Devanarayana and Alfa *EURASIP Journal on Wireless Communications and Networking* (2015) 2015:88
DOI 10.1186/s13638-015-0309-2

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Proactive channel access in cognitive radio networks using statistical radio environment maps

Chamara N Devanarayana[1*] and Attahiru S Alfa[2]

Abstract

Inclusion of statistical knowledge of the primary user (PU) channel usage had shown to be beneficial in dynamic spectrum access. Motivated by this fact, this paper investigated the importance of collecting and using statistics on neighboring secondary users (SUs) in selecting channels in addition to the knowledge of PU channel usage. The paper assumed that PU traffic characteristics of the channels are included in the radio environment map in the form of probabilistic suffix trees, which is a sequence predictor based on Markov property. In the proposed method, an intelligent sequence hopping-based common control channel and a carrier sense multiple access (CSMA)/collision avoidance (CA)-based medium access control (MAC) protocol were introduced. As shown in the paper, selecting channels using statistics of both the neighboring SUs and PUs reduced the number of packet collisions compared to a scheme which only uses PU statistics. Furthermore, the simulation results showed that the scheme proposed had better throughput performance with respect to both the random channel selection scheme and the scheme which only uses PU statistics while having less training complexity.

Keywords: Cognitive radio; Prediction methods; Predictive control; Artificial intelligence

1 Introduction

The vast growth in wireless applications in the past decade has pushed the FCC to search for efficient spectrum allocation mechanisms to avoid spectrum scarcity [1]. The measurement operation carried out by the FCC's Enforcement Bureau in 2002 gave some insight into the real cause of the widely accepted notion of spectrum scarcity [2]. The recommendations made by the spectrum efficiency working group, based on these findings, consisted of promoting flexible use of spectrum development and deployment of advanced technologies and promoting secondary markets for spectrum [2]. A technology known as cognitive radio (CR) was found to have the capability to implement those recommendations [1].

However, the secondary usage of spectrum requires the CRs to opportunistically utilize the spatiotemporally available primary user (PU) spectrum. This utilization brings about several problems as follows:

1. The problem of imperfect sensing (false alarms and missed detection) in the physical layer.
2. Unavailability of a common control channel which is always available.
3. Deciding on which channels to sense to find an opportunity efficiently.
4. Efficient selection of a different channel to continue with the communication when the PU appears in the currently used one (spectrum mobility).
5. Efficient sharing of the spectrum with other secondary users (SUs).

Proactive spectrum access (PSA) can be used as a solution for the third and fourth problems mentioned above. Furthermore, using the statistical information on the PU channel usage, highly available sequence hopping (SH)-based common control channels (CCCs) can be designed. One of the solutions to the problem of imperfect sensing is cooperative sensing. In [3], the authors discuss a semi-distributed cooperative spectrum sensing scheme which

*Correspondence: umdevana@myumanitoba.ca
[1] Department of Electrical and Computer Engineering, University of Manitoba, E1-551 EITC, 75A Chancellors Cir., R3T 5V6 Winnipeg, MB, Canada
Full list of author information is available at the end of the article

reduces this problem. In Sections 1.3 and 1.2, we discuss some available solutions for the problems 2 and 5, respectively. This paper is an extension to the work in [4]. In that paper, we only discussed the usage of probabilistic suffix tree (PST) as a prediction mechanism, where there was only a single SU trying to access the channel.

In the scheme presented, the radio environment map (REM) is enhanced with statistical knowledge of the PU channel usage as in [5]. Furthermore, due to PU channels being highly dynamic, it may not be possible to query the REM for information on the current state of the channels. Even if the channels are not highly dynamic, it might not be possible to query the REM before every time slot due to the control channel saturation problem. Considering these facts, in the scheme presented, each transmitting SU queries the REM once for the probabilistic information before the transmission. Then, the SU will cache the probabilistic information to be used until the end of the communication between itself and the receiver while carrying out periodic sensing. In the work, we presented in this paper, the REM learned the PU channel usage distributions from the past channel usage data using a scheme called the PST algorithm [6], which belongs to the class of variable-order Markov models (VMMs). PSTs only increased the length of the conditioning strings if there was evidence that higher order gives a better prediction result with high probability, although we set a maximum value for the memory in the algorithm [7]. Therefore, the model takes less memory and does not need much bandwidth to transmit. In [8], it was observed that in some channels, the predictability cannot be improved by increasing the Markov order and the largest improvement for other channels happens when going from zeroth order to first order. But the authors admitted that the predictability did not monotonically decrease with the Markov order. In their results, there were evidence for the existence of channels which are more predictable using higher Markov order, for example, 10% of the channels they tested had an optimal Markov order of 8. Therefore, the PST scheme is a good candidate when a CR operates in diverse wireless channels, because of its self-regulation of the Markov order and the convenience of training. Furthermore, the PST is capable of predicting the channel availability conditioned on a string of past channel observations which are of variable length. The second property mentioned in the above line was very useful in our case since the length of consecutive observations the SUs possessed was varying according to the sensing schedule. In a scenario where a carrier sense multiple access (CSMA)/collision avoidance (CA)-based medium access control (MAC) scheme is used, collisions are inevitable. Therefore, time slots become wasted. To alleviate the network from this issue, we used the statistics collected by each SU on the number of successes and failures on each used channel in a given window of time slots when selecting a set of channels to transmit data. We believe that this is the first time SU channel access data are used in PSA. In this scheme, the channel conditions were not considered. It was ignored to keep the model simple and to clearly show the advantage of using available statistics on PUs and SUs. The channel conditions can be obtained by channel probing after the handshake or from the REM itself, and it will enhance the capacity obtained without doubt.

1.1 Existing schemes on PSA

In the literature, there are mainly three categories of PSA schemes as follows: analytic formulations ([9-11]), heuristic schemes based on assumed channel usage ([12-15]), and heuristic schemes based on channel usage learning ([4,16-18]). Most of the learning methods like neural networks (e.g.,[16]) and binary time series models (e.g.,[18]) in the literature fixed the value of the memory to be used in prediction. The hidden Markov models (HMMs) (e.g.,[17]) fixed the number of states in the Markov chain. Although being very good predictors, the training complexities of neural networks, HMMs, and binary time series models are a lot higher than those of the PST, and thus, they cannot be trained online. The analytic formulations (e.g.,[9-11]) used variants of Markov decision process to calculate the sensing and access schedule to maximize the SU reward while keeping the PU interference probability under a given threshold. Although they can calculate the optimum policy, they become impractical even for a small number of channels because of the complexity of policy calculation. The heuristic schemes proposed in the literature although being simple can lead to more SU packet collisions, because all the SUs use the same schedule only varied because of imperfect sensing. Thus, they lead to inefficient channel usage in a CSMA/CA-based protocol.

1.2 Existing medium access control schemes

In [19], the MAC schemes in the literature were categorized according to the spectrum access mode as contention-based, time-slotted, and hybrid. Our scheme falls in the category of hybrid spectrum access protocols. Furthermore, in [20], they were categorized according to the level of coordination performed as direct access-based (DAB) protocols and dynamic spectrum allocation (DSA)-based MAC protocols. In DAB protocols, only the user pair involved in the communication take part in coordinating the access. While in DSA protocols, users coordinate with the neighbors to achieve a network-wide optimum. Although being optimal, DSA protocols have scalability issues. The MAC scheme that we proposed is a DAB with proactive access. The difference in our scheme from these MAC schemes is that being proactive and switching channels inside a time slot when the channel is not available. Interested readers are referred to the survey papers [19] and [20] and the references therein for more information.

1.3 Existing common control channel design schemes

The CCCs proposed for overlay networks can be divided into three main categories [21] as follows:

1. Sequence hopping-based CCC
2. Group-based CCC [22]
3. Dedicated CCC

We used a sequence-based CCC for the initial handshake between two SUs and an SU and REM. The advantage of SH-based CCCs over the group-based ones is having less coordination burden in a highly dynamic PU spectrum access scenario. Then, compared to the dedicated CCCs, it has less probability of being saturated. Our scheme is different from the ones in the literature (e.g.,[23-25]) because of using PU stationary channel usage probabilities in deciding on the channel to be used. The scheme proposed in [25] also uses PU stationary channel usage probabilities to rank the channels. In that paper, each individual SU made a ranking table of PU channels according to their activity. Then, using a linear and a parabolic function, the number of occurrences of each channel in a sequence of fixed length L was decided such that channels with lower activity had more appearances. Because of the deterministic nature of the number of occurrences of the channels, the arrangement of channels in the sequence by each SU plays a huge role on the channel load, which was not discussed in detail in that paper.

One of the drawbacks of the SH-based schemes is the burden of synchronization. Many papers considering decentralized scenario in the literature took the assumption of the network being synchronized (e.g., [23-26]). In [23], although they claimed to have alleviated the network from the need for synchronicity, they assumed the switching boundaries to be synchronous. In other words, the delay between the two sequences is an integer multiple of the time slot length. Our scheme could also handle that form of asynchronicity. For more information on how this synchronicity is maintained, the readers are referred to Section 4.4.

1.4 Main contribution

The main objective of this paper was to find solutions to the MAC layer problems faced by SUs utilizing PU channel usage models acquired from the REM and the SU channel usage individually observed by each SU. These MAC problems are the problem of designing a highly available control channel, the problem of efficiently searching for spectrum holes, the problem of spectrum mobility, and the problem of spectrum sharing among SUs. The contributions of this paper are listed below:

- A synchronized channel hopping-based common control channel was proposed for the initial handshake.
- The theoretical properties of this common control channel were derived.
- A stochastic channel set selection scheme for sensing and access was proposed taking both the PU channel usage and SU channel usage into consideration.
- A probabilistic suffix tree-based prediction scheme was proposed to arrange the channels from best availability to least availability in each time slot.

The rest of this paper is organized as follows. We present the system model used in Section 2. In Section 3, we present a brief explanation of the prediction scheme used in this paper. Then, we discuss the MAC used in this paper in Section 4 followed by the simulation results and conclusion in Sections 5 and 6, respectively.

2 System overview

In the system investigated, we have a centralized time slotted *primary network* (PN) and a decentralized *secondary network* (SN) which opportunistically utilizes the PN resources. In this secondary network, a REM exists which has the ability to collect the PN channel occupancy details. REM is assumed to possess records of past channel usage information of N channels, which are collected from the SUs and a dedicated set of sensors as in [5]. In the event that channel occupancy of PUs change and the statistics are recalculated, they are distributed by the REM to the registered users using the CCC.

The SUs register themselves with the REM and obtain the PU channel usage information. They were assumed to be synchronized among themselves and with the REM. This synchronization is vital in having a universally quiet period (QP) among the SUs. The universally QP enables the use of a less complex sensing method like energy detection as opposed to matched filter detection or cyclostationary detection. The SUs were assumed to be equipped with a widely tunable antenna, a sensing unit and a radio unit which is capable of sensing all the N primary channels and transmitting on the same. The SUs and the dedicated sensors were assumed to be capable of differentiating an SU signal from a PU signal since they are synchronized and have a QP at the beginning of each time slot. Further details on this topic are explained in Section 4.

We assumed unlicensed operation of SUs without any QoS guarantees, which was one of the two DTV band secondary spectrum usage schemes proposed in [27]. In this paper, we adopted the convention of denoting the *busy* period with 1 and the *idle* period with 0. The channel state distribution learning scheme used by the REM is explained in the next section.

3 Channel modeling and prediction

The most commonly used Markov chain in the literature is the one-step Markov chain. Suppose we let the states of the Markov chain be labeled by the window of last m observations. Then, even though the Markov chain does only a one-step prediction, that prediction depends on the last m observations. For example, let '1' and '0' represent the busy and idle states of a particular channel, a fixed-order Markov model of order m has strings of channel observations of the past m time slots as its state labels. Thus, there are 2^m number of states in this particular case. For example, if we assume $m = 2$, then we have $2^2 = 4$ states, and the state labels are $\{00, 01, 10, 11\}$. For $m = 3$, we have $2^3 = 8$ states having state labels $\{000, 001, 010, 011, 100, 101, 110, 111\}$. Since the number of states grows exponentially with m, only Markov chains with lower order can be practically used [6]. In variable-order Markov models as opposed to fixed-order Markov models, we only consider the states which are highly abundant on the training sequence and have a significant impact on the next observation. Therefore, the number of states in a variable-order Markov model is usually less than those in a fixed-order Markov models of the same order [28]. The PST introduced in [6] belongs to the category of variable-order Markov models. A summary of the algorithm is given below for the convenience of the reader.

If we take a variable-order Markov model of order $D = 2$, its states are labeled with binary strings of length $l \leq 2$ (strings are binary since the observation space is binary). The set of states in variable-order Markov model in this case is a subset S of $\Lambda = \{0, 1, 00, 10, 01, 11\}$, chosen by an algorithm run on a past observation sequence. But a sequence in this subset S cannot be a suffix of another sequence in the same set, for example, both 1 and 01 cannot be in set S. A suffix s' of a string s is defined as $s' \in \{s_i s_{i+1} ... s_l | 1 \leq i \leq l\} \cup \{e\}$ where $s = \{s_1 s_2 ... s_l\}$ and e is the null string. In our case $s_i \in \{1, 0\}\ \forall i \in \{1, 2, ..., l\}$. The other requirement about the state space S is that for every string $s \in S$, if the probability of having $\sigma \in \{1, 0\}$ after s ($Probability(\sigma|s)$) is greater than zero, there should be a unique string $\hat{s}$ which is a suffix of $s\sigma$ in S. Where $s\sigma$ is the the string having σ attached to the end of the string s. As mentioned earlier, to find out the states that matter in the variable-order Markov model, we used an algorithm called the PST algorithm, which was introduced in [6]. To use this algorithm, we needed a training sequence of sufficient length which represents channel behavior under normal circumstances, in our case, it was the channel state data gathered by the REM which are binary sequences. An algorithm to discover the *suffix set S* and build the PST using a given training sequence of length m is explained in the papers [6,28]. An algorithmic explanation of the PST algorithm is given in Appendix A for the convenience of the reader. In this paper, the symbol on the right hand side end of any time series string of channel state observations was assumed to be the most recent one.

3.1 Prediction using the probabilistic suffix tree

For a given PST $\bar{T}$ with maximum order D and the channel observations for the previous k ($0 \leq k \leq D$) consecutive slots, we traverse the tree starting from the node e which denotes the empty string. An example tree calculated according to the parameters given in Table 1 is shown in Figure 1. In this parameter list, the first 5 entries are the parameters of the PST algorithm, while σ_1^{10} is a training sequence of length 10. This traversal is done according to the past k channel observations starting with the most recent channel observation and going backwards in time in the past k channel states. Traversing the tree, we will reach a leaf node either using all the k channel state observations or partially using them. When this leaf node is reached, we use the probability distribution of the occurrence of an *idle* slot or a *busy* slot for that particular node as the probability of occurrence of the next channel state. For example in Figure 1, the values given in parentheses, $\{p_{\text{idle}}, p_{\text{busy}}\}$, are the probabilities of the occurrence of an idle slot and a busy slot, where node label gives the longest suffix of the observation string present in the model. In the next section, we will discuss the MAC protocol and the CCC design used.

4 MAC protocol

As discussed in the related work section, most MAC protocols in the literature did not take into account the channel usage characteristics of the PUs, with [9,25] being exceptions. But in those schemes too, the effect of SU access was not taken into account. Therefore, if all the SUs tried to access the optimum channel, the packet collisions bring the throughput down. In this section, we

Table 1 Threshold values used in algorithm and an example training sequence

Description	Value
P_{min}	0.006
α	0
r	1.05
D	3
γ	0.0006
σ_1^{10}	1110010000

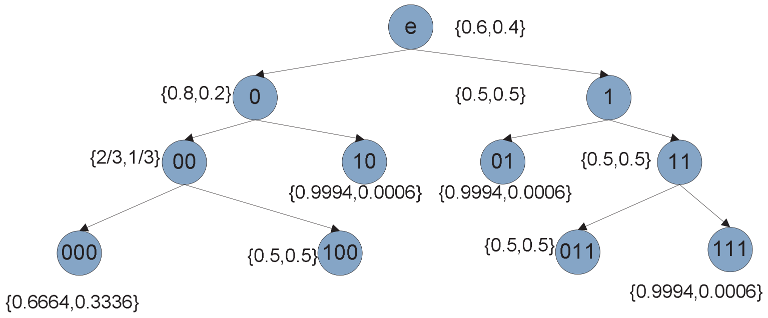

Figure 1 Example probabilistic suffix tree.

discuss how the PU channel usage statistics and the SU channel usage statistics could be used towards increasing the throughput of SUs. First, we explain how the PU channel statistics can be used in the CCC sequence selection process. Then, we explain how both the PU and SU statistics can be used in the data transmission process.

4.1 Proactive common control channel

Most SH CCC schemes in the literature searched for idle channels in a predetermined or random order. So each SU selected channels based on that channel scan order and listened to them to identify the state of the channel. If the channel was idle, the SU used it and if not, the search continued till it finds one and the receiver channel selection coincided with it. In the scheme presented, to reduce the interruptions to the primary and secondary users, we ordered the channels in the ascending order of activity. Thus, the SUs selected the channel with the highest expected availability more often, saving them from interfering with the PU and high variability in the idle channel search delay.

Let p_i denote the stationary probability of PU channel i being idle, where $i \in \{1, 2, \ldots, N\}$. This probability is given by the distribution of events at node *e*. This node *e* represents empty string in the PST. Then, Algorithm 1 was used to generate the channel sequence ϕ_j^t, where ϕ_j^t is the channel selected by SU j at time slot t.

Algorithm 1 Control channel selection algorithm

```
Initialize $\phi_j^t = \emptyset$
$u \Leftarrow rand()$
for $k = \{1, 2, 3, \ldots, N\}$ do
    if $u \leq \frac{\sum_{l=1}^{k} p_i}{\sum_{i=1}^{N} p_i}$ then
        $\phi_j^t = k$
        break;
    end if
end for
```

In the above scheme, it can be seen that the SUs select the channels with less PU activity more often. Furthermore, due to the randomness in the selection process, the packet collisions will be low. The downside to this channel selection method is not having a guaranteed rendezvous time as the schemes [24] and [23].

4.1.1 Analytic formulation of the average time to rendezvous

The time to rendezvous is an important performance metric because it decides the number of attempts an SU has to make to successfully complete a handshake with the receiver which in turn dictates the packet delays. Therefore, time to rendezvous should be as small as possible. In order to derive a formula for the average time to rendezvous for a transmitting SU, first, the probability of successfully contacting the receiver should be calculated. The method of calculation is explained below.

First, the following assumptions were made: the contention window size is CW, all backoff time values are equiprobable with probability $\frac{1}{\mathrm{CW}+1}$ and the total number of SUs is M. The vulnerable time is given by t_v. This is the minimum time gap between the smallest backoff time window and the second smallest backoff time window of contending neighbors on the same channel to avoid a packet collision. In other words, this is the average time duration it takes for nodes in the CSMA/CA network to identify the channel is busy after the first node transmits. This time is equal to the sum of the transmission time duration of the request to send (RTS) packet and the propagation time, in the unit of sub-time slots (smallest synchronous time unit) [29]. The false alarm probability is given by symbol p_{fa} and the probability of selecting channel $k(\leq N)$ is given by p_k^s. According to the CCC selection algorithm above, $p_k^s = \frac{p_k}{\sum_{i=1}^{N} p_i}$. The probability of channel access by an SU is given by a, and the cumulative distribution of the random backoff time slots generated is given by $F(\cdot)$. Using these parameters, the formula for the probability of successful handshake, θ_s, was formulated as given in Equation 1.

$$\theta_s = \sum_{i=1}^{N} p_i p_i^s \left[\sum_{r=0}^{M-1} (1-p_{\text{fa}}) \binom{M-1}{r} \left(ap_i^s\right)^r \left(1-ap_i^s\right)^{M-r-1} \right. \\ \left. \times \sum_{\tau=0}^{\text{CW}} (1-F(\tau+t_v))^r \frac{1}{\text{CW}+1} \right] p_i^s(1-a)\,. \tag{1}$$

The steps of the derivation of this equation are given below.

Proof. First let us find the conditions that should be met in order for an SU to succeed in the handshake. A given SU j can succeed in the handshake if all the conditions below are satisfied:

- SU j has a packet to transmit which is given by probability a.
- The SU j selects channel i which is given by the probability p_i^s.
- Channel i is idle, given by probability p_i.
- Channel i is sensed as idle by j, given by $(1 - p_{\text{fa}})$.
- SU j wins the contention.
- The receiver has selected channel i and is not transmitting, given by $p_i^s \times (1 - a)$.

For the SU j to win the contention, the backoff duration of it should be less than the minimum of all the other SUs trying to transmit in the same channel by atleast the vulnerable time t_v. In this paper, we assumed all the SUs can hear each other. If the cumulative distribution function of the backoff duration is given by $F(\cdot)$ and SU j's backoff duration is τ, the probability, $R(\tau + t_v)_r$, that r number of transmitting SUs backoff times are more than $\tau + t_v$ is given by:

$$R(\tau + t_v)_r = (1 - F(\tau + t_v))^r\ . \tag{2}$$

Then, the distribution of r SUs trying to access channel i can be derived using the binomial distribution with probability of success $a \times p_i^s$ and total number of other SUs $M - 1$ as given in Equation 3.

$$B\left(M-1, ap_i^s\right) = \binom{M-1}{r} \left(ap_i^s\right)^r \left(1-ap_i^s\right)^{M-r-1}\ . \tag{3}$$

Equation 2 gives the probability of winning the contention conditioned on, the backoff time τ of SU j, the number of other SUs trying to access channel i which is r and the given SU j is trying to transmit on channel i. Since the backoff time is uniformly distributed in $[0, \text{CW}]$, we took the expectation of $R(\tau+t_v)_r$ with respect to the backoff duration and the number of other users trying to access channel i. The result is the probability of an SU winning the contention which is given in Equation 4.

$$\zeta_s(i) = \sum_{r=0}^{M-1} \binom{M-1}{r} \left(ap_i^s\right)^r \left(1-ap_i^s\right)^{M-r-1} \\ \times \sum_{\tau=0}^{\text{CW}} (1-F(\tau+t_v))^r \frac{1}{\text{CW}+1}. \tag{4}$$

Then, multiplying $\zeta_s(i)$ by the probability of the receiver being tuned into channel i, ap_i^s, we get the probability of successful handshake in channel i. Then, finally, we took the expectation of it w.r.t. all the channels where the probability of selecting a given channel was $p_i\,(1 - p_{\text{fa}})\,p_i^s$ which gave us the result given in Equation 1.

□

Then, the probability distribution for the number of time slots to achieve rendezvous $TR(q = l)$ was calculated as given in Equation 5.

$$TR(q = l) = (1 - \theta_s\,)^{l-1}\,\theta_s\ . \tag{5}$$

4.1.2 Channel load in the SH control channel

Since this scheme uses a stochastically selected control channel, only the average channel load can be calculated. The probability of r users out of M SUs accessing channel i can be given by $B(M, ap_i^s)$ which was defined in Equation 3. Then, using $B(M, ap_i^s)$, an equation for the maximum average channel load was formulated as shown in Equation 6.

$$L\left(M, \underline{\mathbf{p}}^s, a\right) = \max_{i=\{1,2,\cdots,N\}} \sum_{r=1}^{M} r \cdot B(M, ap_i^s)\ . \tag{6}$$

4.1.3 Degree of rendezvous

This metric gives the number of minimum overlaps between any two channel sequences in a SH-based CCC. In a stochastic channel selection scheme, although a number for the degree of rendezvous cannot be given, the probability for two sequences of length Q to have a minimum of k overlaps can be calculated. Equation 7 gives an expression for this probabilistic degree of rendezvous.

$$\text{DR}(Q, k, \underline{\mathbf{p}}^s) = \sum_{i=1}^{N} p_i^s \sum_{r=k}^{Q} \binom{Q}{r} \left(p_i^s\right)^{2r} \left(1 - \left(p_i^s\right)^2\right)^{Q-r}\ . \tag{7}$$

4.2 Channel set selection for the data transmission

In predicting the status of a channel using the PST, we need to sense the channels. To make the CR less expensive, we used a single half-duplex transceiver and did energy detection-based narrow band sensing. Therefore, in our scheme, only sequential sensing was possible and there is an upper bound m, to the number of channels an

SU can sense in a given time slot. This upper bound is the time needed to fail in transmitting $m - 1$ times and trying transmitting for the mth time, so that there is time for contention window, RTS and clear to send (CTS) messages, data, and the ACK. Therefore, the selection of a set of channels which are both utilized less by the PU and the other SUs is necessary to achieve a better throughput. We used a channel set selection scheme which chose the channels which are highly available for a given SU j with high probability. To find this channel set, we have to calculate the probability of winning the contention on a given idle channel i.

4.2.1 Calculating the probability of winning the contention

This is a subjective usage indicator on an SU j of interest. To calculate this parameter, each SU kept track of the instances it tried to access channel i, $\epsilon_{ij}^{\text{tot}}$ and the instances it acquired channel i, $\epsilon_{ij}^{\text{acq}}$. Then, for the channels that the SU has been accessing, we calculated c_{ij} using the ratio between the two,$\frac{\epsilon_{ij}^{\text{acq}}}{\epsilon_{ij}^{\text{tot}}}$. For the channels that the SU has not tried to access, these parameter values were calculated based on the probability a channel is successfully accessed by an SU, given that each SU chooses each channel with equal probability $\frac{1}{N}$. Thus, for the unused channel i and SU j, the parameter c_{ij} was calculated as given in Equation 8.

$$c_{ij} = \sum_{r=0}^{M-1} (1 - p_{\text{fa}}) \binom{M-1}{r} \left(\frac{a}{N}\right)^{r} \left(1 - \frac{a}{N}\right)^{M-r-1} \times \sum_{\tau=0}^{\text{CW}} (1 - F(\tau + t_{v}))^{r} \frac{1}{\text{CW}+1} . \quad (8)$$

4.2.2 Channel set selection for the data transmission

Using the above calculated parameter c_{ij} and the stationary probability of channel i being idle, p_i, we selected the channel set using Algorithm 2.

Algorithm 2 Channel set selection algorithm

```
Initialize $\mathcal{S}_j = \emptyset$
while $|\mathcal{S}_j| < m - 1$ do
    $\bar{\mathcal{S}}_j = \{1, 2, 3, \ldots, N\} / \mathcal{S}_j$
    $k = \arg\max_{i \in \bar{\mathcal{S}}_j} \{p_i c_{ij}\}$
    $\mathcal{S}_j = \mathcal{S}_j \cup \{k\}$
end while
```

In the following section, we discuss how the communication between two SUs happen.

4.3 Communication between two SUs

In the scheme presented, the communication between two nodes happens in two stages. First, an initial handshake between SUs is completed, and then, the SUs communicate data between each other. In the first stage, the transmitting SU generates the channels to be used at each time slot t, ϕ_j^t, using Algorithm 1. Then, it generates the channel set $\mathcal{S}_j$ to be used for data transmission, using Algorithm 2. After that, the transmitting SU u starts the handshake phase with receiver v. This is explained in the next sub-section.

4.3.1 Handshake between the sender and the receiver

Prior to the handshake with the receiver, in the QP, the transmitter u senses the channel list $\mathcal{S}_j$ first and then senses the control channel ϕ_j^t. Then, in the backoff period, it calculates the statistical availability of the channel set $\mathcal{S}_j \cup \phi_j^t$ at time slot $t + 1$ using the sensing results and the PST models acquired. Then, the channels which have the probability of being idle greater than 0.5 are arranged in the descending order. This ordered list H_u^{t+1} is piggybacked on the RTS packet. If the intended receiver happened to listen to the same channel (i.e., $\phi_u^t = \phi_v^t$) and was able to decode the packet, it transmits a CTS packet. Then, the data communication is done using the same channel ϕ_j^t till the end of the time slot and the receiver sends an ACK. At this stage, the handshake is considered to be successful. If the CTS get timed out, the procedure is continued in the next time slot. A diagram showing an example of transmitter slot structure at handshake phase is given in Figure 2.

4.3.2 Data transmission after the successful handshake

If the handshake was successful at time slot t, the receiver v and transmitter u, knowing the channels to be used in access, sense the channel set H_u^{t+1} in the quiet period of time slot $t + 1$. This sensing is done starting from the worst to the best (i.e., in the reverse order). Therefore, after sensing the best channel, the transmitter u generates the backoff time and if sensed idle, it will send the RTS with the sensed channel status data on set H_u^{t+1} piggybacked on it. Then, if this RTS was received at the receiver v, it replies with a CTS with sensed channel status data on set H_u^{t+1} piggybacked on it. Then, after the data transmission starts at the end of the time slot, the receiver v sends an ACK if the transmission was successful. If the CTS was not received at the transmitter, the transmitter will try the next channel in sequence H_u^{t+1} after the CTS times out. This goes on until the list H_u^{t+1} gets exhausted within time slot $t + 1$. We set this time out period to be equal to the sum of CTS transmission time and propagation delay and the time to transmit the header of the data part and propagation delay. If the transmitter u was unable to contact the receiver v for an entire time slot, then the initial

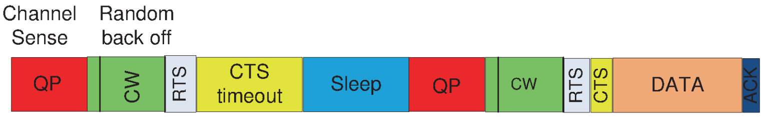

Figure 2 An example of transmitter slot structure at handshake phase.

handshake has to be redone, because the channel status at both ends of the current slot are not known to both u and v.

If the RTS and CTS exchange was successful, then the sensing outcomes of the channels at both ends are available to both u and v. Then, both u and v fuse the sensing results using OR decision fusion and save this result for each channel i separately in a buffer s^i_{uv}, where $i \in \mathcal{S}_j \cup \phi^t_j$. Each user u clears the buffer s^i_{uv} when the channel i is not sensed or the communication between each other is over in that time slot. Then, this buffer content at time slot $t+1$ is used to predict the channel status at time $t+2$. Then, both u and v make their ordered lists H^{t+2}_u and H^{t+2}_v. Both of these lists are the same. Therefore, in time slot $t+2$, the communication is possible. This process goes on till the communication between u and v is complete. A diagram showing an example of transmitter slot structure at data transmission phase is given in Figure 3.

4.4 Registration of new users, synchronization, and model updates

We assume that the REM transmits beacons every T time slots on all the available PU channels. This beacon transmission starts at the QP of the SUs; therefore, it is easy for the SUs to detect this beacon. This beacon contains a synchronizing bit sequence and encoded channel models. The SUs who has registered with the REM can decode these models. In these beacons, the periodic channel hopping pattern of the REM is mentioned so that the SUs newly arriving to the network can follow the REM hopping pattern and communicate with it to get registered. A newly arriving SU should scan the spectrum till it receives the beacon. The period T was determined as the minimum of the time for 5% of SUs to have a clock drift of $\frac{\text{QP}}{2}$ and the inter-arrival time between two new user arrivals to have 95% probability of arrival. Since all the registered SUs get synchronized with the REM from time to time, the assumption of synchronicity holds true.

5 Simulation

In this section, we carry out the numerical evaluation of the proposed scheme. In order to create the channel model according to the probabilistic suffix tree scheme, we needed a training sequence. Real data was not available to us, so we created a set of training data as strings of '1's and '0's. For this, we used a renewal process with *busy* time and an *idle* time generated according to a pseudo self-similar Markovian arrival process (MAP). The used MAP is capable of capturing the correlation between the different channel states. For the simulation purpose, we developed the MAP as given in [30,31]. As mentioned above, this MAP exhibited what is known as 'pseudo long-range-dependent self-similar characteristics'. Self-similarity means that the burstiness of traffic is the same over different time scales. In other words, if we measure the traffic in a particular link and plot, the traffic as a graph of the average amount of packets per 1, 10, 100, 1,000s, etc., the graph will only be scaled in amplitude and the shape will remain the same. The structure of the Markovian arrival process given in [30] has burstiness similar to Ethernet traffic. The MAP distribution is defined by a stochastic matrix D as defined in Equation 9, where D_0 and D_1 are sub-stochastic matrices containing the state transition probabilities to phases within the idle state and busy state, respectively, and d_{01} (d_{10}) contains the probabilities of transition to a phase in busy (idle) state from a phase in idle (busy) state. We fixed the size of the D matrix to be 6 × 6. The parameters of the MAP were calculated such that the channel utilization are 0.2,0.4, and 0.8, and the mean burst length is 10 slots. The theoretical properties of this MAP distribution can be found in [30]:

$$D = \begin{bmatrix} D_0 & d_{01} \\ d_{10} & D_1 \end{bmatrix} \qquad (9)$$

Before training the models, we changed the *busy* slots to *idle* with probability P_m and *idle* slots to *busy* with

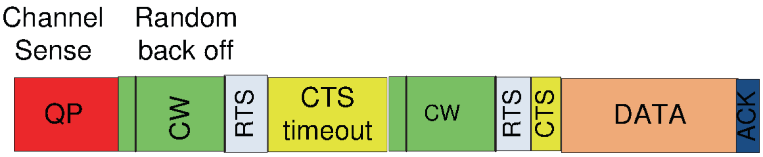

Figure 3 An example of transmitter slot structure at data phase.

probability P_f in the channel state sequence generated. The false alarm probability P_f was calculated for a given missed detection probability, $P_m = 0.1$, assuming an AWGN channel with mean SNR 5 dB at the SU. We then simulated the channel access of 5 and 10 SUs using 10 channels, while varying the channel access probability values a from 0.1 to 0.9.

We used the parameters given in Table 2 when training the PST and in the simulation setup. For comparison, we used a scheme from the literature called the HMM technique given in [17] for predicting the channel state in the random channel selection scheme and the channel selection scheme which uses the channels with lowest PU channel access probability. The results for the random channel selection scheme is named in the graphs as *Random* while the results for the scheme which only uses PU statistics is named as *PU only*. The scheme introduced in this paper is named as *Both stats* in the graphs. Since the HMM model in [17] did not provide any information on the number of states in the HMM or the memory length of sequences they used to determine the parameters, we chose ten states since training can be done in a reasonable time. The length of the string used for training was 20,000 for both PST and HMM schemes.

5.1 Initial handshake

In both the random channel set selection scheme and the channel set selection scheme based on PU activity, we used the control channel selection scheme introduced in [23] (*RCCH-Sync*). In Figures 4 and 5, we plot the average number of time slots taken by each scheme to achieve rendezvous for five and ten SUs, respectively. One can see that our scheme performs as equally well as the *RCCH-Sync* scheme. For this comparison, we used the scheme developed for synchronized networks in [23]. In that scheme, the synchronization offset between the transmitter and the receiver could not even have values which are integer multiples of the time slot length. Our scheme can handle that form of synchronization issue. In [23], authors also developed schemes which can withstand those kind of time offsets, but they had worse time to rendezvous than the synchronous one. In the same figures, we plot the theoretical results. One can see that the theoretical result is quite higher than the simulated value. The reason behind it is, when deriving this formula, we assumed the channels are accessed by the SUs according to the control channel selection scheme in all the time slots, while in the simulation, once the handshake is done, the channels are accessed according to a different distribution which allows SUs to balance its load among all the channels due to the use of SU statistics in data phase channel selection. Furthermore, in the data phase, the receiver

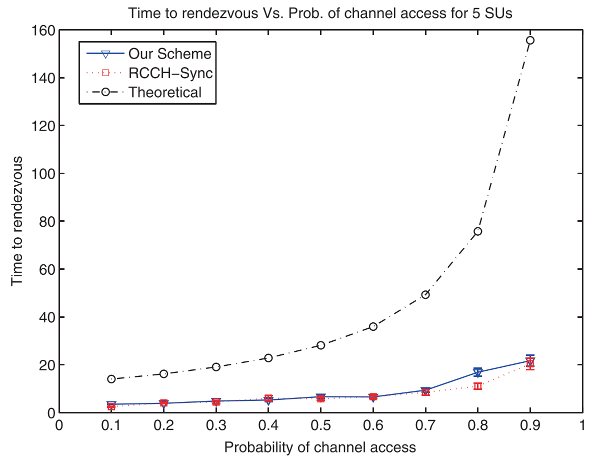

Figure 4 **Average number of control slots vs. SU channel access probability for five SUs.**

Table 2 Simulation parameters

Parameter	Value
D	10
$P_{\min}$	0.006
α	0
r	1.05
γ	0.0006
N	10
Number of SUs	20
k	3
Channels 1 to 4 PU utilization	0.2
Channels 5 to 7 PU utilization	0.4
Channels 8 to 10 PU utilization	0.8
Probability of missed detection	0.1

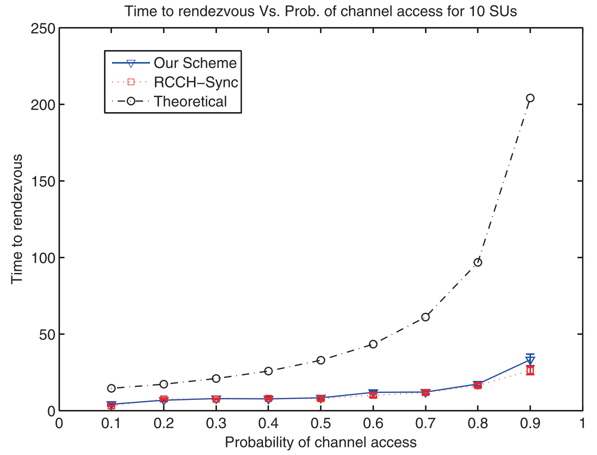

Figure 5 **Average number of control slots vs. SU channel access probability for ten SUs.**

is tuned to the channel the transmitter is using and it is not transmitting. Therefore, these happen with probability 1. But in the derivation of the theoretical result, we associated them with the respective probabilities associated with them in the control phase. Ideally, only the number of users in the handshake phase and the channels free from both PU activity and the SU data transmission activity should be used in this calculation. But due to the complex interactions of SUs, this is not feasible.

We used an exponential backoff scheme for the CSMA/CA access of the SUs. We limited the backoff exponent to 4 for the handshake phase and 3 for the data phase. The time duration details for the control phase is given in Table 3.

The duration of the RTS, CTS, ACK, and DATA parts of the frame for a given data rate R, D bytes of payload, S bytes of channel sensing information is given in Equations 10, 11, 12, and 13. The expressions are taken from [29].

$$t_{\text{RTS}} = 20\ \mu\text{s} + \frac{(22 + (20 + S) \cdot 8)}{R}\,, \tag{10}$$

$$t_{\text{CTS}} = 20\ \mu\text{s} + \frac{(22 + (14 + S) \cdot 8)}{R}\,, \tag{11}$$

$$t_{\text{ACK}} = 20\ \mu\text{s} + \frac{(22 + 14 \cdot 8)}{R}\,, \tag{12}$$

$$t_{\text{data}} = 20\ \mu\text{s} + \frac{(22 + (28 + D) \cdot 8)}{R}\,. \tag{13}$$

5.2 Data phase

In the simulations, once the initial handshake is done, each time slot is divided into three sub-slots. This number is equal to the number of channel switches allowed at each time slot. Then, an SU who acquired the channel can transmit a packet in each sub-slot only if the PU is not present. If an SU transmitted when a PU is present, it is called an interfered sub-slot. This performance measure is plotted against the probability of SU channel access in Figures 6 and 7 for five and ten SUs, respectively. For the random and PU activity-based schemes, we used an HMM for the prediction and in our scheme, we used PST scheme for prediction. In this figure, we can see that our scheme has higher interference than the PU statistics-based one and random one for all SU user channel access probabilities. But the probability of interference to PU is less than 0.005 which is insignificant. A possible reason behind this is, due to the usage of the SU statistics based on past contention failures, a given SU overestimates the level of competition for the highly available channels and try to aggressively use the channels with high PU activity. Fine tuning of the data collection window of contention failures and successes should be able to reduce this problem.

Table 3 Time duration for time slot

Parameter	Value
Transmission rate	1 Mbps
Sensing time per channel	0.05 ms
Channel switch time	1 μs
Slot duration	4.615 ms
CW$_{\text{min}}$	15
Slot time	9 μs
Propagation time	5 μs

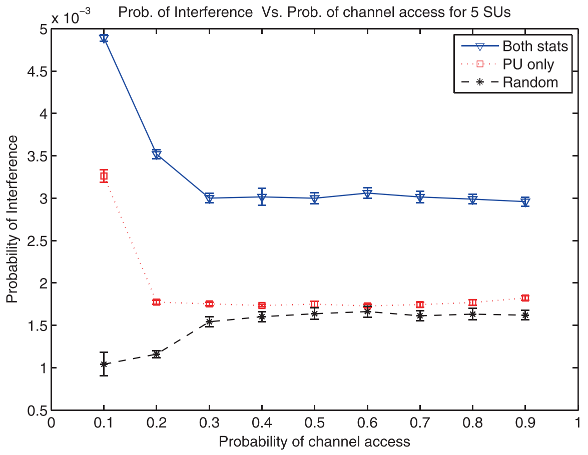

Figure 6 Probability of interference to PU vs. SU channel access probability for five SUs.

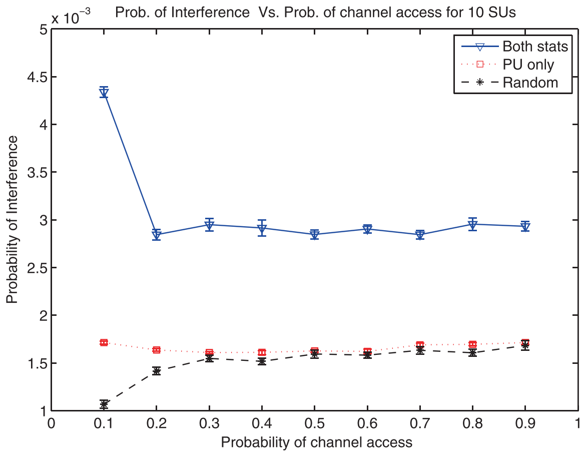

Figure 7 Probability of interference to PU vs. SU channel access probability for ten SUs.

In this simulation, the probability with which an SU begins a transmission was distributed according to the Bernoulli distribution with $a = \{0.1, 0.2, \ldots 0.9\}$ being the probability of success. Then, as long as the number of packets to be transmitted is above zero, packets were added to the transmission queue at the beginning of each main slot according to the discrete uniform distribution having the set of outcomes $\{1, 2, 3\}$. The transmission was assumed to have completed when the queue gets empty. Then, the transmission probability is calculated for each SU by taking the ratio between the number of packets successfully transmitted and the number of attempts made. The graph depicting the fluctuation of the probability of successful transmission is given in Figures 8 and 9 which are plotted for five and ten SUs, respectively. In these figures, we can see that our scheme performs better in ten SU scenarios but becomes inferior to the performance of the random channel selection method when SU transmission probability is 0.4 in the scenario having five SUs. The possible reason behind this is, when the number of transmission attempts are low, the probability of contention failures is not large enough to penalize a channel with low PU activity from getting selected; therefore, most SU select those ending up with higher collisions. Therefore, the probability of success is low. Evidence for this reasoning can be seen in Figures 10 and 11. In these graphs, we plot the probability of the transmissions of two or more SUs colliding with each other for five and ten SUs, respectively. The low probabilities of successful transmission for our scheme in Figures 8 and 9 corresponds to the high collision probabilities in Figures 10 and 11, respectively. Furthermore, although the collision probability of our scheme is more than that of the random scheme, our scheme has better probability of successful transmission. This is due to the fact that our scheme uses channels with both less PU channel activity and less SU channel activity. Since we calculate the SU activity using previous access results, it is not perfect. This imperfectness causes higher collisions than the random scheme which uses all the channels equally, but since it stays biased towards the channels with low PU activity, it results in a better probability of successful transmission.

Finally, in Figures 12 and 13, we plot the average number of SUs, using the same channel at the same time, which is a measure of channel load. The higher the channel load, the higher will be the number of collisions and thus lower throughput.

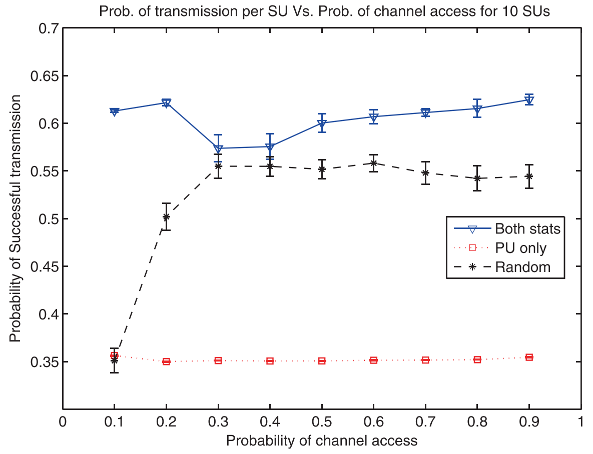

Figure 9 Probability of successful transmission per SU vs. SU channel access probability for ten SUs.

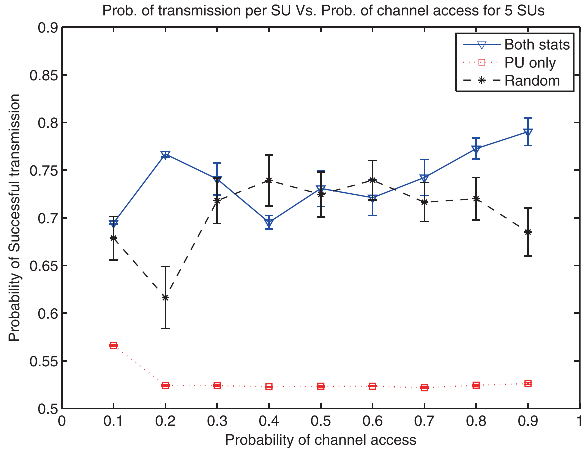

Figure 8 Probability of successful transmission per SU vs. SU channel access probability for five SUs.

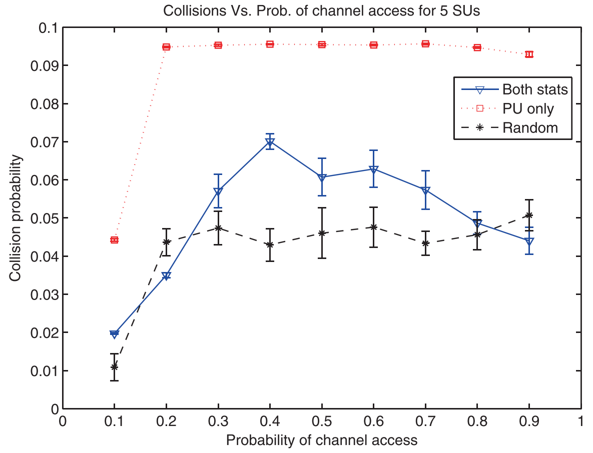

Figure 10 Probability of collision among SUs vs. SU channel access probability for five SUs.

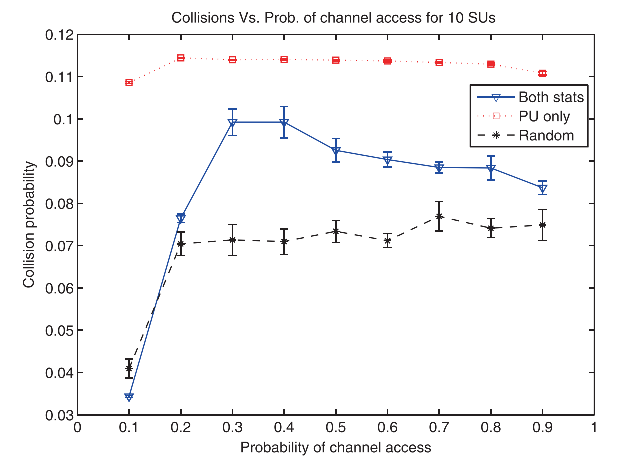

Figure 11 Probability of collision among SUs vs. SU channel access probability for ten SUs.

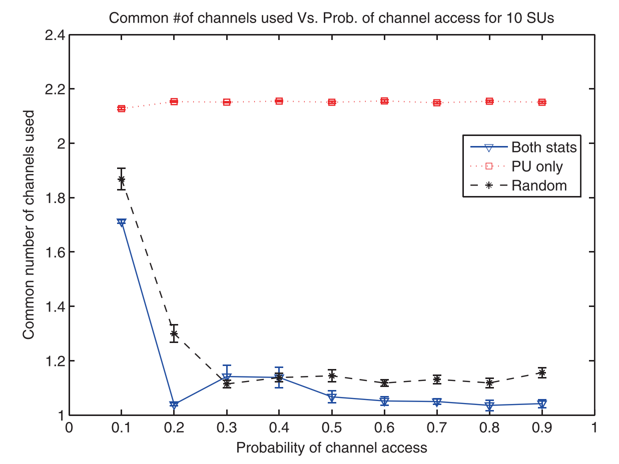

Figure 13 Average number of channels simultaneously used by SUs vs. SU channel access probability for ten SUs.

6 Conclusion

In this paper, we presented a proactive channel access MAC for an *ad hoc* CR network having a REM for initial setup and showed how this MAC can be used to keep the interference experienced by the PUs at a low level while providing better throughput for the SUs. The novel feature of this research was the incorporation of the SU statistics when selecting a channel set to be used. Furthermore, we introduced a SH CCC and derived the theoretical properties of it. This CCC used PU channel access probabilities when selecting the hopping pattern in the control phase. Then, in the data transmission phase, we used a prediction scheme called the PST which has a very low training and memory complexity. We tested our scheme in a scenario with a MAP-based channel state distribution with different channel utilization levels with imperfect sensing and showed that the impact of the channel set selection method has on the throughput and interference to the PUs. In the simulations, our channel set selection scheme had better probability of successful transmission than the random channel set selection scheme and the scheme based only on PU statistics at high SU traffic scenarios while having a PU interference probability less than 0.005. So when considering all of these factors, we can conclude that our MAC scheme is candidate worthy of consideration in proactive channel access in an *ad hoc* setting similar to the one the schemes we tested on.

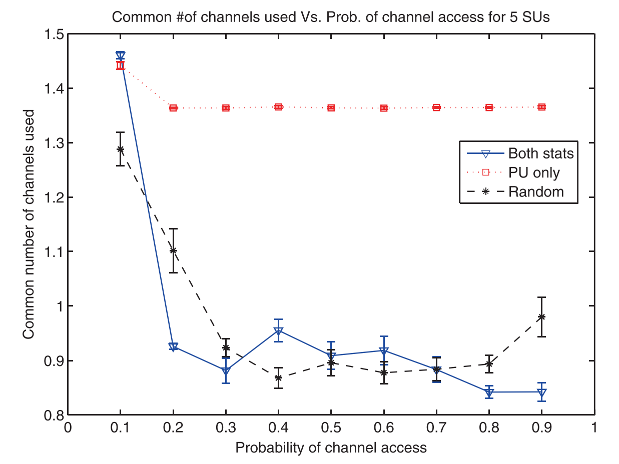

Figure 12 Average number of channels simultaneously used by SUs vs. SU channel access probability for five SUs.

Appendix A
PST learning algorithm

This algorithm consists of two phases. The main purpose of the algorithm was to find out a hypothesis which resembles the distribution which generated the sequence, where the Kullback-Leiber distance [32] between the actual and the empirical distribution is less than ϵ per state with probability $1 - \delta$ where $0 < \epsilon < 1$ and $0 < \delta < 1$ [6]. The Algorithms 3 and 4 mentioned below are based on [6,28]. A training sequence of sufficient length which represented the general channel usage was used by this algorithm to generate an accurate hypothesis on the distribution of the channel states. In the algorithm, empirical probabilities of the occurrence of any string s, $\tilde{P}(s)$, and the occurrence of a particular observation σ given the observation history sequence, $\tilde{P}(\sigma|s)$, were used to create the hypothesis. The formulae for calculating these which were taken from [6] are given in Equations 14 and 15, respectively. We assumed that a training sequence $\sigma_1^m = \sigma_1\sigma_2 \ldots \sigma_m$ of length m having consecutive observations of the channel states is provided.

$$\tilde{P}(s) = \frac{1}{m - D + 1} \sum_{j=D}^{m-1} \chi_j(s), \tag{14}$$

$$\tilde{P}(\sigma|s) = \frac{\sum_{j=D}^{m-1} \chi_{j+1}(s\sigma)}{\sum_{j=D}^{m-1} \chi_j(s)}, \tag{15}$$

where $\chi_j(s)$ is an indicator function which takes value 1 when $s = \sigma_{j-|s|+1} \ldots \sigma_j$ and 0 otherwise, $|s|$ is the length of string s and D is the order of the model. As an example, let $\sigma_1^{10} = 1110010000$ and $s = 00$. When $j = 2$, $\chi_2(00) = 0$ since $\sigma_1\sigma_2 = 11$ is not equal to $s = 00$. When $j = 5$, $\chi_5(00) = 1$, since $\sigma_4\sigma_5 = 00 = s$. All Markov predictors depend on the property that the probability distribution of the next symbol can be approximated by conditioning it on the previous D symbols, which is called by the name 'short memory principle' [7]. If we chose the memory D to be smaller than required, the distribution is incapable of capturing all the dependencies between the symbols, which degrades prediction efficiency [7]. On the other hand, if we chose it to be too high, it over-fits the training sequence and gives a higher prediction error.

The probabilistic suffix tree algorithm has a favorable property of avoiding the over-fitting of the model to the training sequence if the maximum memory kept is too high. Therefore, we can afford to keep D higher than the required since the algorithm takes care of it [7]. But this advantage does not come free. For the algorithm to take care of this issue, more tunable parameters were introduced. This algorithm is governed by five parameters namely $D, P_{\min}, \alpha, r$, and γ. These parameters control the upper bound of the number of states in the variable-order Markov model. The first parameter is the maximum memory of the variable-order Markov model, denoted by D. Value of D is the maximum number of ones and zeros in the strings in set S used to name the states of the Markov chain or the maximum level to which the tree is grown. The second parameter is the minimum probability of abundance $P_{\min}$ in the training sequence, of any binary string s' of length $|s'|(\leq D)$, for s' to be a member of the tree. This is a necessary condition, but it is not sufficient. The third parameter α denotes the minimum value the conditional probability $\tilde{P}(\sigma|s')$ (see Equation 15) can take, for a binary string s' of length $|s'|(\leq D)$ and a symbol $\sigma \in \{1, 0\}$. This is also a necessary condition a string s' should meet in order for it to be a node on the tree. The fourth parameter r is a threshold which determines whether the string s contributes additional information in predicting the next symbol σ than its longest suffix $\hat{s}$ which is a node on the tree (if $s = \sigma_1\sigma_2 \ldots \sigma_{k-1}\sigma_k$, then $\hat{s} = \sigma_2 \ldots \sigma_{k-1}\sigma_k$). The final parameter γ is the probability assigned to $\tilde{P}(\sigma|s)$, if it is zero for any symbol σ and state s. Phase 1 of the algorithm is given in Algorithm 3.

Algorithm 3 PST learning algorithm
Phase 1

```
Initialize $\bar{T}$ and $\bar{S}$ : $\bar{T}$ is a binary tree having a root node
  $e$ denoting empty string & $\bar{S}$ is an empty set
$\bar{S} \Leftarrow \{\sigma|\sigma \in \{1,0\}$ and $\tilde{P}(\sigma) \geq P_{\min}\}$
while $\bar{S} \neq \emptyset$, pick any $s \in \bar{S}$ and do
  remove $s$ from $\bar{S}$
  if $\exists\sigma \in \{1,0\}$ s.t $\tilde{P}(\sigma|s) \geq \alpha$ and $\frac{\tilde{P}(\sigma|s)}{\tilde{P}(\sigma|\text{suffix}(s))} > r$ then
    add the node corresponding to $s$ with all the nodes on the path starting from the deepest node in $\bar{T}$ which is a suffix of $s$ to $s$, to $\bar{T}$
  end if
  if $|s| < D$ then
    $\Sigma' = \{1,0\}$
    while $\Sigma' \neq \emptyset$ do
      remove $\sigma'$ from $\Sigma'$
      if $\tilde{P}(\sigma' s) \geq P_{\min}$ then
        add $\sigma' s$ to $\bar{S}$
      end if
    end while
  end if
  Continue to next $s' \in \bar{S}$
end while
```

As one can see in the above algorithm, we started with a single node tree representing the null string e and as the algorithm progressed, we added nodes which were mandatory for the distribution to be correct. In the algorithm, we add a node v labeled by a string s to the tree $\bar{T}$ if the following criteria were satisfied by the string s. First, the empirical probability of the occurrence of string s in the training sequence $\tilde{P}(s)$ should be larger than a threshold $P_{\min}$. Because of this requirement, we avoided the exponential growth in the number of strings to be tested in the algorithm. Then, we checked whether the probability of occurrence of a symbol $\sigma \in \{1, 0\}$ after the string s, $\tilde{P}(\sigma|s)$ is greater than a threshold α. Finally, we checked to see whether $\tilde{P}(\sigma|s)$ is greater than the probability of getting σ after the longest suffix suffix(s) of s. We checked this by taking the ratio $\frac{\tilde{P}(\sigma|s)}{\tilde{P}(\sigma|\text{suffix}(s))}$ and finding out whether it is greater than r where $r > 1$. We added the extended version of s which is σs where $\sigma = \{1, 0\}$ to the set of strings $\bar{S}$ if the probability of occurrence of σs in the training sequence is greater than $P_{\min}$, immaterial of whether s is included in the tree or not. This was done because it is possible to have a string s' which has a conditional distribution substantially different from its parent.

Parent of a node in our tree was labeled by the longest suffix of that particular node.

After performing Algorithm 3 on the training sequence, there is a possibility that, for some of the nodes, the probability of occurrence of either 1 or 0 after the string s labeling that node ($\gamma_s(1)$ or $\gamma_s(0)$) is 0. Furthermore, there is a possibility that the internal nodes of the tree $\bar{T}$ may only have one of the children with respect to the occurrence of either 1 or 0. We corrected those problems in the phase 2 of the algorithm given in Algorithm 4 [6].

Algorithm 4 PST learning algorithm
Phase 2

```
Initialize $\hat{T}$ to be $\bar{T}$
Extend $\hat{T}$ by adding all missing sons of internal nodes.
For each node in $\hat{T}$ labeled by string $s$ calculate the conditional probability $\gamma_s(\sigma)$ for each $\sigma = \{1, 0\}$ using Equation:
$\gamma_{s'}(\sigma) = \tilde{P}(\sigma|s')(1 - 2\gamma) + \gamma$,
where $s' = s$ if $s$ is in $\bar{T}$ else $s'$ is the longest suffix of $s$ in $\bar{T}$.
```

Although we talked about a Markov chain, the algorithm grows a tree. In this tree, the leaf nodes gave the respective state labels of the equivalent Markov chain. But in the case that the destination state after a transition from a given state is ambiguous, we should add some additional states. Since the tree is sufficient for the predictions to be done, we do not explain it here, the readers are referred to [6] for more information about constructing the equivalent Markov chain.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The work reported in this paper is financially supported by the Natural Sciences and Engineering Research Council of Canada.

Author details

[1] Department of Electrical and Computer Engineering, University of Manitoba, E1-551 EITC, 75A Chancellors Cir., R3T 5V6 Winnipeg, MB, Canada.
[2] Department of Electrical and Computer Engineering, University of Manitoba, E3-504A EITC, 75A Chancellors Cir., R3T 5V6 Winnipeg, MB, Canada.

Received: 6 June 2014 Accepted: 25 February 2015
Published online: 24 March 2015

References

1. FCC, ET Docket No 03-222 Notice of proposed rule making and order, December 2003 (2003)
2. Federal Communications Commission Spectrum Policy Task Force Report of the Spectrum Efficiency Working Group (2002)
3. L Tan, L Le, Joint cooperative spectrum sensing and MAC protocol design for multi-channel cognitive radio networks. EURASIP J. Wireless Commun. Netw. **2014**(1), 101 (2014). doi:10.1186/1687-1499-2014-101
4. C Devanarayana, AS Alfa, in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. Predictive channel access in cognitive radio networks based on variable order Markov Models, (2011)
5. F Bouali, O Sallent, J Perez-Romero, R Agusti, in *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2011 Sixth International ICST Conference On*. Strengthening radio environment maps with primary-user statistical patterns for enhancing cognitive radio operation, (2011), pp. 256–260
6. D Ron, Y Singer, N Tishby, The power of amnesia: learning probabilistic automata with variable memory length. Mach. Learn. **25**(2-3), 117–149 (1997). doi:10.1007/BF00114008
7. D Katsaros, Y Manolopoulos, Prediction in wireless networks by Markov chains. IEEE Wireless Commun. **16**, 56–64 (2009). doi:10.1109/MWC.2009.4907561
8. C Song, D Chen, Q Zhang, in *Communications (ICC) 2010 IEEE International Conference On*. Understand the predictability of wireless spectrum: a large-scale empirical study, (2010), pp. 1–5. doi:10.1109/ICC.2010.5502054. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5502054&isnumber=5501741
9. Q Zhao, L Tong, A Swami, Y Chen, Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. IEEE J. Selected Areas in Commun. **25**(3), 589–600 (2007). doi:10.1109/JSAC.2007.070409
10. Y Chen, Q Zhao, A Swami, Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors. Inf. Theory IEEE Trans. **54**(5), 2053–2071 (2008). doi:10.1109/TIT.2008.920248
11. Q Zhao, S Geirhofer, L Tong, BM Sadler, Opportunistic spectrum access via periodic channel sensing. Signal Process. IEEE Trans. **56**(2), 785–796 (2008). doi:10.1109/TSP.2007.907867
12. J Gu, WS Jeon, JM Kim, Proactive frequency-hopping dynamic spectrum access against asynchronous interchannel spectrum sensing. Vehicular Technol. IEEE Trans. **62**(8), 3614–3626 (2013). doi:10.1109/TVT.2013.2246595
13. Y Song, J Xie, Prospect: A proactive spectrum handoff framework for cognitive radio ad hoc networks without common control channel. IEEE Trans. Mobile Comput. **11**(7), 1127–1139 (2012). doi:10.1109/TMC.2011.140
14. H Kim, KG Shin, Adaptive mac-layer sensing of spectrum availability in cognitive radio networks. Technical report, University of Michigan, (2006). http://www.eecs.umich.edu/techreports/cse/2006/CSE-TR-518-06.pdf
15. L Yang, L Cao, H Zheng, Proactive channel access in dynamic spectrum networks. Phys. Commun. Elsevier. **1**(2), 103–111 (2008). doi:10.1016/j.phycom.2008.05.001
16. VK Tumuluru, P Wang, D Niyato, in *IEEE International Conference on Communications*. A neural network based spectrum prediction scheme for cognitive radio, vol. 294, (2010), pp. 1–5. doi:10.1109/ICC.2010.5502348. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5502348&isnumber=5501741
17. I Akbar, W Tranter, in *Proceedings 2007 IEEE SoutheastCon*. Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case, (2007), pp. 196–201. doi:10.1109/SECON.2007.342884
18. S Yarkan, H Arslan, in *2007 IEEE 66th Vehicular Technology Conference*. Binary time series approach to spectrum prediction for cognitive radio, (2007), pp. 1563–1567. doi:10.1109/VETECF.2007.332
19. P Ren, Y Wang, Q Du, J Xu, A survey on dynamic spectrum access protocols for distributed cognitive wireless networks. EURASIP J. Wirel. Commun. Netw. **2012**(1), 60 (2012). doi:10.1186/1687-1499-2012-60
20. A De Domenico, EC Strinati, M Di Benedetto, A survey on mac strategies for cognitive radio networks. Commun. Surv. Tutorials IEEE. **14**(1), 21–44 (2012). doi:10.1109/SURV.2011.111510.00108
21. BF Lo, A survey of common control channel design in cognitive radio networks. Phys. Commun. Elsevier. **4**(1), 26–39 (2011). doi:10.1016/j.phycom.2010.12.004
22. BF Lo, IF Akyildiz, AM Al-Dhelaan, Efficient recovery control channel design in cognitive radio ad hoc networks. Vehicular Technol. IEEE Trans. **59**(9), 4513–4526 (2010). doi:10.1109/TVT.2010.2073725
23. G-Y Chang, W-H Teng, H-Y Chen, J-P Sheu, Novel channel-hopping schemes for cognitive radio networks. Mobile Comput. IEEE Trans. **13**(2), 407–421 (2014). doi:10.1109/TMC.2012.260

24. K Bian, J-M Park, R Chen, Control channel establishment in cognitive radio networks using channel hopping. Selected Areas Commun. IEEE J. **29**(4), 689–703 (2011). doi:10.1109/JSAC.2011.110403
25. C Cormio, KR Chowdhury, Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping. Ad Hoc Netw. **8**(4), 430–438 (2010). doi:10.1016/j.adhoc.2009.10.004
26. S-S Tan, J Zeidler, B Rao, Opportunistic spectrum access for cognitive radio networks with multiple secondary users. Wireless Commun. IEEE Trans. **12**(12), 6214–6227 (2013). doi:10.1109/TWC.2013.102313.130125
27. H Bogucka, M Parzy, P Marques, JW Mwangoka, T Forde, Secondary spectrum trading in tv white spaces. Commun. Mag. IEEE. **50**(11), 121–129 (2012). doi:10.1109/MCOM.2012.6353691
28. R Begleiter, R El-yaniv, G Yona, On prediction using variable order Markov models. J. Artif. Intell. Res. **22**, 385–421 (2004)
29. B Jang, ML Sichitiu, IEEE 802.11 saturation throughput analysis in the presence of hidden terminals. Netw. IEEE/ACM Trans. **20**(2), 557–570 (2012). doi:10.1109/TNET.2011.2165322
30. S Robert, J-YL Boudec, New models for pseudo self-similar traffic. Perform. Eval. **30**(1-2), 57–68 (1997). doi:10.1016/S0166-5316(96)00055-7
31. A Alfa, V Pla, J Martinez-Bauset, V Casares-Giner, in *NETWORKING 2011 Workshops. Lecture Notes in Computer Science*, ed. by V Casares-Giner, P Manzoni, and A Pont. Discrete time analysis of cognitive radio networks with saturated source of secondary users, vol. 6827 (Springer Berlin, Heidelberg, 2011), pp. 3–12
32. TM Cover, JA Thomas, *Elements of information theory*, 2nd edn. (Wiley-Interscience, New York, New York, USA, 2006)