Analysis and Implementation of Reinforcement Learning on a GNU Radio Cognitive Radio Platform

Yu Ren, Pawel Dmochowski, Peter Komisarczuk
School of Engineering and Computer Science, Victoria University of Wellington
PO Box 600, Wellington, New Zealand
e-mail: {yu.ren,pawel.dmochowski,peter.komisarczuk}@ecs.vuw.ac.nz

***Abstract*—We present a physical cognitive radio system implementation under the GNU Radio platform with the aim of evaluating a reinforcement learning spectrum management scheme. In our experiments we examine the packet transmission success rate of the cognitive user for a variety of channel utilisation parameters. We derive analytical expressions using Markov chain analysis for the learning convergence time and secondary user packet transmission success rate in the general case of large-scale networks. Our results show that the reinforcement learning scheme significantly improves system performance.**

I. Introduction

Cognitive radios (CRs) [1] are being considered as a way to more effectively utilise the licensed spectrum by exploiting temporarily unused spectral resources. In a multi-user context, this concept is often referred to as Dynamic Spectrum Access Networks (DSAN) [2], where unlicensed devices, or Secondary Users (SUs), are permitted to use the spectrum allocated to a licensed Primary User (PU) when it is not occupied, known as *whitespace*. The opportunistic use of whitespace by the SU is conditional on acceptibly low level of interference to the PU.

CRs are defined by the ability to autonomously adapt transmissions to meet QoS requirements depending on the environment. Many approaches, including game theory, reinforcement learning and channel prediction are being considered to effectively exploit whitespace through channel selection. A survey on the issues can be found in [2]–[5], and the references within. The above papers developed and tested their algorithms by means of simulations. There is, however, a need to evaluate these solutions using a hardware platform [6], as many prevalent network simulation environmental assumptions are not valid. Recently, there has been an increased interest in the implementation of CR functionality using a Software Defined Radio (SDR) platform. SDRs enable sensing and transmissions that are reconfigurable in software.

In this paper we study a novel ad-hoc Q-learning dynamic channel selection scheme (DCS) for cognitive radios implemented on GNU Radio [7]. We evaluate the scheme's performance among channels occupied by primary users of different utilisations. Furthermore, a theoretical analysis is conducted to predict performance in a real large-scale network.

This paper is organised as follows. Section II reviews Q-learning, while the Q-learning DCS scheme, which expands on [5], is presented in Section III. The scheme addresses the scalability issues which Reinforcement Learning (RL) algorithms suffer from. In Section IV we discuss CR implementation under the GNU Radio platform. Section V derives the system's analytical performance. Section VI discusses the experimental results. Section VII concludes the paper with suggestions for future work.

II. Q-Learning Overview

Reinforcement learning is an online learning process applicable to the problem of channel selection. In RL scalar rewards are assigned to the outcomes of actions performable by the agent. Through repeated interactions with its environment updating a function based on the reward outcome of its taking an action a at state x, a learner develops an optimum action policy π^* in real-time maximising the value function

$$V^{\pi}(x) = R(x, \pi(x)) + \gamma \sum_{y \in X} P_{xy}(\pi(x)) V^{\pi}(y) \qquad (1)$$

where $R(x, \pi(x))$ is the expected reward for taking the action specified by policy π in state x and $P_{xy}(\pi(x))$ the state transition probability for doing so. The parameter γ, $0 \leq \gamma \leq 1$, is known as the discount factor, so the value function represents the immediate reward R plus the discounted reward for all future actions stemming from this decision.

Q-learning is a model free version of RL. A Q-value is maintained for every pair (x, a). Taking action a in state x results in the Q-value being updated according to

$$Q_{\text{upd}}(x, a) = (1 - \alpha) Q_{\text{curr}}(x, a) + \alpha [r + \gamma \max_{b \in A} Q_{\text{curr}}(y, b)] \qquad (2)$$

where α , $0 \leq \alpha \leq 1$, is the learning rate, A is the available action set and r is the reward.

It is known [8] that as long as all states and actions are repeatedly sampled the Q-values will converge to

$$Q^*(x, a) = Q^{\pi^*}(x, a) = R(x, a) + \gamma \sum_{y \in X} P_{xy}(a) V^{\pi^*}(y) \qquad (3)$$

where π^* is the optimal action policy described earlier. The advantage of Q-learning is that it is guaranteed to converge to a policy within its action set optimal for its general goal, specified by its reward scheme, without requiring any knowledge of

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9170
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9170

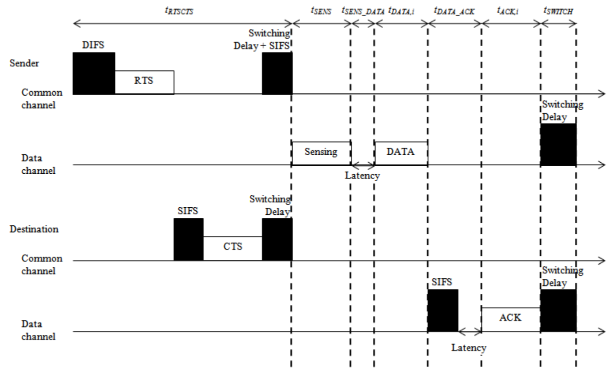

Fig. 1. Simplified ad-hoc cognitive MAC protocol transmission timeline

system mechanics. We consider investigation of Q-learning of significant value. Using Q-learning, the complexity involved in modelling the wireless operating environment can be reduced, such as computing fading and path loss.

III. Q-Learning Spectrum Management Scheme

The goal of the channel Q-learning DCS scheme is to maximise the probability of successful packet transmission of the secondary user. The scheme is presented for an ad hoc network single transceiver radio more suited for the small scale wireless test. Fig. 1 describes the packet transmission between sender and receiver in our *Listen-before-Talking* CSMA/CA cognitive MAC protocol.

The MAC protocol uses a temporary common control channel (CCC) to arrange channel rendezvous. Between transmissions all SUs tune to the CCC, selected for having a long history without a PU being active, and if a SU has a packet to transfer exchanges RTS and CTS with its destination. The temporary CCC approach is used in [9]. The RTS is a modified IEEE 802.11 frame containing the waveform specification the sender has selected. Other secondary users respond by avoiding the channel, as per IEEE 802.11. The receiver echoes the waveform specification in its CTS. The facility exists, if the receiver has earlier sensed a primary user using the data channel to not reply. Together these eliminate the *hidden multi-channel user problem* and *hidden primary user problem*.

Following CTS receipt, the two users switch to the data channel. The sender performs an in-band *Listen-before-Talking* sensing operation. If the channel is occupied by a PU, the sender does not send the DATA packet and the receiver returns to the CCC after timing out with no impact to the licensed user. If the channel is free, the DATA packet is sent, and a successful receipt is replied with an ACK.

The Q-learning DCS is shown in Table I. Depending on the packet transmission success, the channel Q-value is assigned a constant reward RW or a cost CT in (2). The sender selects the available channel not currently in use by another secondary user with the greatest Q-value for transmission. To ensure all channels are sampled, the ϵ-greedy approach to exploration is used, where the sender chooses a random channel with probability ϵ.

TABLE I
Q-Learning Dynamic Channel Selection Scheme

Q-Learning Element	Description	Representation
State-Action Pair	Action: data channel i for packet transmission	Q_i
Reward	Constant positive reward value awarded to channel on successful packet transmission. Constant negative cost incurred on failure	$R = r(s, a) = +RW$ *(successful packet transmission),* $-CT$ *(failed packet transmission)*

The MAC protocol sensing is relied upon to limit the interference caused by the radio. The scheme maximises the packet transmission success within the protocol. Interference to primary users was considered in the experiment, but is not included in this discussion due to space constraints.

IV. Implementation and Evaluation

A. GNU Radio Cognitive Radio Implementation

As discussed in Section I, the CR system was implemented under GNU Radio [7] platform. GNU Radio has seen extensive use by CR and wireless network researchers [10], [11]. In our work we used the Universal Software Radio Peripheral (USRP) as the frontend.

Each radio is equipped with a USRP mounting a single 2.4-2.5GHz ISM band-capable XCVR2450 RF transceiver daughterboard. Frames are modulated in software on a host computer and transmitted to the USRP via a USB2.0 interface. The daughterboard which is not full-duplex uses *auto transmit-receive switching* to transmit and receive the RF signal. Sensing is implemented in the form of a constant false alarm rate (CFAR) power detector [12]. On startup the radio reserves the CCC for sensing and estimates the variance and mean of the noise power. The channel power spectral density (PSD) is calculated using Welch's method [13] and the channel declared occupied if it exceeds 99 standard deviations of the noise level, which is assumed to be stationary.

The cognitive MAC protocol and Q-learning spectrum management scheme are completely implemented on the host computer. The implementation sets up the physical layer signal processing flowgraphs and can change the transmission parameters. Mutator functions can set the frequency, signal bandwidth and waveform amplitude to switch between channels. To change the signal modulation a different physical layer flowgraph is executed and the receiver must be disabled to use *Listen-before-Talking* sensing, which shares the same daughterboard receiver. The ongoing MAC layer transmission spawns a handler thread that is responsible for frames sent by the radio. Received frames trigger a callback function. The handler tunes the receiver and transmitter to the channel, transmits its frame and waits on a condition, to be woken if the callback has received the next frame, whereupon it repeats the process for the next protocol step. A Q-table structure stores and updates the channel Q-values that are referred to in channel selection.

The userspace architecture implementation is limited by the USRP USB interface and jitter caused by interprocess schedul-

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9170
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9170

ing. USB latency, studied in [14], introduces a delay when a block of signal data is transferred across the interface. This makes it impossible to attain the strict timing requirements of common wireless protocols, such as IEEE 802.11. The USB bandwidth limits the combined signal Nyquist bandwidth transferred across to 4MHz. We plan in future work to investigate implementing parts of the CR on the USRP FPGA, which as shown by [15] can dramatically improve timing performance when MAC protocol features are coded directly onto the FPGA. The timing issue is avoided by using exaggerated packet transfer times (see Table II) without loss of generality in the performance trend of the Q-learning scheme.

B. *Experimental Setup*

The wireless test scenario examines the performance of the Q-learning scheme among PU channels with varying utilisations. The test is carried out with a single pair of SUs. One SU is constantly backlogged with identical DATA packet payloads to send. The cognitive radios can select among three 337.5kHz channels to transmit frames which are modulated using Differential Binary Phase Shift Keying (DBPSK). The channels are occupied by PUs transmitting fixed size packets in a stochastic manner, also implemented on USRP devices. PU packet arrivals follow a Poisson traffic model, where the arrival rate λ is varied to give the desired utilisation, $\rho = t_{\mathrm{PU}}\lambda$, where t_{PU} is the time to transmit a packet.

The experimental parameters are described in Table II. The large ACK packet duration is due to the USRP USB interface requiring 512 bytes before a data block is transferred across. CR literature emphasises SUs keep activity in licensed bands short in order to quickly respond once a PU moves into the band. The parameter $t_{\mathrm{PU},i}$ is set significantly above $t_{\mathrm{DATA},i}$. Q-learning parameters are chosen to match those used in [5]. The experiment was set up in a busy laboratory, resulting in rich scatterring environment. The average SINR of PU and SU signals received by other PU and SU stations is 30dB. Despite the interference due to other devices, the channel packet error rate for both SUs and PUs was found to be negligible, and the sensor could be set with negligible false alarm probability and primary user signal detection rate approaching unity.

The experimental setup restricts the number of channels to 3, limited by the number of devices to run as SUs and PUs. Despite this, generality to the multiagent scenario is not lost. Q-learning is known to converge unreliably when there are multiple agents due to a co-adaptation effect where agents are continuously adjusting their strategies due to the influence of other agents, never achieving a stable solution [16]. In our scheme CSMA/CA avoids collisions between secondary users in the data channel, hence the Q-learning judgment of the data channel packet transmission success rate is made with other agents *invisible*. The Q-learning convergence time roughly multiplies with the number of SUs, assuming fair channel contention.

TABLE II
Experiment Notations, Parameters and Timing Values

Category	**Symbol**	**Details**	**Values**
Scenario		Number of SUs	2
	n	Number of available data channels	3
		Common control channel	2.400 GHz
		Available data channels	[2.425GHz, 2.450GHz, 2.475GHz]
		Runtime	350s
Secondary Users			
		Secondary user traffic	Always backlogged
	t_{SENS}	Sensing duration	23ms
	t_{SENS_DATA}	Latency	16ms
	t_{DATA_ACK}	Latency	26ms
Data Channel Usage	$t_{DATA,i}$	Data packet duration	33ms
	$t_{ACK,i}$	ACK packet duration	16ms
	$t_{A,i}$	Successful transmission cycle duration	110ms
	$t_{B_1,i}$	Failed (data transfer) transmission cycle duration	191ms
	$t_{B_2,i}$	Failed (aborted sensing) transmission cycle duration	191ms
		Bitrate	250kb/s
		Size of SU packet	1024 bytes
Data Channels			
Primary Users		Primary user traffic model	Stochastic channels with exponentially distributed ON and OFF times
	$t_{PU,i}$	Packet duration	328ms
	ρ_i	Utilisation of each PU traffic	[0.1, 0.9]
Channel Quality	P_i^E	Packet Error Rate	Negligible
Q-learning			
	α	Learning rate of Q-learning	0.2
	ε	Trade-off between exploration and exploitation	0.1
	γ	Discount factor	0
	RW	Reward	15
	CT	Cost	5

V. Performance Analysis

The expected packet transmission success rate achieved by the Q-learning scheme in the scenario is derived using Markov chain analysis for a general number of channels of different PU utilisations. We assume perfect sensing and an error free CCC as supported by the experiment.

A. *Preliminary Results*

There are three possible outcomes of every transmission by the SU:
A_i: Transmission successful in channel i, lasting t_{A_i}
$B_{1,i}$: Transmission fails - DATA or ACK packet not received correctly, lasting $t_{B_{1,i}}$
$B_{2,i}$: Transmission fails - aborts in sensing because a primary user is detected, lasting $t_{B_{2,i}}$

$P(s_i)$, the probability the channel is sensed clear, requires no ongoing PU packet transmission prior to sensing. The PU is an M/D/1 system and the requirement is equivalent to the probability there are zero packets in the system, $P_{0,i} = 1 - \rho_i$ This must be followed by no new packet arrivals in the sensing period. The probability of no arrivals in time t is easily found for a Poisson process, $F_i(0,t) = e^{-\lambda_i t}$, thus $P(s_i) = P_{0,i} F_i(0, t_{SENS})$

The DATA and ACK packets are sent on the condition that sensing passes. We assume $t_{\text{PU},i} > t_{\text{SENS\_DATA}} + t_{\text{DATA,i}}, t_{\text{DATA\_ACK}} + t_{\text{ACK,i}}$ The PU is not initially transmitting, thus the probability DATA and ACK are correctly received is the probability no new PU packets arrive and no independent packet errors occur,

$$P(p_i|s_i) = F_i(0, t_{\text{SENS\_DATA}} + t_{\text{DATA,i}} + t_{\text{DATA\_ACK}} + t_{\text{ACK,i}})$$
$$(1 - P_i^E(\text{DATA}))(1 - P_i^E(\text{ACK})).$$

The probability of each outcome is thus

$$P(A_i) = P(p_i|s_i)P(s_i)$$

$$P(B_{1,i}) = (1 - P(p_i|s_i))P(s_i)$$

$$P(B_{2,i}) = P(p_i|s_i)(1 - P(s_i)) \qquad (4)$$

B. Packet Transmission Success Rate

The selection of channels can be described by a Markov chain with $n \times n$ transition matrix M where entry m_{ij} is the probability channel j is selected for the current transmission after an attempt in channel i. In the steady-state, the Q-value for channel i converges in this scenario to

$$Q_i{}^* = R(x,a) = (1 - P(A_i))CT + P(A_i)RW. \qquad (5)$$

Denote the set of l channels with the greatest steady-state Q-value as K. One can show that following the convergence of the Q-learning algorithm, the entries of $\boldsymbol{M}$ are given by

$$m_{kk} = m_{ik} = \frac{1-\epsilon}{l} + \frac{\epsilon}{n}, \quad m_{ij} = \frac{\epsilon}{n} \quad (k \in K,\, i,j \notin K)$$

where the other channels are selected using ϵ-greedy exploration.

We note that the Markov chain is regular, and thus the probability channel j is selected for transmission in the long-run approaches w_j that is independent of the starting state. The unique $1 \times n$ vector $\boldsymbol{w} = (w_1 w_2 ... w_n)$ is known as the common row of the limiting matrix of $\boldsymbol{M}$ and can be found from

$$w_1 + w_2 + ... + w_n = 1, \quad \boldsymbol{wM} = \boldsymbol{w}.$$

It is easily shown that for the Q-learning scheme

$$w_j = \frac{\epsilon}{n}, \quad w_k = 1 + \epsilon\left(\frac{1}{n} - \frac{1}{l}\right).$$

As expected, for random channel selection the probability for all channels is $w_j = \frac{1}{n}$.

In the steady-state, the general probability a transmission attempt is successful under the DCS is given by,

$$P(A) = \sum_{i=1}^{i=n} P(A_i) w_i. \qquad (6)$$

A similar approach can be used for $P(B_1)$ and $P(B_2)$.

C. Convergence of Q-Learning Channel Selection Scheme

The convergence speed is a prime concern when considering RL algorithms. The expected discrete-time behaviour of Q-values is described in [16] by the difference equation

$$Q_i(k+1) - Q_i(k) = x_i(k)\alpha(R(i) - Q_i(k)) \qquad (7)$$

where $Q_i(k)$ is the Q-value of channel i after time k, measured in *action epochs* or SU transmission attempts, and $x_i(k)$ is the rate the Q-value is updated. This is the probability the channel is selected, which in ϵ-greedy exploration is

$$x_i(k) = \begin{cases} (1-\epsilon) + \frac{\epsilon}{n} & \text{if } Q_i(k) = \max_i Q_i(k) \\ \frac{\epsilon}{n} & \text{otherwise.} \end{cases}$$

When $x_i = x_i(k)$ is constant, (7) is a first-order linear difference equation with well known explicit form,

$$Q_i(m) = V^m Q_i(0) + R(i)(1 - V^m) \qquad (8)$$

where $V = 1 - x_i\alpha$. It is found on inspecting (8) that Q-values converge in a piecewise exponential fashion. The rate of exponential convergence is set by V and changes from exploitation to exploration at points of intersection where the channel with the greatest Q-value is displaced. In the worst-case scenario, channel i is never the highest and the channel is selected only rarely in exploration. The expected number of action epochs for the contribution of $R(i)$, the steady-state Q-value term, to rise to proportion p is

$$R(i)(1 - V^m) = pR(i)$$
$$t_{conv\_upper}(p) = \frac{\ln(1-p)}{\ln(1 - \frac{\alpha\epsilon}{n})} \qquad (9)$$

which defines an upper bound to the convergence time. In the best-case scenario the channel Q-value is always exploited in which case the lower bound to the convergence time is

$$t_{conv\_upper}(p) = \frac{\ln(1-p)}{\ln(1 - \alpha(1 - \frac{(n-1)\epsilon}{n}))}. \qquad (10)$$

The worst-case convergence time increases linearly with the number of channels. Using the Q-learning parameters in Table II, 95% convergence is guaranteed within the time taken for the CR to transmit 447 packets with 3 channels. The time required is 3144 packets with 21 channels, which is at the upper end the CCC approach can manage [17]. Substituting in a typical network packet transfer time of 500μs, the examples would converge within 0.22s and 1.6s respectively, multiplied again by the number of SUs in contention.

The analytical convergence results were verified in the wireless setup, using initial Q-values of [0,10,5] with the PU

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9170
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9170

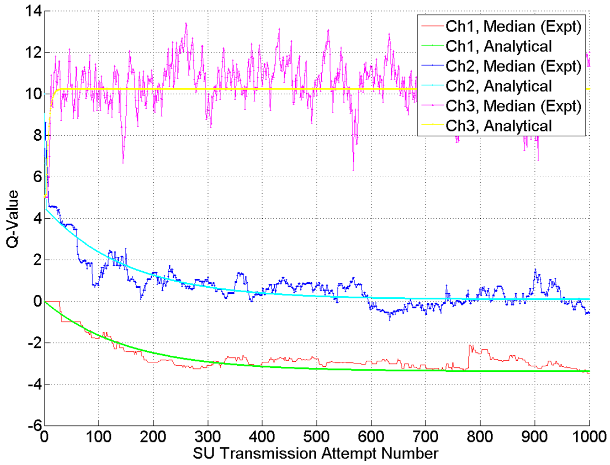

Fig. 2. Channel Q-Values against the number of SU transmission attempts, for channel utilisations: [0.9,0.7,0.2], Q_0=[0,10,5]

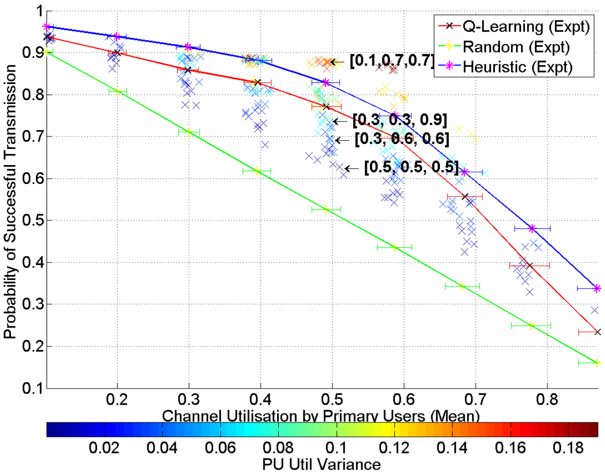

Fig. 3. SU successful transmission probability against mean of the PU channel utilisations

channel utilisations [0.9,0.7,0.2]. Fig. 2 shows the development of the median Q-values over 50 repetitions. The exponential trend is accurately modelled by our system of equations. The local variability is generated by different choices between trials. The Q-values have largely settled to their final values within 500 transmission epochs. The channel ranking has been decided within the first 10 attempts showing the Q-learning scheme can find the best channel much faster than the bounds suggest.

VI. Experimental Results and Discussions

The experiment described in Table II was conducted for all combinations of channel PU utilisations averaging to $[0.1, 0.2, \ldots, 0.9]$, where each PU selects from an utilisation in $[0.1, 0.2, \ldots, 0.9]$. This was repeated with the SU randomly selecting the data channels and using a heuristic DCS scheme. The heuristic scheme is regarded to give good packet transmission success for the scenario [5] and selects the same data channel if the previous packet transmission was successful, otherwise randomly choosing from the other available channels.

Fig. 3 plots the average SU successful transmission probability, $P(A)$, against the average nominal PU channel utilisation. A point on the graph represents the mean taken of all runs where the PU channel combinations average to the utilisation. As an example, the point corresponding to utilisation 0.5 encompasses runs with PU channel utilisations [0.5,0.5,0.5] and [0.1,0.7,0.7]. Each individual Q-learning run lasting $t_{\text{duration}} = 350$s is plotted separately to show the real $P(A)$ value under different utilisations, with points coloured by the variance of the combination utilisations. It was found the PU could not keep up with the packets to be generated at higher utilisations, causing the observed downward drift and increased spread in the actual utilisation. The uncertainties used throughout are one standard deviation in repeated measurements.

The Q-learning scheme consistently outperforms random channel selection. The observed transmission success rate is 1.60 times greater than the random scheme at a nominal average utilisation of 0.6 and 1.58 times at 0.8, but the improvement falls at lower utilisations to 1.04 at 0.1. Performance is slightly worse but comparable to the heuristic scheme which closely follow one another, with $P(A)$ on average across all utilisations smaller by a factor of 0.91. The Q-learning scheme is not expected to be better than the heuristic-based scheme. Q-learning is a model-free RL algorithm offering the ability to autonomously develop a channel selection policy maximising a general goal without any underlying knowledge of the system. It is successful if it achieves comparable results to the heuristic scheme designed to empirically give near-optimal performance in this scenario, with the advantage it will adapt should the scenario change. The Q-learning scheme autonomously develops the heuristic scheme in the limit of a high learning rate when the reinforcement of a single transmission attempt result dominates. This causes the channel to be used again if the attempt was successful or rejected for another.

The analysis in (5) shows the Q-learning scheme will exploit the channel with the least utilisation, which is confirmed by the experimental results. Utilisations [0.1,0.7,0.7] and [0.3,0.3,0.9] exhibit similar performance to when the channel utilisation is 0.1 and 0.3. The proportionate improvement over the random DCS at higher utilisations is explained by the greater difference in whitespace between PUs at these utilisation combinations. Channel 0.6 in [0.6,0.9,0.9]=0.8 (mean) is free four times as often as the other channels, which are of greater utilisation, compared to 1.5 times for [0.1, 0.4, 0.4]=0.3.

The analytical transmission success rate is plotted alongside the experimental results in Fig. 4, with the analysis derived from measured individual transmission outcome probabilities. The predicted performance of the random DCS is identical to that observed, to within uncertainties. While they follow a similar trend the experimental Q-learning success probability is higher. This indicates the scheme successfully exploits local variations in primary user traffic (noise), whereas the Markov chain analysis is done with steady state Q-values. This is seen

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9170
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9170

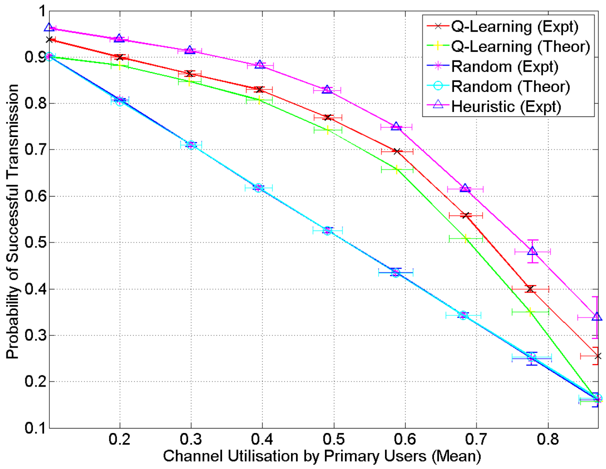

Fig. 4. Mean SU successful transmission probability against mean of the PU channel utilisations

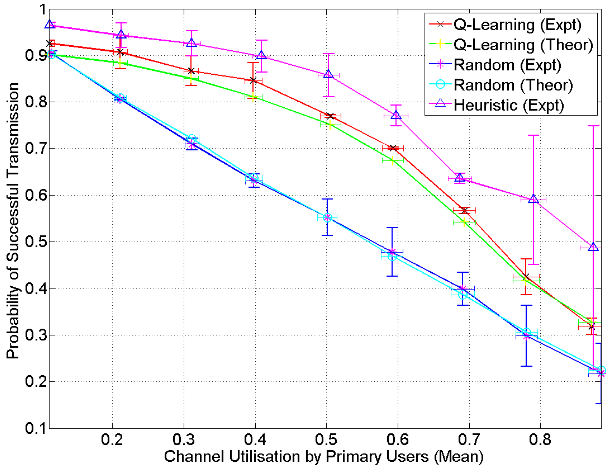

Fig. 5. Mean SU successful transmission probability against mean of the PU channel utilisations, within the first 225 transmission attempts

at utilisations [0.1,0.1,0.1]=0.1 and [0.9,0.9,0.9]=0.9 where $P(A)$ is found to be greater than random channel selection despite the fact the Q-value convergence rule should lead to equal sampling of each channel. The results show the analytical model is indicative of the performance of the Q-learning scheme. The ability of the DCS to learn, for instance a brief quiescent period in a channel, and add to the real success probability, is influenced by the actual traffic and learning rate α tunable in simulation.

Similar results are shown in Fig. 5 where the experimental $P(A)$ is plotted for the first 225 transmission attempts. Recalling the Q-learning convergence time bound to be 447 attempts (Section V-C), we conclude the scheme adapts rapidly. The heuristic scheme is highly dependent on the traffic during this short period at the highest utilisations, which may lead to it oscillating between occupied channels with low $P(A)$ or happening on long unoccupied sections. This is seen in the high uncertainty at these values.

VII. Conclusions and Future Work

In this paper we evaluated a RL spectrum management scheme using a GNU Radio SDR implementation, which overcomes RL scalability and multiagent issues. The scheme was found to converge to a good channel selection policy with comparable rates of packet transmission success to a well-designed heuristic scheme. The performance was accurately modelled by a Markov chain analysis derived for general large-scale multiagent networks. The scheme needs to be reformulated to reinforce metrics more suitable for heterogeneous channels such as goodput and to reduce convergence time by adopting a continuous exploration technique.

References

[1] J. Mitola, “Cognitive radio: Model-based competence for software radios,” Ph.D. dissertation, Dept. of Teleinformatics, KTH, 1999.

[2] 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: The Int. J. of Comp. and Telecom. Networking*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[3] T. Jiang, D. Grace, and Y. Liu, “Performance of cognitive radio reinforcement spectrum sharing using different weighting factors,” in *Proc. ChinaCom*, Aug. 2008, pp. 1195–1199.

[4] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, “A context-aware and intelligence dynamic channel selection scheme for cognitive radio networks,” in *Proc. CrownCom*, June 2009.

[5] K.-L. Yau, P. Komisarczuk, and P. Teal, “Performance analysis of reinforcement learning for achieving context-awareness and intelligence in cognitive radio networks,” in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, oct. 2009, pp. 1046 –1053.

[6] D. Kotz, C. Newport, R. S. Gray, Y. Y. J. Liu, and C. Elliott, “Experimental evaluation of wireless simulation assumptions,” Dartmouth Computer Science, Tech. Rep. TR2004-507, June 2004.

[7] (12/10/2009) GNU Radio. [Online]. Available: http://gnuradio.org/trac

[8] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, pp. 279–292, 1992.

[9] C. Cordeiro and K. Challapali, “C-mac: A cognitive mac protocol for multi-channel wireless networks,” in *New Frontiers in Dynamic Spectrum Access Networks,2007. DySPAN 2007. 2nd IEEE International Symposium on*, April 2007, pp. 147–157.

[10] T. W. Rondeau, “Application of artificial intelligence to wireless communication,” Ph.D. dissertation, Virginia Polytechnic Institute and State University, Sept. 2007.

[11] Z. Yan, Z. Ma, H. Cao, G. Li, and W. Wang, “Spectrum sensing, access and coexistence testbed for cognitive radio using USRP,” in *Proc. ICSCC*, May 2008, pp. 270–274.

[12] W. Lin and Q. Zhang, “A design of energy detector in cognitive radio under noise uncertainty,” in *Proc. ICCS*, Nov. 2008, pp. 213–217.

[13] P. D. Welch, “The use of fast fourier transforms for the estimation of power spectra: A method based on time averaging over short modified periodograms,” in *IEEE Trans. on Audio and Electroacoustics*, vol. 15, 1967, pp. 70–73.

[14] T. Schmid, O. Sekkat, and M. B. Srivastava, “An experimental study of network performance impact of increased latency in software defined radios,” in *WinTECH ’07: Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, 2007, pp. 59–66.

[15] G. Nychis, T. Hottelier, Z.Yang, S.Seshan, and P. Steenkiste, “Enabling MAC protocol implementations on software-defined radios,” in *Proc. 6h USENIX symphosium on Networked systems design and implementation*, April 2009, pp. 91–105.

[16] E. R. Gomes and R. Kowalczyk, “Dynamic analysis of multiagent q-learning with ϵ-greedy exploration,” in *Proc. 26th Annual Int. Conf. on Machine Learning*, 2009, pp. 369–376.

[17] T. Luo, M. Motani, and V. Srinivasan, “Cam-mac: A cooperative asynchronous multi-channel mac protocol for ad hoc networks,” in *BROADNETS*, 2006.

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9170
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9170