Optimal Channel Selection and Sensing Strategy for Cognitive Radio Networks

A Project Report

submitted by

IRENE DIAS

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

June 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Optimal Channel Selection and Sensing Strategy for Cognitive Radio Networks**, submitted by **Irene Dias (EE15M068)**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Sheetal Kalyani
Project Guide,
Associate Professor,
Dept. of Electrical Engineering,
IIT Madras, 600 036.

Place: Chennai

Date : June 9, 2017

ACKNOWLEDGEMENTS

I am greatly indebted to Dr. Sheetal Kalyani for guiding me through the entire course of my M.Tech. project. She always took the time and effort to discuss the problem and to suggest different methods to experiment. Her valuable remarks always gave new directions to my project.

I am thankful to all the professors whose courses helped me improve my knowledge in Wireless Communication and Signal Processing through the course of the two years of my M.Tech. program. Their classes always inspired me to think beyond classrooms into more practical scenarios.

A special thanks to Vishnu and Thulasi who were my fellow bandits in doing this project. This project would not have been possible without their contributions and insighful observations. A special word of thanks to all the wonderful people I met at IIT without whom life would not have been the same.

ABSTRACT

KEYWORDS: Cognitive Radio, Spectrum Models, Reinforcement Learning, Thompson Sampling, Optimal Channel Sensing, Bayesian Approach

Cognitive radios is an efficient tool in maximising the utilisation of available spectrum. Secondary users in a cognitive radio network achieve this by occupying the spectrum holes when the primary user is not transmitting. The performance of CR networks can be further improved by considering factors like number of sensing required to find a vacant channel and time taken for sensing. This is addressed in this thesis where we try to formulate an optimal sensing scheme based on a Bayesian approach which reduces the number of times we have to sense the spectrum before transmission. The proposed algorithm involves a two layer learning strategy-learning which channel to pick and learning the optimal sensing strategy. We have validated the performance of our algorithm in two widely used primary user traffic models, based on Generalised Pareto Distribution and Discrete Time Markov Chain Model. The optimal channel selection and sensing policy implements the Thompson Sampling approach and a conjugate prior based updation rule. The performance of our algorithm is compared against other Reinforcement Learning algorithms in terms of throughput, interference to the primary user and the number of sensing required.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CR	Cognitive Radio
ED	Energy Detector
SNR	Signal to Noise Ratio
ML	Machine Learning
RL	Reinforcement Learning
TS	Thompson Sampling
DSS	Dynamic Spectrum Sensing
pdf	Probability Density Function
CDF	Cumulative Distribution Function
i.i.d	Independent and Identically Distributed

NOTATION

$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{E}$	Probability expectation
$x[n]$	n^{th} sample of the vector observation **x**
$\mathcal{N}(\mu, \sigma^2)$	Normal random variable with mean μ and variance σ^2
$\sim$	Distributed as, for example, $X \sim \mathcal{N}(0, 1)$ denotes X is a zero mean Gaussian random variable with variance 1
χ_2	Chi-squared Distribution

CHAPTER 1

THE CHANGING WIRELESS SCENARIO

1.1 Introduction

With the ever increasing demand for wireless communication, the scarcity of frequency spectrum is a major problem that limits its capabilities. Thus researchers have been trying to implement technologies that can help expand the capabilities of a wireless system even when the available spectrum is less. Surveys conducted by Federal Communications Commission(FCC) clearly indicate the underutilization of the licensed frequency spectrum assigned to licensed users. With the increase in number of applications that require wireless communication, utilizing the underutilized spectrum seems like a feasible approach. Cognitive Radio [1] is an ingenious solution to this problem. The ability to share the less occupied licensed spectrum with unlicensed users when the *primary user* is not transmitting makes cognitive radio an innovative solution especially in the present scenario. By allowing the *secondary users*, i.e. the unlicensed users, to use the unoccupied spectrum, cognitive radio-enabled wireless communication systems efficiently utiilises the channels.

One of the major requirements in the cognitive radio setting is that, secondary users should cause negligible interference to the primary users. This requires the secondary user to sense the channels to detect the presence of primary user traffic before using them. This gives rise to the dilemma of how much time should be spent on sensing the channels and how much time will be left to actually transmit the data [2]. This is because more you sense the channel less is the probability of interference with the primary but we lose out on throughput. Moreover sensing requires energy which is also a constraint in the cognitive network. Therefore by reducing the number of times we sense the channels we gain in energy as well as time. Predicting which channels are more likely to be vacant can improve throughput performance of cognitive radios. This can be done efficiently by learning the traffic patterns on each channel.

The primary user traffic on the channels can be modeled based on the ON and OFF duration of transmission or the busy/idle period which can be assumed to be drawn from particular distributions. The channel occupancy at any instant can also be modeled as a simple independent coin toss model with different channels having different probabilities of success. In this case employing a Bayesian approach for channel selection would give the best possible results. But it has been shown that traffic models explained in Sec. 2.4 capture the dynamics of primary user traffic better [3]. It can also be observed from the channel realization that once a channel is found idle, it will mostly be vacant for the duration of the next frame. This information can be used to skip sensing the channel to check if it is busy.

Finding the algorithms that perform in different channel models provides an intuition as to which learning algorithm(or a combination of different algorithms) can give good performance in the cognitive radio scenario whatever be the traffic model. Arriving at an optimal algorithm is a trade off between throughput, sensing period used and interference to the primary user. This can be done by analyzing the performance of existing algorithms for different channel models and modifying them such as to capture the dynamics of the traffic model better.

1.2 Related work done

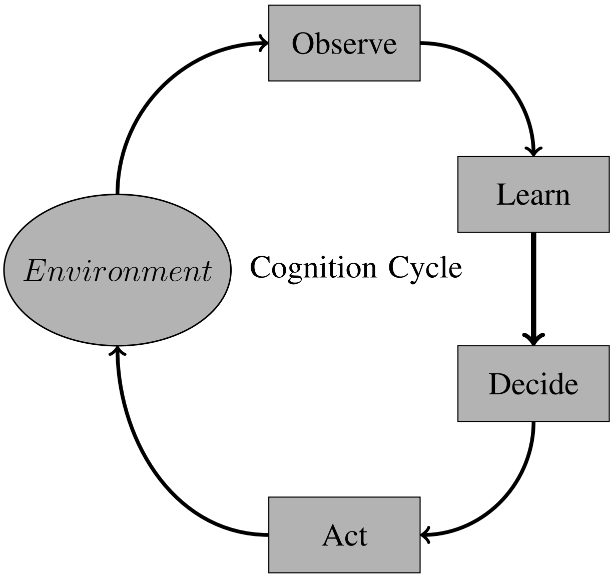


Figure 1.1: Cognition Cycle

From the time Mitola [4] presented the idea of Cognitive Radios (CR) in his doctoral thesis, a lot of research has been done to improve its performance. In his seminal paper he describes a system that works on a cognition cycle which observes, learns, decides and acts from it's environment (Figure 1.1). This has motivated the identification of diverse applications for Cognitive Radios from short range wireless access to wireless backhaul links [5]. The efficiency of a cognitive radio system depends on detecting a vacant spectrum for transmission. Channel occupancy can be determined by different spectrum sensing methods. The advantages and disadvantages of spectrum sensing methods that use Matched Filter (MF), Energy Detector (ED) and cyclostationary feature detection techniques have been analyzed in [6]. But all these detectors, except the ED, have the following disadvantages - they are complex to implement, they require exact knowledge of primary user signals and perfect synchronization with the primary user. Most of the work formalizes the spectrum selection problem in cognitive radios as a multi-objective optimization problem to maximize throughput of the secondary user and minimize either the number of times the channels are sensed [7] or the interference to the primary user [8]. Some works have also considered the optimization problem of minimizing the sensing duration while maximizing secondary user throughput [2] [9] . The secondary user accesses the licensed channels when the primary user is not transmitting. Expriments have shown that the primary user traffic ON/OFF time can be modeled by probability distributions. Most of the work done assumes that the ON/OFF times are sampled from an exponential distribution. But in [10] it has been shown that the generalized Pareto and the hyper-exponential distribution (HED) capture the primary user traffic variations more accurately. The channel occupancy evolution can also be modeled using Hidden Markov Models (HMM) [11] [12] which capture the dynamics of channel transitions. The importance of the learning capability of CR has been described by Mitola in his thesis [4]. Employing machine learning and reinforcement learning algorithms in CR enhances it's learning capability. The primary user traffic models learned can be employed to predict which channels are likely to be idle, which can then used for transmission instead of blindly sensing all the channels [13]. The spectrum selection problem can be mapped to Multi-Armed Bandit (MAB) algorithms as explained in [14] [15].

Reinforcement Learning algorithms like Q Learning [16] and Artificial Neural Networks (ANN) [17] have also been shown to improve CR performance. Some of the other methods for spectrum sensing suggested in [18] include Multi-layer Perceptron based neural networks, Bayesian inference methods, Auto-Regressive(AR) model based method, static neighbour graphs and Moving Average(MA) based methods.

Judicious sensing is quite important for achieving maximum throughput as time utilized for channel sensing is time lost for data transmission. In [19], the authors formulate an optimization problem to maximize throughput given a constraint on the probability of detection and prove that an optimal sensing time exists for a fixed frame size. On the other hand, [20] considers a transmission scheme where the sensing duration is fixed. An expression for the optimal frame size that maximizes normalizes throughput is derived; this determines how often sensing is carried out. However, this expression depends on the parameters of the primary traffic which might not be available to be secondary user or might be changing with time. The above mentioned works assume that the ON/OFF times are sampled from an exponential distribution.

There are many methods that are explained in literature to quantify the performance of algorithms employed in CR networks. The performance of CR is most commonly evaluated in terms of metrics like throughput of primary or secondary users, overall system throughput, interference to the primary user, number of sensing required, etc. [21]. In [8] a new performance metric called the interference efficiency has also been introduced.

In this report, a novel Bayesian approach to pick the optimal skipping duration without prior knowledge about primary user traffic parameters is proposed. This method is used in conjunction with a standard reinforcement learning algorithm to decide which channel to sense. Comparison results with other existing learning algorithms show that the proposed approach performs better in terms of throughput of the secondary user, number of sensing required per frame and number of frame collisions.

1.3 Organization of this thesis

This thesis is organized as follows:

Chapter 2 explains in detail the Cognitive Radio Network. The spectrum sensing technique used and the frame structures considered for transmission in Cognitive Radio are explained in brief. The primary user traffic can be captured using different mathematical models which are explained in Section 2.4.

Chapter 3 tries to introduce Reinforcement Learning (RL) and few widely used Reinforcement Learning algorithms for wireless communication. The update equations of algorithms give an intuition of how the algorithms learn the channel occupancy models by getting rewards from the environment and appropriately using them in the update equations.

Chapter 4 explains the algorithm that is proposed which is based on a Bayesian Learning approach and gives an intuition as to why it should work in the Cognitive Radio setting.

In *Chapter 5*, the simulation setup used for experiments is explained. The performance of different algorithms are analyzed in terms of graphs of throughput of secondary user, interference to the the primary user and the number of sensing required to find an unoccupied channel.

Chapter 6 summarizes the work done and provides some concluding remarks and avenues for future work.

CHAPTER 2

SYSTEM MODEL

2.1 Introduction

This chapter describes in detail the system model employed for the simulations. A typical snap shot of the channel spectrum across time and frequency is shown in Figure 2.1. As can be seen from the figure when a channel is not being used by the primary user, the channel remains idle which is called a *spectrum hole*. Thus cognitive radio improves the spectrum utilization.

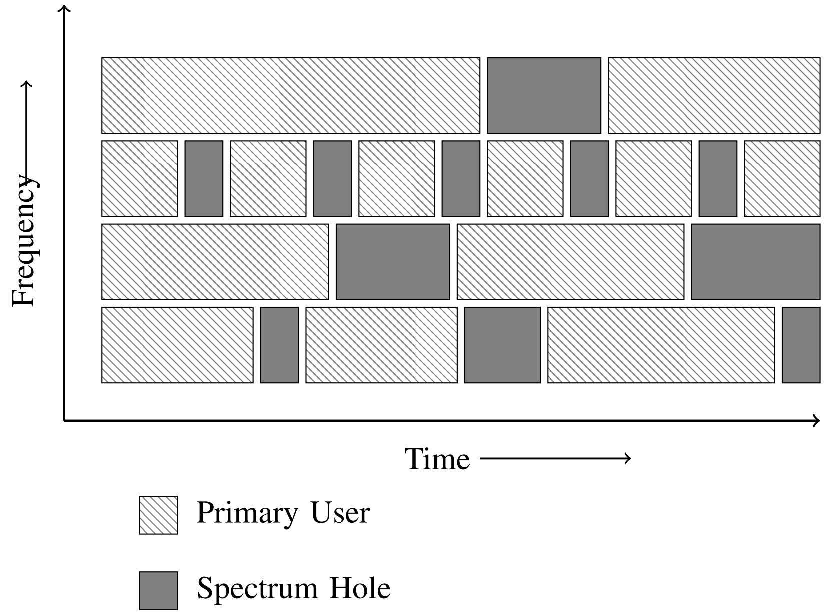


Figure 2.1: Illustration of spectrum utilization by primary user

Another observation that can be seen in a realistic scenario is that the traffic on each channels varies. Therefore it makes sense to learn the traffic model on each channel so that a channel that is mostly unoccupied can be chosen. This is where RL algorithms play a crucial role by learning the dynamics of the channels efficiently.

In our system model, we consider N channels that can be accessed by the secondary user, at any given instant. The primary goal of the secondary user is to transmit data packets so as to maximize its own throughput while causing minimal interference to the licensed users. Before the data is sent, the channel is sensed to check whether it is idle or busy, unless we are positive that the channel is available for transmission. This depends on the channel model that is used. We assume that the sensing operation is done until we find a channel that is idle or we exhaust all available channels. The aim is to reduce the number of sensing operations that need to be performed so that a larger part of the frame is available for data transmission.

The learning algorithm used to predict the current state of the channel returns a ranked list of channels in the order of estimated probability of occupancy. The sensing of channels is carried out in that order. Once a channel is found to be idle, the secondary user transmits its data. Upon transmission, we encounter one of the two scenarios: either collision occurs due to primary transmission or the packet is delivered successfully. If a packet collision occurs, we achieve a low throughput at the receiver end, else, high throughput is achieved. Packets can also be lost due to bad channel conditions. In this case the packet is said to be lost due to channel error. So, if a packet is successfully transmitted a ACK is received and when a packet is lost due to collision or channel error no ACK is sent which is equivalent to a NACK.

2.2 Spectrum Sensing in Cognitive radio networks

Identifying an appropriate spectrum sensing for CR is vital to the performance of CR. Commonly employed detection techniques are

- Matched filter detection
- Cyclostationary feature detection
- Energy detection

In this work, we consider energy detector for spectrum sensing as it is the simplest detector in terms of complexity and implementation [22]. Moreover, it does not require any primary user signal specific information, nor does it need a huge number of samples to determine the presence of primary user. Incidentally, this also happens to be the optimal detector for weak signal detection when the primary user signal arrival and departure times are unknown. The energy detector is explained here. The energy detector considers a simple Hypothesis Testing.

$$y[n] = \begin{cases} w[n] & : \mathcal{H}_0 \\ s[n] + w[n] & : \mathcal{H}_1 \end{cases}, \tag{2.1}$$

where $s[n]$ denotes the unknown primary user signal and $w[n]$ denotes additive white Gaussian noise (AWGN). The null hypothesis $\mathcal{H}_0$ stands for the absence of primary user signal and the alternative hypothesis $\mathcal{H}_1$ denotes the presence of the primary user. The noise is assumed to be from a normal distribution with mean 0 and variance σ_n^2, $w[n] \sim \mathcal{N}(0, \sigma_n^2)$. Similarly, primary user signal is assumed to be from a normal distribution with mean 0 and variance $\sigma_n^2 + \sigma_s^2$, $s[n] \sim \mathcal{N}(0, \sigma_n^2 + \sigma_s^2)$. The energy detector is implemented for a prespecified probability of false alarm, P_f and probability of detection, P_d. For a given value of P_f and number of samples N, the threshold(η) is calculated as,

$$\eta = \chi_2^{-1}(1 - Pf, N) \tag{2.2}$$

The decision made based on the energy (E) received, i.e. the sum of squares of the received samples is given below:

$$E = \sum_{i=1}^{i=N} x_i^2 \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{2.3}$$

2.3 Frame Structure

The data is transmitted by the secondary user in frames. Each frame that is transmitted has a sensing duration and a transmission duration. During the sensing duration the secondary user senses the channels either randomly or based on input from the learning algorithms. It is assumed that each channel sensing takes time τ. Once a channel is sensed free then during the transmission duration data transmission occurs. The sensing in each frame can be done in two ways: single slot sensing and multi-slot sensing. In the first frame type only one channel can be sensed in each frame and if this channel is found to be vacant the user transmits. But this method denies the secondary user the chance to find another free channel that might have been vacant during that time.

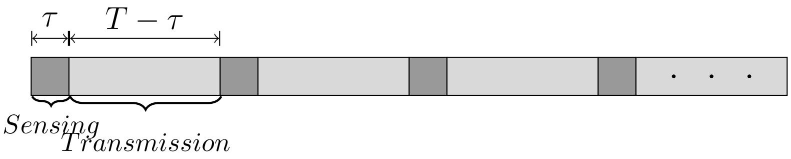


Figure 2.2: Single slot sensing frame

In the multi-slot sensing frames [23] the secondary user keeps on sensing the channels till it finds a vacant channel to transmit. This method has the advantage that the throughput is higher when compared to the case where the secondary user gives up sensing if the first channel sensed is occupied. The disadvantage is the fact that the secondary user has to sense more than one channel before it finds a vacant channel and the duration left for data transmission may vary from frame to frame.

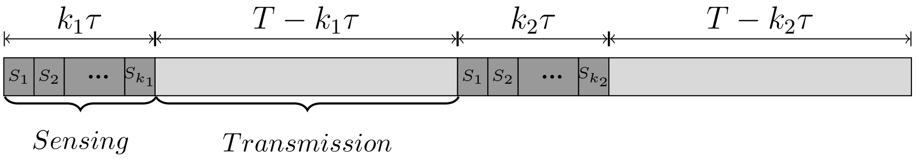


Figure 2.3: Multi slot sensing frame

2.4 Primary User Traffic models

The primary traffic on N channels is modeled to be independent. We take two approaches to model the primary traffic [3]: the discrete-time model and the continuous-time model.

2.4.1 Discrete Time Model

In the Discrete-Time Markov Chain(DTMC) model, the time index set is discrete. The discrete formulation of the primary traffic is adopted in [23–25].The behaviour of the channel can be expressed by a transition probability matrix as given below.

$$P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} \tag{2.4}$$

where p_{ij} represents the probability that the system transitions from state s_i to s_j.

Duty Cycle(Ψ) is defined as the probability that the channel is busy and can be written as $P(S = s_0) = 1 - \Psi$ and $P(S = s_1) = \Psi$. The DTMC model can be used to reproduce any arbitrary DC, Ψ, by selecting the transition probabilities as $p_{01} = p_{11} = \Psi$ and $p_{00} = p_{10} = 1 - \Psi$, which yields,

$$P = \begin{bmatrix} 1-\Psi & \Psi \\ 1-\Psi & \Psi \end{bmatrix} \tag{2.5}$$

For channels with varying load, the P matrix will also vary with time and the duty cycle will become a time varying quantity, $\Psi(t)$.

The channel load under this model is considered to be a random variable, which can be characterized by its PDF. The empirical PDFs of Ψ in real systems can be accurately fit with the Beta distribution(Eqn.2.6) or the Kumaraswamy distribution(Eqn.2.7).

The Beta distribution is given by

$$f_X^B(x; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \qquad x \in (0,1) \tag{2.6}$$

where $\alpha > 0$, $\beta > 0$ and $B(\alpha, \beta)$ is the Beta function.

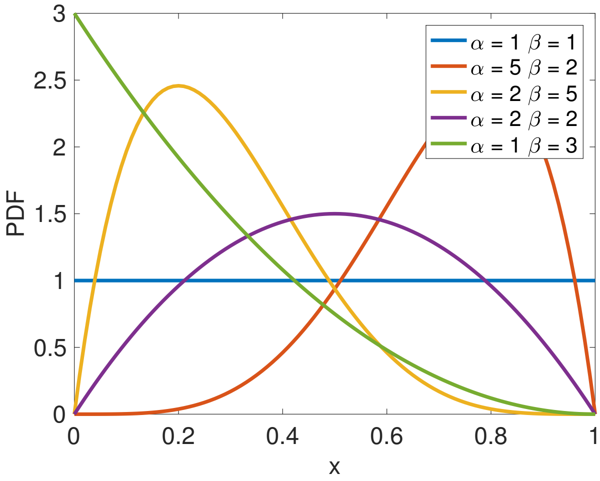


Figure 2.4: Beta distribution for different values of α and β

The Kumaraswamy distribution is given by

$$f_X^K(x; a, b) = abx^{a-1}(1-x^a)^{b-1}, \qquad x \in (0,1) \tag{2.7}$$

where $a > 0$, $b > 0$. For our simulations we have considered the model based on the Beta distribution. A single channel instance for medium traffic condition when using this traffic model is shown in figure 2.5

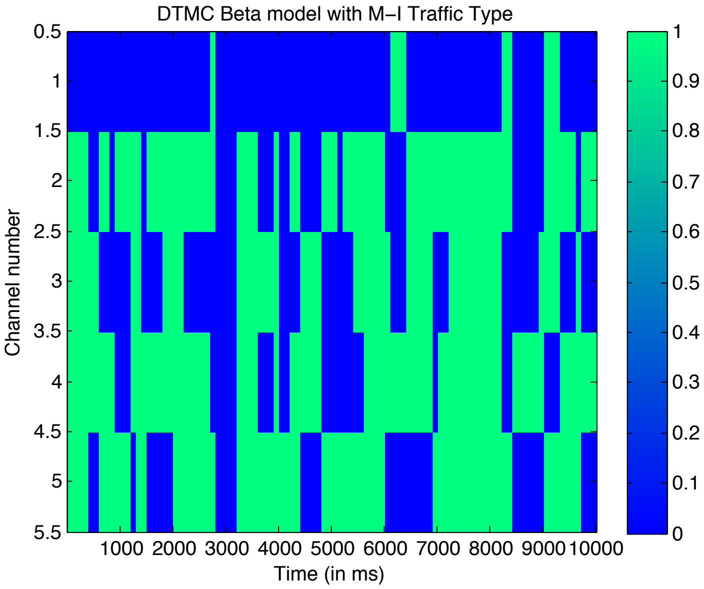


Figure 2.5: Channel occupancy plot for a Discrete Time model with medium traffic load

Various traffic intensities can be modeled using this model [3] if the parameters α and β are chosen as given in Table 2.1

Traffic Type	α	β
Low Traffic(L-I)	$(0, 1]$	$[1, 5]$
Medium Traffic(M-I)	$(0, 1]$	$(0, 1]$
High Traffic(H-I)	$[1, 5]$	$[1, 5]$

Table 2.1: Table with parameter ranges for α and β for different traffic intensities DTMC model

2.4.2 Continuous Time Model

For the continuous-time model, we assume that the on and off times are distributed according to the Generalized Pareto Distribution(GPD). This is a heavy-tailed distribution that allows us to model a wider range of traffic [26]. The probability density function of a GPD is given by

$$f(x|k, \sigma, \theta) = \frac{1}{\sigma}\left(1 + k\frac{x - \theta}{\sigma}\right)^{-1-\frac{1}{k}} \qquad \theta < x; k > 0 \tag{2.8}$$

where k, σ and θ are the shape, scale and location parameters respectively.

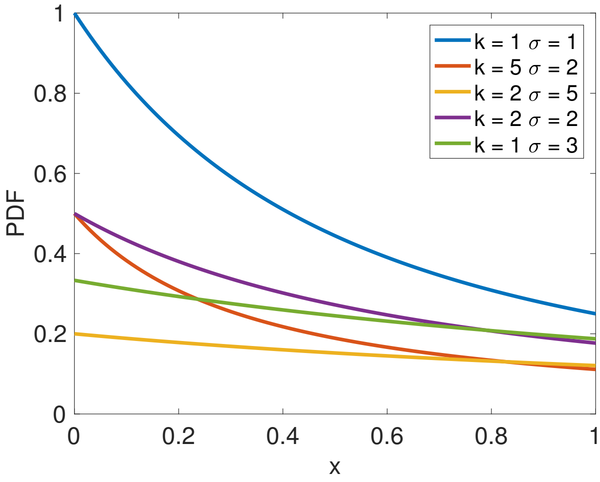


Figure 2.6: Generalised Pareto Distribution for different values of shape parameter and scale parameter with 0 location parameter

Varying traffic load can be generated by varying the parameters of GPD. The channel occupancy simulated using is given in Figure 2.7

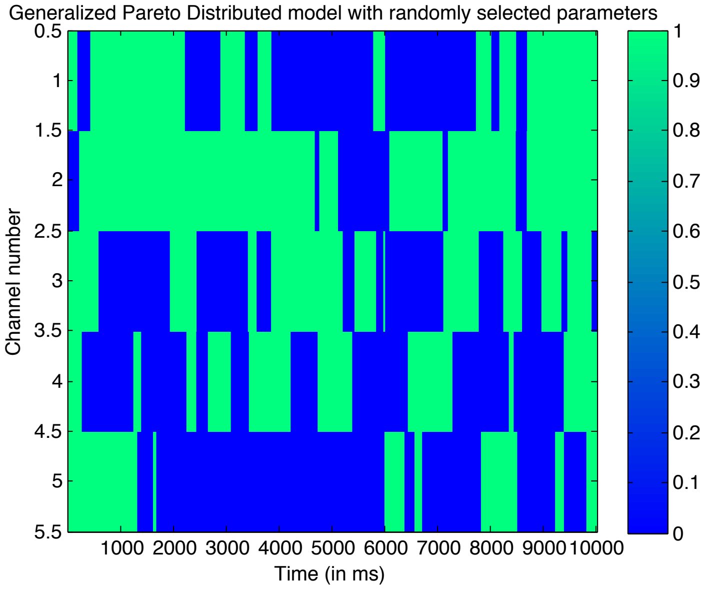


Figure 2.7: Channel occupancy plot when considering a Generalised Pareto Distribution for ON/OFF times

As can be seen from the occupancy plots, some channels are more likely to be idle than others. Therefore, rather than randomly sensing channels to see which are idle and which are busy, a learning algorithm can be employed to learn the occupancy on channels. The next chapter explains a few Reinforcement Learning (RL) algorithms which have been employed for channel selection in cognitive radios.

CHAPTER 3

REINFORCEMENT LEARNING

3.1 Introduction

The past few decades have seen the emergence of learning algorithms and artificial intelligence that is now being employed widely for data analytics in diverse fields from robot manipulations and control systems to medicine and wireless communication.Some diverse applications where RL is employed is given in [27], [28] [29] and [30].Artificial Neural Networks (ANN), Artificial Intelligence (AI) and Genetic algorithms are now being explored widely for it's learning capabilities. But, whatever be the learning algorithm it involves interaction with the environment and formulating future actions based on past learning experiences. Learning can be broadly categorized as supervised, unsupervised and reinforcement learning. In Reinforcement Learning(RL), the agent learns to act appropriately by interacting with the environment [31]. With each action, a reward is obtained which is the only mode of feedback for the learning agent. RL is attractive as it does not require a dataset for training as in machine learning problems. It learns from the observations it gets from the environment and hence, can be used in real-time applications. RL is often employed to solve complex online optimization problems in areas such as robotics, online recommendation etc.

Multi-armed bandits(MABs) are a class of RL problems which can be modeled using a single state. The formulation was inspired from a one-armed slot machine where a gambler wishes to maximize his earning by picking the slot machine that is most likely to give him/her a high reward. A learning agent chooses one of the N actions available to it and updates the value of the action according to the reward it obtains. The next action chosen is a function of the learnt values. A standard problem that is addressed in this setting is the exploration exploitation tradeoff. The algorithm needs to optimally balance utilizing the action that has performed the best so far, and looking for new options that can potentially better. Various algorithms have been proposed to achieve this tradeoff and some of them are listed below.

3.2 MDP based Approach

These algorithms assume a Markov Decision Process (MDP) that satisfies the Markov property [32]. A finite MDP is characterized by its state and action sets. The probability of reaching a state s' and getting a reward r from state s after taking an action a is given by the probability $p(s', r|s, a)$, which is the state transition probability.

3.2.1 Q Learning

The Q Learning Algorithm is a widely used reinforcement learning algorithm. It was first introduced in 1989 by Watkins [33] . The Q Learning algorithm works on a Markov model with multiple states and actions. The algorithm selects the optimal action in each state by learning Q-values, $Q(s, a)$, associated with each action in each state. Q Learning also incorporates an exploration strategy like ϵ-greedy or softmax in its framework. This makes sure that the algorithm does not always exploit. The Q Learning algorithm has been shown to converge to the optimal action for each state [34] . The Q Learning algorithm is as given in Algorithm 1. In the algorithm α is the learning rate, γ is the discount factor. The learning rate decides how much to weigh previously obtained values and the reward obtained in the current time step and the discounting factor discounts the rewards obtained in future.

Algorithm 1 Q Learning

```
Initialization: $Q(s, a) = 0, \forall a \in A, A$ is set of all possible actions and$\forall s \in S, S$ is set of all possible states
for $t = 1, 2, 3, \ldots$ do
    Select the action $a$ based on the exploration strategy
    Take action $a$ selected, $s'$ is new state reached
    Obtain reward $r(s, a, s')$
    Update $Q_{t+1}(s, a) \leftarrow (1 - \alpha)Q_t(s, a) + \alpha * (r + \gamma * \max_{b \in A(s')} Q_t(s', b))$
end for
```

The stateless Q Learning variant [35] assumes a single state with multiple actions. It does not assume multiple states as in the conventional Q-Learning algorithm. The update equation for stateless Q Learning is obtained by setting the discounting parameter γ in the Q Learning algorithm to be 0. This is similar to the Multi-Armed Bandit (MAB) setting for which the algorithms explained in the next section can be employed.

3.3 Frequentist Approaches

In the frequentist inference approach the observer performs a finite number of experiments and the value inferred is approximated to be the value obtained after infinite number of observations. This method assumes values from unobserved samples which can go wrong especially if the basic characteristics of the observations is not stationary.

3.3.1 Upper Confidence Bound Policies

Upper confidence bound policies attaches a quantity called upper confidence bound to the value of each arm and then chooses the one with maximum combined values [36]. The confidence bound gives an indication of how confident we are about the value associated with each arm. The confidence term for a less explored arm will be large. This makes the algorithm pick that arm in the next time instants and thus also incorporates exploration. UCB policies are known to achieve logarithmic regret uniformly over time rather than asymptotically. These policies are easy to implement and are computationally efficient.

Algorithm 2 UCB1 Policy

```
Initialization: Play each arm once.
for $t = 1, 2, 3, \ldots$ do
    Calculate average value $\bar{x}_i$ for $i = 1, \ldots, K$
    Calculate UCB index $\sqrt{\frac{2\log(t)}{N_t(i)}}$ for $i = 1, \ldots, K$; where $N_t(i)$ represents the number of times $i^{th}$ arm is played till time $t$
    Calculate $x_i = \bar{x}_i + \sqrt{\frac{2\log(t)}{N_t(i)}}$ for each arm
    Play the arm with highest $x_i$ value
end for
```

There are other variants of UCB policies called UCB2, UCB-NORMAL [36], etc. which tightens the confidence bounds with the knowledge of new parameters. The UCB-NORMAL algorithm is specifically proposed for the case where the rewards are drawn from normal distributions with unknown mean and variance.

3.3.2 EXP3

EXP3 is proposed for adversarial bandit problems which is a subset of non-stationary bandits, where the reward structure evolves over time [37]. In an adversarial setting the adversary or environment is such that it always tries to minimize the reward that the player gets. EXP3 is given in Algorithm 3. Here, the distribution over the arms is a mixture of uniform distribution and a distribution based on the observed rewards from each arm. EXP3 performs better in an adversarial setting as the algorithm brings in a randomization when it assigns a probability with which to pick an arm. This fetches better rewards than playing a deterministic policy which the adversary can figure out. The scaling by $p_{i_t}(t)$ during weight update ensures that the new estimated reward $\hat{x}_{i_t}(t) = \frac{x_{i_t}(t)}{p_{i_t}(t)}$ gives an unbiased estimator of the rewards.

Algorithm 3 EXP3 with weak regret

```
**Parameters:** $\gamma = \min(1, \sqrt{\frac{Klog(K)}{(e-1)T}})$
**Initialization:** $w_i(1) = 1$ for $i = 1, \ldots, K$
**for** $t = 1, 2, \ldots, T$ **do**
    **for** i = 1, . . . , K **do**
        $p_i(t) \leftarrow (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$
    **end for**
    Pull arm $i_t$ according to the distribution $\mathbf{p}(t)$
    Receive feedback $x_{i_t}(t) \in [0, 1]$
    Set $w_{i_t}(t+1) \leftarrow w_{i_t}(t) \cdot \exp\left(\frac{\gamma}{K}\frac{x_{i_t}}{p_{i_t}}\right)$
**end for**
```

3.4 Bayesian Approach

3.4.1 Conjugate Priors

In estimation theory Bayesian inference is the method where using the prior distribution of the parameter to be estimated, say θ and based on the value of the observation received we update the posterior distribution for θ.

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{3.1}$$

Here, $p(y|\theta)$ is the likelihood function and $p(\theta)$ is the prior distribution on parameter θ. A distribution for which the the posterior distribution for a given likelihood distribution has the same form as the prior is called a $conjugate\ prior$. For example consider the case where the likelihood function is a Bernoulli distribution with parameter θ.

$$p(y|\theta) = {}^{n}C_{k}\theta^{k}(1-\theta)^{n-k} \tag{3.2}$$

$$k = \sum_{i=1}^{n} y_i \tag{3.3}$$

If the prior on θ is a beta distribution with parameters α and β, i.e.,

$$p(\theta) = \frac{\Gamma(\alpha, \beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{3.4}$$

Then, the posterior distribution is given by

$$p(\theta|y) \propto \theta^{k}(1-\theta)^{n-k}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{3.5}$$

$$\propto \theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1} \tag{3.6}$$

which is again a beta distribution with parameters $\alpha + k$ and $\beta + n - k$. Therefore the beta distribution is a conjugate prior for the Bernoulli distribution.

3.4.2 Thompson Sampling

Thompson Sampling comes from a family of randomized probability matching algorithms. It is a Bayesian approach where Bayesian prior of the original distribution is used as tool to encode the current knowledge about the arms. This algorithm is first introduced in [38] and is further analyzed by [39]. Thompson Sampling algorithm has been shown to perform well especially when the reward is the output of Bernoulli trials. In this case the update equations perfectly captures the mean of the distribution.

Algorithm 4 Thompson Sampling

```
Initialization: $\alpha_0^i = 1, \beta_0^i = 1 \,\forall\, i = 1, \ldots, K$
for $t = 1, 2, 3, \ldots$ do
    Sample $\theta^i$ from Beta($\alpha_{t-1}^i, \beta_{t-1}^i$) for all arms
    Select arm with highest value of $\theta^i$
    Pull that arm $I_t$ and receive reward $x_{I_t,t}$
    $\alpha_t^{I_t} \leftarrow \alpha_{t-1}^{I_t} + x_{I_t,t}$
    $\beta_t^{I_t} \leftarrow \beta_{t-1}^{I_t} + (1 - x_{I_t,t})$
end for
```

The above algorithms have been employed in the CR setting in literature. The stateless Q-learning algorithm is employed for choosing the channel to transmit by incorporating some heuristic system information like Inter-cell Interference Coordination (ICIC) signals and Radio Environment Map(REM) in [40]. Q-learning is also employed for channel selection in cognitive radio by [16]. Work done in [41] and [42] use ideas from Thompson sampling to propose an algorithm for channel selection in CR. A variant of the UCB algorithm is also used for channel selection in CR in wireless sensor networks in [43]. These applications mostly look at selecting an optimum channel for transmission. But the proposed algorithm explained in the next chapter uses learning not only for channel selection, but also to find the optimal sensing duration.

CHAPTER 4

PROPOSED APPROACH

4.1 Motivation for the Proposed Approach

Most of the work in CR literature focuses on finding the optimal channel to transmit. Channel sensing to find a vacant channel is repeated every frame. In our algorithm we introduce two stages of learning. Learning to pick the right channel through an MAB formulation and learning an optimal strategy for sensing that channel. A metric, known as value, is maintained for all the arms which indicates quantitatively how beneficial it is to pick that arm. In the context of cognitive radio, this is an indicator of whether a channel is likely to be idle or vacant. When an arm is played, i.e., when a channel is chosen, we receive a reward from the environment in the form of throughput/interference which is used to update the value of that arm. At any instant, the SU can decide to exploit - pick the channel with the maximum value, or explore - pick among other channels so as to learn more about their behaviour. We illustrate our algorithm using the Thompson sampling algorithm [38] although any of the other bandit or RL algorithms can be employed. As we use the multi-slot sensing approach, the Thompson Sampling algorithm returns a ranked list of channels instead of just one optimal arm.

To determine the sensing strategy, one possible approach is to have the secondary user transmit on a channel until it encounters a collision from the primary data. However, this implies sending on a channel is interrupted only when a collision occurs. This increases the interference caused to the primary user. Thus, ideally we should stop transmitting before a collision occurs and switch to another channel. To do so, we need to estimate the underlying traffic distribution of the primary user. The ideal number of frames for which sensing can be skipped can be predicted using a Bayesian approach. A prior distribution for the parameter is maintained which is updated when a data sample is observed. In practical situations, secondary users do not have exact knowledge about the primary traffic; hence, in our algorithm, SU assumes that the primary traffic follows exponential on/off model and maintains a prior for the exponential parameter, θ. The conjugate prior for an exponential distribution is a gamma distribution whose parameters we update to estimate θ. The Generalized Pareto Distribution has a heavier tail when compared to the exponential model; hence, assuming an exponential distribution is justified as it prepares the SU for shorter idle times than the actual traffic.

The OFF time of the channel is a sample from an exponential distribution with parameter θ, i.e., $T_{OFF} \sim Exp(\theta)$. The mean of the distribution is $1/\theta$. Using the conjugate prior we are estimating the parameter θ. Therefore, to determine the number of steps for which sensing can be skipped, we consider the inverse of the sample obtained from the prior distribution. Physically, $\frac{1}{\theta}$ signifies the mean time duration for which the channel stays idle, which is the quantity that has to be estimated.

Let the SU assume that the primary traffic (OFF time) is exponential with parameter θ for a specific channel; let the conjugate gamma prior be parametrized by α and β, i.e., $\theta \sim \mathcal{G}(\alpha, \beta)$.

$$p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta} \tag{4.1}$$

In the cognitive radio context we get a data sample x which denotes the duration for which data is transmitted without experiencing a collision. It is essentially a sample of the primary traffic(OFF time) quantized to the frame size T as we assume that we do not have information of any collisions that occur within one frame duration. The posterior distribution is given by

$$\begin{aligned} p(\theta|x) &\propto p(\theta)l(\theta|x) \\ &\propto \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}\theta e^{-\theta x} \\ &\propto \theta^{\alpha+1-1}e^{-(\beta+x)\theta} \\ p(\theta|x) &\sim \mathcal{G}(\alpha+1, \beta+x) \end{aligned} \tag{4.2}$$

The gamma distribution is hence updated by the above equation. The number of frames to be skipped is then obtained as a function of the posterior.

4.2 Algorithm Description

The steps involved in the choosing the optimal channel skipping policy are listed here. The algorithm operates in two states: the *SENSE* state and the *SKIP* state.

- *SENSE* state: In this state, the algorithm always asks for a ranked list of preferred channels from the Thompson Sampling algorithm. Then, it follows the multi-slot sensing policy to sense the channels to find a vacant channel. Only on finding a vacant channel, will it transmit the data. Also, the duration to skip sensing is found by taking the inverse of a sample from the gamma prior distribution and calculating the number of frames to skip t_{skip}.
- *SKIP* state: In this state the algorithm, primary channel sensing is not performed. The data to be transmitted is sent on the channel which was previously found to be vacant. C_{skip}, a counter for number of skips performed, is also incremented.

The detailed algorithm is as given below:

Step 1: *Initialization* - Set the parameters α_i and β_i of the gamma prior for all channels to 1. Also, set the algorithm in the *SENSE* state and the C_{skip} to 0.

Step 2: *Channel Selection* - If in the *SENSE* state then obtain a new ranked list of preferred channels from TS algorithm and sense the channels sequentially to find a vacant channel to transmit. A sample is drawn from $\mathcal{G}(\alpha_i, \beta_i)\ \forall\ i$ and $t_{skip,i} = 1/\theta_i$ is calculated. Else, if the algorithm is in the *SKIP* state then transmit on channel used in the previous time step and increment C_{skip}.

Step 3: *Obtain feedback* - After frame transmission reward is obtained in the form of a collision indicator which is used to update the TS algorithm and *ACK* is obtained when frame is successfully transmitted. Absence of *ACK* indicates frame is lost due to collision with the primary or because of channel error.

Step 4: *Prior Updation* - Three cases can occur here as given below:

(a) *ACK* received and $C_{skip} < t_{skip,i}$, then continue in *SKIP* state

(b) *ACK* received and $C_{skip} = t_{skip,i}$, update α_i and β_i of gamma prior of channel i used to transmit as given below and reset C_{skip} and go to *SENSE* state

$$\alpha_i \leftarrow \alpha_i + 1$$
$$\beta_i \leftarrow \beta_i + t_{skip,i} * frameLength$$

(c) *ACK* not received, update α_i and β_i of gamma prior of channel i used to transmit as given below and reset C_{skip} and go to *SENSE* state

$$\alpha_i \leftarrow \alpha_i + 1$$
$$\beta_i \leftarrow \beta_i + (C_{skip} - 1) * frameLength$$

Step 5: Goto **Step 2**

The Thompson Sampling algorithm modified to return a ranked list of channels instead of a single channel is given by Algorithm 5.

Algorithm 5 Thompson Sampling (TS) functions

```
Parameters $S_i = 1, F_i = 1 \, \forall \, i \in \{1, \dots, N\}$
function GetRankedList()
    $d_i \sim Beta(S_i, F_i)$
    Sort $d_i$ in descending order
    Return the index of sorted order
end function
function UpdateObservation(channelIndex,collisionOccured)
    for All channel $i$ preceding $channelIndex$ in preference list do
        $F_i \leftarrow F_i + 1$
    end for
    if collisionOccurs then
        $F_{channelIndex} \leftarrow F_{channelIndex} + 1$
    else
        $S_{channelIndex} \leftarrow S_{channelIndex} + 1$
    end if
end function
```

4.3 Intuitive working of the algorithm

Here, we aim to estimate the idle time of the primary user on each of the channels using a Bayesian approach. The duration for which we can forgo sensing a channel is found by drawing a sample from the conjugate distribution and taking the inverse. The β parameter is updated with the duration for which data was successfully transmitted without channel sensing: $(C_{skip}-1) \times T$ when a collision occurs and $t_{skip} \times T$ when there is no collision. Independent of whether a collision occurs or does not occur, α is always incremented by 1 when a β sample is updated; it accounts for the number of samples observed.The mean of the gamma distribution is given by α/β which is the estimate of θ. As mentioned in Section 4.1, the number of frames (OFF duration of primary traffic) that can be skipped is inversely proportional to θ. When no collision occurs, we update β with a higher value; this reduces the mean of the posterior, which implies longer OFF times. On performing this update with each sample that is received, we arrive at an an optimal duration to skip. The motivation behind drawing a sample from the gamma distribution rather than directly using the estimated values is that drawing a sample from a distribution holds the possibility of exploring a higher value for time duration to skip sensing which will help to converge to the best possible skip duration rather than exploiting a skip duration which might actually be lower than the ideal value. As we observe higher number of samples, the variance of the gamma distribution, α/β^2, decreases. This implies that with more number of samples, picking a sample from the gamma distribution is very close to picking its mean value α/β. Also, α/β is the MMSE estimate of θ. Figures 4.1 and 4.2 show the flow of control in the proposed algorithm.

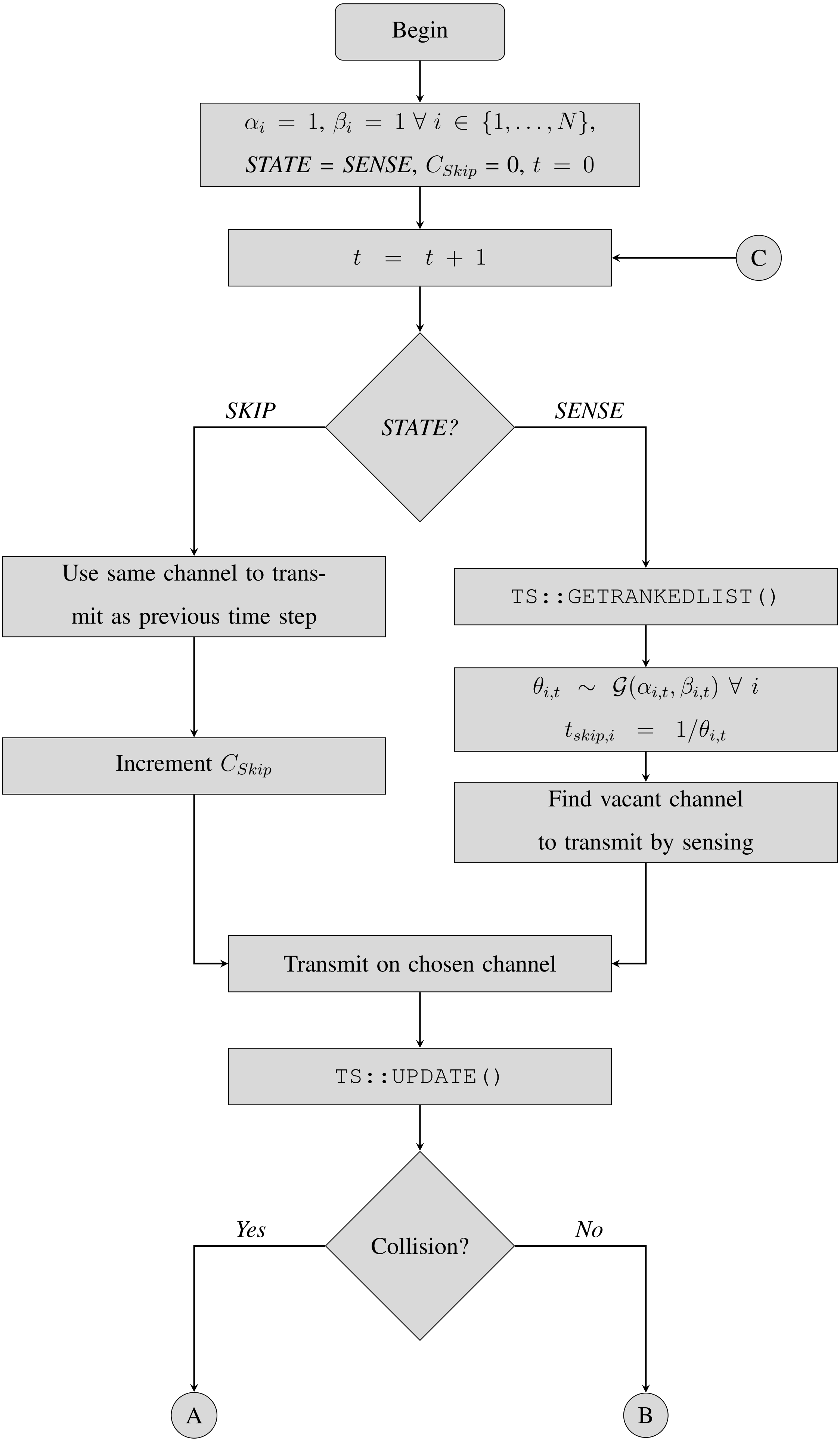

Begin
$\alpha_i = 1, \beta_i = 1 \, \forall \, i \in \{1, \ldots, N\}$,
STATE = *SENSE*, C_{Skip} = 0, $t = 0$
$t = t + 1$
C
STATE?
SKIP
SENSE
Use same channel to transmit as previous time step
TS::GETRANKEDLIST()
$\theta_{i,t} \sim \mathcal{G}(\alpha_{i,t}, \beta_{i,t}) \; \forall \; i$
$t_{skip,i} = 1/\theta_{i,t}$
Increment C_{Skip}
Find vacant channel to transmit by sensing
Transmit on chosen channel
TS::UPDATE()
Collision?
Yes
No
A
B

Figure 4.1: Flow chart of proposed algorithm

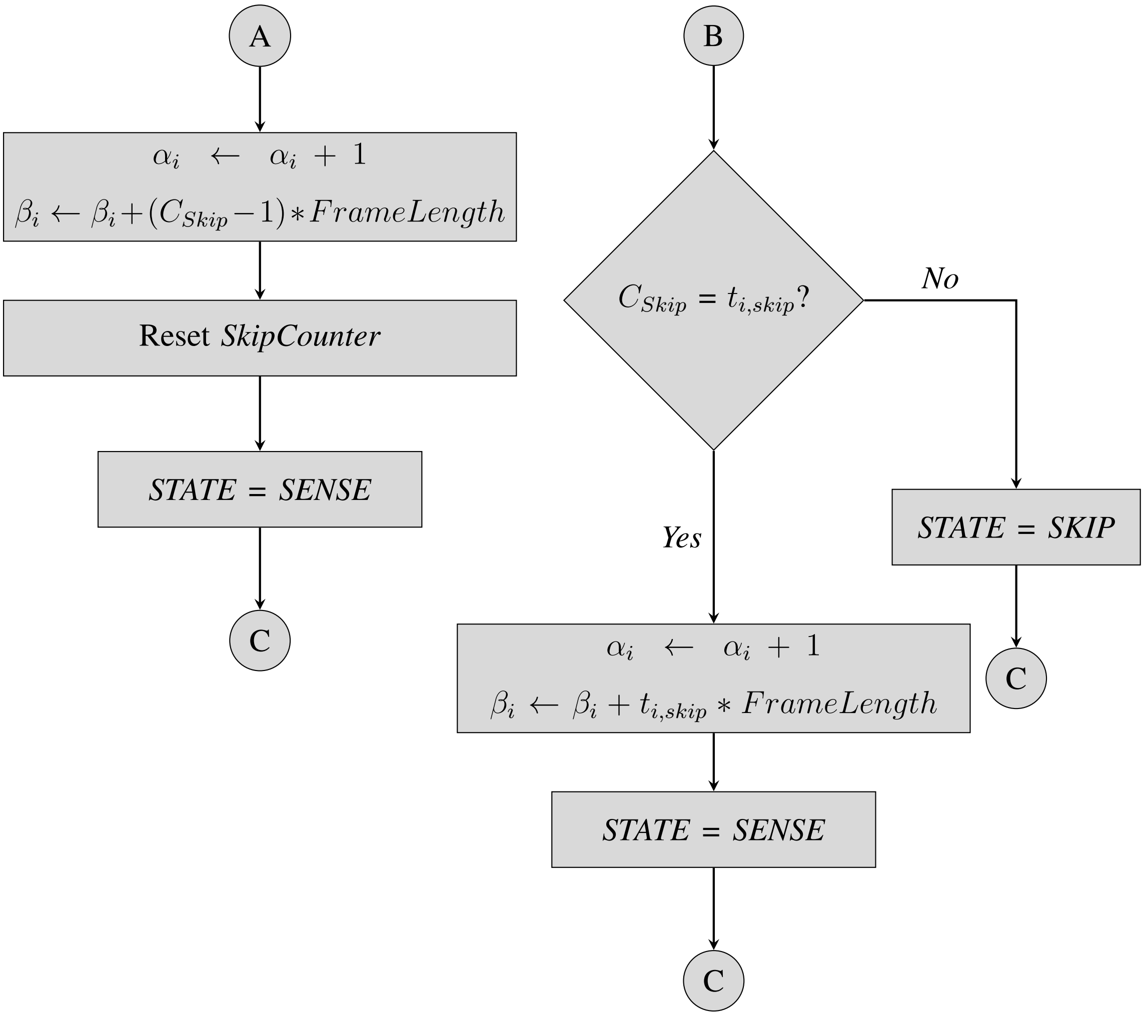

A
$\alpha_i \leftarrow \alpha_i + 1$
$\beta_i \leftarrow \beta_i + (C_{Skip} - 1) * FrameLength$
Reset *SkipCounter*
STATE = *SENSE*
C
B
C_{Skip} = $t_{i,skip}$?
No
Yes
STATE = *SKIP*
C
$\alpha_i \leftarrow \alpha_i + 1$
$\beta_i \leftarrow \beta_i + t_{i,skip} * FrameLength$
STATE = *SENSE*
C

Figure 4.2: Flow chart of proposed algorithm(Contd.)

Each channel maintains a different gamma distribution. So, considering the channel occupancy model in Figure 2.7, channel 5 with a longer OFF time, and hence will have a higher value for t_{skip} and channel 2 with short OFF time will have a lower value for t_{skip} .

CHAPTER 5

SIMULATION RESULTS

5.1 Simulation Setting

For the simulations $N = 10$ channels are considered and the analysis is done for a single secondary user setting. The underlying primary user traffic on the channels is simulated according to the models that are mentioned in section 2.4. The main traffic scenarios simulated are:

- Continuous Time Model based on the GPD ON/OFF time
- Discrete Time Model with Low traffic intensity
- Discrete Time Model with Medium traffic intensity
- Discrete Time Model with High traffic intensity

The values of α and β for the discrete time model are chosen for different traffic intensities in the corresponding ranges given in Table 2.1. For each of the 10 channels randomly chosen parameters in the specified range is chosen to simulate a real time scenario of different traffic levels on different channels. The frame duration is taken to be 100ms as considered in [13], [2] for which the sensing duration is taken to be $6ms$. A smaller frame length of $50ms$ is also considered with a sensing duration of 3ms to analyze the performance of the algorithms when more samples are available to the learning algorithms. The performance of the learning algorithms described in Chapter 3 are evaluated in terms of three metrics:

- Throughput obtained by the secondary user(in Mbps)
- Average number of frame loss
- Average number of sensing required per frame till finding a vacant channel

Let τ be the time SU takes to sense the state of one channel and k be the number of the channels the SU senses before the transmission begins. Then the achievable throughput of each strategy is calculated as

$$Throughput = \frac{T - k\tau}{T} \cdot C \tag{5.1}$$

where

$$C = \log_2 \left(1 + SNR_{sec}\right). \tag{5.2}$$

Here, SNR_{sec} represents the SNR experienced at the receiver of the secondary user. For the experiments, the value of SNR_{sec} is set to $20dB$. Reported results are the averaged metrics over 5000 independent runs for a time duration of 40s. We also provide results of a random agent as a baseline. The random agent picks a channel uniformly at random from the set of channels.

The throughput parameter gives an intuition of how efficiently we are actually able to pick a channel that remains idle for the duration of data transmission.The average number of frame loss takes into account the number of frames lost due to collision with the primary user, due to channel errors and the acknowledgment of frame reception lost due to a bad channel after transmission. The number of sensing per frame is indirectly a measure of how time we can save by not sensing as we will be transmitting during that duration which increases throughput. Less number of sensing required also indicates energy saved.

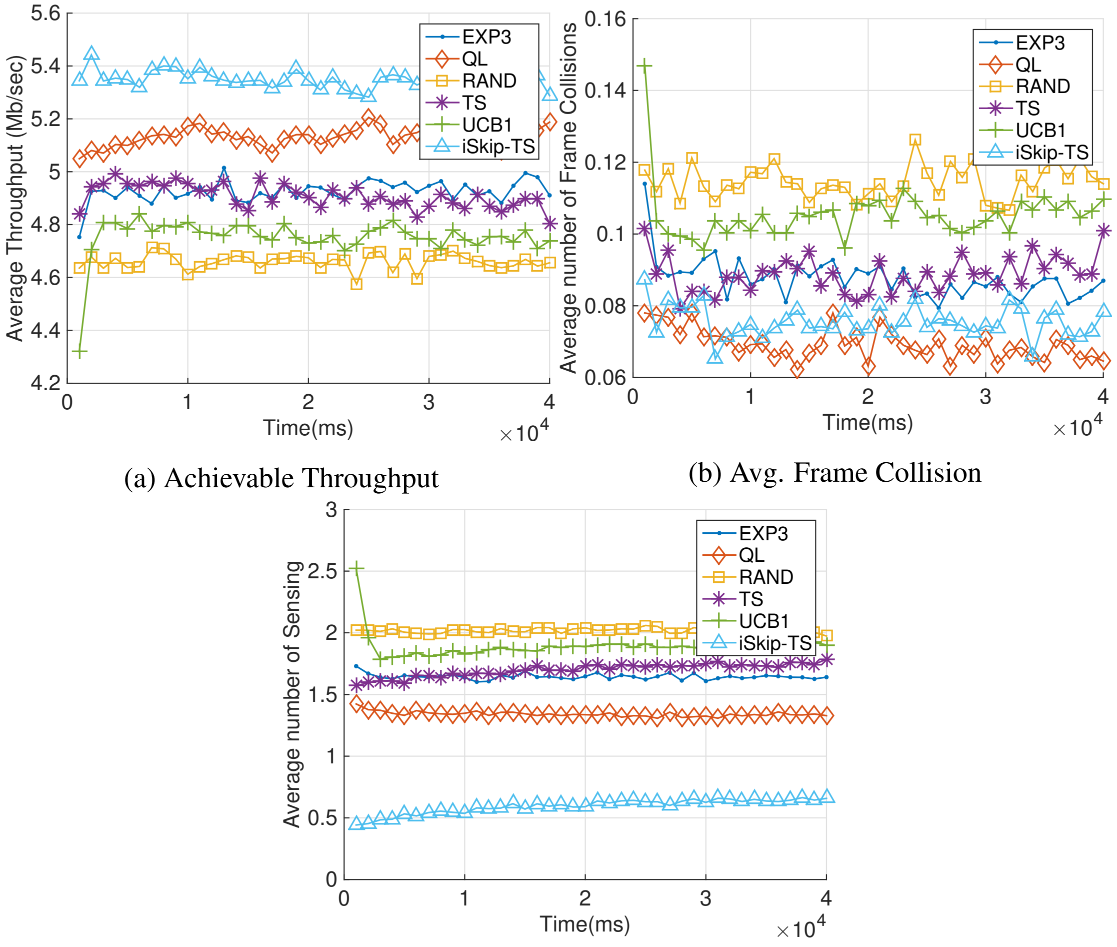
5.2 Simulation Results

5.2.1 Continuous Time Model

The simulation results for the continuous time model are given in Figures 5.1 and 5.2. As can be seen from the graphs, the proposed algorithm performs well in terms of throughput and the number of sensing compared to all other algorithms considered. The number of sensing required is only once in two frames while the number of frame collisions is only around 8% as seen from Figure 5.1b. Even though the number of frame collisions is slightly lower for Q Learning, it loses out in throughput and number of sensing required, which is clearer better for the proposed approach by a fair margin.

Another observation is that, the UCB1 algorithm seems to take more time to learn when the frame duration is 100ms. But in the 50ms case when double the number of samples are available it learns and converges faster.

The simulation using a GPD model represents a stationary environment as the parameters of the model do not vary with time. The proposed method of approximating the GPD model used for simulating the channel with an exponential distribution gives good results in terms of throughput and number of sensing. The proposed algorithm suggests a clearly better sensing scheme without having to sense in each frame without losing out on throughput.

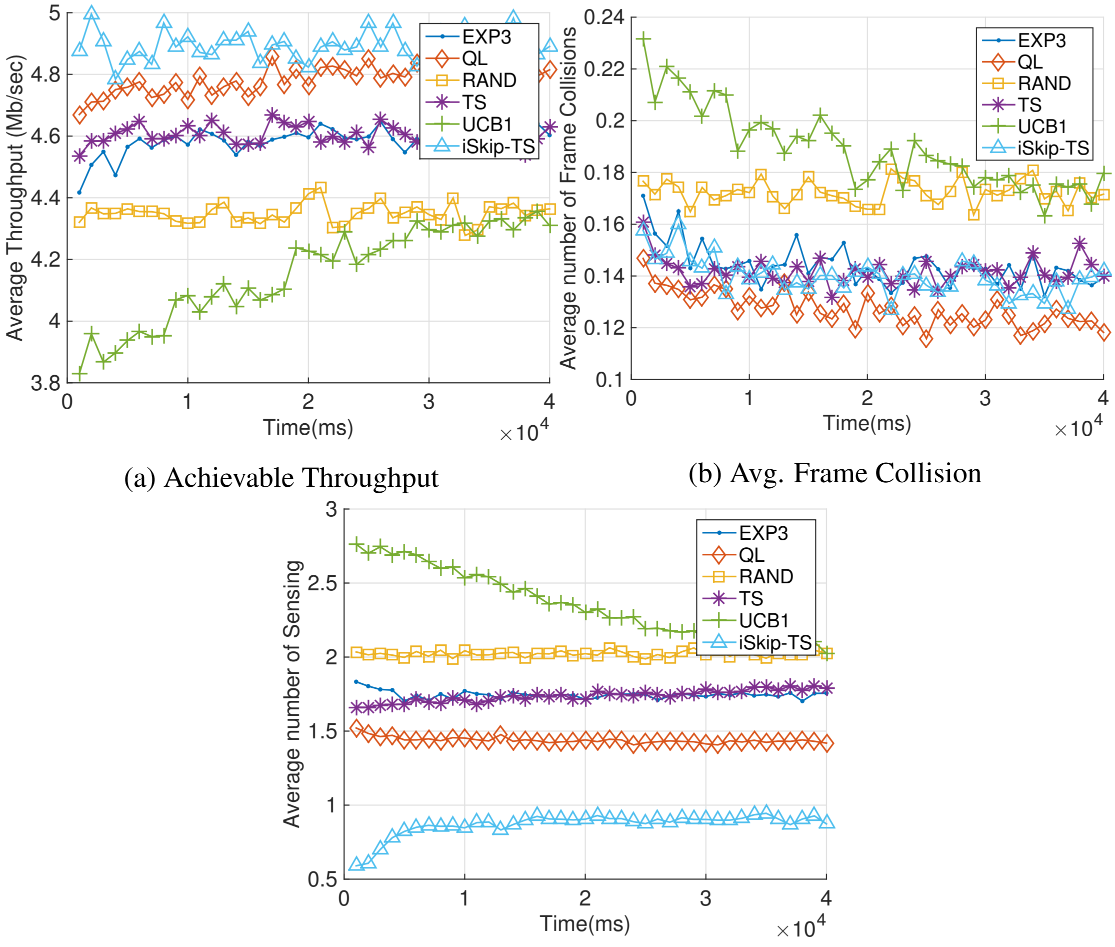


(a) Achievable Throughput

(b) Avg. Frame Collision

(c) Avg. No.of Sensing per frame

Figure 5.1: Plots for frame size 50ms GPD model



(a) Achievable Throughput

(b) Avg. Frame Collision

(c) Avg. No.of Sensing per frame

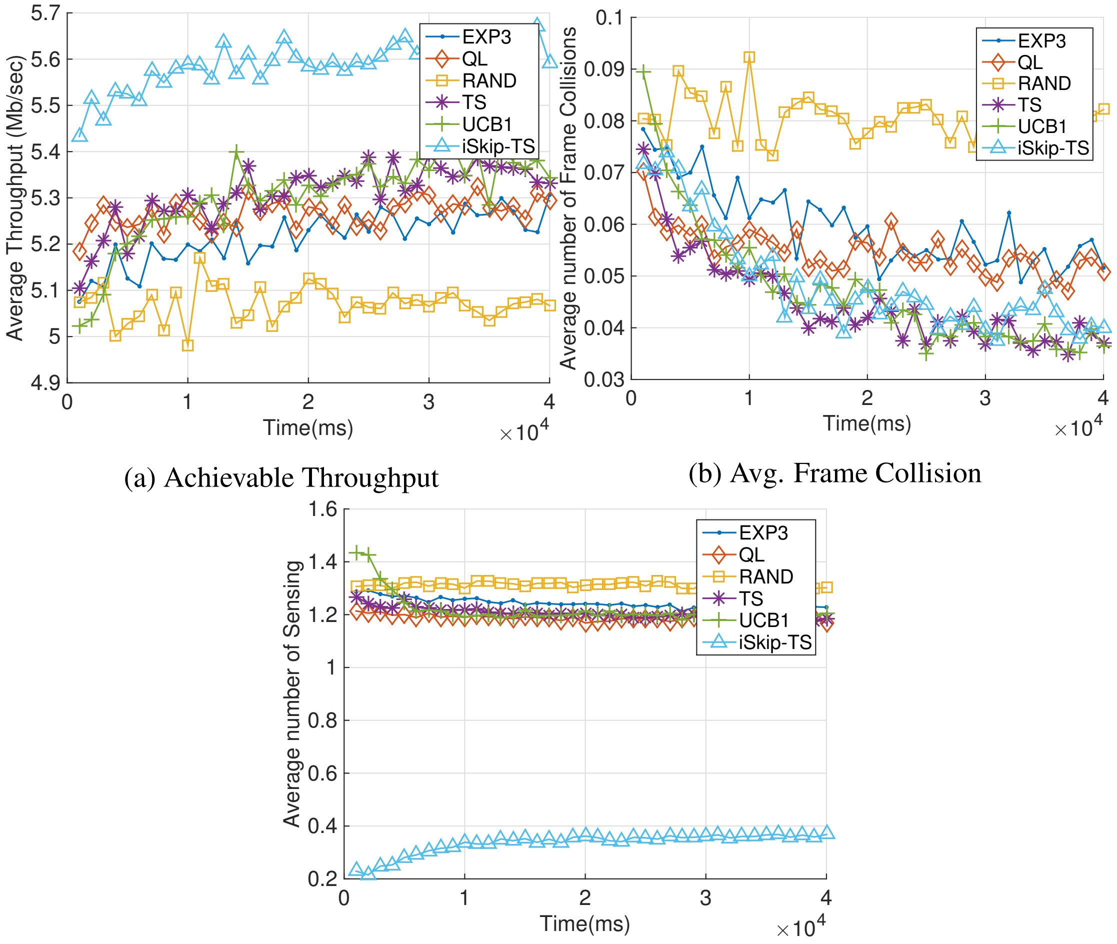
Figure 5.2: Plots for frame size 100ms GPD model

5.2.2 Discrete Time Model

The simulation results for the discrete time model are given in Figures 5.3 to 5.6. The learning curves in this case is not as smooth as in the case of the GPD model especially for the medium and high traffic models as observed in 5.4a and 5.4d. This could be due to the following reason - the proposed algorithm assumes an exponential traffic model for primary users, but the underlying model used for simulating traffic is not. However, it is seen that the algorithm eventually learns some trend in the traffic. And the Thompson Sampling algorithm used for channel selection takes care of the non-stationarity in the environment inherently, but it takes a longer duration to converge.

As is expected the throughput decreases with increase in traffic intensity and more number of channel sensing is required before a free channel can be found for transmission. This is because with increase in traffic the channels will be mostly occupied. But irrespective of the traffic intensity, the proposed approach gives better throughput with less number of sensing. Another interesting observation is the gain margin for the proposed approach in the number of sensing in the low traffic model scenario compared to other algorithms (Figure 5.3). When the frame duration is $50ms$ the learning curves seems to be better compared to the $100ms$ frame, possibly because of more frequent updation of learning algorithms. All the learning algorithms have a performance similar to the random algorithm initially, but it improves with time. When the frame duration is $50ms$ algorithms like UCB and EXP3 seem to have lower number of frame collision. But, our algorithm performs better when considering throughput and number of sensing.

The number of channel sensing required per frame is less than 0.5 for the low traffic DTMC model(Figure 5.3 and Figure 5.5c) which implies sensing only one channel in two frames before finding a vacant channel. Our proposed approach performs well even for a high traffic model even though the gain margin is not very high.

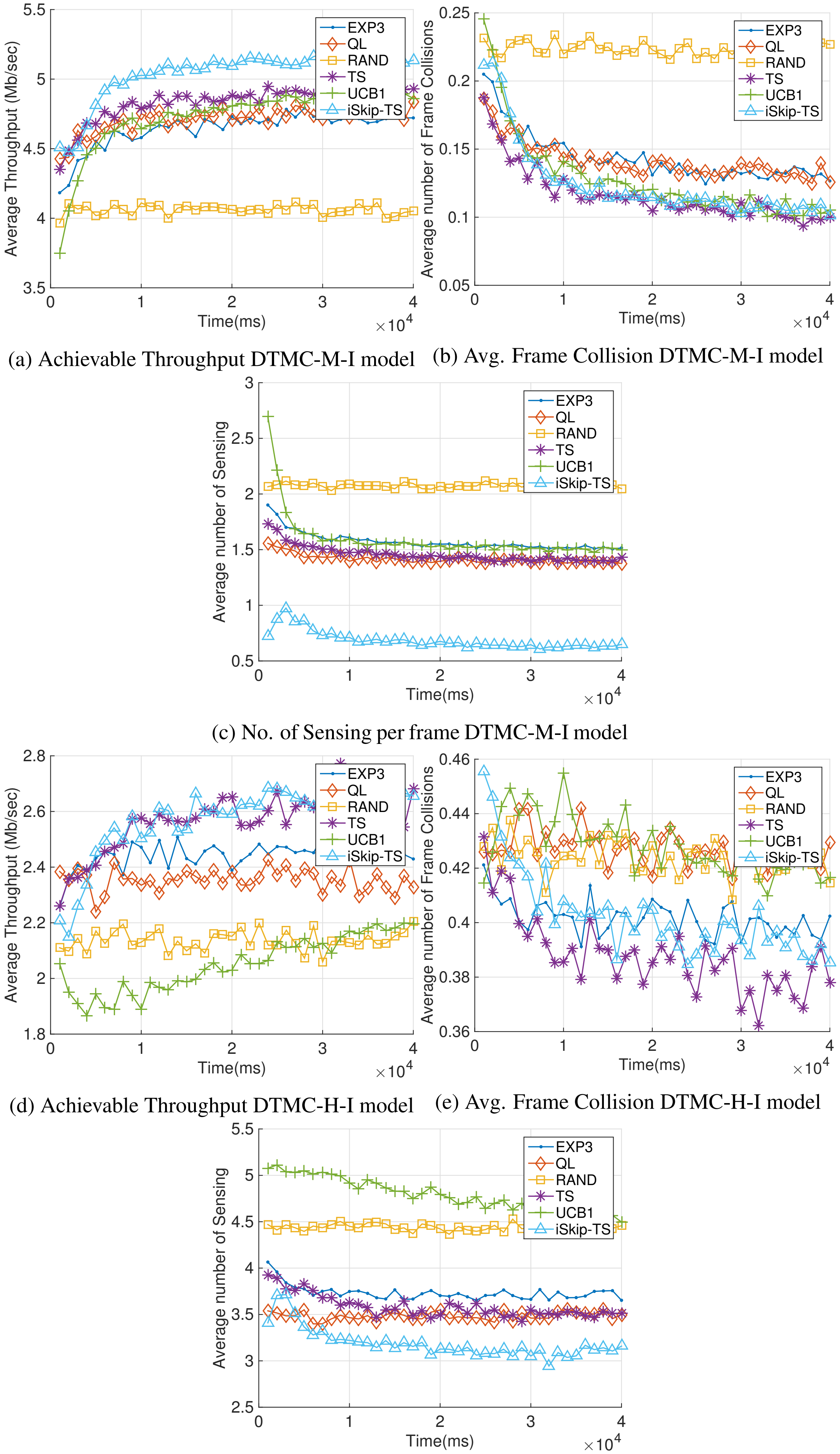


(a) Achievable Throughput

(b) Avg. Frame Collision

(c) Avg. No. of Sensing per frame DTMC-L-I model

Figure 5.3: Plots for frame size 100ms DTMC-L-I model



(a) Achievable Throughput DTMC-M-I model

(b) Avg. Frame Collision DTMC-M-I model

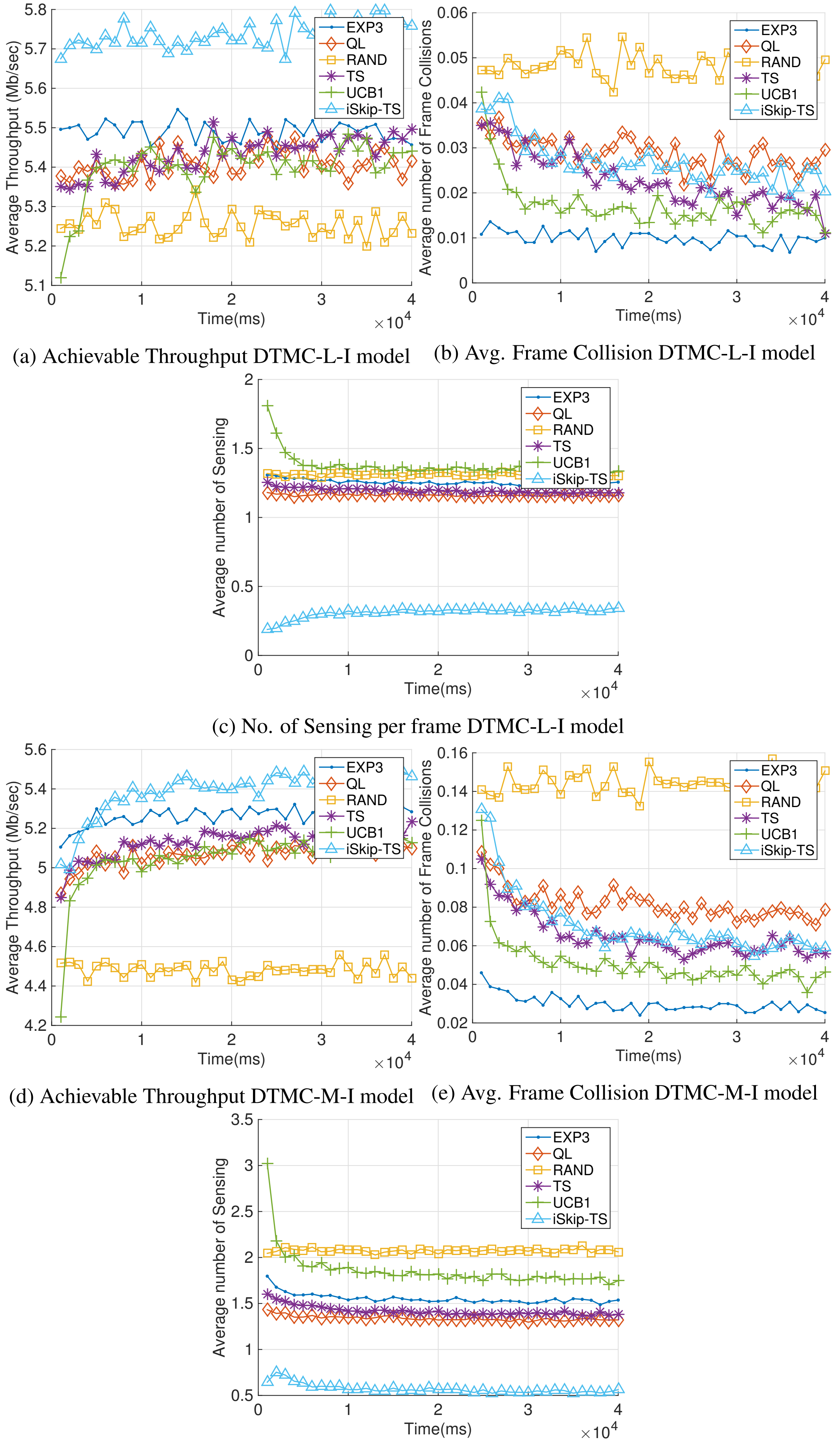
(c) No. of Sensing per frame DTMC-M-I model

(d) Achievable Throughput DTMC-H-I model

(e) Avg. Frame Collision DTMC-H-I model

(f) No. of Sensing per frame DTMC-H-I model

Figure 5.4: Plots for frame size 100ms



(a) Achievable Throughput DTMC-L-I model

(b) Avg. Frame Collision DTMC-L-I model

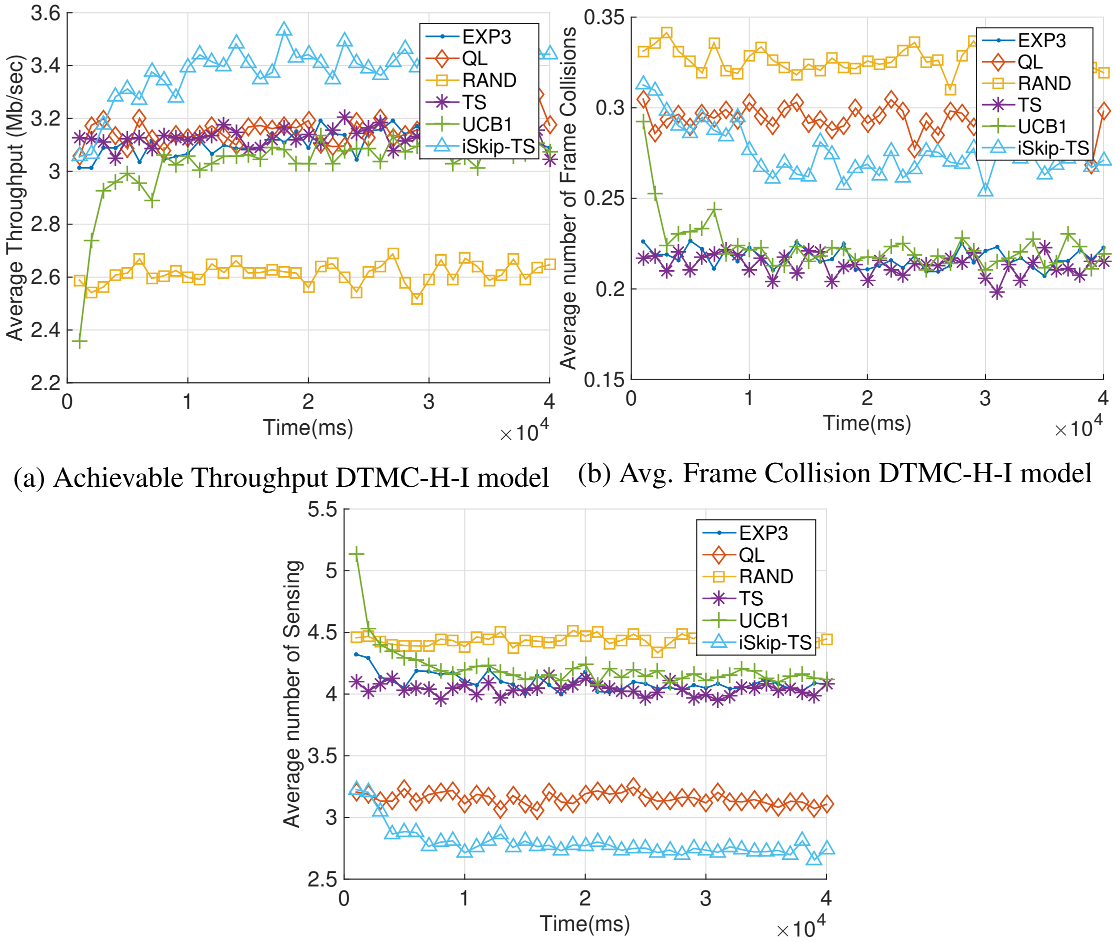
(c) No. of Sensing per frame DTMC-L-I model

(d) Achievable Throughput DTMC-M-I model

(e) Avg. Frame Collision DTMC-M-I model

(f) No. of Sensing per frame DTMC-M-I model

Figure 5.5: Plots for frame size 50ms



(a) Achievable Throughput DTMC-H-I model

(b) Avg. Frame Collision DTMC-H-I model

(c) Avg. No. of Sensing per frame DTMC-H-I model

Figure 5.6: Plots for frame size 50ms

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The importance of Cognitive Radio in the present day world has increased with the ever increasing demand for spectrum resources. Especially with the standardization of 5G and the emergence of Internet of Things(IoT) that envisions everything interconnected with wireless links, ideas like CR and it's improvement gains importance. The proposed approach provides improvement to the performance of CR in terms of reducing the number of sensing required and increased throughput. This work only considered a single secondary user as a starting point. The performance of this algorithm in a multi-user scenario with and without coordination between users is an interesting extension to this work. Similarly, study of the algorithm as a function of number of channels and higher traffic load would be interesting.

REFERENCES

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[2] T. E. Bogale, L. Vandendorpe, and L. B. Le, "Sensing Throughput Tradeoff for Cognitive Radio Networks with Noise Variance Uncertainty," *9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pp. 435–441, 2014.

[3] Miguel, *Cognitive Radio and its Application for Next Generation Cellular and Wireless Networks*, 2012, vol. 116. [Online]. Available: http://www.springerlink.com/index/10.1007/978-94-007-1827-2

[4] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, 2000.

[5] L. D. Nardis and O. Holland, *Deployment Scenarios for Cognitive Radio*, 2014.

[6] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 1, pp. 772–776 Vol.1, 2004.

[7] S. Senthilmurugan and T. G. Venkatesh, "Optimal Channel Sensing Strategy for Cognitive Radio Networks With Heavy-Tailed Idle Times," vol. 3, no. 1, pp. 26–36, 2017.

[8] M. R. Mili and L. Musavian, "Interference Efficiency : A New Metric to Analyze the Performance of Cognitive Radio Networks," vol. 16, no. 4, 2017.

[9] E. Chu, Y. Peh, Y.-c. Liang, Y. L. Guan, and Y. Zeng, "Optimization of Cooperative Sensing in Cognitive Radio Networks :," *Structure*, no. c, pp. 1–6, 2009.

[10] Miguel López-Benítez and Fernando Casadevall, *Spectrum Usage Models for the Analysis, Design and Simulation of Cognitive Radio Networks*, 2012. [Online]. Available: http://www.springerlink.com/index/10.1007/978-94-007-1827-2

[11] I. A. Akbar, W. V. Tech, W. H. Tranter, and W. V. Tech, "Dynamic Spectrum Allocation in Cognitive Radio Using Hidden Markov Models : Poisson Distributed Case," 2007.

[12] Z. Chen, N. Guo, Z. Hu, and R. C. Qiu, "Channel state prediction in cognitive radio, Part II: Single-user prediction," *Conference Proceedings - IEEE SOUTHEASTCON*, no. 1, pp. 50–54, 2011.

[13] J. Yang and H. Zhao, "Enhanced Throughput of Cognitive Radio Networks by Imperfect Spectrum Prediction," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1738–1741, 2015.

[14] W. Jouini, D. Ernst, C. Moy, and J. Palicot, “Multi-armed bandit based policies for cognitive radio’s decision making issues,” *3rd International Conference on Signals, Circuits and Systems, SCS 2009*, pp. 1–6, 2009.

[15] J. Ai and A. A. Abouzeid, “Opportunistic spectrum access based on a constrained multi-armed bandit formulation,” *Journal of Communications and Networks*, vol. 11, no. 2, pp. 134–147, 2009.

[16] H. Li, “Multi-agent Q-Learning of Channel Selection in Multi-user Cognitive Radio Systems : A Two by Two Case,” no. October, pp. 1–6, 2009.

[17] F. Hou, X. Chen, H. Huang, and X. Jing, “Throughput Performance Improvement in Cognitive Radio Networks Based on Spectrum Prediction,” 2016.

[18] X. Xiaoshuang and T. Jing, “Spectrum Prediction in Cognitive Radio Networks,” *IEEE Wireless Communication*, no. April, pp. 90–96, 2013.

[19] Ying-Chang Liang, Yonghong Zeng, E. Peh, and Anh Tuan Hoang, “Sensing-Throughput Tradeoff for Cognitive Radio Networks,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 660–665, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4489760

[20] Y. Pei, “Sensing-Throughput tradeoff in cognitive radio networks : How frequently should spectrum sensing be carried out?” no. 1, pp. 0–4.

[21] Y. Zhao, S. Mao, J. O. Neel, and J. H. Reed, “Performance Evaluation of Cognitive Radios: Metrics, Utility Functions, and Methodology,” *Special Issue on Cognitive Radio, Proceedings of the IEEE*, vol. 97, no. 4, pp. 642–659, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4812016

[22] M. Abdulsattar and Z. Hussein, “Energy detection technique for spectrum sensing in cognitive radio: a survey,” *International Journal of Computer Networks & Communications (IJCNC)*, vol. 4, no. 5, pp. 223–242, 2012. [Online]. Available: http://www.airccse.org/journal/cnc/0912cnc14.pdf

[23] U. G and A. P. Kannu, “Throughput optimal multi-slot sensing procedure for a cognitive radio,” *IEEE Communications Letters*, vol. 17, no. 12, pp. 2292–2295, 2013.

[24] S. Filippi, O. Cappé, and A. Garivier, “Optimally sensing a single channel without prior information: The Tiling Algorithm and regret bounds,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 68–76, 2011.

[25] J. Oksanen and V. Koivunen, “An order optimal policy for exploiting idle spectrum in cognitive radio networks,” *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1214–1227, 2015.

[26] M. López-Benítez and F. Casadevall, “Spectrum usage models for the analysis, design and simulation of cognitive radio networks,” in *Cognitive radio and its application for next generation cellular and wireless networks*. Springer, 2012, pp. 27–73.

[27] K.-l. A. Yau, H. Guan, D. Chieng, and K. Hsiang, “Application of reinforcement learning to wireless sensor networks : models and algorithms,” *Computing*, vol. 97, no. 11, pp. 1045–1075, 2015.

[28] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, “An application of reinforcement learning to aerobatic helicopter flight,” *Education*, vol. 19, p. 1, 2007. [Online]. Available: http://books.google.com/books?hl=en{&}lr={&}id=Tbn1l9P1220C{&}oi=fnd{&}pg=PA1{&}dq=An+Application+of+Reinforcement+Learning+to+Aerobatic+Helicopter+Flight{&}ots=V2maFfjsZY{&}sig=bdOOVtReU3qVDC6hGkHYdgkZKBw

[29] A. G. Barto and R. H. Crites, “Improving Elevator Performance Using Reinforcement Learning,” *Advances in Neural Information Processing Systems*, vol. 8, pp. 1017–1023, 1996. [Online]. Available: http://cseweb.ucsd.edu/users/gary/CSE190/crites-barto.pdf{%}5Cnhttp://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.5519

[30] A. Mukherjee and A. Hottinen, “Learning Algorithms For Energy Efficient MIMO Antenna Subset Selection : Multi-Armed Bandit Framework,” no. Eusipco, pp. 659–663, 2012.

[31] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press Cambridge, 1998, vol. 1, no. 1.

[32] ——, “Reinforcement Learning:An Introduction,” *MIT Press,Cambridge*, 1998. [Online]. Available: https://books.google.com/books?id=CAFR6IBF4xYC{&}pgis=1{%}5Cnhttp://incompleteideas.net/sutton/book/the-book.html{%}5Cnhttps://www.dropbox.com/s/f4tnuhipchpkgoj/book2012.pdf

[33] C. J. C. H. Watkins and P. Dayan, “Technical Note: Q-Learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[34] F. S. Melo, “Convergence of Q -learning : a simple proof,” pp. 1–4.

[35] C. Claus and C. Boutilier, “The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems,” 1998.

[36] P. Auer, “Using Confidence Bounds for Exploitation-Exploration Trade-offs,” vol. 3, pp. 397–422, 2002.

[37] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[38] W. R. Thompson, “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples,” *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[39] S. Agrawal and N. Goyal, “Analysis of thompson sampling for the multi-armed bandit problem.” in *COLT*, 2012, pp. 39–1.

[40] N. Morozs, T. Clarke, and D. Grace, “Heuristically accelerated reinforcement learning for dynamic secondary spectrum sharing,” *IEEE Access*, vol. 3, pp. 2771–2783, 2015.

[41] Y. Gwon, S. Dastangoo, and H. Kung, "Optimizing media access strategy for competing cognitive radio networks," in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 1215–1220.

[42] X. Zhang, L. Jiao, O.-c. Granmo, and B. J. Oommen, "Channel Selection in Cognitive Radio Networks : A Switchable Bayesian Learning Automata Approach," pp. 2362–2367, 2013.

[43] J. Zhu, Y. Song, D. Jiang, and H. Song, "Multi-Armed Bandit Channel Access Scheme with Cognitive Radio Technology in Wireless Sensor Networks for the Internet of Things," *IEEE Access*, vol. PP, no. 99, p. 1, 2016.