Intelligent Spectrum Management based on Transfer Actor-Critic Learning for Rateless Transmissions in Cognitive Radio Networks

Koushik A M*, Fei Hu*†, and Sunil Kumar‡
*Electrical and Computer Engineering, The University of Alabama, USA († Corresponding author)
‡Electrical and Computer Engineering, San Diego State University, San Diego, CA, USA

***Abstract*—This paper presents an intelligent spectrum mobility management scheme for cognitive radio networks. The spectrum mobility could involve spectrum handoff (i.e., the user switches to a new channel) or stay-and-wait (i.e., the user pauses the transmission for a while until the channel quality improves again). An optimal spectrum mobility management scheme needs to consider its long-term impact on the network performance, such as throughput and delay, instead of optimizing only the short-term performance. We use a machine learning scheme, called the Transfer Actor-Critic Learning (TACT), for the spectrum mobility management. The proposed scheme uses a comprehensive reward function that considers the channel utilization factor (CUF), packet error rate (PER), packet dropping rate (PDR), and flow throughput. Here, the CUF is determined by the spectrum sensing accuracy and channel holding time. The PDR is calculated from the non-preemptive M/G/1 queueing model, and the flow throughput is estimated from a link-adaptive transmission scheme, which utilizes the rateless (Raptor) codes. The proposed scheme achieves a higher reward, in terms of the mean opinion score, compared to the myopic and Q-learning based spectrum management schemes.**

***Index Terms*—Cognitive Radio Networks, Spectrum Management, Spectrum Mobility, Spectrum Handoff, Rateless Codes, Transfer Actor-Critic Learning (TACT).**

I. Introduction

The spectrum mobility management is very important in cognitive radio networks (CRNs) [1]. Although a secondary user (SU) does not know exactly when the primary user (PU) will take the channel back, it wants to achieve a reliable spectrum usage to support its quality of service (QoS) requirements. If the quality of the current channel degrades, the SU can take one of the following three decisions: (i) Stay in the same channel waiting for it to become idle again (called stay-and-wait); (ii) Stay in the same channel and adjust to the varying channel conditions (called stay-and-adjust); (iii) Switch to another channel that meets its QoS requirement (called spectrum handoff). Generally, if the waiting time is longer than the channel switching delay plus traffic queueing delay, the SU should switch to another channel [2].

In this paper, we design an intelligent spectrum mobility management (iSM) scheme. To accurately measure the channel quality for spectrum mobility management, we define a channel selection metric (CSM) based on the following three important factors: (i) *Channel Utilization Factor (CUF)* determined based on the spectrum sensing accuracy, false alarm rate, and channel holding time (CHT) [3]; (ii) *Packet Dropping Rate (PDR)* determined by evaluating the expected waiting delay for a SU in the queue associated with the channel; (iii) *Flow throughput* which uses the decoding-CDF [4], along with the prioritized Raptor codes (PRC) [5].

The spectrum management should maximize the performance for the entire session instead of maximizing only the short-term performance. Motivated by this, we design an iSM scheme by integrating the CSM with machine learning algorithms. The spectrum handoff scheme based on the long-term optimization model, such as Q-learning used in our previous work [2], can determine the proper spectrum decision actions based on the SU state estimation (including PER, queueing delay, etc.). However, the SU does not have any prior knowledge of the CRN environment in the beginning. It starts with a trial-and-error process by exploring each action in every state. Therefore, the Q-learning could take considerable time to converge to an optimal, stable solution. To enhance the spectrum decision learning process, we use the transfer learning schemes in which a newly joined SU learns from existing SUs which have similar QoS requirements [6]. Unlike the Q-learning model that asks a SU to recognize and adapt to its own radio environment, the transfer learning models pass over the initial phase of building all the handoff control policies [6], [7].

The transfer actor-critic learning (TACT) method used in this paper is a combination of actor-only and critic-only models [8]. While the actor performs the actions without the need of optimized value function, the critic criticizes the actions taken by the actor and keeps updating the value function. By using TACT, a new SU need not perform iterative optimization algorithms from scratch. To form a complete TACT-based transfer learning framework, we solve the following two important issues: Selection of an expert SU and transfer of policy from the expert to the learner node. We enhance the original TACT algorithm by exploiting the temporal and spatial correlations in the SU's traffic profile, and update the value and policy functions separately for easy knowledge transfer. A SU learns from an expert SU in the beginning; Thereafter, it gradually updates its model on its own. The preliminary results of this scheme appeared in [9].

The CSM concept as well as the big picture of our iSM model is shown in Fig. 1. After the CSM is determined, the TACT model will generate CRN states and actions, which consist of three iSM options (spectrum handoff, stay-and-wait,

or stay-and-adjust).

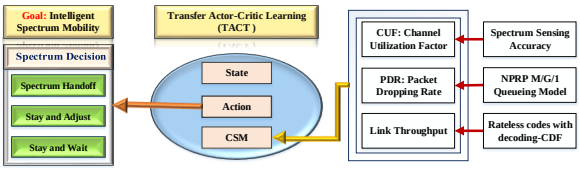


Fig. 1: The big picture of iSM concept.

The main contributions of this paper are:
1) Teaching based spectrum management is proposed to enhance the spectrum decision process. Previously, we proposed an apprenticeship learning based transfer learning scheme for CRN [6], which can be further improved in some areas. For example, the exact imitation of the expert node's policy should be avoided since each node in the network may experience different channel conditions. Therefore, it is helpful to consider a TACT-based transfer learning algorithm which uses the learned policy from the expert SU to build its own optimized learning model by fine tuning the expert policy according to the channel conditions it experiences. More importantly, we connect the Q-learning with TACT to receive the learned policy from the expert node, which greatly enhances the teaching process without introducing much overhead to the expert node.
2) Decoding-CDF with prioritized Raptor codes (PRC) are used to perform the high-throughput spectrum adaptation. Due to mobility, the SU may experience fading and poor channel conditions. In order to improve the QoS performance, we introduce spectrum adaptation by using the decoding-CDF along with machine learning. Initially, the decoding-CDF was proposed for use with the Spinal codes [10], whereas we use the decoding-CDF along with our prioritized Raptor codes (PRC) [5]. Our PRC model considers the prioritized packets and allocates better channels to high-priority traffic.

The rest of this paper is organized as follows. The related work is discussed in Section II. The channel selection metric is described in Section III, followed by an overview of the Q-learning based iSM scheme in Section IV. Our TACT-based iSM scheme is described in Section V. The performance evaluation and simulation results are provided in Section VI, followed by a discussion in Section VII. Finally, the conclusions are given in Section VIII.

II. Related Work

In this section, we review the literature related to our work, which includes the three aspects:

a. Learning-based Wireless Adaptation: The strategy of learning from expert SUs was proposed in our previous work, called the apprenticeship learning based spectrum handoff [6], which was further extended in [11] as the multi teacher apprenticeship learning where the node learns spectrum handoff strategy from multiple nodes in the network. Other related work in this direction includes the concept of docitive learning (DL) [7], [12], reinforced learning (RL) used in CRNs [13], RL-based cooperative spectrum sensing [14], and Q-learning based channel allocation [6], [15], [16]. DL was successfully used for interference management in femtocell [7]. However, it did not consider the concrete channel selection parameters. Also, it does not have clear definitions of expert selection process and node-to-node similarity calculation functions. A channel selection scheme was implemented on GNU radio in [4]. But the CHT and PDR were not used for channel selection. The same drawback exists in [15] and [16]. The TACT learning scheme is superior to RL since it can use both node-to-node teaching and self-learning to adapt to the complex CRN spectrum conditions.

b. Channel Selection Metric: The concept of channel selection metric in CRN was proposed in [6], [17]. A SU selects an idle channel based on the channel conditions and queueing delay. A QoS-based channel selection scheme was proposed in [18], but the channel sensing accuracy and CHT were not considered. Note that the CHT determines the period over which a SU can occupy the channel without interruption from the PU. Further, authors in [19] proposed OFDM based MAC protocol for spectrum sensing and sharing which reduces the sharing overhead, but they did not consider the kind of channel that should be selected by the SU for transmission. Our spectrum evaluation scheme considers the channel dynamics with respect to the interference, fading loss, and other channel variations.

c. Decoding-CDF based Spectrum Adaptation: The rateless codes have been used in wireless communications due to its property of recovering the original data with low error rate. The popular rateless codes include the Spinal codes [10], [20], Raptor codes [21] and Strider codes [22], [23], [24]. The rateless codes for CRNs were proposed in [12], [25]. Authors in [12] proposed a feedback technique for rateless codes using multi-user MIMO to improve the QoS and to provide delay guarantee. Authors in [4] used decoding-CDF with the Spinal codes. In this paper, we use decoding-CDF along with our prioritized Raptor codes (PRC) [5] to perform spectrum adaptation.

III. Channel Selection Metric

In order to select a suitable channel for spectrum handoff, the SU should consider the time varying and spatial channel characteristics. The time-varying channel characteristics comprise of CHT and PDR, which are mainly observed due to PU interruption and SU contentions, and the spatial characteristics comprise of achievable throughput and PER observed due to the SU mobility. As mentioned in Section I, the CSM comprises of CUF, PDR and flow throughput which are described below.

A. Channel Utilization Factor (CUF)

If a busy channel is detected as idle, this misinterpretation is called as false alarm, which is a key parameter of spectrum sensing accuracy. We use the spectrum sensing accuracy and CHT for evaluating the effective channel utilization. From [3], we know that a higher detection probability, P_d, has a low false alarm probability, P_f. Hence we express the spectrum sensing accuracy as

$$M_A = P_d(1 - P_f) \tag{1}$$

If T denotes the total frame length and τ is the channel sensing time, the transmission period is T - τ. We assume that the PU arrival rate λ_{ph} follows the Poisson distribution and the CHT with duration t has the following probability distribution,

$$f(t) = \lambda_{ph} e^{-(\lambda_{ph})t} \tag{2}$$

Since PU's arrival time is unpredictable, it can interfere with the SU's transmission. Hence, the predictable interruption duration can be determined as [26],

$$y(t) = \begin{cases} T-\tau-t, & 0 \le t \le (T-\tau) \\ 0, & t \ge (T-\tau) \end{cases} \tag{3}$$

The SU transmits the data with an average collision duration [26] as,

$$\bar{y}(T) = 1 - \int_0^{T-\tau} (T-\tau-t) f(t)\, d(t) = (T-t) - t(1 - e^{(-\frac{(T-\tau)}{t})}) \tag{4}$$

Hence, the probability that a SU experiences the interference from a PU within its frame transmission duration is given by

$$P_p{}^s = \frac{\bar{y}(T)}{(T-\tau)} = 1 - \frac{t}{(T-\tau)}\left(1 - e^{\left(-\frac{(T-\tau)}{t}\right)}\right) \tag{5}$$

$$\tag{6}$$

The total channel utilization (CUF) is determined by using CHT and probability of interference from PU as,

$$CUF = M_A \frac{(T-\tau)}{T}(1 - P_p{}^s) \tag{7}$$

Substituting the results from (6) in (7), the CUF can be defined as follows,

$$CUF = M_A . \frac{t}{T}\left(1 - e^{\left(-\frac{(T-\tau)}{t}\right)}\right) \tag{8}$$

The CUF can be used to represent the spectrum evaluation results for the selection of an optimal channel. According to IEEE 802.22 recommendations, the probability of correct detection, $P_d = [0.9, 0.99]$ and the probability of false alarm, $P_f = [0.01, 0.1]$. Therefore, the probability of spectrum sensing accuracy is $P_d(1 - P_f) = [0.81, 0.99]$.

B. Non-Preemptive M/G/1 Priority Queueing Model

We use a non-preemptive M/G/1 priority queueing model where a lower priority SU accesses channel without interruption from higher priority SUs. We denote *j*=1 (or *N)* as the highest (or lowest) priority SU. However, any SU transmissions can be interrupted by a PU. When the channel becomes idle, a higher priority SU will be served. When a SU is interrupted by a PU, it can either stay-and-wait in the same channel until it becomes idle again, or handoff to another suitable channel.

Let $Delay_{j,i}$ be the delay of a SU_j connection due to the first $(i-1)$ interruptions. A SU_j packet will be dropped if its delay exceeds the delay deadline d_j. In our previous work [2], we deduced the $PDR_{j,i}^{(k)}$ as the probability of packet being dropped during the i^{th} interruption for channnel k with packet arrival rate, λ, and mean service rate, μ. It equals to the probability of handoff delay $E[D_{j,i}^k]$ being larger than $d_j - Delay_{j,i}$ [2].

$$PDR_{j,i}^{(k)} = \rho_{j,i}^{(k)} . exp(-\frac{\rho_{j,i}^{(k)} \times (d_j - Delay_{j,i})}{E[D_{j,i}^{(k)}]}) \tag{9}$$

Here, $\rho_{j,i}^{(k)}$ is the normalized load of channel k caused by type j SU. It is defined as follows,

$$\rho_{j,i}^{(k)} = \frac{\lambda_i}{\mu_k} \le 1 \tag{10}$$

C. Throughput Determination in Decoding-CDF based Rateless Transmission

After we identify a high-CUF channel, the next step is to transmit the SU's packets in this channel. Even a channel with high CUF can experience the time varying link quality due to the mobility of SU. Therefore, link adaptation is important to avoid frequent spectrum handoffs. Generally, the sender needs to adjust its data rate depending on the channel conditions since a poor link (lower channel SNR) can result in a higher packet loss rate. For example, in IEEE 802.11, the sender uses the channel SNR to select a suitable modulation constellation and forward error correcting (FEC) code rate from a set of discrete values. Such a channel adaptation cannot achieve a smooth rate adjustment since only a limited number of adaptation rates are available. Because channel condition variations can occur on very short time scales (even at the sub-packet level), it is challenging to adapt to the dynamic channel conditions in CRNs.

Rateless codes have shown promising performance improvement in multimedia transmission over CRNs [5]. At the sender side, each group of packets is decomposed into symbols with certain redundancy such that the receiver can reconstruct the original packets as long as enough number of symbols are received. The sender does not need to change the modulation and encoding schemes. It simply keeps sending symbols until an ACK is received from the receiver, signaling that enough symbols have been received to reconstruct the original packets. The sender then sends out the next group of symbols. For a well-designed rateless code, the number of symbols for packets closely tracks the changes in the channel conditions.

In this paper, we employ our unequal error protection (UEP) based prioritized Raptor codes (PRC) [5]. In PRC, more symbols are generated for the higher priority packets than the lower priority packets. As a result, PRC can support higher reliability requirements of more important packets. We describe below how we can achieve cognitive link adaptation through a self-learning of ACK feedback statistics (such as inter-arrival time gaps between two feedbacks). We also show how a SU can build a decoding-CDF by using the previously transmitted symbols and how it can be used for channel selection and link adaptation.

1) CDF-Enhanced Raptor Codes: In rateless codes, after sending certain number of symbols, the sender pauses the transmission and waits for a feedback (ACK) from the receiver. No ACK is sent if the receiver cannot reconstruct the packets, and the sender needs to send extra symbols. Each pause for ACK introduces overhead in terms of the

total time spent on symbol transmission plus ACK feedback [4]. The decoding-CDF defines the probability of decoding a packet successfully from the received symbols. In the CDF-enhanced rateless codes, the sender can use the statistical distribution to determine the number of symbols it should send before each pause. The CDF distribution is sensitive to the code parameters, channel conditions, and code block length. Surprisingly, only a small number of records on the relationship between n (number of symbols sent between two consecutive pauses) and τ (ACK feedback delay) are needed to obtain the CDF curve [4].

In order to speed up the CDF learning process, the Gaussian approximation can be used which provides a reasonable approximation at low channel SNR, and its maximum likelihood (ML) requires only mean (μ) and variance (σ^2). In addition, we introduce the parameter α, which ranges from 0 (means no memory) to 1 (unlimited memory), to represent the importance of past symbols in the calculation. This process has two advantages: the start-up transition dies out quickly, and the ML estimator is well behaved for $\alpha = 1$. The Algorithm 1 defines the Gaussian CDF learning process.

Algorithm 1 : Decoding CDF Estimation by Gaussian Approximation

Input: alpha, *% learning rate [0,1]*
Step-1: Initialization
    NS =1 *% encoded samples*
    sum = 0
    sumsq = sum^2 + 0
Step-2: Update *% updating sum and samples*
    NS = NS*alpha +1
    sum = sum*alpha + NS
    sumsq=sumsq*alpha + NS^2
Step-3: Get CDF: *% estimating CDF by mean & variance*
    mean = sum/NS
    variance= sumsq/NS - mean^2
    estimate CDF

Using Algorithm 1, the decoding-CDF can be estimated by using the following standard equation,

$$F(x) = \int_0^{NS} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(NS-\mu)^2}{2\sigma^2}} dx \tag{11}$$

Here, NS, μ and σ are the number of symbols, mean and variance, respectively.

For the observed link SNR, we can determine the number of symbols that need to be transmitted in order to decode the packet successfully. When the channel condition degrades interms of PER but $PDR \leq PDR_{th}$, the additional symbols are transmitted to adapt to the current channel conditions, which avoids unnecessary spectrum handoff. After the number of transmitted symbols reaches the maximum value, $(NS)_{max}$, the SU should perform spectrum handoff to a new channel. This is called as link adaptation using decoding-CDF.

After determining the number of symbols per packet (NS), which are required to successfully decode a packet, we can calculate the rateless throughput (TH) of channel k in a Rayleigh fading channel as [4],

$$TH_k = \frac{2 \times f_s \times (NS)}{t} \qquad symbols/s/Hz \tag{12}$$

Where f_s and t are the sampling frequency and transmission time, respectively. The value of NS varies over time due to the Rayleigh fading channel and number of symbols per packet estimated using the decoding-CDF curve. Since each node observes either time spreading of digital pulses or time-varying behavior of the channel due to mobility, Rayleigh fading channel is appropriate due to its ability to capture both variations (time spreading and time varying).

The normalized throughput is:

$$(TH_k)_{norm} = \frac{TH_k}{(TH_k)_{ideal}} \tag{13}$$

Here, $(TH_k)_{ideal}$ is the ideal throughput calculated via Shannon capacity theorem.

Now we can integrate the above three models together into a weighted channel selection metric for *ith* interruption in *kth* channel for the SU with priority j [15],

$$U_{ij}^{(k)} = w_1 \star CUF + w_2 \star (1 - PDR_{ij}^{(k)}) + w_2 \star (TH_k)_{norm} \tag{14}$$

Where w_1, w_2 and w_3 are weights representing the relative importance of the channel quality, PDR and throughput, respectively. Here $w_1 + w_2 + w_3 = 1$. Their setup depends on application QoS requirements. For real-time applications, the throughput is more important than PDR. On the other hand, PDR is the most important factor for the FTP applications. For video applications, CHT (part of CUF model) is more important.

IV. Overview of Q-Learning based intelligent Spectrum Management(iSM)

In this paper, the Q-learning scheme is used to compare the performance of our proposed TACT-based learning scheme for intelligent spectrum mobility management. More details about Q-learning based spectrum decisions are available in [2]. The Q-learning uses special Markov Decision Process (MDP), which can be stated as a tuple (S, A, T, R) [13]. Here, S depicts the set of system states; A is the set of system actions at each state; T represents the transition probability, where $T = \{P(s, a, s\prime)\}$, and $P(.)$ the probability of transition from state s to s' when action a is taken; and $R : S \times A \mapsto R$ is the reward or cost function for taking an action $a \in A$ in state $s \in S$. In MDP, we intend to find the optimal policy $\pi^*(s) \in A$, i.e., a series of actions $\{a_1, a_2, a_3, ...\}$ for state s, in order to maximize the total discount reward function.

States: For SU_i, the network state before $(j+1)^{th}$ channel assignment is depicted as $s_{ij} = \{\chi_{ij}^{(k)}, \xi_{ij}^{(k)}, \rho_{ij}^{(k)}, \phi_{ij}^{(k)}\}$. Here k is the channel being used; $\chi_{ij}^{(k)}$ depicts the channel status (idle or busy); $\xi_{ij}^{(k)}$ is the channel quality (CSM); $\rho_{ij}^{(k)}$indicates the traffic load of channel; and $\phi_{ij}^{(k)}$ represents the QoS priority level of SU_i.

Actions: Three actions are considered for iSM scheme - stay-and-wait, stay-and-adjust and spectrum handoff. We denote $a_{ij} = \{\beta_{ij}^{(k)}\} \in A$ as the candidate of actions for SU_i

on state s_{ij} after the assignment of $(j+1)^{th}$ channel, and $\beta_{ij}^{(k)}$ represents the probability of choosing action a_{ij}.

The Q-learning algorithm aims to find an optimal action which minimizes the expected cost of the current policy $\pi^*(s_{i,j}, a_{i,j})$ for $(j+1)^{th}$ channel assignment to SU_i. It is based on the value function $V^{\pi}(s)$ that determines how good it is for a given agent to perform a certain action under a given state. Similarly, we use the action value function, $Q^{\pi}(s,a)$; It defines which action has low cost in the long term. Bellman optimality equation gives the high and discounted long-term rewards [27]. For the sake of simplicity, in further sections we consider $s_{i,j}$ as s, action $a_{i,j}$ as a, and state $s_{i,j+1}$ as s'.

Rewards: The reward R of an action is defined as the predicted reward function for data transmission, for a certain channel assignment. For multimedia data, we use the mean opinion score (MOS) metric. Based on our previous work [2], the MOS can be calculated as follows,

$$R = MOS = \frac{a_1 + a_2 FR + a_3 ln(SBR)}{1 + a_4 TPER + a_5 (TPER)^2} \tag{15}$$

where FR, SBR and TPER are the frame rate, sending bit rate, total packet error rate, respectively. The parameter a_i, $i \in \{1,2,3,4,5\}$ is estimated using the linear regression process. MOS varies from 1 (lowest) to 5 (highest). When the channel status is idle, 'transmission' is an ideal action to take, which would achieve MOS close to 5. On the other hand, when PDR (State: traffic load) or PER (State: channel quality) is high, low MOS would be achieved which reflects poor performance in the acquired channel.

The estimation of expected discounted reinforcement of taking action a in state s, $Q^*(s,a)$ can be written as [2],

$$Q^*(s,a) = E(R_{i,j+1}) + \gamma \sum_{s'} P_{s,s'}(a) \max_{a' \in A} Q^*(s,a) \tag{16}$$

We adopt *softmax policy* for long-term optimization. $\pi(s,a)$, which determines the probability of taking action a, can be determined by utilizing Boltzmann distribution as [2]

$$\pi(s,a) = \frac{exp(\frac{Q(s,a)}{\tau})}{\sum_{a' \in A} exp(\frac{Q(s,a')}{\tau})} \tag{17}$$

Here, Q(s,a) defines the affinity to select action a at state s; it is updated after every iteration. τ is the temperature. The Boltzman distribution is chosen to avoid jumping into exploitation phase before testing each action in every state. The high temperature indicates the exploration of the unknown state-action values, whereas the low temperature indicates the exploitation of known state-action pairs. If τ is close to infinity, the probability of selecting an action follows the uniform distribution, i.e., the probability of selecting any action is equal. On the other hand, when τ is close to zero, the probability of choosing an action associated with the highest Q-value in a particular state is one.

Fig. 2 shows the procedure of using Q-learning for iSM. Here the dynamic spectrum conditions are captured by the states, which are used for policy search in order to maximize the reward function. The optimal policy determines the corresponding spectrum management action in the current round.

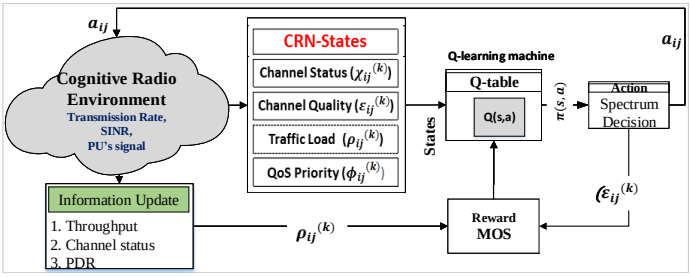


Fig. 2: The Q-learning based iSM.

V. TACT BASED INTELLIGENT SPECTRUM MANAGEMENT(iSM)

The Q-learning based MDP algorithm could be very slow due to two reasons: (1) It requires the selection of suitable initial state/parameters in the Markov chain; (2) It also needs proper settings of Markov transition matrix based on different traffic, QoS and CRN conditions.

Let us consider a new SU which has just joined the network, and needs to build a MDP model. Instead of using trial-and-error to find the appropriate MDP settings, it may find a neighboring SU with similar traffic and QoS demands, and request it to serve as "expert" (or teacher) and transfer its optimal policies. Such teaching or transfer based scheme can considerably shorten the learning (or convergence) time.

We use the TACT model for the knowledge transfer between SUs, which consists of three components: actor, critic and environment [8][9]. For a given state, the actor selects and executes an action in a stochastic manner. This causes the system to transition from one state to another with a reward as feedback to the actor. Then the critic evaluates the action taken by the actor in terms of *time difference (TD)* error, and updates the value function. After receiving the feedback from the critic, the actor updates the policy. The algorithm repeats until it converges.

To apply TACT in our spectrum management scheme, we solve the following two issues:

(1) Selection of the Expert SU: We consider a distributed network without a central coordinator. When a new SU joins the network, it performs the localized search broadcasting the Expert-Seek messages. The nearby nodes may be located in the area covered by the same PU(s), and thus have similar spectrum availability. The SU should select a critic SU based on its relevance to the application, level of expertise, and influence of an action on the environment. To find the expert SU, the SUs share the following three types of information among them, i.e., channel statistics (such as CUF), node statistics (node mobility, modulation modes, etc.), and application statistics (QoS, QoE, etc.). The similarity of the SUs can be evaluated in an actor SU by using the manifold learning [6], which uses the Bregman Ball concept to compare the complex objects. The Bregman ball comprises of a center ($\mu(k)$) and a radius ($R(k)$). The data point X_p which lies inside the ball possesses strong similarity with $\mu(k)$. We define their distance as [6],

$$B(\mu_k, R_k) = \{X_t \in X : D_{\phi}(X_t, \mu_k) \leq R_k\} \tag{18}$$

Here $D(p,q)$ is known as the Bregman Divergence, which is the manifold distance between two signal points (the expert SU and learning SU). If the distance is less than a specified threshold, we conclude that p and q are similar to each other. All distances are visualized in Gephi (a network analysis and visualization software) [28], as shown in Fig. 3. The similarity calculation between any two SUs includes three metrics: (1) The application statistics, which mainly refer to the QoS parameters such as the data rates, delay, etc.; (2) The node statistics, which include the node modulation modes, location, mobility pattern, etc.; (3) The channel statistics, which include the channel parameters such as bandwidth, SNR, etc. The SU with the highest similarity value with the learning SU is chosen as the expert SU. In Fig. 3, SU3 is selected as the expert SU (i.e., the critic) since it has stronger similarity to the learning SU (SU1) compared to the rest of the SUs.

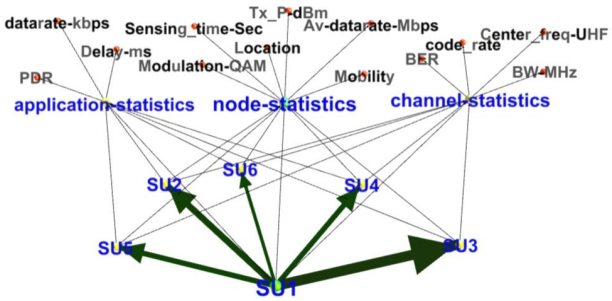


Fig. 3: Gephi-simulated expert SU search.

(2) The Knowledge Transfer via TACT Model: The actor-critic learning updates the value function and policy function separately, which makes it easier to transfer the policy knowledge compared to the other critic-only schemes, such as Q-learning and greedy algorithm. We implement the TACT-based iSM as follows:

(i) Action Selection: When a new SU joins the network, the initial state is s_{ij} in channel k. In order to optimize the performance, the SU chooses suitable actions to balance two explicit functions: a) searching for the new channel if the current channel condition degrades (exploration), and b) finding an optimal policy by sticking to the current channel (exploitation). This also enables the SU to not only explore a new channel but also to find the optimal policy based on its past experience. The probability of taking an action a in state s is determined, as mentioned in equation (17).

(ii) Reward: The MOS from equation (16) is evaluated as the reward resulting out of an action $a \in \{A\}$ taken in state $s \in \{S\}$.

(iii) State-Value Function Update: Once the SU chooses an action in channel k , the system changes the state from s to s' with a transition probability,

$$P(s'|s,a) = \begin{cases} 1, & s' \in S \\ 0, & otherwise \end{cases} \tag{19}$$

The total reward for the taken action would be $R_{s.a}$. The time difference (TD) error can be calculated from the difference between (i) the state-value function, $V(s)$ estimated in the previous state, and (ii) $R_{s,a} + V(s')$ at the critic [29],

$$\begin{aligned} \delta(s,a) &= R_{s,a} + \gamma \sum_{s' \in S} P(s'|s,a)V(s') - V(s) \\ &= R_{s,a} + \gamma V(s') - V(s) \end{aligned} \tag{20}$$

Subsequently, the TD error is sent back to the actor. By using TD error, the actor updates its state-value function as

$$V(s') = V(s) + \alpha(\nu_1(s,m))\delta(s,a) \tag{21}$$

Where $\nu_1(s,m)$ indicates the occurrence time of state s in these m stages. $\alpha(.)$ is a positive step-size parameter that affects the convergence rate. $V(s')$ remains as $V(s)$ in case of $s \neq s'$.

(iv) Policy Update: The critic would employ the TD error to evaluate the selected action by the actor, and the policy can be updated as [28],

$$p(s,a) = p(s,a) - \beta(\nu_2(s,a,m))\delta(s,a) \tag{22}$$

Here $\nu_2(s,a,m)$ denotes the occurrence time of action a at state s in these m stages. $\beta(.)$ denotes the positive step size parameter defined by $(m * logm)^{-1}$ [8]. Equations (17) and (22) ensure that an action in a specific state can be selected with a higher probability, if we reach the highest minimum reward, i.e., $\delta(s,a) < 0$.

If each action is executed for infinite times in each state and the learning algorithm follows a greedy exploration, the value function $V(s)$ and the policy function $\pi(s,a)$ will ultimately converge to $V^*(s)$ and π^*, respectively, with a probability of 1.

(v) Formulation of Transfer Actor-Critic Learning: Initially, the expert SU shares its optimal policy with the new SU. Let $p(s,a)$ denote the likelihood of taking action a in state s. When the process eventually converges, the likelihood of choosing a particular action a in a particular state s is relatively higher than that of other actions. In other words, if the spectrum handoff is performed based on a learned strategy by SU_i, the reward will be high in the long term. However, in spite of the similarities between the two SUs, they might have some differences, such as in the QoS parameters. This may make an actor SU take more aggressive action(s). To avoid these problems, the transferred policy should have a decreasing impact on the choice of certain actions, especially after the SU has taken its action and learned an updated policy. This is the basic idea of TACT-based knowledge transfer and self-learning.

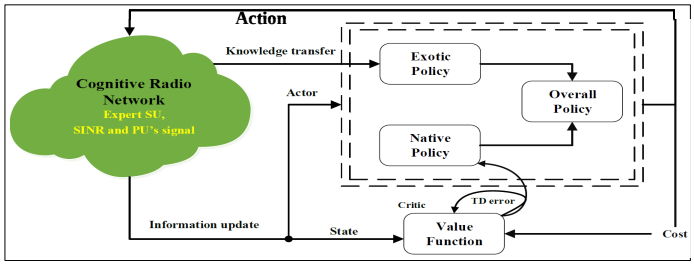


Fig. 4: TACT based SU-to-SU teaching.

The new policy update follows TACT principle (see Fig. 4), in which the overall policy of selecting an action is divided into

a *native policy*, p_n and an *exotic policy*, p_e. Assume at stage m, the state is s and the chosen action is a. The overall policy can be updated as [8]:

$$p_o^{(m+1)}(s,a) = [(1-\omega(\nu_2(s,a,m))p_n^{(m+1)}(s,a) + \omega(\nu_2(s,a,m))p_e^{(m+1)}(s,a)]_{-p_t}^{p_t} \quad (23)$$

Where $[x]_a^b$ with $b > a$, indicates the Euclidean distance of interval [a,b], i.e., $[x]_a^b = a$ if $x < a$; $[x]_a^b = b$ if $x > b$ and $[x]_a^b = x$ if a$\leq x \leq b$. In this scenario, a = $-p_t$ and b = p_t. In addition, $p_0^{(m+1)}(s,a) = p_0^{(m)}(s,a)$, $\forall a \in A$ but $a \neq a_{ij}$. And $p_n(s,a)$ updates itself according to equation (22).

During the initial learning process, the exotic policy $p_e(s,a)$ is dominant. Therefore, when the SU enters a state s, the presence of $p_e(s,a)$ forces it to choose the action, which might be optimal based on the expert SU. Subsequently, the proposed policy update strategy can improve the performance. We define $\omega \in (0,1)$ as the transfer rate, and $\omega \mapsto 0$ as the number of iterations goes to ∞. Thus the impact of exotic policy $p_e(s,a)$ is decreased. Algorithm 2 describes our proposed TACT-based iSM scheme.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme, including the channel selection, decoding CDF and enchanced TACT learning model.

A. Channel Selection:

We first examine our channel selection scheme (described in Section III), including the effect of spectrum sensing accuracy (M_A) and CHT. We setup the parameters as shown in Table I.

Parameters	Values
Number of time slots, (T)	100
False Alarm Probability,P_f	[0.01,0.1]
Detection probability, P_d	[0.9,0.99]
Exponential distribution rate $\lambda_{pi}, i = 0,1$	[0.02,1]
Temperature, τ	1000
Discount factor, γ	0.001
Transfer rate, ω	0.7
The number of channels	10
Learning rate, α (decoding CDF)	[0.9,0.8,0.99]
Packet aggregation cost, n_f	10

TABLE I: Simulation Parameters

We consider N=10 PUs, each of them possessing one primary channel, and randomly select the probability parameters given in Table I. Fig. 5a and 5b represent M_A and CHT, respectively. By considering both M_A and CHT, the SU determines the CUF for each channel and ranks them in the decreasing order as shown in Fig. 5c.

Fig. 6 shows the normalized throughput of the system that can be achieved by our channel selection scheme (BIGS) for different frame rates and PU idle durations (CHT). Here, BIGS refers to the channel sensing using Bayesian Inference with Gibbs Sampling [3]. For comparison, we also show the normalized throughput achieved by a random channel selection (RCS) scheme. Our scheme achieves better throughput than RCS because it selects the channel with high sensing accuracy as well as high CHT, whereas RCS does not consider the CHT and is also prone to channel miss detection and false alarm.

In Fig. 7, we compare the normalized throughput of our channel selection model with [16] and [17]. In our scheme, the SU senses the channel and ranks them based on the channel sensing accuracy and CHT. Similarly, authors in [17] performed the channel sensing based on the energy detection, and categorized the channels based on their CHT. In addition, they considered the directional antenna whereas we use the omni-directional antenna. Therefore, [17] has higher channel sensing accuracy than our scheme as the interference level is much lower in directional communication as compared to the omni communication. As a result, the throughput of [17] is higher than ours. To compare our schemne with [16], we consider that the channel can use one band at a time and

Algorithm 2 : TACT-based Spectrum Decision Scheme

Input: Channel, Node and Application statistics
Output: best policy $\pi(s,a)$ of SU_i

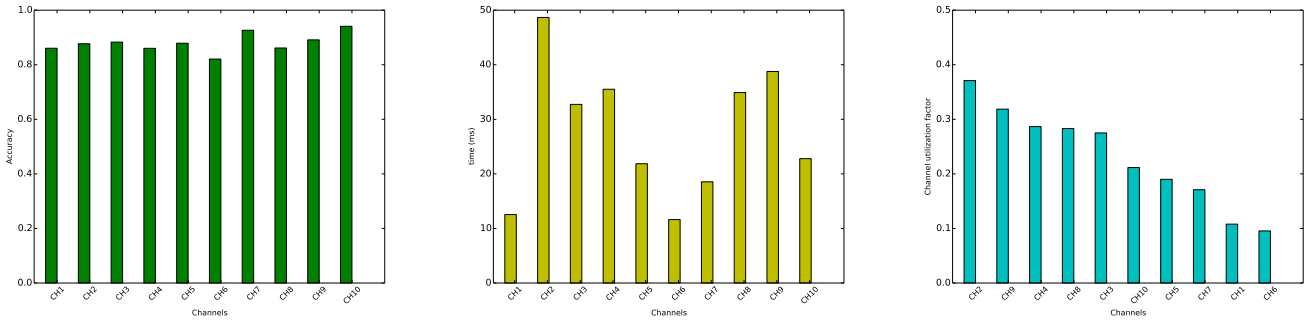
Part-I

Initialization
if node is new **then**
  if there is expert **then**
    Perform TACT algorithm from Part-II
  else
    Determine the channel k status and CUF from (8).
    Find PDR from (9) and $(TH)_{norm}$ from (13).
    Calculate $U_{ij}^{(k)}$ using (14) and select the best channel
    Perform Q-learning itself
  end if
else
  Perform TACT algorithm from Part-II
  if channel condition is below the threshold **then**
    Perform one of the three actions: stay-and-wait, stay-and-adjust, or Handoff
  end if
end if

Part-II

Input: Channel, Node and Application statistics
Output: best policy $\pi(s,a)$ of SU_i

Initialize $V^{\pi}(s)$ arbitrarily.
Exchange node information among node i and its neighbors.
Using manifold learning to find the expert.
Get the expert policy, i.e., exotic policy $P_e(s,a)$, from expert SU.
Initialize native policy, $p_n(s,a)$.
Repeat:
  Choose an action based on the initial policy $\pi^{(0)}$.
  Calculate MOS, update TD error using (20), state-value function (21), and native and overall policy using (22) and (23), respectively.
  Update the strategy function using (17).
end



(a) Spectrum sensing accuracy (b) PU idle duration (CHT) (c) Channel utilization factor

Fig. 5: The channel selection parameters.

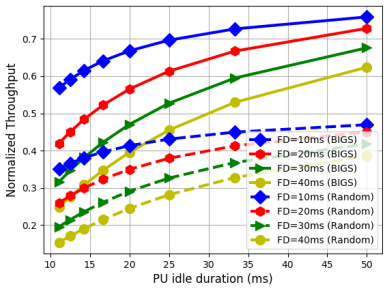


Fig. 6: Comparison of the proposed and random channel selection schemes. Here, FD represents the frame duration.

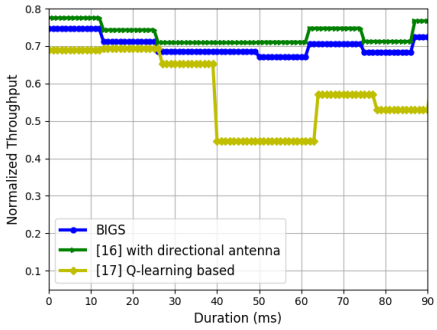


Fig. 7: Comparison of the proposed channel selection scheme with [16] and [17].

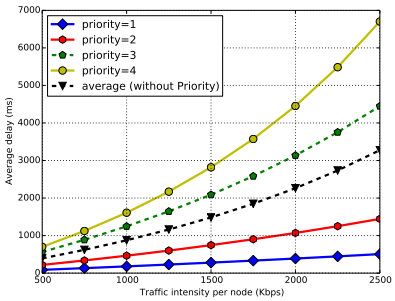


Fig. 8: Average delay for the non-preemptive M/G/1 priority queueing model and non-prioritized model.

also assume that the Q-learning has achieved the optimal condition. Alongside we also consider that SU communicates in its current channel until it is occupied by other users. Since the channel selection is random in [16], the SU may select a channel with small CHT even when a channel with longer CHT is available. Therefore, though its sensing accuracy is close to ours, the throughput is lower. Channel selection based on the channel ranking is very important to achieve smooth communication and to avoid frequent spectrum handoffs.

B. Average Queueing Delay:

We assume that the service time of SUs follows the exponential distribution, and the number of channels is 10. The maximum transmission rate of each channel is $3Mbps$, and the PER varies from 2% to 10%. Different priorities are assigned to the SUs depending on the delay constraint of their flow. The highest priority (priority = 1) is assigned to the interactive voice data with rate of $50Kbps$ and strict delay constraint of $50ms$. Priority 2 is assigned to the interactive Skype call with rate of $500Kbps$ and delay constraint of $100ms$. Priority 3 is assigned to the video-on-demand streaming data with rate of $> 1Mbps$ and delay constraint of $1sec$. Finally, the lowest priority (priority = 4) is assigned to the data without any delay constraint (e.g., file download). Fig. 8 shows that the non-preemptive M/G/1 priority queueing model outperforms the non-prioritized model. The idle channels are assigned based on the priority of the applications in priority model. The higher priority user(s) (such as voice data and real-time video) will get more channel access opportunities, which decreases their average queueing delay, whereas the lower priority user(s) experiences a longer average waiting time. In the non-prioritized model, all the applications are given the same priority, which leads to an increase in the average delay. Therefore, the priority based queueing model is suitable for SUs with different delay constraints.

C. Decoding CDF Learning:

In this section, we examine the performance of decoding CDF with Raptor codes over a range of symbols for different SNR values. Fig. 9 shows the plot of decoding CDF using Algorithm 1 for the SNR values from -5dB to 25dB. For higher

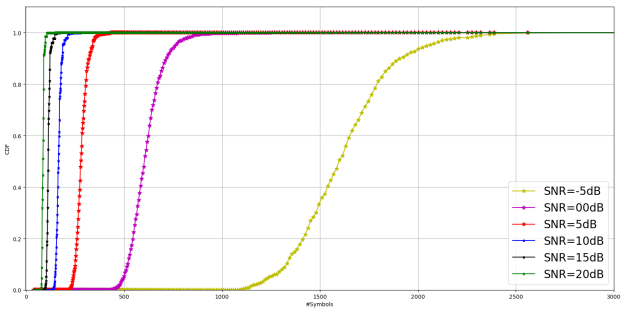


Fig. 9: Estimated CDF for different SNR levels.

(lower) SNR, we require less (more) symbols to decode a transmitted packet. The Rayleigh fading channel is used.

Using the decoding CDF, we examine the throughput for Raptor codes in Fig. 10. For better visualization, Fig. 11 zooms in a section of Fig. 10. As mentioned before, the decoding CDF enables us to find the optimal feedback strategy, i.e., when to pause for feedback and how many symbols should be transmitted before the next pause. The throughput is examined for a SU moving at a speed of 10 m/s over Rayleigh fading channel at 2.4GHz (channel $SNR = 15dB$) within a time range of 100 ms with a packet aggregation cost $n_f = 10$, which decides the number of packets to be aggregated to send an ACK. The throughput is estimated offline using Algorithm 1 with learning rate parameter, α, set to 0.9. It can be seen from Fig. 10 and 11 that α need not to be close to 1 to obtain a good performance. The throughput achieved by the Raptor codes is almost half of the Shannon capacity [4]. The decoding CDF performance is close to that of the ideal learning which is determined based on receiving ACKs from the receiver.

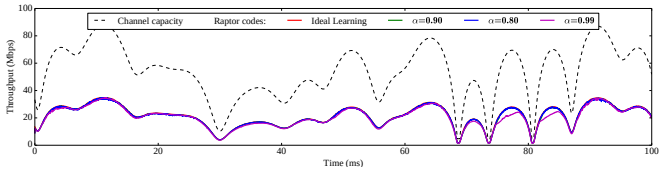


Fig. 10: Channel throughput estimation for Raptor codes for Rayleigh fading channel.

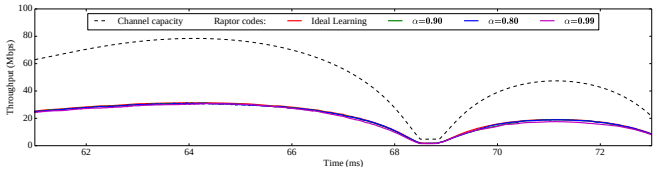


Fig. 11: Zoomed-in section of Figure 10 (for time 61-73 ms).

D. TACT Enhanced Spectrum Management Scheme:

In this section, we study the performance of our TACT-based spectrum mobility scheme. For 10 available channels with capacity of 3$Mbps$ each, we assume there are 10 different PUs with different data rates for transmission which can interrupt the SU transmission. Different SUs contending for the channel access also have different data rates. We study the performance of a SU which is supporting a Skype video call at *500 Kbps* and has a priority of 2. All SUs use the Raptor codes, and the expert SU teaches a new SU about its transmission strategy based on the decoding-CDF profile. We consider the following four cases. Case 1: The newly joined SU moves very slowly at <5mph; Case 2: The SU moves fast (>50mph) and experiences different channel conditions; Case 3: The SU moves fast but does not use the decoding CDF and pause control for transmission. Instead, it manually changes the symbol sending rate based on the current channel conditions; Case 4: The SU moves fast and uses the decoding CDF. We use the low-complexity MOS metric to estimate the received quality.

In Fig. 12 (for Case 1), the Q-learning based spectrum decision scheme outperforms the myopic approach, because the former takes spectrum decisions to maximize the long-terms reward (i.e., MOS) whereas the latter considers only the immediate reward. Further, our proposed TACT-based scheme outperforms the Q-learning scheme since the newly joined SU can learn from the expert SU, and thus spends less time in estimating the channel dynamics. Without the expert node, the node in Q-learning scheme learns everything by itself, and thus needs more time to converge to a stable solution. Fig. 13 shows the result for fast moving SU for Case 2, which experiences channel condition variations with time. Our proposed TACT scheme still performs better than the Q-learning scheme.

Fig. 14 depicts the Case 3 where the SU moves fast but does not use the decoding-CDF concept for Raptor codes. Since the SU is moving fast, it experiences different channel conditions. Once the SU attains the convergent state it achieves a high MOS value. But this does not guarantee that it will stay in the optimal state during the entire communication due to variations in channel conditions. Without the use of decoding-CDF, the SU is unable to adapt to the channel variations which results in the lower MOS value of around 4. In Fig. 15 (Case 4), the SU uses the CDF curve to learn the strategy of transmitting more symbols with lower overhead, and achieves a higher MOS of around 4.4. In both cases we can see that the MOS drops due to the change in channel condition at time slot 7. But CDF helps to quickly improve the MOS value to around 4.4.

Figure 16 shows the effect of transfer rate, ω on learning performance. We observe that the transfer rate has impact only at the beginning. Higher the transfer rate ($\omega = 0.8$), faster the adaptation to the network with less MOS variations. Whereas lower the transfer rate ($\omega = 0.2$), slower is the adaptation to the network and more are fluctuations in the MOS value. The performance converges after some iterations as the SU gradually builds up its own policy using the expert node.

Figure 17 shows that our TACT based spectrum decision scheme outperforms the Q (or RL) scheme [2] and the apprenticeship based transfer learning scheme [6]. In AL scheme, the student node uses the expert node's policy for its own spectrum decision. This model works well if both the student and expert nodes experience the same channel and traffic conditions. Our TACT based model, on the other hand, can tune the expert policy according to its own channel conditions in a few iterations.

VII. Discussion

Main concern in transfer learning approach is the overhead introduced by the expert search and the transfer of its knowledge (optimal policy) to the learner node. The proposed TACT learning-based spectrum decision requires a 'learner node' to communicate only with the closest neighbors, since only these nearby nodes are likely to have similar PU traffic distribution and channel conditions. This communication with neighbors can be easily achieved by the MAC (medium access control) protocols. It is also possible to piggyback this information exchange in the node discovery messages. Similarly, route discovery messages could also be used for this purpose. In this process, the learner node has more involvement and does not put much burden of transfer of the expert strategies on most other nodes in the network.

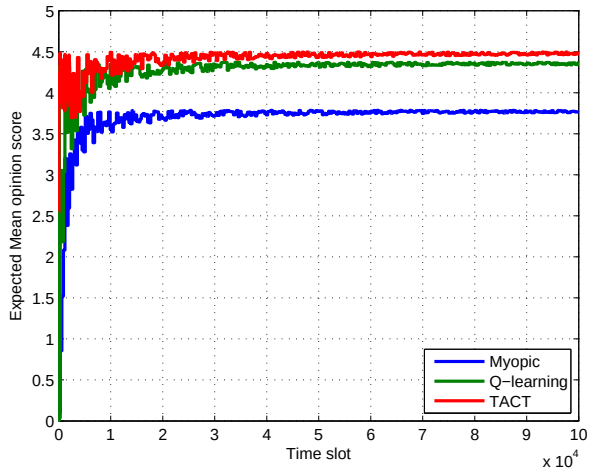


Fig. 12: The MOS performance for slow moving node.

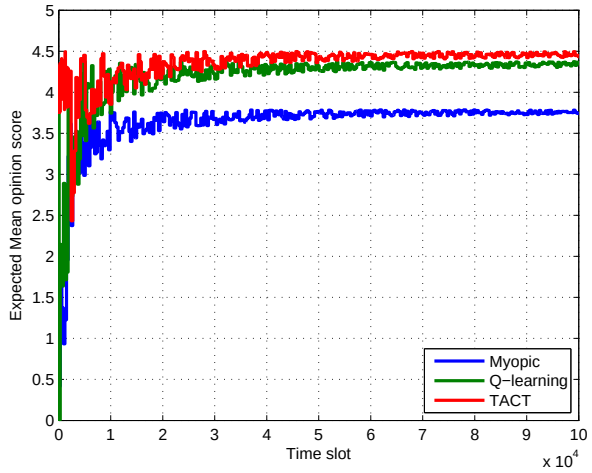


Fig. 13: The MOS performance for fast moving node.

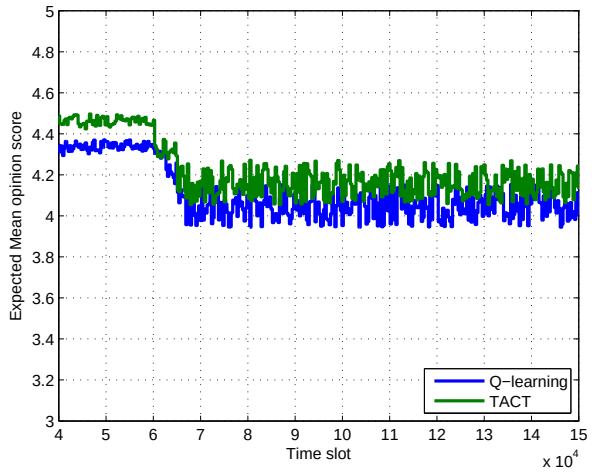


Fig. 14: The MOS performance comparison without the decoding-CDF.

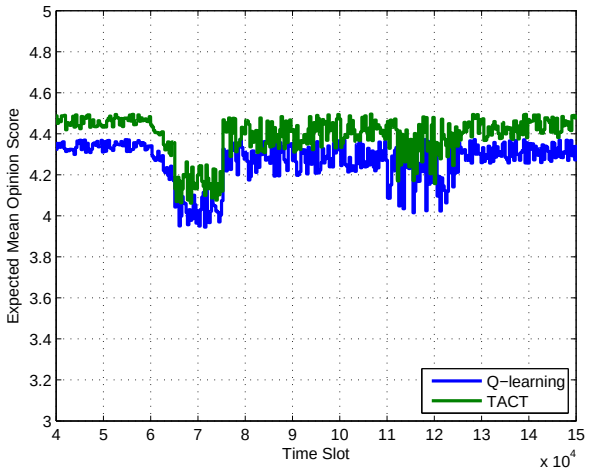


Fig. 15: The MOS performance with the use of decoding-CDF

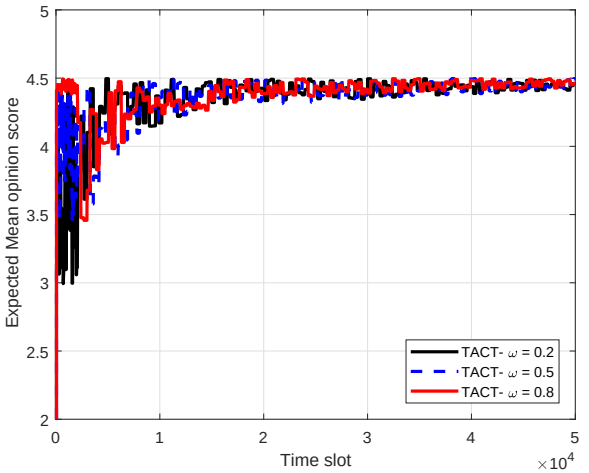


Fig. 16: The effect of transfer rate, ω on learning performance.

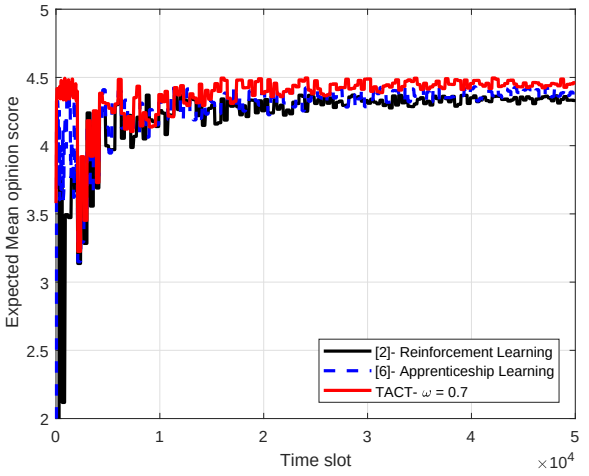


Fig. 17: The comparison of our TACT model with RL [2] and AL [6].

In fact, a node which is new to the network needs to exchange the control messages with its neighbors to find an expert node only in the beginning. If there is a new transmission task for an existing node, it might be able to use the policy it has learned over the previous transmissions without the need of triggering a new round of expert search. More importantly, the policy $\pi(s,a)$ is just an array of size 4 ($\approx 20 bytes$), which does not add much overhead to the packet size.

VIII. Conclusions

An intelligent spectrum management scheme was designed by using the TACT based learning algorithm. The primary goal of this scheme was to make an intelligent spectrum handoff and stay-and-wait decision for the rateless multimedia transmissions in dynamic CRN links. The spectrum desicion scheme requires a good knowledge of channel quality. For accurate channel quality evaluation of a link, we calculated the CUF. To adapt to the dynamic CRN channel conditions, we used the CDF-enhanced, UEP-based Raptor codes to achieve intelligent link adaptation. A good link adaptation strategy can significantly reduce the spectrum handoff events. The proposed cognitive learning scheme can also be useful in other CRN tasks, such as multimedia streaming over CRN, and dynamic route establishment.

In future, we intend to further enhance our TACT-based model, by using the budget-limited teaching process, in order to efficiently transfer the important parameters from an expert SU to a learning SU within the given time constraints. The expert search model will be based on the manifold learning and NMF (non-negative matrix factorization) pattern extraction/recognition schemes, so that a more suitable expert SU can be found in the neighborhood of a learning SU.

IX. Acknowledgment of Support and Disclaimer

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-13-1-046. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

References

[1] F. C. Commission, "Spectrum Policy Task Force Report," Nov. 2002.

[2] Y. Wu, F. Hu, S. Kumar, Y. Zhu, A. Talari, N. Rahnavard, and J. D. Matyjas, "A Learning-based QoE-Driven Spectrum Handoff Scheme for Multimedia Transmissions over Cognitive Radio Networks," *IEEE J. Selected Areas in Communications*, vol. 32, no. 11, pp. 2134–2148, 2014.

[3] X. Xing, T. Jing, Y. Huo, H. Li, and X. Cheng, "Channel quality prediction based on Bayesian inference in cognitive radio networks," in *IEEE INFOCOM*, 2013, pp. 1465–1473.

[4] P. A. Iannucci, J. Perry, H. Balakrishnan, and D. Shah, "No symbol left behind: A link-layer protocol for rateless codes," in *18th ACM Annual Intl. Conf. Mobile Computing and Networking*, 2012, pp. 17–28.

[5] Y. Wu, S. Kumar, F. Hu, Y. Zhu, and J. D. Matyjas, "Cross-layer Forward Error Correction Scheme Using Raptor and RCPC Codes for Prioritized Video Transmission over Wireless Channels," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 1047–1060, 2014.

[6] Y. Wu, F. Hu, S. Kumar, J. D. Matyjas, Q. Sun, and Y. Zhu, "Apprenticeship Learning based Spectrum Decision in Multi-Channel Wireless Mesh Networks with Multi-Beam Antennas," *IEEE Trans. Mobile Computing*, vol. 17, no. 2, pp. 314–325, 2017.

[7] L. Giupponi, A. Galindo-Serrano, P. Blasco, and M. Dohler, "Docitive networks: An emerging paradigm for dynamic spectrum management," *IEEE Trans. Wireless Communication*, vol. 17, no. 4, pp. 47–54, 2010.

[8] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: A Transfer Actor-Critic Learning Framework for Energy Saving in Cellular Radio Access Networks," *IEEE Trans. Wireless Communications*, vol. 13, no. 4, pp. 2000–2011, 2014.

[9] A. Koushik, F. Hu, J. Qi, and S. Kumar, "Cognitive Spectrum Decision via Machine Learning in CRN," in *Conf. Information Technology: New Generations*. Springer, 2016, pp. 13–23.

[10] J. Perry, H. Balakrishnan, and D. Shah, "Rateless Spinal Codes," in *10th ACM Workshop on Hot Topics in Networks*, 2011, p. 6.

[11] Y. Wu, F. Hu, Y. Zhu, and S. Kumar, "Optimal spectrum handoff control for CRN based on hybrid priority queuing and multi-teacher apprentice learning," *IEEE Trans. Vehicular Technology*, vol. 66, no. 3, pp. 2630–2642, 2017.

[12] X. Chen and C. Yuen, "Efficient resource allocation in a rateless-coded MU-MIMO cognitive radio network with QoS provisioning and limited feedback," *IEEE Trans. Vehicular Technology*, vol. 62, no. 1, pp. 395–399, 2013.

[13] Y. Ren, P. Dmochowski, and P. Komisarczuk, "Analysis and implementation of reinforcement learning on a GNU radio cognitive radio platform," in *5th Intl. Conf. Cognitive Radio Oriented Wireless Networks and Communications, Cannes, France*, 2010.

[14] B. F. Lo and I. F. Akyildiz, "Reinforcement Learning based Cooperative Sensing in Cognitive Radio Ad Hoc Networks," in *21st IEEE PIMRC*, 2010, pp. 2244–2249.

[15] N. Hosey, S. Bergin, I. Macaluso, and D. O'Donohue, "Q-Learning for Cognitive Radios," in *Proc. China-Ireland Information and Communications Technology Conf., ISBN 9780901519672*, 2009.

[16] Z. Chen and R. C. Qiu, "Q-learning based Bidding Algorithm for Spectrum Auction in Cognitive Radio," in *IEEE Southeastcon*, 2011, pp. 409–412.

[17] Y. Dai and J. Wu, "Sense in Order: Channel Selection for Sensing in Cognitive Radio Networks," in *8th IEEE Intl. Conf. Cognitive Radio Oriented Wireless Networks (CROWNCOM)*, 2013, pp. 74–79.

[18] L. Zappaterra, "QoS-Driven Channel Selection for Heterogeneous Cognitive Radio Networks," in *ACM Conf. CoNEXT Student Workshop*, 2012, pp. 7–8.

[19] L. Wang, K. Wu, J. Xiao, and M. Hamdi, "Harnessing frequency domain for cooperative sensing and multi-channel contention in CRAHNs," *IEEE Trans. Wireless Communications*, vol. 13, no. 1, pp. 440–449, 2014.

[20] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah, "Spinal Codes," in *ACM SIGCOMM Conf. Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2012, pp. 49–60.

[21] A. Shokrollahi, "Raptor Codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[22] A. Gudipati and S. Katti, "Strider: Automatic Rate Adaptation and Collision Handling," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, 2011, pp. 158–169.

[23] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[24] U. Erez, M. D. Trott, and G. W. Wornell, "Rateless Coding for Gaussian Channels," *IEEE Trans. Information Theory*, vol. 58, no. 2, pp. 530–547, 2012.

[25] X.-Y. Zhi, Z.-Q. He, and W.-L. Wu, "A Novel Cooperation Strategy based on Rateless Coding in Cognitive Radio Network," *Intl. J. Advancements in Computing Technology*, vol. 4, no. 8, pp. 333–347, 2012.

[26] W. Tang, M. Z. Shakir, M. A. Imran, R. Tafazolli, and M.-S. Alouini, "Throughput Analysis for Cognitive Radio Networks with Multiple Primary Users and Imperfect Spectrum Sensing," *IET Communications*, vol. 6, no. 17, pp. 2787–2795, 2012.

[27] A. D. Redish, S. Jensen, A. Johnson, and Z. Kurth-Nelson, "Reconciling Reinforcement Learning Models with Behavioral Extinction and Renewal: Implications for Addiction, Relapse, and Problem Gambling." *Psychological Review*, vol. 114, no. 3, p. 784, 2007.

[28] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," 2009.

[29] V. R. Konda and V. S. Borkar, "Actor-Critic–Type Learning Algorithms for Markov Decision Processes," *SIAM J. Control and Optimization*, vol. 38, no. 1, pp. 94–123, 1999.