximately 49.5% and 43.2%, respectively. Note that the average probabilities of successful transmissions increases sharply after 250000 epochs because action exploration stops. The results indicate that our proposed approach outperforms the G-30 and random approach after learning, even when G-30 uses more memory.

Figures 9 shows the corresponding average rewards of CR

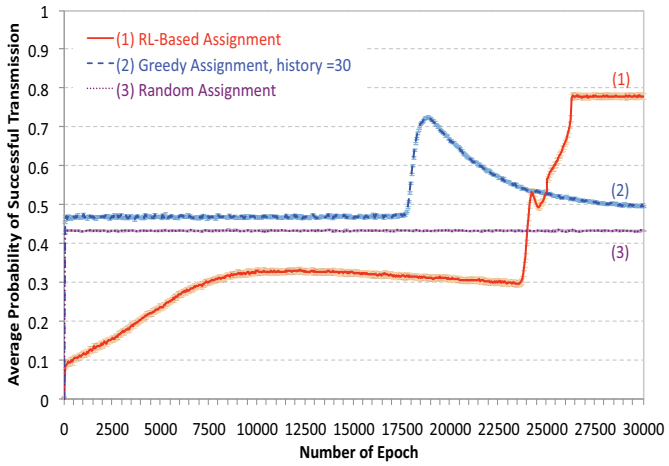

Figure 8: Average probability of successful transmission for large topology.

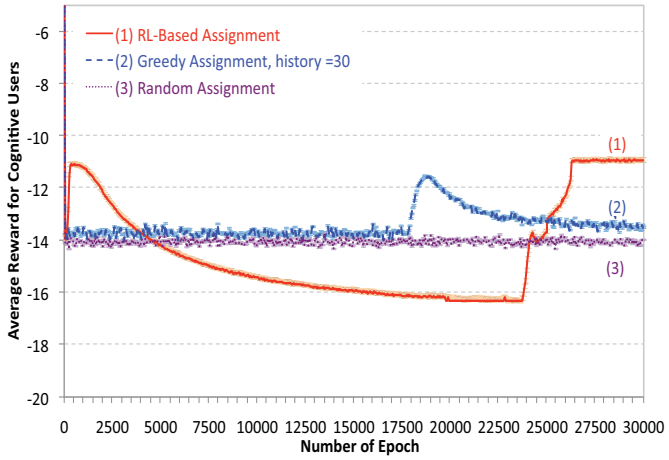

Figure 9: Average reward of CR users for large topology.

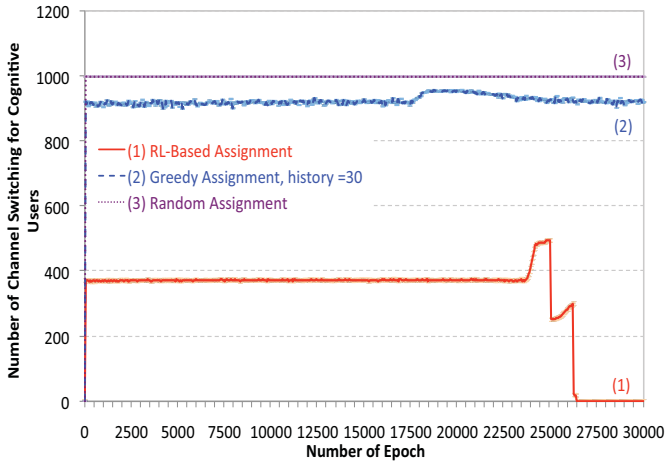

Figure 10: Average number of channel switches for large topology.

users when applying the same three schemes to the large topology. The results show that after learning, the RL-based scheme receives the largest average reward of approximately −10.9, while the G-30 gets an average reward of approximately −13.6 and the random scheme gets a negative average reward of approximately −14.0. The results indicate that our RL-based approach reduces the negative rewards to the largest extent when unsuccessful transmission must occur in a cognitive radio network.

Figures 10 shows the corresponding average number of channel switches of CR users when applying the three schemes to the large topology. The results show that after learning, the RL-based scheme tends to decrease channel switching to 0, while G-30 keeps the channel switches to approximately 906 and the random scheme keeps the channel switches to approximately 991. The results indicate that our proposed approach can keep the channel switches very low and converge to an optimal solution.

In the large topology, our reinforcement learning technique shows two key characteristics: (1) the learning curve initially underperforms in both the probability of successful transmission and the reward earned by the CR user, respectively , and (2) it outperforms the competing schemes towards the later end of the simulation duration, after about 25,000 time slots. In both these cases, the learning technique assumes a flat slope that indicates fewer fluctuations and a convergence-like behavior. Moreover, the number of channel switches is significantly lower than the others, which directly results in higher network throughput. The RL-based scheme, as well as the competing schemes, are subject to the complex network behavior and exhibit different suboptimal peaks before converging on a static value. We note that the authors are not aware of simulation studies of such a large-scale network being previously performed.

5. FUNCTION APPROXIMATION FOR RL-BASED COGNITIVE RADIO

A key limitation on the effectiveness of reinforcement learning is the size of the table needed to store the state-action values. The requirement that an estimated value be stored for every state limits the size and complexity of the learning problems that can be solved. Problems with large state spaces, such as large-scale cognitive radio problems, are typically difficult to solve.

Function approximation [5], which stores an approximation of the entire table, is one way to solve this problem. Many function approximation techniques exist, including coarse coding [9] and tile coding [4] (also known as CMAC [19]), and there are guarantees on their effectiveness in some cases. A limitation of these techniques is that they cannot handle continuous state-action spaces with high dimensionality [17]. Sparse Distributed Memories (SDM) [11] can also be used to reduce the amount of memory needed to store the state-action value table. This approach applied to reinforcement learning, also called Kanerva Coding [17], represents a function approximation technique that is particularly well-suited to problem domains with high dimensionality.

In order to implement Kanerva Coding to our reinforcement learning based approach, a collection of k *prototype state* (prototypes) $\vec{p} = (\vec{sp}, \vec{pw})$ is randomly selected from the state space of every CR user. A state s and a prototype p are said to be *adjacent* if their Euclidean distance is no more than a threshold. Normally, we set the threshold as the ratio of the number of states over the number of prototypes. A state is represented as a collection of binary features, each of which equals 1 if and only if the corresponding prototype is adjacent. A value $\theta(i)$ is maintained for the ith feature, and an approximation of the value of a state is then the sum of the θ values of the adjacent prototypes. In this way, Kanerva Coding can greatly reduce the size of the value table that needs to be stored in the RL-based approach. The computational complexity of the approach depends entirely on the number of prototypes, which is not a function of the number of the dimensions of the state space.

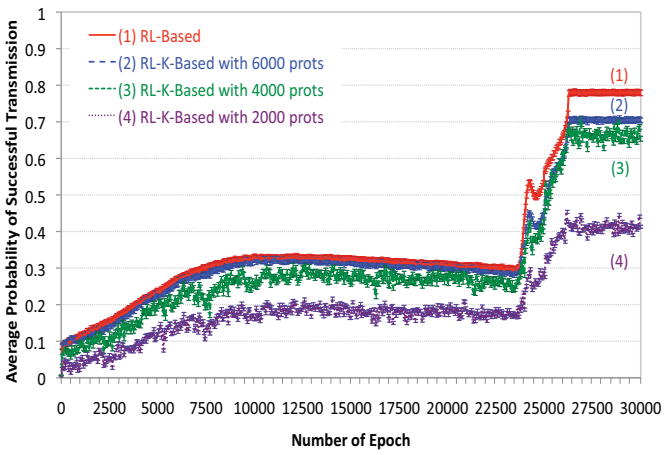

Figure 11: Average probability of successful transmission for large topology.

5.1 Results

We evaluate our reinforcement learning-based approach with Kanerva-based function approximation (RL-K-based) by applying it to the large topology. We compare the performance of the RL-K-based scheme with the RL-based scheme. In the RL-K-based approach, the learning parameters are same as the RL-based approach. The number of prototypes varies over the following values: 2000, 4000, 6000. Note that the number of states in the large topology is 8000, that is, 400 (channels) × 20 (power values).

Figures 11 shows the average probability of successful transmission when applying the RL-K-based scheme with varying numbers of prototypes to the large topology. The results show that after learning, the RL-K-based scheme with 2000, 4000 or 6000 prototypes transmits successful packets with an average probability of approximately 41.5%, 65.2%, and 70.5%, respectively. The results indicate that although the performance of the RL-K-based scheme is worse than the RL-based scheme, the RL-K-based scheme can use less memory. For example, the RL-K-based scheme uses 2/3 of memory of the pure RL scheme with a loss of only 7.9% of successful transmissions with 6000 prototypes, and uses 1/2 size of memory with the loss of only 13.0% of successful transmissions with 4000 prototypes. The results also show that if the number of prototypes is too small, the performance is similar to randomly choosing channels and power values. In our future work, we will focus on using prototype optimization techniques to improve the efficiency of Kanerva-based function approximation for reinforcement

learning based cognitive radio.

6. CONCLUSIONS

Cognitive radio is a new paradigm that attempts to opportunistically transmit in licensed frequencies, without affecting the existing primary users of these bands. To realize this capability, such a radio must predict specific interferer transmission patterns and adaptively change its operational parameters, such as transmit power and spectrum. These tasks, collectively referred to as spectrum management, are difficult to achieve in a dynamic distributed environment, in which CR users may only make local decisions, and react to changes in the environment. In this paper, we described a novel spectrum management approach based on multi-agent reinforcement learning for CR ad hoc networks with decentralized control. Our approach uses value functions to measure the desirability of choosing different transmission parameters, and enables efficient assignment of spectrum and transmit powers by maximizing long-term rewards.

We evaluated our approach by applying it to several real-world scenarios. By comparing the communication performance with random and greedy spectrum assignment, we showed that our reinforcement learning-based approach outperforms the other approaches. We also employed Kanerva-based function approximation to improve our approach's ability to handle large cognitive radio networks. By evaluating its effect on communication performance, we showed that function approximation can effectively reduce the memory used for large networks with little loss of performance. We therefore conclude that our reinforcement learning based spectrum management can significantly reduce interference to licensed users, while maintaining a high probability of successful transmissions in a cognitive radio ad hoc network.

7. REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury. Crahns: Cognitive radio ad hoc networks. *Ad Hoc Networks Journal (Elsevier)*, 7(5):810–836, July 2009.

[2] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury. Spectrum management in cognitive radio ad hoc networks. *IEEE Network*, 23(4):6–12, July 2009.

[3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation / dynamic spectrum access / cognitive radio wireless networks: a survey. *Computer Networks Journal (Elsevier)*, 50:2127–2159, September 2006.

[4] J. Albus. *Brains, Behaviour, and Robotics*. McGraw-Hill, 1981.

[5] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proc. of the 12th Intl. Conf. on Machine Learning*. Morgan Kaufmann, 1995.

[6] FCC. Spectrum policy task force report. *ET Docket No. 02-155*, Nov. 2002.

[7] G. J. Gordon. Stable function approximation in dynamic programming. In *Proc. of Intl. Conf. on Machine Learning*, 1995.

[8] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, February 2005.

[9] G. Hinton. Distributed representations. *Technical Report, Department of Computer Science, Carnegie-Mellon University, Pittsburgh*, 1984.

[10] IEEE Std 802.11b Specification. Ieee std 802.11b-1999/cor 1-2001, 2001.

[11] P. Kanerva. *Sparse Distributed Memory*. MIT Press, 1988.

[12] J. Mitola III. Cognitive radio for flexible mobile multimedia communication. In *Proc. IEEE International Workshop on Mobile Multimedia Communications (MoMuC) 1999*, pages 3–10, November 1999.

[13] J. Proakis. *Digital Communications*. McGraw-Hill Science/Engineering/Math, August 2000.

[14] B. Ratitch and D. Precup. Sparse distributed memories for on-line value-based reinforcement learning. In *Proc. of the European Conf. on Machine Learning*, 2004.

[15] G. L. Stüber. *Principles of mobile communication (2nd ed.)*. Kluwer Academic Publishers, 2001.

[16] R. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Proc. of Conf. on Neural Information Processing Systems*, 1995.

[17] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. Bradford Books, 1998.

[18] M. D. Waltz and K. S. Fu. A heuristic approach to reinforcment learning control systems. In *IEEE Transactions on Automatic Control, 10:390-398.*, 1965.

[19] C. Watkins. Learning from delayed rewards. *Ph.D thesis, Cambridge Univeristy, Cambridge, England*, 1989.

[20] M. Wooldridge. An introduction to multiagent systems. In *John Wiley Sons Ltd, ISBN 0-471-49691-X*, 2002.