www.ietdl.org

Published in IET Communications
Received on 5th September 2013
Revised on 13th May 2014
Accepted on 23rd May 2014
doi: 10.1049/iet-com.2013.0796

ISSN 1751-8628

Physical layer-optimal and cross-layer channel access policies for hybrid overlay–underlay cognitive radio networks

Ashok K. Karmokar, Sivasothy Senthuran, Alagan Anpalagan

Department of Electrical and Computer Engineering, Ryerson University, 350 Victoria Street, Toronto, ON, M5B 2K3, Canada
E-mail: alagan@ee.ryerson.ca

Abstract: The authors study the opportunistic spectrum access techniques for hybrid overlay–underlay cognitive radio networks. A secondary user (SU) chooses a channel, transmission mode and adjusts its power so that the interference limit is not crossed and its throughput is maximised. The authors assume that multiple primary user (PU) channels are available and the SU conducts spectrum sensing to access the channels. The objective is to maximise the throughput by switching between the overlay and underlay transmission modes. Using finite-horizon partially observable Markov decision process framework, the authors first study the optimal policies, where the PU is assumed to be in busy, concurrent or idle state, and the SU either stays idle or transmits with any of the two designed power levels. Although the PU's states are hidden, their activity statistics, transmission ranges and interference thresholds are assumed to be known. Via Monte Carlo simulation, the authors evaluate the performance of physical layer optimal policy (PLOP) and cross-layer policy (CLAP) and compare them with a fully observable optimal policy. The beliefs in each slot for both policies are updated using the forward algorithm based technique. Simulation results show that the proposed CLAP is more throughput efficient than the conventional PLOP.

Nomenclature

Notation for common variables

N	number of PU channels
T_{f}, T_{slot}	duration of a radio frame and time-slot in seconds, respectively
T_{s}, T_{d}	sensing and data transmission time, respectively
H	number of time-slot in a radio frame (also horizon)
P_{switch}	the probability of sensing next out-of-band channel after PU reappears in the current channel
$P_{o,u}$	the probability of switching to underlay mode from overlay mode
P_{MD}, P_{FA}	misdetection and false alarm probabilities, respectively
P_{fe}	feedback error probability
$\mathcal{S}$, s_i, s^t	set of states (with dimension 1-by-3), ith state and state at time-slot T,^{t} respectively
$\mathcal{U}$, u_i	set of actions (with dimension 1-by-3) and ith action, respectively
$\mathcal{O}$, o_i	observation vector and ith observation, respectively
$\mathcal{P}_s(u)$	transition probability matrix
$P_{s_i,s_j}(u)$	probability of transition from state s_i to s_j for action u
P_{st}	self-transition probability
$P(o_j/s_i, u_k)$	observation probability of o_j given state s_i and action u_k
$R_{s_i}(u)$	the reward value in state s_i when action u is chosen
N	number of time-slot to go
$O_{1:m}$	observation sequence
$f_{0:m}(s_i)$	the probability of state s_i given the observations
$\boldsymbol{f}_{0:m}$	the 3-by-1 column vector containing probability of state $s_i \in S$ given the observations
$\boldsymbol{O}_{\mathrm{d}}$	diagonal observation matrix (with dimension 3-by-3)
$\boldsymbol{B}$	normalisation factor in belief update formula
ζ_i	ith threshold for energy detector
$X(y)$	test statistics for the energy detector's received signal y
$P(o_i\|s_j)$	probability of observing o_i in state s_j
$\boldsymbol{O}_{\mathrm{f}}$	feedback observation vector (with dimension 1-by-2)
q_1, q_2	no collision and collision observation
$\boldsymbol{Q}$	feedback observation matrix
q_{ik}	probability of q_k feedback in state s_i

1 Introduction

In order to cope with the ever increasing bandwidth demand in the future generation wireless networks, cognitive radio is thought to be one of the most promising technologies [1].

Wireless spectrum is limited a natural resource and it is severely under-utilised in some bands (TV transmission, amateur radio and so on). As well, it is extremely crowded in consumer radio communications band because of current static allocation of wireless spectrum. Following a study conducted by the federal communications commission (FCC), the cognitive radio network (CRN) technologies have been incepted for the dynamic and opportunistic utilisation of the under-utilised spectrum. An opportunistic secondary user (SU) can reuse a free piece of spectrum (also called a spectrum hole) that is licensed to a primary user (PU). In order to reuse spectrum holes, an SU must first carry out-of-band sensing [2]. If a spectrum hole is found by the sensor, the SU can then use it for its data transmission. However, it should also conduct in-band sensing periodically so that it can vacate the acquired channel when the incumbent user re-appears and starts transmission [3]. In IEEE 802.22 Wireless Regional Area Networks (WRANs), PUs should be detected within 2s of their reappearance with the sensing error probabilities no greater than 0.1 [4]. In order to avoid interference among multiple sensors and achieve reliable sensing, it is also necessary that all the SUs sense the channel during quiet period. In this period, all SUs should postpone their transmissions so that any sensor monitoring the channel, may observe the presence/absence of PU signals without interference.

Channel access techniques based on physical layer (PHY) and medium access control (MAC) layer channel sensing have been discussed in the literature. In [5], a decentralised MAC protocol for *ad hoc* CRNs has been proposed that senses channel in each time-slot and takes opportunistic channel access decisions. Zhang and Tsang in [6] extended the partially observable Markov decision process (POMDP)-based optimal and myopic greedy suboptimal techniques of [5] to cooperative sensing CRNs. In [7], Liang *et al.* studied the problem of designing the sensing slot duration to maximise the achievable throughput. A comparative study of energy detection and feature detection in-band spectrum sensing techniques in WRANs has been studied in [4]. Although the above works consider that the PU's state may be in one of the two states, namely, busy and idle, Senthuran *et al.* in [8] consider a third state (where concurrent transmission is possible) that a PU user may occupy, and have studied opportunistic access strategies for three-state CRNs. An information theoretic perspective of three paradigms, namely, underlay, overlay and interweave has been discussed in [9].

The following works also deal with the capacity maximisation schemes that intelligently combine the underlay and overlay modes for the purpose. In [10], Khoshkholgh *et al.* analysed the achievable capacity for overlay, underlay and mixed access strategies. A hybrid strategy that combines overlay and underlay spectrum access schemes is studied in [11] utilising a double-threshold energy detection method and Markov chain model. The SUs can switch between full-access and partial-access modes dynamically. Bansal *et al.* studied a joint overlay and underlay power allocation scheme for OFDM-based cognitive radio systems in [12] in order to maximise the transmission capacity. A hybrid cognitive radio system by combining both the underlay and the overlay modes is studied to maximise the average throughput of a secondary network in [13]. In [14], an overlay/underlay spectrum sharing techniques is studied for multi-operator environment in CRNs. Using continuous time Markov chain model, Nair *et al.* [15] analysed the hybrid spectrum access scheme which combines overlay and underlay spectrum sharing schemes to improve the system throughput. A spectrum access and power adaptation technique on a single PU channel is discussed in [16] using three hidden states of the PU channel.

Motivated by the above studies, we study the spectrum access and power adaptation techniques for a CRN with three-states. Because, in overlay mode, the SU remains 'off' when the PU channel is in 'busy' mode. However, if the SU switches to the underlay mode, it can still transmit with low power provided the interference threshold is maintained. Moreover, since multiple channels are available, the SU can find another free channel. In this paper, we first study the optimal channel access and power adaptation strategy by formulating the problem as a finite-horizon POMDP that optimises the throughput and avoids the interference to the PUs. The optimal alpha vectors for the formulated POMDP problem is computed using the incremental-pruning algorithm. Then the instantaneous optimal policies in a particular time-slot are obtained from the tabulated optimal alpha vectors and the belief of the states that are estimated using the forward algorithm from the spectrum sensing in each time-slot for physical layer optimal policy (PLOP). Since the PLOP requires channel to be sensed in every time-slot (which causes loss in both the energy and the data transmission time), we propose a novel cross-layer policy (CLAP)-based algorithm that updates the state belief using spectrum sensing result in the first time-slot and then using ACK/NAK information (obtained from data link-layer) of the previous time-slot in the rest of the frame. When the PU reappears in the current channel, the SU takes the decision whether to switch in underlay mode or sense another channel for spectrum hole. We study the throughput and collision performance results via simulation for different number of PU channels, PU channel mixing rates and error occurrence both in sensing and in feedback.

The summary of our contributions are given below:

- We formulate the channel access, transmission mode switching and power allocation techniques for CRNs, where multiple PU channels are available.
- We formulate the problem as a POMDP and discuss its components. Incremental pruning based algorithm is presented to find the optimal alpha vectors.
- We propose a CLAP and compare its performance with traditional PLOP. We also compared both policies with benchmark fully observable optimal policy (FOOP).
- We have carried out Monte Carlo simulations to show the effect of various system parameters, such as number of PU channels, self-transition probabilities, sensing error and feedback error.

The paper is organised as follows. In Section 2, we describe the system model for the problem. The formulation of the problem as a POMDP is given in Section 3 and its solution technique to obtain the optimal alpha vectors is discussed in Appendix 1. Depending on belief tracking, two policies: PLOP and CLAP are described in Section 4. The FOOP is also discussed in this section. The belief tracking algorithm, namely forward algorithm, is discussed in Section 5. We provide simulation results in Section 6 and conclude in Section 7.

2 System model

In this paper, a hybrid underlay–overlay CRN is considered, where an SU is intelligently accessing channels that are

licensed to the PUs. The SU adjusts its transmission power so that it can concurrently transmits its information data with the PU transmission in underlay mode. However, the transmitter power should be small enough so that the interference perceived by the PU is below some acceptable threshold. In overlay mode, the SU senses the channel to find the spectrum holes, where the PU is absent. The SU can use higher transmission power. However, the SU should periodically sense the channel, and either leave the channel free or switch to underlay mode when the presence of PU is detected. In the first case, the PU needs to carry out-of-band spectrum sensing to find out spectrum opportunity. In the latter case, the PU can continue to transmit with lower power by switching to underlay transmission mode. Let us assume that P_{switch} denotes the probability of sensing the next out-of-band channel. Therefore, the probability of switching to the underlay mode is just $P_{o,u} = \text{Prob}(\text{Underlay}|\text{Overlay}) = 1 - P_{\text{switch}}$.

Let us assume that the time is discretised into a finite number of time-slots, and duration of each time-slot is T_{slot} seconds. A radio frame of duration $T_{\text{f}} = H \times T_{\text{slot}}$ consists of H discrete time-slots, where H is the horizon of the transmission. In the time-slots where spectrum sensing is done, a time-slot consists of sensing time T_{s} seconds and data transmission time $T_{\text{d}} = T_{\text{slot}} - T_{\text{s}}$ seconds, as shown in Fig. 1*a*. Data transmission time also includes the control and feedback signal times. When spectrum sensing is not carried out in a particular time-slot, the whole time-slot T_{s} is used for data transmission. Later, we will note that in some transmission policy, the spectrum sensing is not necessary in all but the first few time-slots. Let us assume that the number of possible PU channels in the geographic region is N.

The system evolves as follows: at the beginning of a radio frame, an SU decides how many and which channels to sense, and in what sequence. It then carries out spectrum sensing for the first channel. If the sensed channel is found empty, SU stops sensing the next channels. Otherwise, it senses the next channel in the sequence. Note that our studied model is general enough to accommodate any PHY channel sensing schemes. The performance of the channel sensing scheme is assumed to only affect the accuracy of the mis-detection and false alarm probabilities, P_{MD} and P_{FA}. If no channels are found empty, the SU picks up one channel and transmits in underlay mode. Otherwise, after one empty channel has been picked up, the SU starts transmitting on that channel for the remaining of the time-slots. In the next time-slot of the frame, the SU carries out in-band spectrum sensing and transmits packets if it is still free. Otherwise, either it starts the out-of-band sensing with probability P_{switch} or switches to underlay mode with probability $P_{o,u}$. In the next frame, the SU again starts the out-of-band or in-band channel sensing depending on its strategy and the availability status of the previously occupied channel, and then it repeats the transmission process. Upon receiving the data packet, the receiver sends an ACK/NAK feedback via automatic repeat request (ARQ) technique for the transmission. Let P_{fe} denote the probability of feedback error. However, in our proposed CLAP, the SU does not need to sense all the time-slots. Rather, it uses the already available ACK/NAK feedback information as discussed later. In the following, we discuss the problem formulation assuming that the channel is sensed in every time-slot. The modification for CLAP is discussed in respective section.

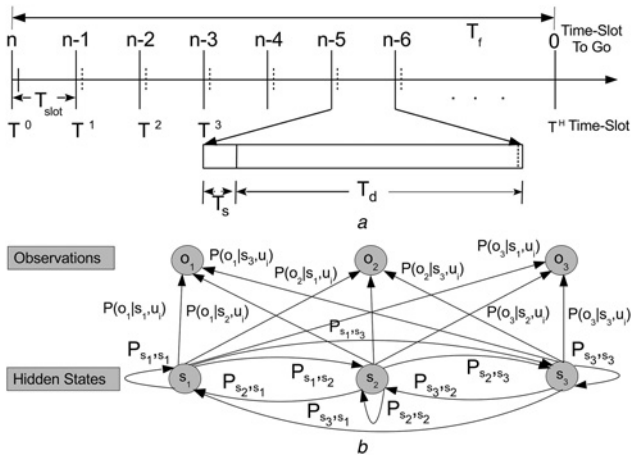

Fig. 1 *Time and state dynamics for the system*

a A pictorial view of a radio frame and the slot structure, where in the beginning part of a slot is for sensing and remaining part is for data transmission. The tth time-slot is denoted as T^t and the number slots to go in the frame is represented as n

b The state diagram and the corresponding observations for the POMDP problem

3 Problem formulation

In this paper, our goal is to intelligently access available PU channels via switching between channels and modes so that the achievable throughput in the horizon is maximised. We are concerned with the optimum utilisation of the channel and adaptation of power in order to maximise the throughput and avoid the collision occurrence in a CRN. It can be noted that the exact instantaneous state of the PU is unknown to the SU. However, the SU wants to adapt the transmission with respect to PU activity. Thus, the problem can be inherently formulated as a POMDP. A POMDP problem is composed of the following ingredients: a set of states $\mathcal{S}$, a set of actions $\mathcal{U}$, transition probability matrices $\mathcal{P}_s$, a set of observations $\mathcal{O}$, observation probability matrices $\mathcal{Q}$ and reward matrix $\mathcal{R}$ [17]. Below we discussed each component individually:

3.1 System states

Assume that $\mathcal{S} = \{s_1, s_2, s_3\}$ is the set of activity states of the PUs as seen by the SU, where s_1, s_2 and s_3 correspond to busy, concurrent and idle states, respectively.

The brief descriptions of the states are given below:

(1) *Busy state* s_1*:* In this state, we assume that the channel is occupied by the PU. Thus, it does not expect any kind of overlay channel access by the SU. Because when an SU tries to access the channel with full transmission power whereas PU is in state s_1, it will cause undesirable interference to PU.

(2) *Concurrent state* s_2*:* In this state, the SU can transmit simultaneously with the PU using underlay mode, possibly using lower transmission power than that in state s_3. There will be two possible scenarios where this state may be possible:

- When the receiver of the PU network has a specified interference tolerance limit and SU transmitter is able to transmit using lower power so that the interference threshold is not violated. We assume that the PU broadcasts its QoS requirement when transmitting its data information and SU knows its and PU transmission ranges [8].

doi: 10.1049/iet-com.2013.0796

• When the SU transmitter uses dirty paper coding (DPC) techniques to zero-force interference for its own receiver's signal and uses a part of its power to amplify and relay PU signal to compensate the interference it causes.

In any case, it is clear that the SU transmitter has to use lower transmission power in this state than the full transmission power possible.

(3) *Idle state s_3:* In this state, the channel is assumed to be completely free to be used by the SU and the SU can use overlay transmission mode. The SU can use any power actions depending on its number of packets in the buffer and channel gain between the SU transmitter and receiver pair. Although all actions are allowable in state s_3, higher power action is most preferable since SU can transmit with higher rate with this action.

3.2 Transmission power actions

During a radio frame, at each time-slot, the controller of the cognitive transmitter takes the decision whether or not to access the channel. If it decides to access a channel, it also decides which mode to use, namely, either overlay or underlay. Depending on the above decision, the controller also decides what power it will use to transmit data to the cognitive receiver. Let $\mathcal{U} = \{u_1, u_2, u_3\} = \{\text{zero power, low power, high power}\}$ be the set of power actions, where each action corresponds to a specific transmission rate for the corresponding power. Note that action u_1 corresponds to no transmission. The SU must use it in state s_1. Any actions can be chosen in state s_3. However, in state s_2, action u_3 is not permitted.

3.3 State transition matrix

The underlying states can transit from one to another over time. The transition probability matrix for a particular action captures this probability of switching from one state to another. For action $u \in \mathcal{U}$, it can be written as $\mathcal{P}_s(u) = \left[P_{s_i,s_j}(u)\right]_{S\times S}$. Note that, for the problem at hand, state transitions of the PU activity are independent of the SU's choice of action. Therefore, we can write, $P_{s_i,s_j}(u) = P_{s_i,s_j}, \forall u \in \mathcal{U}$. The SU can learn the system state transitions matrix by any statistical method.

3.4 State observation and its probability

The spectrum sensor at the SU transmitter senses the channel and takes decision on the hidden state of the system. The sensing outcomes on the hidden system state form the observation vector $\mathcal{O}_s = \{o_1, o_2, o_3\} = \{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$, where $o_i = \hat{s}_i$ is the spectrum sensor's outputs for corresponding hidden states $s_i \in \mathcal{S}$. The relationship between the spectrum sensor outputs and the hidden states is shown in Fig. 1*b*, where the probability of a state being in a particular state for a particular action is expressed in terms of observation probabilities $P(o_j|s_i, u_k)$, $s_i \in \mathcal{S}$, $o_j \in \mathcal{O}$, $u_k \in \mathcal{U}$.

3.5 Reward matrix

At the beginning of a time-slot, the system moves to the next state according to the state transition probability matrix of the hidden core process (PU activity process) and an observation of the state is received through spectrum sensor. Then, the controller at the SU transmitter takes a power action, based on the updated belief of the state. The system receives a reward (when no interference occurs) or incurs a cost (when interference occurs) depending on the action choice. The controller chooses the best action that is expected to give the highest reward. Let $R_{s_i}(u)$ represent the reward value in state s_i when action u is chosen. Please note that the reward values for each state should be assigned in such way so that the controller chooses the best action. We discuss more on the reward values in Section 6. Unless specified otherwise, we use superscript t, $t = 0, 1, \ldots, H$ to denote a variable at time-slot T^t throughout this paper. For example, we represent the state of the PU at time-slot T^t by s^t. On the other hand, subscript n is used to denote n time-slots to reach the end of the horizon as shown in Fig. 1. A list of commonly used symbols is given in *I*.

4 Solution techniques

In this section, we discuss three possible policies that can be used to obtain the actions in each time-slot and state. Although first policy gives us optimal policy for the problem, the second and the third are based on belief state. Note that a policy gives us the action to be taken in each state and in all the time-slots in a frame.

4.1 Fully observable optimal policy

First, let us consider the optimal policy when, in each time-slot, the states are fully observable to the SU transmitter. We termed this policy as FOOP. In state s_i, $1 = 1, 2, 3$ the optimal policy u_i, $1 = 1, 2, 3$ is applied. That is, in each time-slot, we assume that the scheduler is provided with the hidden states of the PUs' channels. This policy is certainly infeasible in many practical cases. However, it serves as a benchmark of performance of the two policies discussed below. Both policies use optimal alpha vectors of the formulated POMDP problem as given in Appendix 1.

Note that in the Appendix 1, the alpha vectors are determined using the transition probability, observation probability and rewards models for the problem. Both policies discussed below compute action for each time-slot of the SU communications. They update belief state vector [We call forward probability distribution over states as belief state vector for PLOP and CLAP. For POMDP formulation model and solution, we use information vector to distinguish between two.], which is defined as the updated probability distribution over all the states starting from an initial belief states, using sensor outcome and/or ACK/NAK information.

4.2 Physical layer optimal policy

In this policy, the spectrum sensor at the SU transmitter senses (either in-band or out-of-band as necessary) the channels in each time-slot. The steps for finding the instantaneous optimal policy in each slot are as follows: (1) in every time-slot, the scheduler obtains the observation and it updates the belief state vector using the forward algorithm. The initial belief vector for the first slot is initialised randomly, (2) the value of belief vector is plugged into the respective slot's alpha vector sets and (3) the alpha vector that maximises the value function for the

belief vector is the optimal alpha vector and the corresponding action is the optimal action.

4.3 Cross layer policy

It can be noted that in PLOP, the spectrum sensor at the SU transmitter needs to sense the channels in all the time-slots in order to update belief vector. There are two disadvantages of doing this: first, sensing in all the time-slots consumes bandwidth, and second it also spends the power for sensing process. Especially since the ACK/NAK information for the previous time-slot is already known at the link-layer, using the cross-layer interaction, the scheduler at the PHY can use those already available information to reduce the burden of sensing and eliminate two consequent disadvantages.

We propose a novel CLAP using the forward algorithm that estimates and maintains the probability distribution of the states, also called belief, from both the sensing observation result and the feedback ACK/NAK observations. We assume that the transmitter senses the channel in the first time-slot and thereafter it uses feedback observation during the rest of the slots to update the belief of the states. The belief of the states in each time-slot is then used to pick up the alpha vector that gives maximum rewards and associated action using (8).

5 Forward algorithm

We discuss the forward algorithm that deals with the updating and propagating the belief for both PLOP and CLOP in the following subsections.

5.1 Hidden state belief estimation

It can be noted that spectrum sensor is the only source of observations in PLOP. In CLAP, however, we have two sources from where we can obtain observations: spectrum sensor and ACK/NAK feedback. Let $O_{1:m} := o_1, o_2, \ldots, o_m$ denotes a given sequence of observations (either from sensor or from feedback). Therefore, we can write the probability of state s_i given the observations as

$$f_{o:m}(s_i) = P(s^m = s_i | o_1, o_2, \ldots, o_m, \pi), \quad \forall s_i \in \mathcal{S} \tag{1}$$

where π is the belief of the states at time-slot T^0. When we know the current belief and obtain the new observation, we can iteratively update the belief of the hidden states by using the following relation

$$\boldsymbol{f}_{o:m} = O_d \mathcal{P}_s \boldsymbol{f}_{o:m-1} \tag{2}$$

where $\boldsymbol{f}_{0:m}$ is the 3-by-1 column vector containing probability of state $s_i \in \mathcal{S}$ given the observations, and $\boldsymbol{O}_\mathrm{d}$ is the 3-by-3 diagonal observation matrix whose diagonal elements $\boldsymbol{O}_\mathrm{d}[i,i]$ are the observation probabilities for states $s_i \in \mathcal{S}$ and other elements are zeros. After normalising the probabilities so that the sum is equal to 1, we can write the normalised belief vectors as follows

$$\hat{\boldsymbol{f}}_{o:m} = \beta \boldsymbol{O}_\mathrm{d} \mathcal{P}_s \hat{\boldsymbol{f}}_{o:m-1} \tag{3}$$

where $\beta = \left(1/\left(\sum \boldsymbol{O}_\mathrm{d} \mathcal{P}_s \boldsymbol{f}_{o:m-1}\right)\right)$ is the normalising factor. Below we discuss the update of belief using both the sensor and ACK/NAK observations.

5.2 Belief update using sensor observation

In order to utilise unused spectrum opportunistically and at the same time avoid interference to the returning PUs, spectrum sensing is done periodically. Among the different spectrum sensing methods, energy detection is the most popular because of its simple design and smaller sensing time [4]. Without loss of generality, we continue our discussion using energy detection technique, where the collected energy after sampling and frequency domain operations forms the test statistics, $X(y)$ of the energy detector's received signal y. The probability density function (pdf), $p(y)$ of the test statistic can be approximated by a Gaussian distribution [7]. The SU transmitter senses the channel energy and compares with two different thresholds ζ_1 and ζ_2 to determine the state of the PU [8]. The probability of false alarm $P(\hat{s}_1|s_3)$, which is the probability of the sensor falsely declaring the presence of primary signal in lower state when it is actually in upper state, can be written as $P(\hat{s}_1|s_3) = \Pr(X(y) > \zeta_1 | s_3)$. The probability of mis-detection $P(\hat{s}_3|s_1)$, which is the probability of the sensor incorrectly declaring the presence of primary signal in state that is higher than actual state, can be written as $P(\hat{s}_3|s_1) = \Pr(X(y) < \zeta_2 | s_1)$. Similarly, other false alarm probabilities $P(\hat{s}_2|s_3)$ and $P(\hat{s}_1|s_2)$, and mis-detection probabilities $P(\hat{s}_2|s_1)$ and $P(\hat{s}_3|s_2)$ can be evaluated using appropriate thresholds and pdf. The diagonal observation matrix $\boldsymbol{O}_\mathrm{d}$, for observation $\hat{s}_1$ in (2), can be written as $\boldsymbol{O}_\mathrm{d} = \mathrm{diag}(P(o_1/s_1), P(o_1/s_2), P(o_1/s_3))$.

5.3 Belief update using ACK/NAK observation

In a particular time-slot, in state s^t when the controller takes a particular action u^t, the secondary transmitter obtains an observation using the feedback from the receiver. When the receiver receives the packet sent from the transmitter successfully, it sends an ACK. Therefore, when an ACK is received, the transmitter assumes that no collision occurred with PU's transmission. The absence of ACK is interpreted as collision with the PU's transmission.

Hence, the set of feedback observations can be written as $\mathcal{O}_f = \{q_1, q_2\} = \{\text{No Collision, Collision}\}$. Although the actual states of the PU's activity are hidden, the SU obtains an idea of them using the previous observations. The relationship between the actual state and the observation for each action can be expressed in terms of 3×2 feedback observation matrix and can be expressed as $\mathcal{Q} = [q_{ik}]$, where $q_{ik} = P(q^t = q_k | s^t = s_i, u^t)$, $q_k \in \mathcal{O}_f$, $s_i \in \mathcal{S}$. The diagonal observation matrix $\boldsymbol{O}_\mathrm{d}$ can be written for ACK (no collision) feedback as follows: $\boldsymbol{O}_\mathrm{d} = \mathrm{diag}(q_{11}, q_{21}, q_{31})$.

6 Simulation results

In this section, we present simulation results for the PLOP and CLAP using Monte Carlo technique averaged over 10^6 time-slots. As discussed before, the belief of the states is updated using sensing information in each time-slot for PLOP. For the CLAP case, it is updated using channel sensing information in the first time-slot and using previous ACK/NAK information in the other time-slots. We compare both policies with the FOOP. The throughput results for FOOP are obtained assuming that states are completely known and the action that gives maximum throughput without any collision are chosen. For both PLOP and CLAP cases, the action for a particular time-slot is obtained by substituting the respective updated belief in (8). Note

doi: 10.1049/iet-com.2013.0796

that the alpha vectors are obtained using incremental pruning algorithm as described in Appendix 1. The algorithms for FOOP, PLOP and CLAP are given in Appendix 2. For our problem in hand, we found that the computation complexity is very minimal. The computations of the alpha vectors are very fast. The computation of the actions for FOOP, PLOP and CLAP are very fast as well.

We assume that spectrum sensing requires 10% (approximately 1 ms [4]) of time in a slot. The duration of each slot is $T_{slot} = 10$ ms [18]. The horizon length (frame size), H is varied from 2 to 20. The number of radio frames for each scenario is found according to the frame size, H. Although any activity statistics of PU are valid for the problem, without loss of generality, we consider three cases where the self-transition probabilities, $P_{\text{st}} = P_{s_i,s_i}, s_i \in \mathcal{S}$ are 0.99, 0.97 and 0.95. When in state s_1 and s_3, the adjacent transition probability, $P_{s_i,s_j}, |i-j| = 1$ is $0.8(1 - P_{s_i,s_i})$ and other transition probability is $0.2(1 - P_{s_i,s_i})$. When in state s_2, both adjacent states are equally probable, that is, $P_{s_i,s_j} = 0.5(1 - P_{s_j,s_i}),\ |i-j| = 1$.

We use the following rewards so that the occurrence of collision among transmission of PU and SU can be avoided: $R_{s1}(u_1) = 0,\ R_{s_2}(u_2) = 1,\ R_{s_3}(u_2) = 1,\ R_{s_3}(u_3) = 2$, and others are -1. We use positive reward and negative reward for an action when it is expected and is not expected in a state, respectively.

That is, the negative value discourages the scheduler for taking actions when a higher positive reward exists. For example, when in state s_1, action u_2 and u_3 are not permitted as both of these two actions will introduce interference to the PU. Therefore, the rewards corresponding to these actions in state s_1 are negative. The reward for action u_1 is zero, which is better than negative reward, so the scheduler chooses u_1 over u_2 and u_3. In state s_2, our expected action is u_2 because u_3 will introduce interference and u_1 will miss spectrum utilisation opportunity. Likewise, in state s_3, expected action is u_3, but u_2 is permitted (but discouraged) since it does not either introduce interference or miss spectrum opportunity. The reward is lower because the throughput obtained using u_2 is less than that using u_3. Action u_1 is not expected in state s_3 as discussed above. Note that the reward matrices are not the same as throughput matrices. We assume that we receive throughput of 1 and 2 for actions u_2 and u_3, respectively, when no collision happened. In state $s_i \in \mathcal{S}$ collision occurs for action $u_j \in \mathcal{U}$ when $i < j$. We use 1 and 0 to express collision and successful events to find average collision per-slot. We have found that the policy does not depend on the initial belief and therefore, it is initialised to steady-state vector. It is worth to mention that when short-term fading is considered, the throughput rate can take many values as determined by the employed modulation and coding schemes.

Unless specified otherwise, we use following: number of PU channels, $N = 10$ and self-transition probabilities, $P_{\text{st}} = 0.99$. We show the throughput and the collision performances as a function of different horizon lengths for different number of PU channels in Figs. 2–6. In all these figures, the solid, dashed and dotted curves are for number

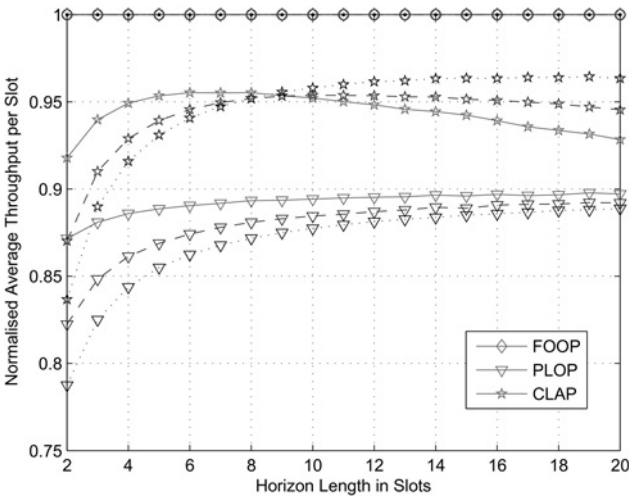

Fig. 3 *Throughput against horizon size when sensing error is zero, but feedback error is non-zero*

The solid, dashed and dotted curves are for number of PU channels, $N = 2$, $N = 5$ and $N = 10$, respectively

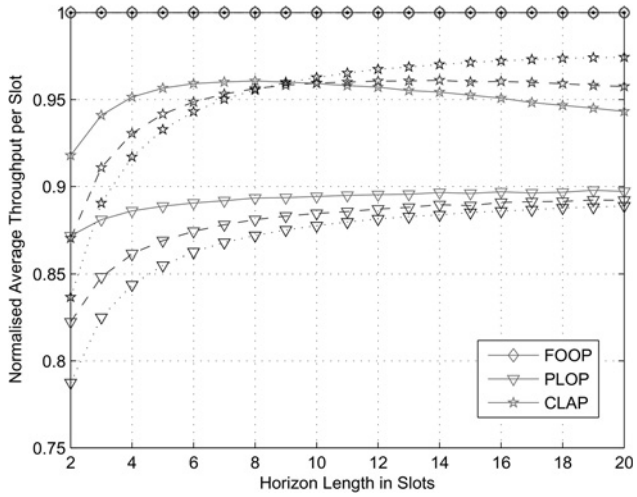

Fig. 2 *Throughput against horizon size when sensing/feedback error is zero*

The solid, dashed and dotted curves are for number of PU channels, $N = 2$, $N = 5$ and $N = 10$, respectively

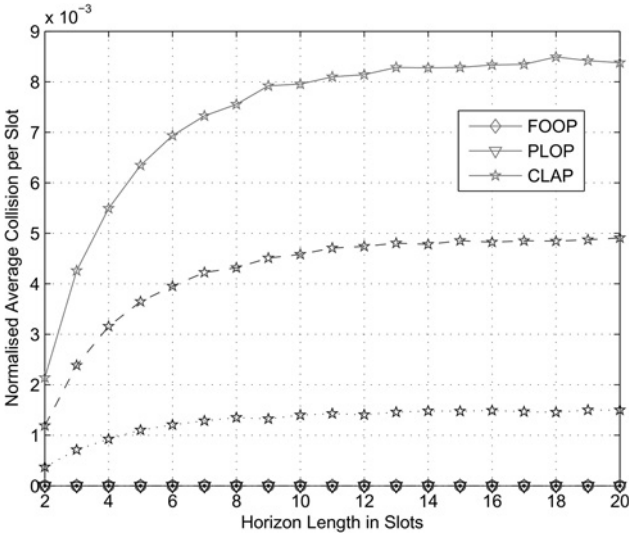

Fig. 4 *Collision against horizon size when sensing error is zero, but feedback error is non-zero*

The solid, dashed and dotted curves are for number of PU channels, $N = 2$, $N = 5$ and $N = 10$, respectively

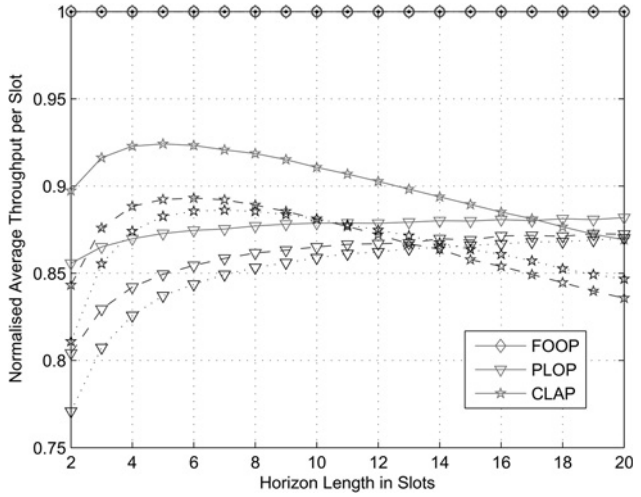

Fig. 5 *Throughput against horizon size for non-zero sensing and zero feedback error*

The solid, dashed and dotted curves are for number of PU channels, $N = 2$, $N = 5$ and $N = 10$, respectively

of PU channels, $N = 2$, $N = 5$ and $N = 10$, respectively. In Fig. 2, we first show the normalised throughput per slot for FOOP, PLOP and CLAP, where it is assumed that the sensing and feedback errors are zero. Note that maximum possible normalised throughput is 1, which we obtain for FOOP for any numbers of PUs channels and any horizon lengths.

For PLOP, the throughput increases as the horizon length increases. Within a radio frame, the belief is updated in each time-slot with the new sensing observation. More observation data translates into better belief. Therefore, the average throughput for PLOP increases as the number of time-slots in a frame (which is the horizon) increases. However, for CLAP, the average throughput is determined by the two factors. For this reason, there is a break over point for CLAP plots. It can be seen that the throughput first increases and then decreases. Since each curve has a break over point, the horizon length can be intelligently chosen depending upon the number of PU channels. First since there is no sensing (and no corresponding time-slot loss) after first slot, the average throughput should increase when the number of time-slot in a frame increases. Owing to the effect of throughput loss in the first slot is averaged out over the horizon length and becomes smaller for increasingly larger horizon lengths. There is another factor is the accuracy of the belief. Although the sensing observation is very accurate indication about the channel state, ACK/NAK is not that accurate (since ACK/NAK are outdated information). Moreover, the belief is more accurate as the horizon length increases in this case (no error in sensing and feedback). Thus, the accuracy of the belief degrades for larger horizon lengths for CLAP.

When the number of PU channels N increases, the throughput increases for larger horizon lengths. However, it is opposite for lower horizon length. Here two opposing factors are working as well. First, the throughput increases with the number of PUs. As for larger N, the SU transmitter has more probability of getting free channels (rather than staying with one channel). However, when the number of PUs are larger, it takes more time-slots to obtain refined belief (since the states are initialised to equally probably at the beginning). That is why the throughput initially decreases for the smaller horizon length and then gradually improved as the horizon length increases. When the feedback errors are 5%, the throughput is not changed for PLOP in Fig. 3 because PLOP depends on sensing observation. However, it is decreased at higher horizon length for CLAP. Since the accuracy of belief for CLAP depends on feedback observation. Therefore, the accuracy of ACK/NAK would drop when the accuracy of feedback degrades. For the above reason, the collision is still zero for PLOP (since sensing error is zero), but it is non-zero for CLAP (since feedback error is non-zero and it is less accurate indication of the hidden state). However, it is less than 1% as can be seen from Fig. 4.

Now let us see the cases when the sensing error is 5% and feedback error is zero in Figs. 5 and 6. It can be noticed that both the PLOP and CLAP throughput and collision performances are affected by sensing error. However, in all the above cases, it can be noticed that the CLAP attains better performance, especially at lower horizon lengths. The

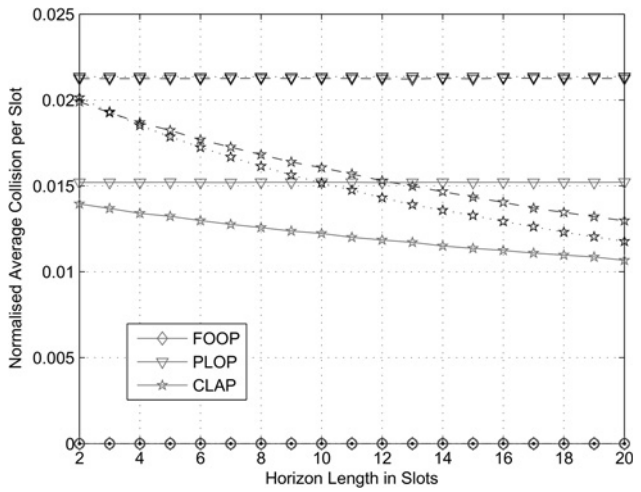

Fig. 6 *Collision against horizon size for non-zero sensing and zero feedback error*

The solid, dashed and dotted curves are for number of PU channels, $N = 2$, $N = 5$ and $N = 10$, respectively

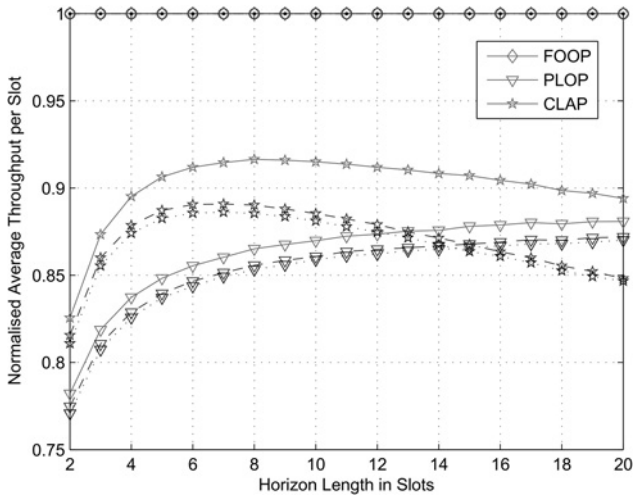

Fig. 7 *Throughput against horizon size for non-zero sensing and zero feedback error*

The solid, dashed and dotted curves are for self-transition probabilities, $P_{st} = 0.95$, $P_{st} = 0.97$ and $P_{st} = 0.99$, respectively

doi: 10.1049/iet-com.2013.0796

effect of different self-transition probabilities is shown in Fig. 7. It can be seen that the throughput performance is better for lower value of P_{st}. This is because when self-transition probability increases the channel is going to stay in the same state for longer time. When it is in the state s_1, it is stuck there for longer time and less chance to switch in better state in another channel that permit transmission with higher rate. As a result, the average throughput decreases when the value of P_{st} increases. It is seen that at a certain value of horizon length the CLAP curves cross PLOP curves in the downward direction. The crossing value of the horizon length increases as the state memory decreases (i.e. when the state mixing rate increases.). That is, CLAP maintains better performance in the longer horizon length region than PLOP for the faster channel mixing.

7 Conclusion

In this paper, we investigated the throughput and the collision performances of a hybrid overlay–underlay CRN. It was assumed that the PU has three hidden states and multiple PU channels are available. We provided POMDP-based formulation and incremental-pruning algorithm based solution technique to find the optimal alpha vectors needed to compute policy for a particular belief vector. For a particular belief vector in a particular time-slot, the power adaptation actions and overlay–underlay transmission mode are found by plugging the belief vector in the alpha vector sets and by finding maximum value of the scalar products of them. We studied two policies: in order to track the belief, whereas first technique (termed as PLOP) requires channel to be sensed in each time-slot, second technique (termed as CLAP) requires channel sensing in the first time-slot only. In the other time-slots, ACK/NAK feedback from previous slot is used in the latter technique. We evaluated the throughput and the collision performances of both policies and compared them with FOOP. Simulation results showed that the proposed CLAP is more throughput efficient than the PLOP, especially when the frame size is smaller. The collision rate for both policies are also well below the specified threshold by the standards.

8 Acknowledgment

This work is funded by Ryerson University RPDF and NSERC grants. The material of this paper has been presented in part at the IEEE GLOBECOM 2012 conference.

9 References

1 Amanna, A., Reed, J.: 'Survey of cognitive radio architectures'. Proc. IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp. 292–297
2 Ye, Z., Feng, Q., Shen, Y.: 'Scheme for opportunistic spectrum access in cognitive radio', *IET Commun.* 2013, **7**, (11), pp. 1061–1069
3 Zhao, Q., Geirhofer, S., Tong, L., Sadler, B.: 'Opportunistic spectrum access via periodic channel sensing', *IEEE Trans. Signal Process.*, 2008, **56**, (2), pp. 785–796
4 Kim, H., Shin, K.G.: 'In-band spectrum sensing in IEEE 802.22 WRANs for incumbent protection', *IEEE Trans. Mob. Comput.*, 2010, **9**, (2), pp. 1766–1779
5 Zhao, Q., Tong, L., Swami, A., Chen, Y.: 'Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: a POMDP framework', *IEEE J. Sel. Areas Commun.*, 2007, **25**, (3), pp. 589–600
6 Zhang, T., Tsang, D.: 'Optimal cooperative sensing scheduling for energy-efficient cognitive radio networks'. Proc. IEEE INFOCOM'2011, Shanghai, China, June 2011, pp. 2723–2731
7 Liang, Y.-C., Zneg, Y., Peh, E.C., Hoang, A.T.: 'Sensing-throughput tradeoff for cognitive radio networks', *IEEE Trans. Wirel. Commun.*, 2008, **7**, (4), pp. 1362–1337
8 Senthuran, S., Anpalagan, A., Das, O.: 'Throughput analysis of opportunistic access strategies in hybrid underlay–overlay cognitive radio networks', *IEEE Trans. Wirel. Commun.*, 2012, **11**, pp. 2024–2035
9 Goldsmith, A., Jafar, S.A., Maric, I., Srinivasa, S.: 'Breaking spectrum gridlock with cognitive radios: an information theoretic perspective', *Proc. IEEE*, 2009, **97**, (5), pp. 894–914
10 Khoshkholgh, M.G., Navaie, K., Yanikomeroglu, H.: 'Access strategies for spectrum sharing in fading environment: overlay, underlay, and mixed', *IEEE Trans. Mob. Comput.*, 2010, **9**, (12), pp. 1780–1793
11 Jiang, X., Wong, K.-K., Zhang, Y., Edwards, D.: 'On hybrid overlay–underlay dynamic spectrum access: double-threshold energy detection and Markov model', *IEEE Trans. Veh. Technol.*, 2013, **62**, (99), pp. 1–6
12 Bansal, G., Duval, O., Gagnon, F.: 'Joint overlay and underlay power allocation scheme for ofdm-based cognitive radio systems'. 2010 IEEE 71st Vehicular Technology Conf. (VTC 2010-Spring), 2010, pp. 1–5
13 Oh, J., Choi, W.: 'A hybrid cognitive radio system: a combination of underlay and overlay approaches'. 2010 IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), 2010, pp. 1–5
14 Shashika Manosha, K., Rajatheva, N., Latva-aho, M.: 'Overlay/underlay spectrum sharing for multi-operator environment in cognitive radio networks'. 2011 IEEE 73rd Vehicular Technology Conf. (VTC Spring), 2011, pp. 1–5
15 Nair, S., Schellenberg, S., Seitz, J., Chatterjee, M.: 'Hybrid spectrum sharing in dynamic spectrum access networks'. 2013 Int. Conf. Information Networking (ICOIN), 2013, pp. 324–329
16 Karmokar, A., Senthuran, S., Anpalagan, A.: 'POMDP-based cross-layer power adaptation techniques in cognitive radio networks'. 2012 IEEE Global Communications Conf. (GLOBECOM), 2012, pp. 1380–1385
17 Littman, M.L.: 'A tutorial on partially observable Markov decision processes', *Elsevier J. Math. Psychol.*, 2009, **53**, pp. 119–125
18 Hoang, A.T., Liang, Y.-C., Zeng, Y.: 'Adaptive joint scheduling of spectrum sensing and data transmission in cognitive radio networks', *IEEE Trans. Commun.*, 2010, **58**, (1), pp. 235–246
19 Smallwood, R.D., Sondik, E.J.: 'The optimal control of partially observable Markov processes over a finite horizon', *Oper. Res.*, 1973, **21**, (5), pp. 1071–1088
20 Cassandra, A.R., Littman, M.L., Zhang, N.L.: 'Incremental pruning: a simple, fast, exact method for partially observable Markov decision processes'. Proc. Uncertainty in Artificial Intelligence'97, Providence, RI, USA, July 1997, pp. 54–61

10 Appendix

10.1 Appendix 1: optimal solution techniques

The finite-horizon POMDP problem formulated in Section 3 can be solved using different techniques, such as incremental pruning algorithm. Incremental pruning algorithm gives us optimal alpha vectors, which forms the basis of the optimal policy in a POMDP problem. We use the optimal alpha vectors in both PLOP and CLAP to obtain the action for a given belief state.

10.1.1 Information state: Information state of hidden state $s \in \mathcal{S}$ can be tracked and updated with the new observation $o \in \mathcal{O}$ in each time-slot and it provides sufficient statistics for POMDP problem. It is the probability associated with hidden state s and denoted by $I(s)$. $I(s)$ is a continuous positive real number and bounded by 0 and 1. Thus an information state vector I is just a probability distribution over the set of states $\mathcal{S}$. The information state at time-slot T^t can be calculated when the sequence of observations and actions so far are known using relation

$$I(s^t) = P\left(s^t | u^{t-1}, o^{t-1}, \ldots, u^0, o^0, s^0\right) \tag{4}$$

where o^0 and u^0 are the observation and action at the starting time-slot T^0. Please note that this conditional probability (4) is

essentially a filtering task. Thus, the new information state can be calculated recursively from the previous information state and the new observation-action pair. As we mentioned before, at any time-slot, the information vector is a sufficient statistic of the past sequence of observation and actions.

Let $I(s^t)$ denotes the previous information state at time-slot T^t, and the agent perceives observation o^t. Now, if it takes action u^t, then the new information state at time-slot T^{t+1} can be written as

$$I(s^{t+1}) = \alpha P(o^t|s^{t+1}, u^t) \sum_{s^t} P_{s^t,s^{t+1}}(u^t) I(s^t) \tag{5}$$

where the factor α is a normalising constant that makes the information state sum equal to 1. We can write it as $\alpha = \left(\sum_{s^{t+1}} P(o^t|s^{t+1}, u^t) \sum_{s^t} P_{s^t,s^{t+1}}(u^t) I(s^t)\right)^{-1}$. Then the information state transformation function $T(I/u,\ o) = I(s^{t+1})$ becomes

$$I(s^{t+1}) = \frac{P(o^t|s^{t+1}, u^t) \sum_{s^t} P_{s^t,s^{t+1}}(u^t) I(s^t)}{\sum_{s^{t+1}} P(o^t|s^{t+1}, u^t) \sum_{s^t} P_{s^t,s^{t+1}}(u^t) I(s^t)} \tag{6}$$

Therefore, transformation function $T(I/u,\ o)$ updates the information state $I(s^t)$, $\forall s^t \in \mathcal{S}$ at time-slot T^t to the information state $I(s^{t+1})$, $\forall s^{t+1} \in \mathcal{S}$ at time-slot T^{t+1}. Once the observation probabilities are known, the term $P(o^t|s^{t+1}, u^t)$, $\forall s^{t+1} \in \mathcal{S}$ can be found by $P(o^t|s^{t+1}, u^t) = \sum_{s^t \in \mathcal{S}} P(o^t|s^t, u^t) P_{s^t,s^{t+1}}(u^t)$. We can note that the observation probability of a state depends only on the underlying hidden state and it is independent of the action. Therefore, we can again write $P(o^t|s_i, u^t) = P(o^t|s_i)$, $\forall o^t \in \mathcal{O},\ s_i \in \mathcal{S}$.

Initialisation:

Number of PU channels N, Number of Simulation Slots $N_{Sim} = 10^6$, Frame size H

Generate N_{Sim} actual states and sensed states for all the channels randomly

Compute number of frames, $N_F = \frac{N_{Sim}}{H}$

for Number of Frames $i := 1$ to N_F **do**

  for slots $j := 1$ to H **do**

    while Sense channels sequentially until a channel is accessed or all the channels are sensed **do**

      if Sensed State $= s_1$ **then**

        Sense next channel

      else if Sensed State $= s_3$ **then**

        Access channel in overlay mode

      else

        Sense next channel with probability P_{switch}; otherwise, access channel in underlay mode.

      end if

    end while

    Calculate the action as follows:

    For FOOP: reveal actual state and find corresponding action

    For PLOP: update belief using sensing observation and find action by plugging the belief in optimal alpha vectors

    For CLAP: update belief using sensing observation in first time-slot and ACK/NAK observation in subsequent time-slot, and find action as PLOP.

    Compute throughput and collision from the action for each policy

  end for

end for

Fig. 8 *Algorithm to compute actions for FOOP, PLOP and CLAP*

doi: 10.1049/iet-com.2013.0796

10.1.2 Value function: The value function is an important function in POMDP problem that maps the information state to expected discounted total reward. Assume that $V_n(I)$ is the maximum expected reward that the system can accrue during the lifetime of the process when the current information vector is $I_{1\times 3}$ and there are n slots remaining before the process terminates. The recursive equation for the value function can be written as

$$V_n(I) = \max_{u\in\mathcal{U}}\left[I.\mathcal{R}(u) + \gamma\sum_{o\in\mathcal{O}} P(o|I,u)V_{n-1}[T(I|u,o)]\right] \quad (7)$$

where $0 < \gamma \leq 1$ is a discount factor for the expected reward, $\mathcal{R}(u)$ is the reward column vector of dimension 3-by-1 for action u, and $P(o|I,u) = \sum_{s_i=s_1}^{s_3} I(s_i)\sum_{s_j=s_1}^{s_3} P_{s_i,s_j}(u)P(o|s_j,u)$ is the probability of observing output o if the current information vector is I and action u is taken. Therefore, the value for an information state I is the value of the best action that can be taken from I of the expected immediate reward for that action plus the expected discounted value of the resulting information state. It is shown that the value function $V_n(I)$ is piecewise, linear and convex [19] and can be written as

$$V_n(I) = \max_k\left[\sum_{i=1}^{i=3}\alpha_i^k(n)I(s_i)\right] \quad (8)$$

for some set of vectors, called α-vectors, $\alpha^k(n) = \left[\alpha_1^k(n), \alpha_2^k(n), \alpha_3^k(n)\right]$, $k = 1, 2, \ldots$. The exact numbers of α-vectors depend on the numbers of action-observation pairs. We can see from the above relation (8) that once the alpha vectors for various time-slots in the horizon are calculated, the policy that maximises $V_n(I)$ for a given information vector is the optimal policy.

10.1.3 Algorithm for calculating alpha vectors: There are several algorithms in the literature to find the optimal policy for the formulated finite-horizon discounted reward POMDP problem. Some of the algorithms are as follows: enumeration, witness, incremental pruning and so on. In all the algorithms, the algorithm calculates the best policy for a given information vector in each time-slot during the finite horizon. In [20], Cassandra *et al.* have discussed different techniques for calculating the alpha vectors for (8). Their relative merits and limitations are also presented there. Incremental pruning algorithm is shown to perform the best in [20]. As such, although any algorithm can be used for the considered problem, we use incremental pruning algorithm in this paper to calculate the alpha vectors over the horizon. The basic idea behind this algorithm is to form transformed value function sets for a particular action and all observations, and then combining all choices incrementally in observation by observation manners (e.g. value functions for first and second observations are combined first, then the results are combined with third, and so on) to find dominant sets. The process is repeated for the other actions and then the alpha vector sets are obtained by combining and eliminating dominated sets.

10.2 Appendix 2: algorithm to compute actions for FOOP, PLOP and CLAP

Below we present the algorithm used to compute different performance parameters. In each case, we assume that the total number of simulations slots is the same and the same sets of actual states are used in all the simulations. We also assume that the optimal alpha vectors are computed using incremental pruning algorithm. We repeat the simulations for different frame sizes, different number of PU channels, and different sensing and feedback errors (see Fig. 8):