Spectrum hole identification and exploitation in cognitive radio networks

Vamsi Krishna Tumuluru

2011

Vamsi Krishna Tumuluru. (2011). Spectrum hole identification and exploitation in cognitive radio networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46302>

<https://doi.org/10.32657/10356/46302>

Spectrum Hole Identification and Exploitation in Cognitive Radio Networks

Vamsi Krishna Tumuluru

School of Computer Engineering

A thesis submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2011

Abstract

Cognitive radio networks are envisaged as the solution for effective spectrum management for the future communications. In a cognitive radio network, the unlicensed users are permitted to access the unused portions of the licensed spectrum without interfering with the licensed user services. Two mechanisms are instrumental in achieving the above objective, namely, spectrum sensing and dynamic spectrum access. The spectrum sensing mechanism enables the unlicensed users to identify the unused portions of the licensed spectrum (referred as idle channels or *spectrum holes*) whereas the dynamic spectrum access mechanism facilitates sharing and access of the detected spectrum holes among the unlicensed users. While the spectrum sensing is a physical layer issue, the dynamic spectrum access is a medium access control (MAC) issue. This dissertation aims to address some critical problems related to the design of the cognitive radio networks. Specifically, this dissertation encapsulates the original research contributions made in the course of our research on the cognitive radio networks. The focus of this research is mainly with the MAC layer of the cognitive radio networks. The key MAC layer issues addressed in this dissertation are given below.

The first issue is at the sensing overhead which can be expressed as the energy wasted by the sensing mechanism in sensing the busy portions (i.e., busy channels) of the licensed spectrum. This issue is tackled by using novel channel status prediction schemes. Different from the existing schemes, channel status prediction schemes proposed in this dissertation do not require a priori knowledge of the licensed user's traffic characteristics. We show that our prediction schemes greatly reduce the sensing energy and also improve spectrum utilization significantly.

The second issue is at the scheduling overhead which can be expressed as the amount of control information exchanged to allocate the available channels to the unlicensed users in centralized CRN (i.e., infrastructure-based CRN). We emphasize the importance of addressing this problem and propose a queue-aware frame-based opportunistic spectrum scheduling scheme in this direction. We show that the proposed scheduling scheme incurs significantly lower scheduling overhead and greatly improves the throughput of the unlicensed users compared to the existing scheduling schemes.

For the third problem, it is important to consider that the unlicensed user traffic can have different service requirements (i.e., priorities) and hence it is necessary to design the dynamic spectrum access schemes accordingly. Therefore, we propose various dynamic spectrum access schemes which perform admission control and spectrum handoff according to the priority of unlicensed users under different networking scenarios (e.g., centralized and distributed).

The above problems are critical to the design of the CRN as they affect the efficiency of the sensing mechanism and that of the dynamic spectrum access scheme. While some works in the literature have attempted to address the issue of sensing overhead, none of the existing works have considered the issues of scheduling overhead and prioritized spectrum handoff. Hence, the research findings presented in this dissertations add great value and enhance the performance of the cognitive radio networks. Therefore, in this dissertation, we describe different research problems in the MAC layer of the CRN and explain our approach in solving these problems along with extensive performance analysis and simulations.

Acknowledgments

I express my sincere thanks to my advisors Dr. Ping Wang and Dr. Dusit Niyato for their valuable guidance and cooperation during my research. I also thank them for their accessibility and openness to various discussions that I had over the course of my association with them. It has been an enriching and a very enjoyable experience working under both of them.

I express my gratitude to Dr. Amitabha Das, my ex-advisor, for his valuable guidance and encouragement which laid the foundation for my research. I also express special thanks to Prof. Sundararajan, EEE, NTU for his precious time and suggestions during the initial stages of my work.

Finally, I am thankful to all my friends in CeMNeT laboratory (SCE, NTU), especially our team members for creating a cheerful environment to carry out my research.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Research Objective	1
1.2 Motivation and Background	1
1.2.1 Cognitive Radio Networks: An Overview	4
1.3 Research Focus	6
1.3.1 Channel Status Prediction	6
1.3.2 Opportunistic Spectrum Scheduling	7
1.3.3 DSA Policies for Prioritized SU Traffic	7
1.4 Organization of the Chapters	8
2 Literature Review and Background Theory	9
2.1 MAC-based Spectrum Sensing: Related Work	9
2.1.1 Cooperative Sensing Schemes	10
2.1.2 Reward-based Sensing Schemes	10
2.1.3 Prediction-based Sensing Schemes	11
2.2 Dynamic Spectrum Access schemes: Related Work	12
2.2.1 Classification of CRN Architecture	13
2.2.2 DSA schemes for Centralized CRN	16
2.2.3 DSA schemes for Distributed CRN	18
3 Channel Status Prediction for Cognitive Radio Networks	21
3.1 Research Highlights	21
3.2 Neural Network based Prediction	22
3.2.1 MLP Predictor Design	22
3.2.2 MLP Predictor Training	23

3.3	Channel Status Prediction using HMM	28
3.3.1	Hidden Markov Model	28
3.3.2	HMM Predictor	29
3.4	Simulation and Analysis	34
3.4.1	Performance of the Predictors Under Stationary Traffic Conditions .	35
3.4.2	Performance of the Predictors Under Non-stationary Traffic Conditions	38
3.4.3	Performance Enhancement due to Channel Status Prediction	40
3.5	Implementation Issues	43
3.5.1	Model Complexity of MLP Predictor	43
3.5.2	Computational Complexity of MLP Predictor	44
3.5.3	Model Complexity of HMM Predictor	44
3.5.4	Computational Complexity of HMM Predictor	45
3.6	Qualitative Comparison	46
3.6.1	Performance Comparison	46
3.6.2	Design Comparison	47
3.7	Summary	47
4	Queue-aware Frame-based Opportunistic Scheduling Scheme for CRN	48
4.1	Research Highlights	48
4.2	System Model and Assumptions	49
4.3	Opportunistic Spectrum Scheduling Scheme	51
4.3.1	Estimation of Packet Departures	52
4.3.2	Scheduling Algorithm	56
4.4	Analytical Framework	57
4.4.1	Calculation of State Transition Matrix	58
4.4.2	Performance Measures	62
4.5	Results and Discussion	63
4.5.1	Comparison of Simulation and Analysis	63
4.5.2	Selecting the Optimal Frame Size	68
4.5.3	Comparison of the Proposed Scheduling Scheme with Slot-based Scheduling	69
4.6	Summary	73
5	Dynamic Spectrum Access schemes for Prioritized SU Traffic	74
5.1	Research Highlights	74
5.2	System Model	75
5.2.1	DSA schemes for Centralized CRN	77
5.2.2	DSA schemes for Distributed CRN	79

5.3	Performance Analysis	81
5.3.1	Analytical Models for Centralized CRN	81
5.3.2	Analytical Models for Distributed CRN	87
5.3.3	Performance Measures	90
5.4	Results and Discussion	93
5.4.1	Throughput	94
5.4.2	Mean Handoff Delay	97
5.4.3	Optimal Sub-channel Reservation	99
5.4.4	Summary of Performance Analysis	100
5.5	Summary	100
6	Conclusions and Future Work	101
6.1	Summary of the Contributions	101
6.1.1	Reducing the Spectrum Sensing Overhead	101
6.1.2	Reducing the Scheduling Overhead in Centralized CRN	102
6.1.3	Call Admission and Spectrum Handoff for Prioritized SU Traffic . .	102
6.2	Future Research	103
6.2.1	CRN for Real PU Scenarios	103
6.2.2	Research Challenges for Centralized CRN	103
6.2.3	Research Challenges for Distributed CRN	104
	Publications	105
A	List of Abbreviations	106
B	List of Symbols	107
	Bibliography	115

List of Figures

1.1	U.S. frequency allocations.	3
1.2	Measurement of 0 – 6 GHz spectrum utilization at Berkeley Wireless Research Center.	4
2.1	Classification of CRN based on network topology.	13
2.2	Centralized CRN.	14
3.1	MLP predictor training.	24
3.2	HMM predictor training.	30
3.3	Plot of the log likelihood probability $\log(P(\mathbf{s}_{hmm}^\kappa \cdot))$ at each iteration during HMM training.	33
3.4	Performance of the MLP predictor in predicting the busy channel status for various traffic scenarios.	36
3.5	Overall performance of the MLP predictor in predicting the channel status for various scenarios.	36
3.6	Performance of the HMM predictor in predicting the busy channel status for various traffic scenarios.	38
3.7	Overall performance of the HMM predictor in predicting the channel status for various scenarios.	39
4.1	System model: Queue-aware frame-based opportunistic spectrum scheduling scheme.	50
4.2	Markov chain evolution for state $\mathbf{x}_n^{u,v}$ inside a frame.	52
4.3	Structure of the frame in the proposed scheduling scheme.	56
4.4	Convergence of probability of queue size $q^1 = q'$ (i.e., $\Pi_{q^1=q'}$) for different values of q'	64
4.5	Mean queue size for different mean arrival rates.	65
4.6	Mean waiting time for a packet in the queue for different mean arrival rates.	66
4.7	Average SU throughput for different mean arrival rates.	67
4.8	Blocking probability for different mean arrival rates.	67
4.9	Average SU throughput under different frame size N_{slots} for various R_c	68

4.10	Average channel throughput $\bar{T}$ under different frame sizes N_{slots} for various network settings.	70
4.11	Blocking probability under different frame sizes N_{slot} for $V = 8$ users, $N_c = 2$ channels and varying Q_v	72
5.1	System model.	75
5.2	State transitions for centralized DSA schemes without buffer.	83
5.3	State transitions for centralized DSA schemes with buffer, $l_q = 0$ and $m_q = 0$	86
5.4	State transitions for centralized DSA schemes with buffer, $l_q + m_q > 0$	86
5.5	State transitions for the DSA-D1 scheme.	87
5.6	State transitions for the DSA-D2 scheme.	90
5.7	Effect of PU arrival rate on SU_1 throughput, under all DSA schemes.	94
5.8	Effect of PU arrival rate on SU_2 throughput, under all DSA schemes.	95
5.9	Effect of PU arrival rate on forced termination probability P_{F1} , under DSA schemes without buffer.	96
5.10	Effect of PU arrival rate on forced termination probability P_{F2} , under DSA schemes without buffer.	96
5.11	Effect of PU arrival rate on the mean handoff delay of SU_1 calls, under DSA schemes with buffer.	98
5.12	Effect of PU arrival rate on the mean handoff delay of SU_2 calls, under DSA schemes with buffer.	98
5.13	Simulation: Optimal sub-channel reservation under DSA-C1 scheme.	99

List of Tables

3.1	Performance of the MLP predictor in predicting the idle channel status . .	37
3.2	Performance of the HMM predictor in predicting the idle channel status . .	38
3.3	Example of a licensed channel with non-stationary traffic distribution . . .	39
3.4	Different channel models for the primary user system	42
3.5	Percentage improvement in the spectrum utilization due to channel status prediction	42
3.6	Percentage reduction in sensing energy for different traffic intensities ρ and different mean inter-arrival times t_{inter}	43
3.7	Performance of HMM predictor for different number of states N_S	45
3.8	Qualitative comparison of the two channel status prediction schemes	46
4.1	Slot dependent parameter b_n for different channel settings in the network .	69

Chapter 1

Introduction

1.1 Research Objective

Spectrum is a vital and scarce resource, and hence it should be managed carefully. This dissertation aims to address key issues in the design of *cognitive radio networks* (CRNs) which are fast emerging as an effective and efficient solution for spectrum management for the future communications. Specifically, this dissertation deals with our original research contributions aimed to enhance the capabilities of the medium access control (MAC) layer of the CRN.

1.2 Motivation and Background

In the recent years, there has been tremendous increase in the demand for spectrum due to the rapid proliferation of wireless technologies and wireless markets. For example, the Internet broadband markets are expanding despite the limited spectrum (around the 2.4 GHz band) due to the competition between the 3G cellular networks and the WiFi networks [1]. Further, various service sectors such as healthcare, education, transportation systems, home automation and power grids have also began investing on wireless communications to enhance their service delivery and to optimize their operational performance according to their consumer demographics. According to a recently published report by the National Science Foundation (NSF) [2], the number of wireless devices is expected to reach 100 billion devices by the year 2025. However, sufficient spectrum is not available

under the current static spectrum allocation regime to handle such large scale wireless communications as most of the spectrum has been licensed for exclusive use (i.e., reserved for the government entities or sold to non-government entities). For example, in Fig. 1.1 [3] which shows the spectrum allocation in the U.S., the spectrum between 54 – 88 MHz and 117 – 212 MHz have been licensed to TV broadcast operators (who are non-government enterprises). Such trends endanger the quality-of-service (QoS) that can be achieved for the future wireless services. According to the recently published report [4] by the Federal Communications Commission (FCC), the spectrum regulatory authority in the United States, the traffic distribution across the radio spectrum is extremely uneven. It is observed that the unlicensed portions of the spectrum (2.4GHz and 5GHz bands) in which most wireless networks operate (e.g. WiFi, WiMaX), are heavily occupied whereas the licensed portions of the spectrum (e.g. UHF band) are sporadically used. The spectrum utilization in the licensed spectrum was observed to vary between 15% to 85% at different geographical regions and time zones [4] (e.g., spectrum measurements at Berkeley, Fig. 1.2 [5], show low utilization of spectrum above 3 GHz). The above observations motivate the design of wireless networks in which the unlicensed users explore and exploit unused portions of the licensed spectrum while safeguarding the transmissions of the licensed users.¹ Such wireless networks are referred to as the *cognitive radio networks* (CRNs), [6]. The CRNs offer the capability for the future communications through the following provisions:

- (1) Alleviate the problems faced due to spectrum scarcity in the wireless system
- (2) Inspire the growth of cutting edge technologies and networks which can dynamically optimize the QoS requirements of the wireless users
- (3) Generate new sources of revenue (e.g., by opening wireless markets in the licensed spectrum, like UHF band of the TV spectrum).

¹A licensed user is a device which legitimately operates in a licensed spectrum.

Recently, the FCC approved the deployment of CRNs in the TV spectrum, subject to specific operating constraints which were determined based on a series of experiments [7]. Similar experiments were conducted to classify other portions of the licensed spectrum for the deployment of CRNs in the future [8–12].

Figure 1.2: Measurement of 0 – 6 GHz spectrum utilization at Berkeley Wireless Research Center.

1.2.1 Cognitive Radio Networks: An Overview

In the CRN, the licensed spectrum is logically divided into several licensed channels (i.e. frequency bands) which can be specified by their respective center frequency and bandwidth. The licensed users have higher priority in accessing the channels² compared to the unlicensed users. For this reason, the licensed users are also called *primary users* (PUs) while the unlicensed users are called *secondary users* (SUs). The channels not used by the primary users are called *spectrum holes* or *white spaces* [13]. The availability of spectrum holes varies both temporally and spatially. This phenomenon is termed as *spectrum*

²A licensed channel will be simply referred to as a channel in the rest of the report.

heterogeneity [13]. Spectrum heterogeneity complicates coordination among the secondary users during data transmission and affects network connectivity. Therefore, unlike the conventional wireless networks which operate in a homogeneous physical medium, the CRN design should consider the fluctuating availability of the spectrum holes. Due to this reason, the wireless devices and protocols used in conventional wireless networks (e.g. WiFi and WiMaX) are not directly applied to a CRN scenario.

Spectrum heterogeneity can be tackled by adding two functions to a conventional wireless device, namely *spectrum sensing* (to detect the spectrum holes) and *spectrum hole management*. Both *spectrum sensing* and *spectrum hole management* are active research areas in CRN. Several spectrum sensing schemes are available in the literature, e.g., [14–17]. A reliable spectrum sensing scheme should have low probability of missed detection (i.e. wrongly detecting the channel as idle while the PU transmission is present) and low probability of false alarm (i.e. wrongly detecting the channel as busy while there is no PU activity). Spectrum hole management is handled by the medium access control (MAC) layer. The functions of the MAC layer are given as follows:

- MAC-based spectrum sensing
- Dynamic spectrum access (DSA)

The MAC-based spectrum sensing deals with the planning and management of the sensing device using MAC layer protocols (e.g. channel selection during sensing). The DSA scheme facilitates the sharing of the detected spectrum holes among the secondary users. The DSA scheme is implemented either as a scheduling algorithm (in an infrastructure-based network) or as a MAC protocol (in an ad hoc network). A detailed review on the DSA policies for various CRN architectures was published in [18]. Similarly, the higher layer protocols (routing, transport, etc.) are also modified to address spectrum heterogeneity [19].

1.3 Research Focus

In this dissertation, we focus on the research problems pertaining to the MAC layer of the CRN. Specifically, the following research problems were undertaken.

- (1) Channel status prediction
- (2) Opportunistic spectrum scheduling
- (3) DSA policies for prioritized SU traffic

1.3.1 Channel Status Prediction

In most CRNs, a SU is equipped with a single transceiver which can sense/access only one channel at a time due to the hardware constraint [20]. On the other hand, due to the energy constraint, the secondary users may not have the willingness to waste energy to sense the part of spectrum which is very likely to be busy. Further, it takes a significant time to sense all the channels in order to determine the spectrum holes within a slot (the longer the sensing time, the better is the sensing accuracy, e.g. [21]). Therefore, the key issue is to let the secondary users efficiently and effectively sense the channels in the licensed spectrum without wasting much energy. The sensing energy and sensing time can be treated as *sensing overhead*.

A MAC-based spectrum sensing scheme can alleviate the sensing overhead. In this line of work, the channel status prediction is one of the emerging approaches. Consider a slotted transmission scenario in which the PU status (idle/busy) remains constant over the slot duration. A SU predicts the channel status based on the past sensing results of that channel (i.e. sensing history). In a given slot, the SU senses the channel only if the predicted status is idle. For this purpose, two novel channel status prediction schemes were analyzed as part of our research. One of the prediction schemes uses neural networks whereas the other prediction scheme is based on hidden Markov models (HMM). Further details are provided in Chapter 3.

1.3.2 Opportunistic Spectrum Scheduling

The most elegant way of designing a CRN is by introducing a central controller (e.g., infrastructure like base stations) which can monitor and govern the actions of the secondary users. Such infrastructure-based CRN is called centralized CRN. Consider a scenario in which the PU activity changes on a slot-by-slot basis. In every slot, the SUs contact the central controller and obtain the schedule for their data transmissions on the available channels (i.e., spectrum holes). This communication constitutes the scheduling overhead. In a centralized CRN, the scheduling overhead is a cause for major concern because it reduces the efficiency of DSA scheme. The scheduling overhead grows with the number of channels and with the number of SUs in the CRN. As most CRNs assume a low bandwidth common control channel for communicating the schedule information (e.g., [22]), the effective transmission time on the data channels (i.e. spectrum holes whose bandwidth is relatively large compared to the common control channel) gets reduced, resulting in lower throughput.

To tackle the scheduling overhead and improve the SU throughput, a novel queue-aware frame-based opportunistic spectrum scheduling scheme was developed as part of our research. Further details are provided in Chapter 4.

1.3.3 DSA Policies for Prioritized SU Traffic

In most CRN, the SU traffic is treated homogeneous. However, in reality the SU traffic can be composed of transmissions having different QoS requirements. In other words, the SU traffic can have transmissions from SUs belonging to different priority classes (e.g. real-time data has higher priority over non real-time data). Hence, one part of our research focused on DSA policies for managing prioritized SU traffic. Under the prioritized SU setting, call³ admission and spectrum handoff (i.e. reassigning new channels to the SU calls which paved way for a new PU call) should take the priority classes of the SUs into account.

³The PU and SU traffic are treated as calls.

To this effect, we developed different DSA policies for centralized (i.e. infrastructure-based) and distributed (i.e. ad hoc) CRN, and each of which manages the prioritized SU traffic differently. Each DSA scheme uses a different spectrum handoff mechanism. To the date, SU priority based spectrum handoff was not considered in the existing literature. Further, very few works have analyzed distributed CRN using the system model described in Chapter 5.

1.4 Organization of the Chapters

The remaining chapters are organized as follows.

The next chapter reviews the research related to the MAC layer of CRN from the literature. Subsequent chapters deal with the key contributions of our research.

Chapter 3 provides detailed insights into the design of the channel status prediction schemes and also demonstrates the importance of channel status prediction to spectrum sensing.

Chapter 4 elaborates the uniqueness of the design and analysis of the queue-aware frame-based opportunistic spectrum scheduling scheme. An analytical model based on discrete time Markov chain (DTMC) is constructed to analyze the system performance. Simulations are carried out to validate the analysis.

Chapter 5 exclusively deals with design and analysis of the DSA schemes developed for managing the prioritized SU traffic. Analytical models based on continuous time Markov chain (CTMC) are constructed to analyze the system performance under various DSA schemes. Simulations are carried out to validate the analysis.

Finally, Chapter 6 provides the conclusion for the dissertation and offers directions for future research.

Chapter 2

Literature Review and Background Theory

This chapter provides a comprehensive overview of the research pertaining to the design of the MAC layer of the CRN, based on the existing research. The literature review is presented along the lines of the MAC layer functions described in the previous chapter, which include

- MAC-based spectrum sensing
- Dynamic spectrum access (DSA)

2.1 MAC-based Spectrum Sensing: Related Work

In the MAC-based spectrum sensing, a MAC protocol controls the usage of the sensing device in a manner to reduce the sensing overhead (i.e., sensing energy). Various types of MAC-based spectrum sensing schemes are found in the literature. For the simplicity of presentation, they are classified into three categories, as given below.

- Cooperative sensing schemes
- Reward-based sensing schemes
- Prediction-based sensing schemes

Although this dissertation deals with only the last category (refer Chapter 3), each of the above categories is explained with examples for the benefit of the readers.

2.1.1 Cooperative Sensing Schemes

The cooperative sensing scheme is mostly applied to scenarios in which the PU activity remains constant for a fixed time duration called a slot [17, 23–26]. The secondary users (SU) are assumed to be aware of the slot structure and hence they synchronize their actions according to the slot. The SUs implement a MAC protocol which includes a sensing period during which all the SUs randomly sense the channels and share the sensing results with one another on a common control channel. Thereby, the SUs gain knowledge of the spectrum holes in the slot quickly compared to individually performing full sensing of the licensed spectrum.

Zhang *et al* [17] proposed a cooperative sensing scheme which improved the reliability of the sensing mechanism through a process of voting among the SUs based on their sensing results. The optimal sensing scheme for a CRN with large number of SUs was also discussed in [17].

In [23], two cooperative sensing schemes namely random sensing scheme (RSP) and negotiation-based sensing scheme (NSP), were proposed. Under RSP, the SUs randomly sense the channels and report on the common control channel. The RSP was shown to be particularly useful when the number of SUs was more than the number of channels. In the NSP, the SUs only sense among the channels which were not previously reported. Similar distributed sensing schemes were also considered in [24, 25].

Ganesan *et al* [26] also demonstrated that the cooperative sensing minimizes the interference to the PUs. According to [26], due to cooperative sensing, a SU which detects the PU transmission early can alert other SUs in its vicinity and thereby avoid harmful interference to the PU.

2.1.2 Reward-based Sensing Schemes

The reward-based sensing schemes are mostly applied to slotted PU transmission scenario. Under the reward-based sensing, whenever a SU senses an idle channel and successfully accesses it, a reward is accrued. Hence, the objective of the SU is to maximize the total

expected reward for a finite number of slots. For a single SU, the reward-based sensing schemes rely on the statistics of the channel usage by the PUs, such as sensing history and collision history (i.e. number of occasions the SU transmission interfered with the PU) which were collected over a finite time duration. Combining these statistics with their objective function (i.e. maximizing the total expected reward), the reward-based sensing schemes try to determine the optimal channel selection order for sensing. Several reward-based sensing schemes are available in the literature [20, 27–31].

Jia *et al* [20] proposed a reward-based sensing scheme in which a pair of SUs (sender and receiver) defined an optimal stopping criterion for the sensing operation based on the estimated total reward. The total expected reward was set to the maximum bandwidth that could be supported by the hardware of the SU pair. Jiang *et al* [27] also followed a similar approach, however, the total expected reward was based on the bandwidth requirements of the SU as well as the signal-to-noise ratio (SNR) measured for the sensed channels. Motamedi *et al* [28] developed a reward-based sensing scheme which modeled the channel usage by the PUs as a multi-armed bandit problem. Similar approach was also followed by Lai *et al* [29]. However, the sensing scheme in [29] can also be applied if the PU traffic distribution is not known a priori. Zhao *et al* [30] used a partially observable Markov decision process (POMDP, [32]) to model the licensed channel's traffic statistics due to PU activity. In [30], sensing errors were also considered. Further, Zhao *et al* presented a constrained POMDP formulation for the optimal sensing scheme in an unslotted PU transmission scenario [21]. Liu *et al* [31] formulated the sensing scheme as a constrained optimization problem under the assumption of perfect sensing.

2.1.3 Prediction-based Sensing Schemes

Prediction-based sensing schemes are applied when the PU activity changes on a slot-by-slot basis. Under the prediction-based sensing schemes, the states of the channels due to PU activity (busy/idle) are predicted prior to sensing using the sensing history (i.e. past sensing results) of the respective channels. The sensing operation is performed according

to the predicted states of the channels. The accuracy of a channel status predictor can be described by the probability of error in predicting the channel status. It must be noted that the accuracy of the predictor inevitably depends on the accuracy of the history data. Cooperative sensing (e.g. [17]) can be applied to generate reliable history data for the predictor design. Several channel status prediction schemes are found in the literature, [33–36].

Wen et al. [33] modeled the channel status predictor using an autoregressive model based on the Kalman filter ([37]). This model required knowledge of the PU’s traffic characteristics, i.e. mean arrival rate and mean service rate.

In [34], a linear filter model followed by a sigmoid transform function was used to predict the channel busy probability based on the past sensing observations. This model also requires prior knowledge of the PU’s traffic characteristics, besides the history data should be deterministic.

In [35], a hidden Markov model (HMM) based channel status predictor was proposed. The primary user traffic follows Poisson process with 50% traffic intensity (i.e., 50% channel time is occupied by the primary users). The secondary user used the whole time slot if the slot was predicted idle. However, in [35] the accuracy of prediction was not provided. Another HMM based predictor is also proposed in [36], but it only deals with deterministic traffic scenarios, making it not applicable in practice.

2.2 Dynamic Spectrum Access schemes: Related Work

The implementation of the DSA scheme in the CRN should consider two major factors, namely, the network topology which defines the level of coordination among the SUs in the network, and the stationarity of the PU traffic which affects the the channel allocation in the network (i.e., the SU channel assignment must consider the time varying nature of the PU activity). Under a given network topology, we classify the existing DSA schemes based on the stationarity of the PU traffic. For convenience, we categorize them as *discrete time DSA scheme* and *continuous time DSA scheme*. Under the former category, the PU

activity (busy/idle) remains constant for fixed duration (called as *slot*) whereas in the latter category the PU activity changes at any time instant.

Prior to explaining the various existing DSA schemes according to the aforementioned factors, we provide an overview of the CRN based on the various network topologies.

2.2.1 Classification of CRN Architecture

Fig. 2.1 shows the classification of the CRN according to the network topology. The network topology of the CRN can be broadly classified as centralized CRN (i.e., infrastructure-based network) and distributed CRN (i.e., ad hoc network).

Figure 2.1: Classification of CRN based on network topology.

A centralized CRN is an infrastructure based network in which a central controller exists for the coordination among the SUs within its coverage area which is referred to as the *cell*. The central controllers belonging to different cells may further be controlled by a superior entity in the network hierarchy, for example, a spectrum server which maintains the database of the spectrum occupancy over a large geographic region spanning several cells [38]. Fig. 2.2 depicts a model of the centralized CRN. Within each cell, the central controller monitors the traffic on the licensed channels (both PU and SU traffic) and allocates the vacant channels to the SUs using a suitable DSA scheme. Several DSA schemes exist

in the literature for the centralized CRN [9, 22, 39–48]. The SUs communicate with their corresponding central controller using control messages (e.g., bandwidth request) over a low bandwidth common control channel. Such a common control channel may be established within the licensed spectrum depending on the interference experienced by the PUs [49]. Alternatively, the common control channel may be established in the unlicensed region of the spectrum¹ (e.g., in the 2.4 GHz band [22]). However, the biggest limiting factor for a common control channel assumption is the control channel saturation problem [50], which could reduce the achievable user density and throughput if a fairly large number of spectrum holes are available or rapidly occurring. Therefore, load balancing is a crucial design issue in the centralized CRN.

Figure 2.2: Centralized CRN.

The centralized CRN offers greater coordination and throughput for the SUs owing to the presence of the central controller. However, the centralized CRN incurs infrastructure

¹Note that the actual bandwidth used for data transmission by the SUs is much larger compared to the control channel.

cost. Further, a central controller may not be always available for SU coordination, in such a scenario we implement the distributed CRN. A distributed CRN behaves in an ad hoc manner and it is generally difficult to achieve coordination among the SUs. The coordination among the SUs depends on the control channel configuration, accordingly, the distributed CRN can be further classified as follows.

- *Distributed CRN with global control channel*

Similar to the centralized CRN, several works on distributed CRN assume a low bandwidth global (i.e., common) control channel to exist for the coordination of the SUs [20,22,23,51,52]. However, unlike the centralized CRN, the distributed CRN has to address issues like the hidden and exposed terminals, and time synchronization. These issues can be addressed using a suitable MAC protocol, however, this introduces control overhead. For example, to avoid hidden and exposed terminals, control messages such as RTS (request-to-send) and CTS (clear-to-send) were exchanged by the sender and destination SUs on the global control channel in [20].

- *Distributed CRN with local control channels*

In some works on distributed CRN [24,25,53], the SUs within a one (or two) hop neighborhood may be sharing a common set of available channels. Of these channels, the SUs periodically choose the channel with the lowest PU traffic load as their local control channel (e.g., local control channel chosen by vote [24]). SUs within a group sharing local control channels in multiple groups act as relay nodes (also referred as bridge nodes) between these groups. These networks also deploy MAC protocols to address issues such as control channel configuration, group coordination, and hidden and exposed terminals. Compared to the global control channel case, this network topology incurs more control overhead [18].

- *Distributed CRN without dedicated control channel*

In some works [21,54–57], the SUs in the distributed CRN do not configure a separate control channel for control information exchange. Instead, the SUs randomly access

the chosen idle channels using techniques such as the CSMA/CA or the IEEE 802.11e EDCF [58]. It is very difficult to address critical issues such as the multichannel hidden terminal problem (i.e., hidden terminal problem occurring when a SU performs spectrum handoff), transmitter-receiver coordination and time synchronization due to the absence of MAC protocols.

2.2.2 DSA schemes for Centralized CRN

2.2.2.1 Centralized CRN using Discrete Time DSA schemes

The discrete time DSA scheme under a centralized CRN is implemented as scheduling algorithm. The scheduling algorithm allocates the channels to the SUs for the slot duration. At the beginning of the slot, the central controller processes the transmission requests from the SUs and broadcasts the schedule information containing the channel-user assignments in the current slot to the SUs. Thereafter, the SUs switch to their respective assigned channels (i.e. spectrum holes which have high bandwidth compared to the common control channel) and transmit data for the remaining slot duration. Several centralized DSA schemes under this category are found in the literature [9, 22, 39–43]. All these works can be referred as slot-based scheduling schemes.

In the IEEE 802.22 network or WRAN (wireless regional area network) [9], expected to provide DSA for SUs in the TV spectrum, in each slot, the scheduling cycle is divided into uplink (i.e, SUs to central controller) and downlink (i.e., central controller to SUs) schedules. In each slot, the central controller broadcasts the uplink and downlink schedules on every idle channel. The central controller assigns exclusive bandwidth (portions of an idle TV channel) to the SUs based on their bandwidth requests for the uplink transmission.

A scheduling algorithm based on integer linear programming (ILP) was presented in [22]. The channels serve as links to connect the SUs with one another. The ILP formulates the transmission schedule for the SUs such that a unique link is activated in at any time instant within a slot. Using the ILP, the scheduling scheme [22] eliminates interference to the SUs.

Tang *et al* [39] described several scheduling algorithms which maximize the transmission capacity of the SUs. The approach in [39] achieves fairness for the SUs while allocating

the channels. The scheduling algorithms in [39] consider various factors including the link capacity, signal-to-interference-and-noise ratio (SINR) and traffic demand of the SUs.

Scheduling algorithms which maximize the transmission capacity of the SUs were also presented in [40]. These scheduling algorithms consider factors like the interference, fading and packet waiting time.

In [41], the scheduling algorithms were proposed using ILP. The scheduling algorithms in [41] consider different factors including the packet waiting time, maximum frequency, interference temperature, and throughput.

Tian *et al* [42] developed a scheduling algorithm which achieves proportional fairness for the SUs. The scheduling algorithm in [42] considers factors like the packet waiting time and interference caused by the SU to the PU's receiver.

Rashid *et al* [43], described a scheduling algorithm which assigns the idle channels to the SUs based on whether the SNR measured at the receiving end (i.e. SUs in the downlink and central controller in the uplink) is the highest for a given channel.

2.2.2.2 Centralized CRN using Continuous Time DSA schemes

The centralized CRN using a continuous time DSA scheme is usually treated as a call admission control system which receives calls from the PUs and SUs. A PU call is given higher priority compared to a SU call during channel assignment. In such centralized CRN, the central controller continuously monitors the activity on the channels which were assigned to the SUs. If a PU transmission is detected on any of these channels then the SUs transmitting on that channel are either reassigned to other idle channels (if available) or terminated. The process of reassigning a displaced SU transmission is referred to as *spectrum handoff*. Several centralized DSA schemes under this category are found in the literature [44–48].

According to the DSA scheme in [44], an incoming SU call is queued till a new transmission opportunity is found. Also, the SUs which experienced unsuccessful handoff were dropped.

Zhang in [45] described a DSA scheme similar to [44]. However, the analytical model used in [45] was different from that used in [44]. Zhang also considered the case with buffer in which the incoming SU calls are queued in buffer in case all the channels are busy (i.e. busy due to ongoing PU and SU activity on different channels).

According to Tang *et al.* [46], the SUs are queued in case of unsuccessful handoff and reassigned channels when they become available. Thus, the SU transmissions were prevented from forced termination due to a PU arrival. General distributions were used to characterize the PU and SU traffic in [46].

In [47], Zhu *et al.* proposed a DSA scheme which considered channel reservation for SU handoff. According to [47], an incoming SU call is assigned a channel only when the number of idle channels in the licensed spectrum exceeds the number of channels reserved for SU handoff.

The DSA scheme in [48] assigned as much possible bandwidth to the SUs in order to improve the spectrum utilization. This bandwidth assignment was repeated whenever the system state changed.

In all the above works, the performance of the DSA scheme was analyzed using continuous time Markov chain (CTMC). The performance of the secondary users was evaluated in terms of the blocking probability and probability of forced termination for the SU calls.

2.2.3 DSA schemes for Distributed CRN

2.2.3.1 Distributed CRN using Discrete Time DSA schemes

Under this category, the DSA scheme is implemented as a MAC protocol. Most distributed CRN usually assume a low bandwidth common control channel to exist for control information exchange among the SUs (e.g., [23]). The MAC protocol usually consists of two phases namely, a contention phase and a data transmission phase. During the contention phase the SUs contend for the spectrum holes over the common control channel. This is followed by the successful SUs taking over the spectrum holes during the data transmission phase. The MAC protocol operates on a slot-by-slot basis. Several such MAC protocols are found in the literature [24, 25, 52–57].

A MAC protocol called HD-MAC was proposed in [24] for multi-hop distributed CRN. According to [24], the SUs sharing a common channel form a group. The SUs belonging to multiple groups act as gateways between the groups. During the contention phase, the SUs negotiate with one another to acquire the channels. During the data transmission phase, the SUs which successfully negotiated will transmit their data packets.

Chen *et al* [25] proposed a MAC protocol according to which the SUs form clusters with every cluster controlled by a group leader. The members of a cluster report the spectrum holes to their group leader. The data transmission over the spectrum holes is based on the result of the contention phase which is broadcast by the group leader at the beginning of the MAC protocol.

Ma *et al* [52] proposed a MAC protocol called DOSS which reserves few narrow bandwidth channels called *busy tone channels* for the purpose of eliminating the collisions among the SUs selecting the same channels. Data transmission between a SU pair requires these SUs to negotiate a common data transmission channel over the common control channel. Ma *et al* also proposed a MAC protocol in [54] in which, a SU transmits data on a channel as long as that channel is found to be idle at the receiving SU. This transmission scheme was called cross channel communication in [54].

In [53], a multichannel MAC protocol called C-MAC was proposed for the distributed CRN. According to the C-MAC, the SUs maintain a common channel called the rendezvous channel (RC) on which the details of the channel selection of all the SUs is periodically updated. To transmit data, a SU simply listens to the RC to know the exact channel on which the destination SU is currently operating.

In [55–57, 59, 60], the MAC protocols were modeled after the IEEE 802.11 DCF (distributed coordination function) mechanism [58]. To transmit data, a SU first performs spectrum hole detection and then senses the channel for a short duration called (distributed inter-frame space) DIFS to detect SU transmissions. If the channel continues to be idle, the SU senses the channel for some more time randomly chosen based on a contention window (CW) [58]. For analysis, [55–57] calculated the saturation throughput of the secondary

user using a Markov chain, similar to the analysis in [58]. In [59,60], the average number of packets in the SU's queue and throughput were calculated in the analysis using $M/G^Y/1$ queuing model, similar to [61].

2.2.3.2 Distributed CRN using Continuous Time DSA schemes

Very few works have dealt with the design of distributed CRN under this category. Some of the existing works in the literature ([21,51,62,63]) are described below.

According to the MAC protocol for the ECMA-392 standard [51], proposed for providing DSA for SUs in TV spectrum, an SU can transmit on a frame shared by several SUs using two methods, namely PCA (prioritized contention access) or CRP (channel reservation protocol). Under the PCA, an SU transmits its packets on a channel based on its random backoff time. The higher an SUs priority, the shorter is the interval from which it chooses the backoff value randomly. Under the CRP, an SU reserves a few slots of the frame for contention-free access. In [51], SUs perform spectrum handoff when they detect a PU activity during the frame.

In [21,62,63], the SU transmission was assumed to be slotted while the PU transmission was considered unslotted. Further, it was assumed that all channels in the licensed spectrum are homogeneous (i.e. all channels have the same PU traffic distribution). These works were proposed from a single SU's perspective (i.e., SU coordination was not considered). The SU transmission was assumed to be slotted. The SU senses a channel periodically at the beginning of each slot and transmits for the remaining slot duration, whenever the channel is detected to be idle. At the end of each slot, the SU should receive an acknowledgement message (ACK) from the destination SU, this indicates that there was no PU activity during the slot. Otherwise, it is assumed that a collision occurred between the SU and PU during the slot. Therefore the objective of these works ([21,62,63]) was to maximize the SU throughput given its transmission duration (fixed number of slots), subject to the constraint on the interference permissible to the PUs. This objective was achieved using constrained partially observable Markov decision process (CPOMDP).

Chapter 3

Channel Status Prediction for Cognitive Radio Networks

This chapter demonstrates the utility of channel status prediction to spectrum sensing. Besides reducing the sensing energy, these channel status prediction schemes also improve the spectrum utilization. Specifically, two prediction schemes are explored. For the first scheme, a neural network approach using the multilayer perceptron (MLP) network [64] is proposed, whereas for the second scheme, a statistical approach using the hidden Markov model (HMM) [65] is introduced.

3.1 Research Highlights

The key aspects of the proposed channel status prediction schemes are presented below.

- (1) We assume that the PU activity (i.e., busy or idle) stays constant for a fixed time duration called the *slot*. In other words, the PU activity varies on a slot-by-slot basis. We assume ideal spectrum sensing¹. The proposed channel status prediction schemes predict the channel status in a given slot based on finite sensing history (e.g., SUs share their sensing history). Sensing operation is performed only when the predicted channel status is idle.
- (2) Unlike the reward based sensing schemes [20, 27–31], which also perform channel status prediction (statistically), the proposed channel status prediction schemes do

¹The effect of sensing errors will be considered in the future work

not require prior knowledge of PU traffic characteristics such as the mean ON and OFF times of the PU activity on the licensed channels, or the transition probabilities of the PU activity on the licensed channels.

- (3) Unlike the reward based sensing schemes [20, 27–31] and the existing channel status prediction schemes [33–36], we consider both stationary and non-stationary PU traffic scenarios. Further, we analyze the performance of the proposed schemes using various performance measures such as probability of wrongly predicting the busy channel status and probability of wrongly predicting the idle channel status.

The layout of this chapter is as follows. First, the design of the two channel status predictors is presented. Following this, the effect of the two prediction schemes over spectrum sensing is shown using simulations. Thereafter, a discussion is provided on the implementation issues along with a qualitative comparison of the two prediction schemes.

3.2 Neural Network based Prediction

Neural networks are nonlinear parametric models which create a mapping function between the input and output data. The advantage of neural networks over statistical models is that they do not require a priori knowledge of the underlying distributions of the observed process. In cognitive radio networks (CRNs), it is difficult to obtain the statistics of channel usage by the PUs. Therefore, the neural networks offer an attractive choice for modeling the channel status predictor. Once the neural networks are trained, the computational complexity is significantly reduced. The neural network model, multilayer perceptron (MLP), has been used in various applications, e.g., system identification and time series prediction [66, 67].

3.2.1 MLP Predictor Design

The MLP network is a multilayered structure consisting of an input layer, an output layer, and a few hidden layers. Let a layer be denoted as $o \in \{1, \dots, O\}$, where O denotes the

total number of layers in the MLP network. Excluding the input layer, every layer contains some computing units (referred to as neurons) which calculate a weighted sum of the inputs and perform a nonlinear transform on the sum. The nonlinear transform is implemented using a hyperbolic tangent function. Neurons belonging to different layers are connected through adaptive weights. The output of a neuron g in the o^{th} layer, denoted by p_g^o , can be represented as

$$p_g^o = \frac{1 - \exp(-\nu_g^o)}{1 + \exp(-\nu_g^o)} \quad (3.1)$$

where

$$\nu_g^o = \sum_f p_f^{o-1} \omega_{g,f}^o. \quad (3.2)$$

Equation (3.2) represents weighted sum of the inputs coming from the output of the neurons in the $(o - 1)^{th}$ layer using the adaptive weights (or parameters) $\omega_{g,f}^o$ connecting the respective neurons. Equation (3.1) represents the nonlinear transform on ν_g^o . This nonlinear transform gives an output in the range of $[-1, +1]$. If the inputs are coming from the input layer, p_g^{o-1} in equation (3.2) is replaced with the corresponding input. The total number of inputs in the input layer is referred to as the order of the MLP network and is denoted by τ .

The number of hidden layers and the number of neurons in each layer depend on the application. For channel status prediction problem, we found the MLP network with two hidden layers to be sufficient. The first hidden layer has 15 neurons, the second hidden layer has 20 neurons, and the output layer has only one neuron. The order of the MLP predictor (τ) which represents the length of the observation sequence (or slot's status history) in our problem, is set to 4.

3.2.2 MLP Predictor Training

The status of a channel is predicted by its corresponding MLP predictor which is trained using the sensing history of that channel². The sensing history of a channel u is generated by

²Similarly, a separate HMM predictor is used for each channel when using the HMM based prediction scheme.

observing the channel u for a finite time duration, say L_{mlp} slots. Let the sensing history be denoted as $\mathbf{s}_{mlp}^u = [s_1^u, s_2^u, \dots, s_{L_{mlp}}^u]$ where s_n^u ($n \in \{1, \dots, L_{mlp}\}$) represents sensing result (i.e. PU channel activity) of the channel u in the slot n . For simplicity of presentation, the superscript u denoting channel index is omitted in the rest of the chapter. Numerically, s_n can be represented as 1 if the PU activity is not detected (i.e. it indicates a spectrum hole) and as -1 if PU activity is detected. Thus $\mathbf{s}_{mlp}$ resembles a binary series containing symbols 1 and -1 . The MLP predictor training requires shorter training patterns which are denoted by the input vector $\mathbf{s}_{mlp}^\tau = [s_n, s_{n-1}, \dots, s_{n-\tau+2}, s_{n-\tau+1}]$ of length τ slots and its corresponding desired value s_{n+1} . The parameters of the MLP predictor are updated using the batch backpropagation (BP) algorithm [64]. The MLP predictor training process is illustrated in Fig. 3.1.

Figure 3.1: MLP predictor training.

For each input vector presented to the input layer of the MLP, the outputs of the neurons in each layer are calculated proceeding from the first hidden layer to the output layer using Eqs. (3.1) and (3.2). This computation is called forward pass. The output of the neuron in the output layer p_1^O is referred to as the MLP output and is denoted by

$\hat{s}_{n+1}$. $\hat{s}_{n+1}$ is treated as an estimate of the corresponding desired value s_{n+1} . The difference between the desired value and its estimate is called as the *error* (ϵ_n) which can be expressed as follows:

$$\epsilon_n = s_{n+1} - \hat{s}_{n+1} = s_{n+1} - p_1^O. \quad (3.3)$$

The objective of the training algorithm is to minimize this error ϵ_n by adapting the parameters $\omega_{g,f}^o$ such that the MLP output approximately represents the desired value. In other words, the MLP predictor tries to create a mapping function between the input vector and the desired value. According to the BP algorithm [64], it is easier to minimize the mean square error than to directly minimizing the error ϵ_n . The mean square error criterion can be expressed as

$$E = \frac{1}{2} \sum \epsilon_n^2. \quad (3.4)$$

Based on the BP algorithm [64], the parameters are updated as follows:

$$\omega_n = \omega_{n-1} + \Delta\omega_n \quad (3.5)$$

$$\Delta\omega_n = -\mu' \cdot \frac{\partial E}{\partial \omega_n} + \eta' \cdot \Delta\omega_{n-1}. \quad (3.6)$$

In Eqs. (3.5) and (3.6), ω_n represents the parameter $\omega_{g,f}^o$ at time instant n (also denoted by ω_{g,f_n}^o) while μ' and η' represent the learning rate and the momentum term respectively. μ' can be chosen from the range $(0, 1)$ while η' can be chosen from the range $(0.5, 0.9)$ as in [37, 64]. In the simulations, the values of μ' and η' are set to 0.2 and 0.9, respectively. The partial derivative $\partial E / \partial \omega_n$ in equation (3.6) is calculated successively for each neuron by proceeding backwards from the output layer to the input layer. This computation is called backward pass. The partial derivative $\partial E / \partial \omega_n$ can be expressed in terms of the variables ϵ_t , p_g^o , ν_g^o and ω_{g,f_n}^o using chain rule as follows:

$$\frac{\partial E}{\partial \omega_n} = \frac{\partial E}{\partial \omega_{g,f_n}^o} = \frac{\partial E}{\partial \epsilon_n} \cdot \frac{\partial \epsilon_n}{\partial p_g^o} \cdot \frac{\partial p_g^o}{\partial \nu_g^o} \cdot \frac{\partial \nu_g^o}{\partial \omega_{g,f_n}^o}. \quad (3.7)$$

The partial derivatives $\partial p_g^o / \partial \nu_g^o$ and $\partial \nu_g^o / \partial \omega_{g,f_n}^o$ in (3.7) are calculated by the following expressions, based on equations (3.1) and (3.2)

$$\frac{\partial p_g^o}{\partial \nu_g^o} = (1 - p_g^o) \cdot (1 + p_g^o) \quad (3.8)$$

$$\frac{\partial \nu_g^o}{\partial \omega_{g,f_n}^o} = p_f^{o-1} . \quad (3.9)$$

The partial derivative $\partial E / \partial \omega_n$ is calculated in two ways depending on whether the neuron g has a desired output value or not [64]. For the neuron in output layer O , a desired output value s_{n+1} exists. Therefore, $\partial E / \partial \omega_n$ for the parameter $\omega_n = \omega_{1,f_n}^O$ connecting the output layer neuron to a neuron f in the preceding layer ($O - 1$) is expressed as

$$\frac{\partial E}{\partial \omega_n} = \frac{\partial E}{\partial \omega_{g,f_n}^o} = \frac{\partial E}{\partial \omega_{1,f_n}^O} = \frac{\partial E}{\partial \epsilon_n} \cdot \frac{\partial \epsilon_n}{\partial p_1^O} \cdot \frac{\partial p_1^O}{\partial \nu_1^O} \cdot \frac{\partial \nu_1^O}{\partial \omega_{1,f_n}^O} . \quad (3.10)$$

From equation (3.3), $\partial \epsilon_n / \partial p_1^O$ is given by

$$\frac{\partial \epsilon_n}{\partial p_1^O} = -1 . \quad (3.11)$$

From equation (3.4), $\partial E / \partial \epsilon_n$ is given by

$$\frac{\partial E}{\partial \epsilon_n} = \epsilon_n . \quad (3.12)$$

Substituting $\nu_g^o = \nu_1^O$, $p_f^{o-1} = p_1^{O-1}$ and $\omega_{g,f_n}^o = \omega_{1,f_n}^O$ in (3.8) and (3.9), we obtain

$$\frac{\partial p_1^O}{\partial \nu_1^O} = (1 - p_1^O)(1 + p_1^O) \quad (3.13)$$

$$\frac{\partial \nu_1^O}{\partial \omega_{1,f_n}^O} = p_f^{O-1} . \quad (3.14)$$

Substituting (3.11)-(3.14) in (3.10), $\partial E / \partial \omega_n$ for the parameter $\omega_n = \omega_{1,f_n}^O$ connecting the output layer neuron to a neuron f in the preceding layer ($O - 1$) is given by

$$\frac{\partial E}{\partial \omega_n} = \frac{\partial E}{\partial \omega_{1,f_n}^O} = (\epsilon_n) \cdot (-1) \cdot ((1 - p_1^O) \cdot (1 + p_1^O)) \cdot (p_f^{O-1}) . \quad (3.15)$$

For a neuron g in a hidden layer o , a desired output value does not exist. In this case, $\partial E / \partial \omega_n$ is calculated in terms of an error term called local gradient, denoted by $\delta_g^o(n)$.

The local gradient $\delta_g^o(n)$ for a neuron g in the hidden layer o can be expressed as

$$\delta_g^o(n) = \frac{\partial E}{\partial \nu_g^o} = \frac{\partial E}{\partial p_g^o} \cdot \frac{\partial p_g^o}{\partial \nu_g^o} . \quad (3.16)$$

In order to calculate the local gradient $\delta_g^o(n)$ at time instant n for every neuron g in a hidden layer o , $\delta_g^o(n)$ is initialized at the output layer neuron and calculated recursively for every neuron g in layer o by proceeding backwards from the output layer to the first hidden layer. The initialization can be given by

$$\begin{aligned}\delta_1^O(n) &= \frac{\partial E}{\partial v_1^O} = \frac{\partial E}{\partial p_1^O} \cdot \frac{\partial p_1^O}{\partial \nu_1^O} = \frac{\partial E}{\partial \epsilon_n} \cdot \frac{\partial \epsilon_n}{\partial p_1^O} \cdot \frac{\partial p_1^O}{\partial \nu_1^O} \\ &= (\epsilon_n) \cdot (-1) \cdot ((1 - p_1^O) \cdot (1 + p_1^O)) .\end{aligned}\quad (3.17)$$

The recursive equation for $\delta_g^o(n)$ is given by

$$\delta_g^o(n) = (1 - p_g^o) \cdot (1 + p_g^o) \sum_h \delta_h^{o+1}(n) \cdot \omega_{h,g}^{o+1} \quad (3.18)$$

where h denotes a neuron in layer $o + 1$. The partial derivative $\partial E / \partial \omega_n$ for a parameter $\omega_n = \omega_{g,f}^o$ connecting the neuron g in a hidden layer o to a neuron f in the layer $(o - 1)$ is expressed in terms of $\delta_g^o(n)$ as

$$\frac{\partial E}{\partial \omega_n} = \frac{\partial E}{\partial \omega_{g,f}^o} = \frac{\partial E}{\partial p_g^o} \cdot \frac{\partial p_g^o}{\partial \nu_g^o} \cdot \frac{\partial \nu_g^o}{\partial \omega_{g,f}^o} = \delta_g^o(n) \cdot p_f^{o-1} . \quad (3.19)$$

For batch BP algorithm [64], the parameter ω (or $\omega_{g,f}^o$) is not updated at the instant n , only $\Delta \omega_n$ is calculated. After showing all patterns, the parameter ω is updated by the average value of $\Delta \omega_n$ as follows:

$$\omega_{new} = \omega_{old} + \frac{1}{L_{mlp} - \tau} \sum_{n=1}^{L_{mlp} - \tau} \Delta \omega_n \quad (3.20)$$

where L_{mlp} is the length of the entire binary series $\mathbf{s}_{mlp}$.

The BP algorithm [64] is repeated until the minimum of the mean square error or the maximum number of iterations is reached. Once training is complete, we test the MLP predictor by randomly observing τ successive slots $\mathbf{s}_{mlp}^\tau$ and computing the MLP output $\hat{s}_{n+1}$. By using a decision threshold at the MLP output, the predicted value can be expressed as binary symbol

$$\begin{aligned}\text{if } \hat{s}_{n+1} \geq 0 \text{ then } \hat{s}_{n+1} &= +1 \\ \text{if } \hat{s}_{n+1} < 0 \text{ then } \hat{s}_{n+1} &= -1 .\end{aligned}\quad (3.21)$$

The accuracy of the MLP predictor is evaluated in Section 3.4.

3.3 Channel Status Prediction using HMM

3.3.1 Hidden Markov Model

Consider a system having N_S states, the set of states can be denoted as $\mathcal{S} = \{S_1, \dots, S_{N_S}\}$. At a time instant n the system enters state Q_n depending on some probabilities associated with the state transitions. If the state transitions follow the Markov property, then the probability of a state transition can be expressed as

$$P(Q_n = S_{h'} | Q_{n-1} = S_{g'}, Q_{n-2} = S_{f'}, \dots) = P(Q_n = S_{h'} | Q_{n-1} = S_{g'}) . \quad (3.22)$$

Suppose that the states are associated with E discrete symbols, and the set of symbols is denoted as $\mathcal{P} = \{P_1, \dots, P_E\}$. After every state transition, a symbol $s_n (\in \mathcal{P})$ is emitted by state $Q_n (\in \mathcal{S})$ depending on some probability distribution. Suppose that only the symbol sequence is observable while the state sequence is hidden, this gives rise to the Hidden Markov Model (HMM). Therefore, the HMM can be formally stated as a statistical model in which the observed process is assumed to be generated in response to another stochastic process which is hidden and follows the Markov property. In our case, the observed process is the sensing history of a channel.

The HMM has found remarkable application in speech recognition [65] and bioinformatics [68]. While HMM based channel status prediction was proposed in the literature [35, 36], these works do not give a detailed analysis of the HMM predictor design. Therefore, we explain the HMM predictor design in more detail.

In order to model the HMM it is necessary to specify the following

- The number of emitted symbols, E
- The number of states, N_S
- The observation sequence, $\mathbf{s}_{hmm}^\kappa = [s_1, s_2, \dots, s_\kappa]$
- The state transition probabilities, $A_{g', h'} = P(Q_n = S_{h'} | Q_{n-1} = S_{g'})$ subject to the conditions $A_{g', h'} \geq 0$ and $\sum_{h'=1}^{N_S} A_{g', h'} = 1$

- The symbol emission probabilities, $B_{h'}(P_{e'}) = P(s_n = P_{e'} | Q_n = S_{h'})$ subject to the conditions $B_{h'}(P_{e'}) \geq 0$ and $\sum_{e'=1}^E B_{h'}(P_{e'}) = 1$, $1 \leq h' \leq N_S$
- Initial state distribution, $\boldsymbol{\pi} = [\pi_1, \dots, \pi_{N_S}]$, where $\pi_{g'} = P(Q_{n=1} = S_{g'})$ and satisfies the conditions $\pi_{g'} \geq 0$ and $\sum_{g'=1}^{N_S} \pi_{g'} = 1$

The HMM can be denoted by the notation $\mathbf{H} = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$, where $\mathbf{A}$ is a $N_S \times N_S$ state transition matrix containing the probabilities $A_{g',h'}$ where g' denotes the rows and h' denotes the columns, and $\mathbf{B}$ is a $N_S \times E$ emission matrix containing the probabilities $B_{h'}(P_{e'})$ where h' denotes the rows and e' denotes the columns.

3.3.2 HMM Predictor

The HMM prediction scheme is illustrated in Fig. 3.2. Consider the following sequence of sensing results $[s_1, s_2, \dots, s_\kappa]$ where s_n represents the PU activity on a channel in the slot n (where $n \in \{1, \dots, \kappa\}$). s_n takes values 1 and -1 for the sensing statuses *busy* and *idle* respectively. The objective of the HMM predictor is to predict the symbol $s_{\kappa+1}$ based on the past κ observations. To predict the symbol $s_{\kappa+1}$, the HMM should be able to generate the observation sequence $\mathbf{s}_{hmm}^\kappa = [s_1, s_2, \dots, s_\kappa]$ with maximum likelihood probability. Hence, the parameters $\mathbf{H} = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ are adapted to maximize the likelihood probability of generating the observation sequence, i.e., maximize the probability $P(\mathbf{s}_{hmm}^\kappa | \mathbf{H})$. Once training is completed, the joint probability of observing the sequence $\mathbf{s}_{hmm}^\kappa$ followed by a busy slot or an idle slot at instant $n = \kappa + 1$ is calculated. In other words, the joint probabilities $P(\mathbf{s}_{hmm}^\kappa, 1 | \mathbf{H})$ and $P(\mathbf{s}_{hmm}^\kappa, -1 | \mathbf{H})$ are calculated. The slot occupancy at instant $n = \kappa + 1$ is predicted according to decision rule given by

$$\begin{aligned} &\text{if } P(\mathbf{s}_{hmm}^\kappa, 1 | \mathbf{H}) \geq P(\mathbf{s}_{hmm}^\kappa, -1 | \mathbf{H}) \text{ then } \hat{s}_{\kappa+1} = +1 \\ &\text{if } P(\mathbf{s}_{hmm}^\kappa, 1 | \mathbf{H}) < P(\mathbf{s}_{hmm}^\kappa, -1 | \mathbf{H}) \text{ then } \hat{s}_{\kappa+1} = -1 \end{aligned} \quad (3.23)$$

where $\hat{s}_{\kappa+1}$ is the predicted value.

Figure 3.2: HMM predictor training.

3.3.2.1 HMM Training

The Baum Welch Algorithm (BWA) [65] is an iterative method to estimate the HMM parameters $\psi = [\pi, \mathbf{A}, \mathbf{B}]$ such that the probability $P(\mathbf{s}_{hmm}^\kappa | \psi)$ is maximized. To estimate the parameters $\psi = [\pi, \mathbf{A}, \mathbf{B}]$, the BWA defines the following variables

- Forward variable $\alpha_n(g') = P(s_1, s_2, \dots, s_n, \phi_n = S_{g'} | \mathbf{s}_{hmm}^\kappa)$, for $1 \leq g' \leq N_S$
- Backward variable $\beta_n(g') = P(s_{n+1}, s_{n+2}, \dots, s_\kappa, Q_n = S_{g'} | \mathbf{s}_{hmm}^\kappa)$, for $1 \leq g' \leq N_S$
- $\chi_n(g', h') = P(Q_n = S_{g'}, Q_{n+1} = S_{h'} | \mathbf{s}_{hmm}^\kappa)$ for $1 \leq g', h' \leq N_S$, the probability of being in state $S_{g'}$ at instant n and in state $S_{h'}$ at instant $n+1$ given the observation sequence $\mathbf{s}_{hmm}^\kappa$ and the model $\psi = [\pi, \mathbf{A}, \mathbf{B}]$
- $\sigma_n(g') = P(Q_n = S_{g'} | \mathbf{s}_{hmm}^\kappa)$ for $1 \leq g' \leq N_S$, the probability of being in state $S_{g'}$ at instant n given the observation sequence $\mathbf{s}_{hmm}^\kappa$ and the model $\psi = [\pi, \mathbf{A}, \mathbf{B}]$

The estimation formulas for the parameters of the model $\psi = [\pi, \mathbf{A}, \mathbf{B}]$ are expressed in terms of the variables $\chi_n(g', h')$ and $\sigma_n(g')$ as follows:

$$A_{g', h'} = \frac{\sum_{n=1}^{n=\kappa-1} \chi_n(g', h')}{\sum_{n=1}^{n=\kappa-1} \sigma_n(g')} \quad (3.24)$$

$$B_{g'}(P_{e'}) = \frac{\sum_{n=1, s_n=P_{e'}}^{n=\kappa} \sigma_n(g')}{\sum_{n=1}^{n=\kappa} \sigma_n(g')} \quad (3.25)$$

$$\pi_i = \sigma_1(g') . \quad (3.26)$$

In Eq. (3.24), the numerator represents the expected number of transitions from state g' to state h' over duration $\kappa - 1$ while the denominator represents the expected number of times a transition is made from state g' . The numerator in (3.25) represents the expected number of transitions from state g' and symbol $P_{e'}$ is observed after the transitions. In the Eqs. (3.24)-(3.26), $\chi_n(g', h')$ and $\sigma_n(g')$ are calculated as follows:

$$\chi_n(g', h') = \frac{\alpha_n(g') \cdot A_{g', h'} \cdot B_{h'}(s_{n+1}) \cdot \beta_{n+1}(h')}{P(\mathbf{s}_{hmm}^\kappa | \cdot)} \quad (3.27)$$

$$\sigma_n(g') = \sum_{h'=1}^{N_S} \chi_n(g', h') . \quad (3.28)$$

The forward and backward variables in equations (3.27) and (3.28) are calculated recursively. The forward variable $\alpha_n(g')$ is calculated as follows.

Initialization:

$$\alpha_1(n) = \pi_{g'} \cdot B_{g'}(1), \quad 1 \leq g' \leq N_S . \quad (3.29)$$

Recursion:

$$\alpha_{n+1}(h') = \left[\sum_{g'=1}^{N_S} \alpha_n(g') \cdot A_{g', h'} \right] \cdot B_{h'}(s_{n+1}), \quad \begin{array}{l} 1 \leq g' \leq N_S \\ 1 \leq n \leq \kappa - 1. \end{array} \quad (3.30)$$

Termination:

$$P(\mathbf{s}_{hmm}^\kappa | \cdot) = \sum_{g'=1}^{N_S} \alpha_\kappa(g'), \quad 1 \leq g' \leq N_S . \quad (3.31)$$

The backward variable is calculated as follows.

Initialization:

$$\beta_\kappa(g') = 1, \quad 1 \leq g' \leq N_S . \quad (3.32)$$

Recursion:

$$\beta_n(g') = \sum_{h'=1}^{N_S} A_{g', h'} \cdot B_{h'}(s_{n+1}) \cdot \beta_{n+1}(h'), \quad \begin{array}{l} 1 \leq g' \leq N_S \\ \kappa - 1 \geq n \geq 1. \end{array} \quad (3.33)$$

Equation (3.31) provides the formula for the probability of observing the sequence $\mathbf{s}_{hmm}^\kappa$ given the model $\theta = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$. The parameters $\theta = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ are re-estimated using the equations for a maximum of I_{max} iterations or till the maximum $P(\mathbf{s}_{hmm}^\kappa | \theta)$ is reached.

To avoid the possibility of underflow³, the forward and backward variables are scaled (or normalized) while calculating Eqs. (3.29)-(3.33). The scaling operation [65], can be expressed as follows:

$$\hat{\alpha}_n(g') = \left(\prod_{t=1}^n c_t \right) \cdot \alpha_n(g'), \quad \begin{matrix} 1 \leq g' \leq N_S \\ n = 1, 2, \dots, \kappa \end{matrix} \quad (3.34)$$

$$\hat{\beta}_n(g') = \left(\prod_{t=n+1}^{\kappa} c_t \right) \cdot \beta_n(g'), \quad \begin{matrix} 1 \leq g' \leq N_S \\ n = \kappa, \kappa - 1, \dots, 1 \end{matrix} \quad (3.35)$$

where $\hat{\alpha}_n(g')$ and $\hat{\beta}_n(g')$ represent the scaled forward and backward variables respectively while c_t represents the scaling coefficient, which can be calculated as

$$c_t = \frac{1}{\sum_{g'=1}^{N_S} \alpha_t(g')} . \quad (3.36)$$

Accordingly the variables $\chi_n(g', h)$ and $\sigma_n(g')$ are calculated by replacing $\alpha_n(g')$ and $\beta_n(g')$ with their scaled versions $\hat{\alpha}_n(g')$ and $\hat{\beta}_n(g')$ in Eqs. (3.27) and (3.28). Because of using the same coefficients for the forward and the backward variables, the estimation formulas given in Eqs. (3.24)-(3.26) remain unchanged. However, the likelihood probability $P(\mathbf{s}_{hmm}^\kappa | \theta)$ cannot be truly estimated due to the scaling operation. Instead, the log likelihood probability $\log(P(\mathbf{s}_{hmm}^\kappa | \theta))$ can be calculated as follows:

$$\begin{aligned} \log(P(\mathbf{s}_{hmm}^\kappa | \theta)) &= \log \left[\sum_{g'=1}^{N_S} \alpha_\kappa(g') \right] \\ &= \log \left[\sum_{g'=1}^{N_S} \left(\hat{\alpha}_\kappa(g') / \left(\prod_{t=1}^{\kappa} c_t \right) \right) \right] \\ &= \log \left[\sum_{g'=1}^{N_S} \hat{\alpha}_\kappa(g') \right] - \log \left[\prod_{t=1}^{\kappa} c_t \right] . \end{aligned} \quad (3.37)$$

³The computations (3.29)-(3.33) involves multiplication of probabilities which causes the variables $\alpha_n(g')$ and $\beta_n(g')$ to tend to zero as n tends to a large value.

As $\sum_{g'=1}^{N_S} \hat{\alpha}_n(g') = 1$ at any time instant n due to normalization

$$\log(P(\mathbf{s}_{hmm}^\kappa | \psi)) = - \sum_{t=1}^{\kappa} \log(c_t) . \quad (3.38)$$

Thus when scaling is used, the modified BWA estimates the model $\psi = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ such that the log likelihood probability of generating the observation sequence $\mathbf{s}_{hmm}^\kappa$, $\log(P(\mathbf{s}_{hmm}^\kappa | \psi))$ is maximized.

During the training, the observation sequence length κ is set to 80, the number of states N_S in the HMM is set to 10, and the maximum number of iterations I_{max} is set to 10. Initial values of the probabilities for state transition ($\mathbf{A}$), state emission ($\mathbf{B}$) and state initial distribution $\boldsymbol{\pi}$ are chosen arbitrarily. However, the conditions for $A_{g',h'}$, $B_{h'}(P_{e'})$ and $\pi_{g'}$ given in Section 3.3.1 should be satisfied. The log likelihood of generating an observation sequence $\mathbf{s}_{hmm}^\kappa$, $\log(P(\mathbf{s}_{hmm}^\kappa | \psi))$ by the HMM at every iteration of the training is shown in Fig. 3.3.

Figure 3.3: Plot of the log likelihood probability $\log(P(\mathbf{s}_{hmm}^\kappa | \psi))$ at each iteration during HMM training.

3.3.2.2 HMM Testing

After training, the joint probabilities $P(\mathbf{s}_{hmm}^\kappa, 1| \cdot)$ and $P(\mathbf{s}_{hmm}^\kappa, -1| \cdot)$ are calculated as follows:

$$P(\mathbf{s}_{hmm}^\kappa, 1| \cdot) = \sum_{g'=1}^{N_S} \alpha_{\kappa+1}^1(g') \quad (3.39)$$

$$P(\mathbf{s}_{hmm}^\kappa, -1| \cdot) = \sum_{g'=1}^{N_S} \alpha_{\kappa+1}^{-1}(g') \quad (3.40)$$

where $\alpha_{\kappa+1}^1(g')$ and $\alpha_{\kappa+1}^{-1}(g')$ are given by

$$\alpha_{\kappa+1}^1(g') = \left[\sum_{h'=1}^{N_S} \alpha_\kappa(h') \cdot A_{h',g'} \right] \cdot B_{g'}(s_{\kappa+1} = 1), \quad 1 \leq g' \leq N_S \quad (3.41)$$

$$\alpha_{\kappa+1}^{-1}(g') = \left[\sum_{h'=1}^{N_S} \alpha_\kappa(h') \cdot A_{h',g'} \right] \cdot B_{g'}(s_{\kappa+1} = -1), \quad 1 \leq g' \leq N_S. \quad (3.42)$$

Based on $P(\mathbf{s}_{hmm}^\kappa, 1|\kappa)$ and $P(\mathbf{s}_{hmm}^\kappa, -1| \cdot)$ the predicted value is found using (3.23). When scaling is used, the log likelihood probabilities $\log(P(\mathbf{s}_{hmm}^\kappa, 1|\mathbf{s}_{hmm}^\kappa))$ and $\log(P(\mathbf{s}_{hmm}^\kappa, -1| \cdot))$ are calculated similar to equation (3.38) and the channel status at the time instant $\kappa + 1$ is predicted in favor of the maximum of the two values. The accuracy of the HMM predictor is evaluated in Section 3.4.

3.4 Simulation and Analysis

For the purpose of simulation, the PU traffic on a channel is assumed to follow Poisson process⁴. The ON and OFF times of the channel are drawn from geometric distributions. For different traffic scenarios, the traffic intensity ρ and the mean inter-arrival time t_{inter} of the traffic bursts are varied. The traffic intensity is related to the mean inter-arrival time as follows.

$$\rho = \frac{\text{mean ON time}}{\text{mean ON+OFF time}} = \frac{t_{serv}}{t_{inter}} \quad (3.43)$$

where t_{serv} is the mean time that the PU is active on a channel for each traffic burst. A minimum of 50% traffic intensity is maintained for a channel.

⁴In the simulation, we use Poisson process to generate the primary user traffic. However, the channel status predictors are not restricted to Poisson process and are applicable to any traffic distribution.

The accuracy of the two prediction schemes is evaluated using two performance measures, $P_p^e(Overall)$ and $P_p^e(Busy)$, for various traffic scenarios. $P_p^e(Busy)$ represents the wrong prediction probability (i.e., misdetection probability) given the PU activity is really present. $P_p^e(Busy)$ is an important measure from the PU's standpoint because it indicates the level of interference to the PU. Whereas, $P_p^e(Overall)$ represents the overall wrong prediction probability. $P_p^e(Overall)$ is an important measure from a SU's perspective because the goal of the SU is to minimize the interference to the PUs while maximizing its own transmission opportunities. We also investigate the probability of wrongly predicting the idle channel status (i.e., the channel is predicted to be busy when it is actually idle or so-called false-alarm probability), denoted by $P_p^e(Idle)$. For both prediction schemes, it is observed that for a given mean inter-arrival time t_{inter} and traffic intensity ρ , there are only small changes in the probability $P_p^e(Idle)$. Hence $P_p^e(Idle)$ is shown for different mean inter-arrival times t_{inter} but only for traffic intensity of $\rho = 0.5$.

3.4.1 Performance of the Predictors Under Stationary Traffic Conditions

First, the performance of the two predictors under stationary⁵ traffic conditions is evaluated. The traffic settings are same as those used in [35]. The length of the testing data for both the predictor models under each traffic setting is chosen as 30,000 slots. Fig. 3.4 shows the performance of the MLP predictor in predicting the busy channel status for various traffic scenarios. It can be seen that the MLP predictor performance improves when the traffic intensity ρ increases. This is because, for a given mean inter-arrival time t_{inter} , as ρ increases the number of slots occurring with busy channel status also increases which leads to more correlation in the PUs' channel occupancy data. However, when ρ is equal to 50% there is little correlation in the PU' channel occupancy data. Hence, the worst case scenario occurs when the traffic intensity ρ is 0.5 (mean ON time=mean OFF time). The correlation also decreases as the mean inter-arrival time decreases.

⁵Stationary means that for each simulation scenario, the primary user traffic statistics (i.e., the traffic intensity ρ and the mean inter-arrival time t_{inter}) remain unchanged over time.

Figure 3.4: Performance of the MLP predictor in predicting the busy channel status for various traffic scenarios.

Figure 3.5: Overall performance of the MLP predictor in predicting the channel status for various scenarios.

Fig. 3.5 shows the overall performance of the MLP predictor for various traffic scenarios. It can be seen that $P_p^e(Overall)$ is slightly higher than $P_p^e(Busy)$ under the same traffic scenario. This is because of the inclusion of the wrong predictions when channel status is idle. Table 3.1 shows the performance of the MLP predictor in predicting the idle channel status. The traffic intensity is set to 50%. It can be observed that the probability of wrongly predicting the idle channel status, $P_p^e(Idle)$, ranges from 4.8% to 10.1% when the mean ON/OFF time varies from 11 slots to 5 slots, respectively.

Table 3.1: Performance of the MLP predictor in predicting the idle channel status

Mean ON time (slots)	Mean OFF time (slots)	$P_p^e(Idle)$
5	5	0.101477
8	8	0.063528
9	9	0.058236
10	10	0.052713
11	11	0.047875

The performance of the HMM predictor in predicting the busy channel status for different traffic scenarios is shown in Fig. 3.6. Similar to the result shown in Fig. 3.4, for a given mean inter-arrival time t_{inter} , the performance improves as the traffic intensity ρ of the channel increases.

Fig. 3.7 shows the overall performance of the HMM predictor for various traffic scenarios. Similar to the MLP predictor, the HMM predictor also shows a slightly higher value for $P_p^e(Overall)$ than the probability $P_p^e(Busy)$ under the same traffic scenario. This is due to the same reason as observed for the MLP predictor. Table 3.2 shows the probability of wrongly predicting the idle channel status $P_p^e(Idle)$ for the HMM predictor when we set the traffic intensity to 50% and vary the mean ON/OFF time. It is observed that $P_p^e(Idle)$ ranges from 5.1% to 10.3% when the mean ON/OFF time varies from 11 slots to 5 slots, respectively.

Figure 3.6: Performance of the HMM predictor in predicting the busy channel status for various traffic scenarios.

Table 3.2: Performance of the HMM predictor in predicting the idle channel status

Mean ON time (slots)	Mean OFF time (slots)	$P_p^e(Idle)$
5	5	0.103467
8	8	0.067164
9	9	0.062685
10	10	0.056119
11	11	0.051153

3.4.2 Performance of the Predictors Under Non-stationary Traffic Conditions

Consider a licensed channel with different PU traffic distributions at different time intervals (e.g., each interval has 6000 slots) as shown in Table 3.3. Consider the training data for the MLP predictor to be generated during the interval $[t_0, t_1]$. The testing data for both the predictors is obtained during the interval $[t_1, t_6]$. The performances of the MLP and the HMM predictors are analyzed by calculating the percentage of wrong predictions in the

Figure 3.7: Overall performance of the HMM predictor in predicting the channel status for various scenarios.

duration $[t_1, t_6]$. The percentages of wrong predictions for the MLP and HMM predictors are 6.31% and 7.08% respectively. The performance of the MLP predictor can be improved by retraining the MLP for a short duration in each interval. For instance, when the MLP is retrained using 1000 observations at the beginning of each interval, the percentage of wrong predictions reduces to 4.07%.

Table 3.3: Example of a licensed channel with non-stationary traffic distribution

Time interval	t_{inter}	ρ
$[t_0, t_1]$	20	0.7
$[t_1, t_2]$	10	0.5
$[t_2, t_3]$	18	0.667
$[t_3, t_4]$	16	0.625
$[t_4, t_5]$	22	0.5
$[t_5, t_6]$	18	0.5

3.4.3 Performance Enhancement due to Channel Status Prediction

The advantages of applying the two channel status prediction schemes to spectrum sensing is demonstrated using two performance measures (to be explained)

- Percentage improvement in spectrum utilization, denoted by $SUT_{imp}(\%)$,
- Percentage reduction in sensing energy, denoted by $SE_{red}(\%)$.

3.4.3.1 Improvement in Spectrum Utilization

Consider a cognitive radio network containing N_c channels with different PU traffic distributions. Every channel is logically divided into slots and the slot size is constant over all the channels. Each SU is able to sense only one channel during a slot due to the hardware constraint. We also assume that every SU stores a short history of the sensing results for every channel. This information can be collected from neighbors over a common control channel (e.g. using cooperative means [17]).

Two types of SUs are considered, CR_{sense} device and $CR_{predict}$ device. Both device types are assumed to use the same sensing mechanism and have the same level of sensing accuracy. A CR_{sense} device randomly selects a channel at every slot and senses for PU activity on that channel, while a $CR_{predict}$ device individually predicts the PU activity of all channels based on their respective sensing history, before sensing. The channel to be sensed by the $CR_{predict}$ device is randomly selected among those channels with idle predicted status.

Spectrum utilization (SUT) can be defined as the ratio of the number of idle slots discovered by the secondary user to the total number of idle slots available in the system over a finite period of time (e.g., 30,000 slots).

$$SUT = \frac{\text{Number of idle slots sensed}}{\text{Total number of idle slots in } N_c \text{ channels}} . \quad (3.44)$$

The percentage improvement in spectrum utilization due to channel status prediction can be expressed as

$$SUT_{imp}(\%) = \frac{SUT_{predict} - SUT_{sense}}{SUT_{sense}} \quad (3.45)$$

where SUT_{sense} and $SUT_{predict}$ represent the spectrum utilization for the CR_{sense} and $CR_{predict}$ devices respectively. Substituting (3.44) in (3.45), $SUT_{imp}(\%)$ can be given by

$$SUT_{imp}(\%) = \frac{I_{predict} - I_{sense}}{I_{sense}} \quad (3.46)$$

where I_{sense} and $I_{predict}$ represent the number of idle slots sensed by the CR_{sense} and the $CR_{predict}$ devices respectively.

In the simulation, the system with different number of channels N_c (the channels are added sequentially according to Table 3.4) is considered. Table 3.5 shows that a $CR_{predict}$ device using MLP and HMM predictors can discover more idle slots than a CR_{sense} device. The percentage of improvement in spectrum utilization is more than 68% when the $CR_{predict}$ device uses MLP (HMM) predictor.

Table 3.4: Different channel models for the primary user system

Channel index	ρ	t_{inter}
1	0.5625	16
2	0.6667	18
3	0.5	10
4	0.6818	22
5	0.5	20
6	0.7	10
7	0.5	18
8	0.5	22
9	0.5	16
10	0.7	20

Table 3.5: Percentage improvement in the spectrum utilization due to channel status prediction

N_c	I_{sense}	$I_{predict,MLP}$	$I_{predict,HMM}$	$SUT_{imp}(\%),MLP$	$SUT_{imp}(\%),HMM$
3	11994	21811	20240	81.85	68.75
4	11060	23607	22047	113.44	99.34
5	11785	25771	24143	118.68	104.86
6	11105	25751	24097	131.89	116.99
7	11578	26297	24628	127.13	112.71
8	11866	26820	25155	126.02	111.99
9	12165	26978	25284	121.77	107.84
10	11658	26985	25309	131.47	117.09

3.4.3.2 Reduction in Sensing Energy

Consider a single channel scenario, a CR_{sense} device senses all the slots whereas a $CR_{predict}$ device only senses when the PU activity in the slot is predicted to be idle. In other words, when the slot status is predicted to be busy, the sensing operation is not performed, thereby sensing energy is saved. Both device types are assumed to use the same sensing mechanism and have the same level of sensing accuracy. If we assume one unit of sensing energy is required to sense one slot, then the total sensing energy required for a CR_{sense} device in a finite duration of time (e.g., 30,000 slots) can be given by

$$SE_{sense} = (\text{Total number of slots in the duration}) \times (\text{unit sensing energy}) \quad (3.47)$$

while the total sensing energy required by the $CR_{predict}$ device can be given by

$$SE_{predict} = SE_{sense} - ((B_{predict}) \times (\text{unit sensing energy})) \quad (3.48)$$

where $B_{predict}$ is the total number of busy slots predicted by the $CR_{predict}$ device.

Therefore, using Eqs. (3.47) and (3.48), the percentage reduction in the sensing energy can be given by

$$SE_{red}(\%) = \frac{SE_{sense} - SE_{predict}}{SE_{sense}} = \frac{B_{predict}}{\text{Total no. of slots}}. \quad (3.49)$$

Table 3.6 shows the percentage of reduction in the sensing energy for different traffic settings when $CR_{predict}$ devices with MLP predictor and HMM predictor are used respectively. It can be seen that as ρ increases for a given t_{inter} more busy slots are predicted and hence more sensing energy is saved.

Table 3.6: Percentage reduction in sensing energy for different traffic intensities ρ and different mean inter-arrival times t_{inter}

t_{inter}	ρ	$B_{predict,MLP}$	$B_{predict,HMM}$	$SE_{red}(\%),MLP$	$SE_{red}(\%),HMM$
10	0.5	15514	15479	51.71	51.6
10	0.7	21672	21811	72.24	72.7
16	0.5	15663	15636	52.21	52.12
16	0.5625	17608	17606	58.69	58.69
18	0.5	15885	15840	52.94	52.8
18	0.6667	21167	21230	70.56	70.77
20	0.5	15734	15757	52.45	52.8
20	0.7	21778	21871	72.59	72.9
22	0.5	15274	15242	50.91	50.81
22	0.6818	20795	20843	69.32	69.48

3.5 Implementation Issues

In this section, the aspects related to model selection and computational complexity are discussed.

3.5.1 Model Complexity of MLP Predictor

The MLP predictor can predict the channel status based on a small number of inputs (channel statuses of the past slots) τ . From the simulations, we found that for channel status prediction τ in the range of $[4, 10]$ slots is sufficiently long. However, there is no straightforward method to identify the optimal number of hidden layers and the optimal

number of neurons in each layer. In the simulations, the size of the MLP predictor is chosen arbitrarily. By selecting sufficient number of neurons and appropriate learning rate, the channel status predictor can be implemented using the MLP network. A detailed analysis of the parameter selection for the MLP network is explained in [64].

3.5.2 Computational Complexity of MLP Predictor

The MLP uses two computation phases during training, namely the *forward pass* and the *backward pass*. However, only the *forward pass* is required after training. Unlike the HMM model, the MLP predictor is trained only once and the training is done offline. The computational complexity of the MLP predictor is directly related to the network size. Consider the *forward pass*, with the addition of a neuron to the layer o , the number of multiplication and addition operations increases by $(L_{o-1} + L_{o+1})$ and $(L_{o-1} - 1) + L_{o+1}$ respectively, where L_{o-1} and L_{o+1} are the number of neurons in the layers preceding and succeeding layer o , respectively. However, if a layer o is added to the MLP network, the number of multiplication and addition operations increases by $L_o(L_{o-1} + L_{o+1})$ and $L_o(L_{o-1} - 1) + L_{o+1}(L_o - 1)$ respectively, where L_o denotes the number of neurons in layer o . The MLP predictor may suffer from the local minima problem which slows down the training. However, the faster versions of the BP algorithm can be used to overcome the local minima problem [69].

3.5.3 Model Complexity of HMM Predictor

In the simulations, the number of states N_S in the HMM predictor is kept the same when modeling different observation sequences $\mathbf{s}_{hmm}^\kappa = \{s_1, s_2, \dots, s_\kappa\}$ and predicting the symbol $s_{\kappa+1}$ following the sequences. However, it is not an optimal solution, since $N_S = 10$ might not yield the maximum likelihood probability $P(\mathbf{s}_{hmm}^\kappa | \cdot)$ for some observation sequences. To determine the optimal number of states for modeling an observation sequence is extremely challenging and even infeasible in practice because of the large number of trials needed. Hence for practical reason, the number of states is fixed during the simulations.

Table 3.7 shows the performance of the HMM predictor for different states N_S . For this experiment, the length of the observation sequences $\kappa = 80$, maximum number of iterations $I_{max} = 10$ and the traffic model has mean inter-arrival time $t_{inter} = 20$ and $\rho = 0.5$. It can be seen that the predictor performance improves with the increase of the number of states. However, as the number of states N_S increases, the number of model parameters to be estimated $\theta = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ also increases.

Table 3.7: Performance of HMM predictor for different number of states N_S

N_S	$P_p^e(Busy)$	$P_p^e(Overall)$	$P_p^e(Idle)$
3	0.097910	0.107110	0.089215
6	0.061735	0.064784	0.058853
10	0.059035	0.062576	0.055687
15	0.058110	0.061891	0.054536

3.5.4 Computational Complexity of HMM Predictor

To predict the channel status at the $(\kappa + 1)^{th}$ time instant we have to compute the estimates of the model $\theta = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ such that the probability $P(\mathbf{s}_{hmm}^\kappa | \theta)$ is maximized. Unlike the MLP predictor, the HMM predictor training is done in real time (online learning). As shown in Fig. 3.2, the estimation and prediction steps are repeated when a new observation $\mathbf{s}_{hmm}^\kappa$ is made. Hence, it is necessary that all computations are completed before the occurrence of the $(\kappa + 1)^{th}$ slot for the HMM predictor to be applicable.

The length of the observation sequence is chosen as $\kappa = 80$ slots so that the state transition probabilities and state emission probabilities are reliably estimated. By increasing the value of κ , a better estimate of the model $\theta = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ can be obtained. However, if larger values of κ are used in realtime then the HMM has to wait for a longer duration to collect the observation symbols. Alternately, we may investigate approaches whereby κ can be iteratively reduced.

The computational complexity of the HMM predictor is related to the number of states N_S and the size of the observation sequence κ . Consider the computation of the forward variable $\alpha_n(g)$ which is computed after training, it takes approximately $N_S^2 \kappa$ calculations.

3.6 Qualitative Comparison

The HMM based channel status prediction schemes [35, 36] mentioned in Section 2 do not provide details of the model like number of states N_S and length of the observation sequence κ , hence making it difficult for comparison with the two proposed channel status prediction schemes. In this section, the pros and cons of the two proposed channel status prediction schemes is provided through a qualitative comparison of their performance and implementation issues which arise during the design. Table 3.8 shows a comparison of the properties of the two proposed prediction schemes.

Table 3.8: Qualitative comparison of the two channel status prediction schemes

Property	MLP predictor	HMM predictor
Adaptive parameters	weights $\omega_{h,g}^o$	probabilities $\mathbf{A}, \mathbf{B}$ and $\boldsymbol{\pi}$
No. of slot observations	τ	κ
No. of adaptive parameters	$\sum_o L_o L_{o+1}$ ^a	N_S
Training algorithm	BP algorithm [64]	BWA [65]
Training criteria	minimize mean square error E	maximize $P(\mathbf{s}_{hmm}^\kappa \cdot)$
Mode of training	offline, only once	online, repeated
Problems during training	local minima problem	no problems
No. of calculations in prediction after training	$\sum_o (L_o L_{o+1} + L_{o+1}(L_o - 1))$	$N_S^2 \kappa$

^a: L_o denotes the number of neurons in the layer o of the MLP predictor, o ranges from input layer to the last hidden layer. For the input layer, L_o simply denotes the number of inputs.

3.6.1 Performance Comparison

It is evident from the simulations that the two prediction schemes perform similarly under the same traffic scenario while the performance of MLP predictor is slightly better than that of the HMM predictor. This is because, the MLP predictor is a fully trained model whereas the HMM predictor is not optimally designed to cater to all observation sequences, because the number of states in the HMM is fixed.

Further, the MLP predictor requires fewer past observations (or slot occupancy history) than the HMM predictor to predict the channel status in the next slot.

The MLP predictor is trained only once whereas the HMM predictor is trained repeatedly (see Fig. 3.2). However, for time varying traffic scenario, the MLP predictor can be retrained periodically for better performance.

3.6.2 Design Comparison

During the training, the MLP predictor can get trapped in local minima but the HMM predictor faces no such problem. However, the local minima problem can be remedied by using modified versions of the BP algorithm [69].

Both the prediction schemes require considerable computations and memory to store the parameters for the training process. However, after training, the memory requirement for the MLP predictor is significantly reduced. The number of parameters in the MLP predictor is higher than that of the HMM predictor. However, the number of computation after training is lower for the MLP predictor than that for the HMM predictor. In our simulations, the MLP predictor has 380 parameters while the HMM has 130 parameters while the number of calculations after training is 724 for MLP predictor and 8000 for HMM predictor.

3.7 Summary

This chapter highlighted the benefits of channel status prediction to spectrum sensing (i.e. role in reducing sensing overhead and improving spectrum utilization). Adaptive schemes offer scope for the design of the channel status predictor, especially when the PU traffic statistics are difficult to be determined. Two such adaptive schemes for channel status predictor design were explored. Based on these prediction schemes the sensing energy could be reduced by more than 50% and the spectrum utilization could be improved by more than 68% for the PU traffic settings considered in this work. The two prediction schemes were also qualitatively compared based on the implementation issues and performance analysis.

Chapter 4

Queue-aware Frame-based Opportunistic Scheduling Scheme for CRN

This chapter describes the design and performance analysis of the new queue-aware frame-based opportunistic spectrum scheduling scheme for the centralized CRN. The proposed scheduling scheme has a dual objective, i.e., to maximize SU throughput and minimize the scheduling overhead.

4.1 Research Highlights

The proposed scheduling scheme is different from the existing scheduling schemes [22, 39–43, 52–56]¹ in various aspects which are explained below.

- (1) Unlike the existing scheduling schemes [22, 39–43, 52–56] in which the central controller schedules the SUs' transmissions on a slot-by-slot basis, the proposed scheduling scheme schedules the SUs transmissions over a larger time duration called *frame* which consists of several slots. Therefore the proposed scheduling scheme requires lesser scheduling overhead per frame compared to the slot-based scheduling schemes [22, 39–43, 52–56].
- (2) None of the existing scheduling schemes [22, 39–43, 52–56] have ever considered the queue status of the SU (i.e. the instantaneous number of packets in the SU's transmit

¹These DSA policies were referred as discrete time DSA policies in the Chapter 2

buffer) while allocating the licensed channels to the SUs. These scheduling schemes [22, 24, 25, 39–43, 52–56] work on the principle that the SU always has the packet to transmit (i.e., throughput always increases when more bandwidth is allocated to that SU). For example, consider the case when a SU which does not have sufficient number of packets in its buffer, is allocated a large bandwidth. This results in poor bandwidth utilization and lower throughput. Different from these scheduling schemes, the proposed scheduling scheme takes the queue status of the SU into consideration while scheduling its transmission to achieve higher throughput.

In the following section, the system model and the related assumption are explained. Following this, the design of the new queue-aware frame-based opportunistic spectrum scheduling scheme is presented. Thereafter, the performance analysis and results are discussed.

4.2 System Model and Assumptions

Consider a CRN in which the spectrum scheduling is performed by a central controller referred in this work as the central scheduler (CS), as shown in Fig. 4.1. The CRN is composed of V SUs sharing N_c channels. A low rate common control channel is assumed to be available to the SUs and the CS to exchange the control information (e.g. transmission requests and schedule). Such a control channel configuration is commonly assumed in the literature (e.g., [41]). Slotted transmission is assumed on the channels. The channel condition (i.e., PU activity, signal-to-noise ratio (SNR) and fading) is assumed to vary on a slot-by-slot basis. Scheduling is performed over a large time duration called a frame which is composed of several slots (say N_{slot} slots). The scheduling operation is done at the beginning of each frame. Let each SU be equipped with an adaptive modulation and coding (AMC) mechanism to cope with the time varying channel quality. The SU is assumed to have one transceiver (i.e., can transmit on one channel at a time). Thus, the CS can at most allocate one channel to a SU.

Figure 4.1: System model: Queue-aware frame-based opportunistic spectrum scheduling scheme.

Let each SU have a transmission queue of size Q_v packets. The packet arrivals are assumed to occur independently for each SU and follow a batch Bernoulli process with a mean arrival rate of λ_v packets per slot. The mean arrival rate λ_v can be expressed as

$$\lambda_v = \sum_{n_p=0}^{N_p} n_p \cdot P(A_p = n_p) \quad (4.1)$$

where $P(A_p = n_p)$ represents the probability of n_p packet arrivals in a slot and N_p denotes the maximum number of packet arrivals.

The probabilities of state transition for the PU activity and the channel quality are assumed to be known by the SUs based on the observation of history data. Similar assumptions about the PU activity and channel quality can be found in [30] and [70], respectively. The PU activity in slot n is denoted as s_n . s_n takes the value 0 when channel is being occupied by the PU, and 1 when the channel is idle. Similar to [30], the primary user activity is modeled as a two-state Markov chain with transition probability matrix $\mathbf{P}(s_{n+1}|s_n)$ given by

$$\mathbf{P}(s_{n+1}|s_n) = \begin{pmatrix} P(0|0) & P(1|0) \\ P(0|1) & P(1|1) \end{pmatrix} \quad (4.2)$$

where $P(s''|s')(s', s'' \in \{0, 1\})$ represents the transition probability from state s' in the current slot to state s'' in the next slot. For a system with N_c channels, let $\mathbf{P}(s_{n+1}^u|s_n^u)$ denote the transition probability matrix corresponding to channel u for $u \in \{1, \dots, N_c\}$.

Similar to [70], we assume that the channel quality (SNR) at the receiving SU follows Nakagami- m distribution. According to [70], for a given target packet error rate (PER) and average SNR $\bar{\gamma}$, the instantaneous SNR range for channel u corresponding to user v can be partitioned into C SNR regions $[\gamma_c^{u,v}, \gamma_{c+1}^{u,v})$ ($c \in \{0, 1, \dots, C-1\}$), where $\gamma^{u,v}$ denotes the instantaneous SNR. Each SNR region is denoted by mode $c^{u,v} \in \{0, 1, \dots, C-1\}$. The SNR region $[\gamma_0^{u,v}, \gamma_1^{u,v})$ corresponds to the mode $c^{u,v} = 0$ in which the transmission is not allowed. When the channel quality (i.e., SNR of user v 's transmission on channel u) is measured in slot n , its corresponding SNR region is determined and subsequently the corresponding modulation and coding mode is found. The mode in slot n is denoted by $c_n^{u,v}$. Based on mode $c_n^{u,v}$, the AMC of SU v selects a transmission rate $R_{c_n} = b_n c_n^{u,v}$, where b_n is a parameter specified based on system requirements (choice of b_n is explained in Section 4.3.1). [70] provides a procedure to model the fading channel quality in a slot as a finite state Markov chain (FSMC) with C states. The transition probability matrix for the mode $c_n^{u,v}$ is denoted by $\mathbf{P}(c_{n+1}^{u,v}|c_n^{u,v})$ for every channel u associated with SU v . $P(c_{n+1}^{u,v} = c''|c_n^{u,v} = c')$ denotes the probability that the mode status changes from mode c' in slot n to mode c'' in slot $n+1$. It is worthy to note that the FSMC model characterizes the change in the channel quality for the SUs under the condition that the channel is not occupied by the PUs. When the PU is active, SUs are prohibited to transmit on the corresponding channel.

4.3 Opportunistic Spectrum Scheduling Scheme

In this section, the proposed opportunistic spectrum scheduling scheme is presented. At the beginning of each frame, the SUs observe the PU activity and channel quality (i.e., SNR) for every channel. Based on these observations and the current queue status, every SU estimates the expected number of packet departures (i.e., the number of packets to be transmitted) over the frame corresponding to each channel (to be explained in Section

4.3.1). After calculating the expected number of packet departures corresponding to each channel, the SU sends a control packet to the CS indicating the user index, indexes of the channels and their corresponding expected number of packet departures. Once obtaining the control packets from all the SUs, the CS executes the scheduling algorithm (to be explained in Section 4.3.2) and broadcasts a control packet containing the schedule to all the SUs. The schedule contains the index of every channel and the index of the SU assigned to the channel. After receiving the schedule, the SUs assigned a channel will perform channel sensing at the beginning of each slot of their respective channels and start data transmission when the channel is idle.

4.3.1 Estimation of Packet Departures

In this section, the Markov chain (MC) formulation to calculate the expected number of packet departures $T_F(u, v)$ over the frame F for SU v corresponding to channel u , is presented. $T_F(u, v)$ represents the expected number of packet departures by SU v if channel u is to be assigned by the central scheduler. A new state vector $\mathbf{x}_n^{u,v} = [s_n^u, c_n^{u,v}, q_n^v]$ ($n = 1, 2, \dots, N_{slot}$) is introduced for this calculation, where s_n^u represents the primary user activity (i.e., busy or idle) on channel u at slot n in a frame, $c_n^{u,v}$ represents the mode for the channel u for user v at slot n , and q_n^v represents the queue status of user v at the beginning of slot n in the frame. The Markov chain in Fig. 4.2 shows the evolution of state $\mathbf{x}_n^{u,v}$ from the beginning of the frame (i.e., slot $n = 1$) to the end of the frame (i.e., slot $n = N_{slot}$). Since the MC formulation is the same for all channel-user pairs, for simplicity of presentation, the state vector for any channel-user pair is indicated as $\mathbf{x}_n = [s_n, c_n, q_n]$.

Figure 4.2: Markov chain evolution for state $\mathbf{x}_n^{u,v}$ inside a frame.

Let $T_n(u, v)$ denote the expected number of packet departures in the slot n for a given channel-user pair (u, v) (for presentation simplicity, it is denoted as T_n for the rest of

chapter). In order to calculate T_n , the probability distribution of state vector $\mathbf{x}_n$ in every slot of the frame should be known. However, state $\mathbf{x}_n$ is known only for the first slot in the frame. State $\mathbf{x}_n$ for the remaining slots in the frame is found based on the MC shown in Fig. 4.2. The probability distribution for state vector $\mathbf{x}_n$ in slot n is denoted by

$$\mathbf{\Pi}_n = [\Pi_{(s_n=s', c_n=c', q_n=q')}], \quad (4.3)$$

for $s' \in \{0, 1\}$, $c' \in \{0, 1, \dots, C-1\}$, and $q' \in \{0, 1, \dots, Q_v\}$, where $\Pi_{(s_n=s', c_n=c', q_n=q')}$ represents the probability that the values of state variables s_n , c_n and q_n are s' , c' , and q' , respectively. The state vector for the first slot in the frame is denoted as $\mathbf{x}_1 = [s_1, c_1, q_1]$. According to the proposed scheduling scheme, the value of this state vector is obtained from the observation of the channel and queue status at the beginning of each frame. Let $\hat{s}$, $\hat{c}$ and $\hat{q}$ denote the observed values for the primary user activity, channel mode, and initial queue status, respectively. The corresponding probability distribution is denoted by

$$\mathbf{\Pi}_1 = \left[\Pi_{(s_1=s', c_1=c', q_1=q')} = \begin{cases} 1 & s' = \hat{s}, c' = \hat{c}, q' = \hat{q} \\ 0 & \text{otherwise.} \end{cases} \right], \quad (4.4)$$

for $s' \in \{0, 1\}$, $c' \in \{0, 1, \dots, C-1\}$, and $q' \in \{0, 1, \dots, Q_v\}$.

The probability distribution of state vector $\mathbf{x}_n$ in the remaining slots of the frame is calculated based on state transition matrix $\mathbf{P}(\mathbf{x}_{n+1}|\mathbf{x}_n)$. Let a state transition occur at the end of every slot with probability $P(\mathbf{x}_{n+1}|\mathbf{x}_n)$, also denoted as $P(s_{n+1} = s'', c_{n+1} = c'', q_{n+1} = q'' | s_n = s', c_n = c', q_n = q')$ where $s', s'' \in \{0, 1\}$, $c', c'' \in \{0, 1, \dots, C-1\}$ and $q', q'' \in \{0, 1, \dots, Q_v\}$. State transition matrix $\mathbf{P}(\mathbf{x}_{n+1}|\mathbf{x}_n)$ for the MC is given by

$$\mathbf{P}(\mathbf{x}_{n+1}|\mathbf{x}_n) = \begin{pmatrix} P_{(0,0,0|0,0,0)} & P_{(0,0,1|0,0,0)} & \cdots & P_{(1,C-1,Q_v|0,0,0)} \\ P_{(0,0,0|0,0,1)} & P_{(0,0,1|0,0,1)} & \cdots & P_{(1,C-1,Q_v|0,0,1)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(0,0,0|1,C-1,Q_v)} & P_{(0,0,1|1,C-1,Q_v)} & \cdots & P_{(1,C-1,Q_v|1,C-1,Q_v)} \end{pmatrix} \quad (4.5)$$

where the transition probability $P_{(s'',c'',q''|s',c',q')}$ is calculated as follows:

$$\begin{aligned}
 P_{(s'',c'',q''|s',c',q')} &= P(s_{n+1} = s'', c_{n+1} = c'', q_{n+1} = q'' | s_n = s', c_n = c', q_n = q') \\
 &= \frac{P(s_{n+1} = s'', c_{n+1} = c'', q_{n+1} = q'', s_n = s', c_n = c', q_n = q')}{P(s_n = s', c_n = c', q_n = q')} \\
 &= \frac{P(q_{n+1} = q'' | s_n = s', c_n = c', q_n = q', s_{n+1} = s'', c_{n+1} = c'')}{P(s_n = s', c_n = c', q_n = q')} \\
 &\quad \cdot P(s_n = s', c_n = c', q_n = q', s_{n+1} = s'', c_{n+1} = c'') \\
 &= P(q_{n+1} = q'' | s_n = s', c_n = c', q_n = q', s_{n+1} = s'', c_{n+1} = c'') \\
 &\quad \cdot P(s_{n+1} = s'', c_{n+1} = c'' | s_n = s', c_n = c', q_n = q'). \tag{4.6}
 \end{aligned}$$

In (4.6), state variables s_{n+1} , c_{n+1} and q_{n+1} are independent of one another. Further, s_{n+1} does not depend on q_n and c_n , similarly c_{n+1} does not depend on q_n and s_n . Therefore, (4.6) can be expressed as

$$\begin{aligned}
 P_{(s'',c'',q''|s',c',q')} &= P(q_{n+1} = q'' | s_n = s', c_n = c', q_n = q') \\
 &\quad \cdot P(s_{n+1} = s'' | s_n = s') \cdot P(c_{n+1} = c'' | c_n = c'). \tag{4.7}
 \end{aligned}$$

In (4.7), the value of $P(s_{n+1} = s'' | s_n = s')$ is obtained from (4.2) and the value of $P(c_{n+1} = c'' | c_n = c')$ is obtained from the FSMC model for the corresponding channel-user pair. Conditional probability $P(q_{n+1} = q'' | s_n = s', c_n = c', q_n = q')$ is calculated as follows.

The queue status q_{n+1} at the beginning of slot $n + 1$ for any secondary user can be expressed as

$$q_{n+1} = q_n - d_n + a_n \tag{4.8}$$

where a_n represents number of packets entering the queue in slot n whereas d_n represents the number of packet departures in slot n . The number of packet departures is calculated from

$$d_n = \min(q_n, b_n s_n c_n) = \min(q', b_n s' c'). \tag{4.9}$$

From (4.8) and (4.9), we have $a_n = q'' - q' + \min(q', b_n \cdot s' \cdot c')$, then the conditional probability $P(q_{n+1} = q'' | s_n = s', c_n = c', q_n = q')$ is obtained from

$$P(q_{n+1} = q'' | s_n = s', c_n = c', q_n = q') = \begin{cases} P(A_p = a_n) & \text{if } q'' < Q_v, 0 \leq a_n \leq N_p \\ \sum_{a'_n=a_n}^{N_p} P(A_p = a'_n) & \text{if } q'' = Q_v, 0 \leq a_n \leq N_p \\ 0 & \text{otherwise.} \end{cases} \quad (4.10)$$

Note that some of the arriving packets will be dropped if the queue is full.

Based on the above derivation for $\mathbf{P}(\mathbf{x}_{n+1} | \mathbf{x}_n)$, the probability distribution of state vector $\mathbf{x}_n$ in the remaining slots in the frame is found recursively from

$$\mathbf{\Pi}_{n+1} = \mathbf{\Pi}_n \cdot \mathbf{P}(\mathbf{x}_{n+1} | \mathbf{x}_n) \quad (4.11)$$

for $n = 1, 2, \dots, N_{slot} - 1$. Therefore, the expected number of packet departures T_n in slot n ($n \in \{1, 2, \dots, N_{slot}\}$) is calculated as follows:

$$T_n = \sum_{s', c', q'} \Pi_{(s_n=s', c_n=c', q_n=q')} \cdot \min(q_n = q', b_n \cdot s' \cdot c') \quad (4.12)$$

where $s' \in \{0, 1\}$, $c' \in \{0, 1, \dots, C - 1\}$ and $q' \in \{0, 1, \dots, Q_v\}$. The expected number of packet departures for the entire frame $T_F(u, v)$ (or simply T_F) for any channel-user pair is given by

$$T_F = \sum_{n=1}^{N_{slot}} T_n. \quad (4.13)$$

In the above derivation, b_n represents the number of packets that can be transmitted in a slot for the lowest transmission mode ($c_n = 1$). b_n depends on the effective transmission time in a slot after scheduling. Note that in the proposed scheduling scheme, a portion of the first slot in a frame is used for control information exchange (as shown in Fig. 4.3). Therefore, the value of b_1 is different from those in other slots. The parameter b_n is calculated as follows:

$$b_n = \begin{cases} T_p \cdot R_1 / L_p & \text{if } n = 1 \\ T_s \cdot R_1 / L_p & \text{if } n > 1 \end{cases} \quad (4.14)$$

where R_1 represents the transmission rate of any licensed channel when the transmission mode is $c_n = 1$, L_p represents the packet size in bits, T_s denotes the slot duration and

$T_p = T_s - T_{ov}$ is the transmission time remaining in the first slot of the frame. Here, T_{ov} denotes the time taken in exchanging the scheduling overhead, and it is given by $T_{ov} = L_{ov}/R_c$, where L_{ov} denotes the size of the scheduling overheads in bits whereas R_c is the transmission rate of the common control channel. L_{ov} is calculated as $L_{ov} = L_{ov}^{up} + L_{ov}^{down}$, where L_{ov}^{up} and L_{ov}^{down} are the uplink and downlink scheduling overheads expressed in bits, respectively. L_{ov}^{up} and L_{ov}^{down} can be obtained from

$$L_{ov}^{up} = V \cdot (N_c \cdot (u_{index} + L_{T_F}) + v_{index}) \quad (4.15)$$

$$L_{ov}^{down} = N_c \cdot (u_{index} + v_{index}) . \quad (4.16)$$

In (4.15), L_{T_F} denotes the number of bits needed to represent the expected number of packet departures over the frame $T_F(u, v)$ corresponding to the channel-user pair identified by the index values u and v , respectively. u_{index} and v_{index} denote the number of bits of index values u and v , respectively.

Figure 4.3: Structure of the frame in the proposed scheduling scheme.

4.3.2 Scheduling Algorithm

The CS allocates the channels to the secondary users according to the following algorithm.

- (1) Let $\mathcal{CS}_{u,v} = \{T_F(u, v)\}$ denote the set containing the expected number of packet departures over the frame for each channel-user pair, where u represents the channel index and v represents the user index.
- (2) Find the channel-user pair (u, v) with the highest value of $T_F(u, v)$. Ties are broken randomly if multiple channel-user pairs share the highest value. Let (u^*, v^*) denote the winning pair. Allocate channel u^* to user v^* .
- (3) Remove all the T_F values associated with channel u^* and user v^* from the set $\mathcal{CS}_{u,v}$.
- (4) Repeat the above from step 2 till either all channels or all users have been allocated.

4.4 Analytical Framework

In this section, an analytical model is developed to analyze the performance (e.g., throughput and packet loss) of the proposed scheduling scheme in a homogeneous network environment in which all SUs are considered to be identical. By finding the stationary distribution for the state of a tagged SU (e.g. user 1), various performance measures can be obtained.

Let $\mathbf{s}_F = [s_F^1, \dots, s_F^{N_c}]$ represent the PU activity on the N_c channels at the beginning of the frame F . Let $\mathbf{c}_F^v = [c_F^{1,v}, \dots, c_F^{N_c,v}]$ and q_F^v indicate the modes corresponding to N_c channels and the queue status observed by SU v at the beginning of the frame F , respectively. Here, $s_F^u \in \{0, 1\}$, $c_F^{u,v} \in \{0, 1, \dots, C - 1\}$ and $q_F^v \in \{0, 1, \dots, Q_v\}$ where $u \in \{1, \dots, N_c\}$ and $v \in \{1, \dots, V\}$ denote the channel index and the SU's index, respectively. Therefore, the state of the system observed by the v^{th} SU is represented as $\mathbf{y}_F^v = [s_F^1, \dots, s_F^{N_c}, c_F^{1,v}, \dots, c_F^{N_c,v}, q_F^v]$ or simply $\mathbf{y}_F^v = [\mathbf{s}_F, \mathbf{c}_F^v, q_F^v]$.

Let $\mathbf{y}_F^1$ represent the state of the tagged SU. The state distribution (probability distribution) of $\mathbf{y}_F^1$ is denoted by

$$\boldsymbol{\Pi}_F = [P(\mathbf{y}_F^1)] = [P(s_F^1, \dots, s_F^{N_c}, c_F^{1,1}, \dots, c_F^{N_c,1}, q_F^1)] \quad (4.17)$$

The evolution of state $\mathbf{y}_F^1$ is modeled as a discrete time Markov chain (DTMC) in which a state transition occurs at the end of the frame. The state transition matrix of the DTMC is

denoted by $\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1)$. The state distribution is updated by using the following expression

$$\mathbf{\Pi}_{F+1} = \mathbf{\Pi}_F \cdot \mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1) \quad \text{where } F = 1, 2, \dots \quad (4.18)$$

The stationary distribution is reached when the condition $\mathbf{\Pi}_{F+1} = \mathbf{\Pi}_F$ is satisfied. The stationary distribution is denoted by $\mathbf{\Pi}$.

4.4.1 Calculation of State Transition Matrix

The state transition matrix $\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1)$ in Eq. (4.18) is calculated as follows:

$$\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1) = \sum_{u'=0}^{N_c} P(u' | \mathbf{y}_F^1) \mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1, u'). \quad (4.19)$$

In the above equation, $P(u' | \mathbf{y}_F^1)$ represents the conditional probability of assigning channel u' to the tagged SU given the state of the tagged user $\mathbf{y}_F^1$, and $\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1, u')$ represents the transition probability matrix for the state $\mathbf{y}_F^1$ given that channel u' is assigned to the tagged SU. In (4.19), $u' = 0$ means that no channel is assigned to the tagged SU. The calculation of the conditional probabilities $\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1, u')$ and $P(u' | \mathbf{y}_F^1)$ is explained in the Sections 4.4.1.1 and 4.4.1.2, respectively.

4.4.1.1 Calculation for $\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1, u')$

The frame-by-frame state transition matrix $\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1, u')$ can be expressed in terms of the slot-by-slot transition matrix $\mathbf{P}(\mathbf{y}_{n+1}^1 | \mathbf{y}_n^1, u')$ as follows:

$$\mathbf{P}(\mathbf{y}_{F+1}^1 | \mathbf{y}_F^1, u') = \prod_{n=1}^{N_{slot}} \mathbf{P}(\mathbf{y}_{n+1}^1 | \mathbf{y}_n^1, u') \quad (4.20)$$

where $\mathbf{y}_n^1$ denotes the state of the tagged SU in the n^{th} slot of the frame F . $\mathbf{y}_n^1$ can be represented as $\mathbf{y}_n^1 = [s_n^1, \dots, s_n^U, c_n^{1,1}, \dots, c_n^{U,1}, q_n^1]$ or simply $\mathbf{y}_n^1 = [\mathbf{s}_n, \mathbf{c}_n^1, q_n^1]$, where $\mathbf{s}_n = [s_n^1, \dots, s_n^U]$, $\mathbf{c}_n^1 = [c_n^{1,1}, \dots, c_n^{U,1}]$ and q_n^1 are the PU activities (for N_c channels), corresponding channel modes and queue status, respectively, of the tagged SU at the beginning of the n^{th} slot of the frame F . Note that state $\mathbf{y}_F^1$ observed at the beginning of the frame is the same as state $\mathbf{y}_n^1$ observed in the slot $n = 1$.

The probability $P(\mathbf{y}_{n+1}^1 | \mathbf{y}_n^1, u')$ is given by

$$\begin{aligned}
 P(\mathbf{y}_{n+1}^1 | \mathbf{y}_n^1, u') &= \\
 P(q_{n+1}^1 = q'' | s_n^1 = s'_1, \dots, s_n^{u'} = s'_{u'}, \dots, s_n^{N_c} = s'_U, c_n^{1,1} = c'_1, \dots, c_n^{u',1} = c'_{u'}, \dots, c_n^{N_c,1} = c'_{N_c}, \\
 q_n^1 = q', u') \cdot \prod_{u=1}^{N_c} P(c_{n+1}^{u,1} = c''_u | c_n^{u,1} = c'_u) \cdot \prod_{u=1}^{N_c} P(s_{n+1}^u = c''_u | s_n^u = s'_u) \quad (4.21)
 \end{aligned}$$

where $s''_u, s'_u \in \{0, 1\}$, $c''_u, c'_u \in \{0, \dots, C-1\}$ and $q'', q' \in \{0, 1, \dots, Q_v\}$, for $u \in \{1, \dots, u', \dots, N_c\}$.

In (4.21), $P(s_{n+1}^u = s''_u | s_n^u = s'_u)$ is obtained from (4.2) whereas $P(c_{n+1}^{u,1} = c''_u | c_n^{u,1} = c'_u)$ is obtained from the FSMC model described in Section 4.2. Under the condition that the channel u' is assigned to the tagged SU, the parameter q_{n+1}^1 depends only on $s_n^{u'}$, $c_n^{u',1}$ and q_n^1 . Therefore, Eq. (4.21) is modified as

$$\begin{aligned}
 P(\mathbf{y}_{n+1}^1 | \mathbf{y}_n^1, u') &= P(q_{n+1}^1 = q'' | s_n^{u'} = s'_{u'}, c_n^{u',1} = c'_{u'}, q_n^1 = q') \\
 &\cdot \prod_{u=1}^{N-c} P(c_{n+1}^{u,1} = c''_u | c_n^{u,1} = c'_u) \cdot \prod_{u=1}^{N_c} P(s_{n+1}^u = s''_u | s_n^u = s'_u) . \quad (4.22)
 \end{aligned}$$

The probability $P(q_{n+1}^1 = q'' | s_n^{u'} = s'_{u'}, c_n^{u',1} = c'_{u'}, q_n^1 = q')$ in Eq. (4.22) is calculated by substituting parameters $s_n^{u'} = s'_{u'}$, $c_n^{u',1} = c'_{u'}$, $q_n^1 = q'$ and $q_{n+1}^1 = q''$ in Eq. (4.10).

4.4.1.2 Calculation for $P(u' | \mathbf{y}_F^1)$

The conditional probability $P(u' | \mathbf{y}_F^1)$ is calculated as the probability that channel u' is assigned to the tagged SU under all possible combinations of the states of other SUs in the system as follows:

$$\begin{aligned}
 P(u' | \mathbf{y}_F^1) &= \sum_{\forall \mathbf{y}_F^2, \dots, \mathbf{y}_F^V} P(u', \mathbf{y}_F^2, \dots, \mathbf{y}_F^V | \mathbf{y}_F^1) \\
 &= \sum_{\forall \mathbf{c}_F^2, q_F^2, \dots, \mathbf{c}_F^V, q_F^V} P(u', [\mathbf{s}_F, \mathbf{c}_F^2, q_F^2], \dots, [\mathbf{s}_F, \mathbf{c}_F^V, q_F^V] | [\mathbf{s}_F, \mathbf{c}_F^1, q_F^1]) \\
 &= \sum_{\forall \mathbf{c}_F^2, q_F^2, \dots, \mathbf{c}_F^V, q_F^V} P(u' | [\mathbf{s}_F, \mathbf{c}_F^1, q_F^1], [\mathbf{s}_F, \mathbf{c}_F^2, q_F^2], \dots, [\mathbf{s}_F, \mathbf{c}_F^V, q_F^V]) \\
 &\cdot \prod_{v=2}^V P([\mathbf{s}_F, \mathbf{c}_F^v, q_F^v] | \mathbf{s}_F) . \quad (4.23)
 \end{aligned}$$

In (4.23), the conditional probability $P([\mathbf{s}_F, \mathbf{c}_F^v, q_F^v] | \mathbf{v}_F)$ is given by

$$\begin{aligned} P([\mathbf{s}_F, \mathbf{c}_F^v, q_F^v] | \mathbf{s}_F) &= P(s_n^1 = s'_1, \dots, s_n^{N_c} = s'_{N_c}, \mathbf{c}_F^v, q_F^v | s_n^1 = s'_1, \dots, s_n^{N_c} = s'_{N_c}) \\ &= \frac{P(s_n^1 = s'_1, \dots, s_n^{N_c} = s'_{N_c}, \mathbf{c}_F^v, q_F^v)}{\sum_{s_n^1 = s'_1, \dots, s_n^{N_c} = s'_{N_c}} P(s_n^1 = s'_1, \dots, s_n^{N_c} = s'_{N_c}, \mathbf{c}_F^v, q_F^v)} . \end{aligned} \quad (4.24)$$

In the above equation, $P(s_n^1 = s'_1, \dots, s_n^{N_c} = s'_{N_c}, \mathbf{c}_F^v, q_F^v)$ belongs to the distribution of state $\mathbf{y}_F^v = [\mathbf{s}_F, \mathbf{c}_F^v, q_F^v]$ corresponding to SU v (where $v \in \{2, \dots, V\}$). As all the SUs are homogeneous, this probability distribution is the same as $\mathbf{\Pi}$, which can be obtained from Eq. (4.18). The conditional probability $P(u' | [\mathbf{s}_F, \mathbf{c}_F^1, q_F^1], [\mathbf{s}_F, \mathbf{c}_F^2, q_F^2], \dots, [\mathbf{s}_F, \mathbf{c}_F^V, q_F^V])$ describes the probability that the proposed scheduling algorithm assigns channel u' to the tagged user² under the specified condition. For presentation simplicity, it is denoted as $P(u' | \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$. The algorithm to obtain $P(u' | \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$ is given as follows.

- (1) Let $\mathcal{CS}_{u,v} = \{T_F(u, v)\}$ denote the set containing the expected number of packet departures over the frame T_F of all the channel-user pairs in the system, where $u \in \{1, \dots, N_c\}$ and $v \in \{1, \dots, V\}$ denote the channel and SU indexes, respectively.
- (2) Obtain conditional probabilities $P(m_1, \dots, m_{t-1}, m_t = (u', 1) | \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$, where $M = \{m_1, \dots, m_{t-1}, m_t = (u', 1)\}$ is one of the possible outcomes from the proposed scheduling algorithm, representing any ordered sequence of channel-user pairs in which the t^{th} element is the pair $(u', 1)$. The pair $(u', 1)$ means that the channel u' is assigned to user 1 (i.e., tagged SU). Here, $t \in \{1, \dots, M_{uv}\}$, and $M_{uv} = \min(N_c, V)$ represents maximum length of the sequence. To calculate $P(m_1, \dots, m_{t-1}, m_t = (u', 1) | \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$, we check if the elements of the sequence can be assigned sequentially using the scheduling algorithm described in Section 4.3.2, as follows:

²Note that the CS allocates a channel-user pair based on the expected number of packet departures over a frame $T_F(u, v)$ of all channel-user pairs. $T_F(u, v)$ of a channel-user pair (u, v) depends on the state of that user v .

Initialization:

$$P(m_1|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V) = \begin{cases} 1 & \text{if } T_F \text{ for the channel-user pair } m_1 \text{ is the only highest value in } \mathcal{CS}_{u,v} \\ 1/K & \text{if } m_1 \text{ shares the highest value of } T_F \text{ in } \mathcal{CS}_{u,v} \text{ with other } K-1 \text{ pairs} \\ 0 & \text{otherwise} \end{cases} \quad (4.25)$$

Recursion: For every $t' \in \{2, \dots, t\}$

$$P(m_1, \dots, m_{t'-1}, m_{t'}|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V) = P(m_{t'}|m_1, \dots, m_{t'-1}, \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V) \cdot P(m_1, \dots, m_{t'-1}|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V) \quad (4.26)$$

To obtain $P(m_{t'}|m_1, \dots, m_{t'-1}, \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$, elements of $\mathcal{CS}_{u,v}$ associated with the channels and users assigned up to the $(t' - 1)^{th}$ pair in the sequence M are removed from the set, and resulting set is denoted as $\mathcal{CS}'_{u,v}$.

$$P(m_{t'}|m_1, \dots, m_{t'-1}, \mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V) = \begin{cases} 1 & \text{if } T_F \text{ for the channel-user pair } m_{t'} \text{ is the only highest value in } \mathcal{CS}'_{u,v} \\ 1/K & \text{if } m_{t'} \text{ shares the highest value of } T_F \text{ in } \mathcal{CS}'_{u,v} \text{ with other } K-1 \text{ pairs} \\ 0 & \text{otherwise} \end{cases} \quad (4.27)$$

- (3) Probability $P(u'|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$ is found by summing all possible conditional probabilities

$$P(m_1, \dots, m_{t-1}, m_t = (u', 1)|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V).$$

- (4) Finally, the conditional probability that no channel is assigned to user 1, denoted by $P(u' = 0|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V)$ is calculated as follows:

$$P(u' = 0|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V) = 1 - \sum_{u=1}^{N_c} P(u' = u|\mathbf{y}_F^1, \mathbf{y}_F^2, \dots, \mathbf{y}_F^V). \quad (4.28)$$

4.4.2 Performance Measures

In this section, the performance measures are obtained based on the stationary distribution $\mathbf{\Pi}$ of state of the tagged SU. The stationary distribution is also denoted by $\mathbf{\Pi} = [P(s^1, \dots, s^{N_c}, c^{1,1}, \dots, c^{N_c,1}, q^1)]$.

- Queue distribution, $\mathbf{\Pi}_{q^1}$: The queue distribution gives the information about the probability of finding q' packets in the queue. The queue distribution is obtained from the stationary distribution using the following expressions

$$\begin{aligned} \mathbf{\Pi}_{q^1} &= [\Pi_{q^1=q'}] \quad \text{where } q' \in \{0, 1, \dots, Q_v\} \\ \Pi_{q^1=q'} &= \sum_{\forall s^1, \dots, s^{N_c}, c^{1,1}, \dots, c^{N_c,1}} P(s^1, \dots, s^{N_c}, c^{1,1}, \dots, c^{N_c,1}, q^1 = q'). \end{aligned} \quad (4.29)$$

- Average SU throughput, η : The average SU throughput η represents the average number of packets which are transmitted successfully (i.e., without packet transmission error) by that SU per slot. η (expressed in packets per slot) for the tagged SU (user 1) is given by $\eta = \eta_F / N_{slot}$, where η_F represents the average SU throughput over a frame. η_F is given by

$$\eta_F = \sum_{\forall \mathbf{y}_F^1} \left[\sum_{u'=1}^{N_c} T_F^{\mathbf{y}_F^1}(u', 1) P(u' | \mathbf{y}_F^1) \right] P(\mathbf{y}_F^1) (1 - PER). \quad (4.30)$$

In the above equation, PER is the packet error rate and $T_F^{\mathbf{y}_F^1}(u', 1)$ represents the expected number of packet departures over the frame $T_F(u', 1)$ for the tagged SU corresponding to channel u' and the observed state $\mathbf{y}_F^1$. (Calculation of $T_F(u, v)$ for any channel-user pair (u, v) given any observed state is explained in Section 4.3.1.)

- Blocking probability, P_B : The probability of blocking represents the probability that a arriving packet is dropped at the queue (as the queue status is full). The blocking probability P_B is related to the average SU throughput η by

$$P_B = 1 - \eta / (\lambda_v (1 - PER)). \quad (4.31)$$

- Mean waiting time of a packet, $\bar{W}$: Using Little's law [71], the mean waiting time of a packet in a SU's queue $\bar{W}$ is expressed as $\bar{W} = \bar{q}/\lambda'_v$, where $\lambda'_v = \lambda_v(1 - P_B)$ is the effective arrival rate of the packets successfully entering a SU's queue and $\bar{q}$ represents the mean queue size, given by

$$\bar{q} = \sum_{q'=0}^{Q_v} q' \cdot \Pi_{q^1=q'} \quad (4.32)$$

4.5 Results and Discussion

In this section, firstly, the accuracy of the analytical model is evaluated by comparing the performance measures with those obtained from the simulation. The simulations were performed using MATLAB. The performance of the proposed scheduling scheme is also done with the slot-based scheduling schemes. A slot-based scheduling scheme (e.g., [41]) can be realized by setting the frame size $N_{slot} = 1$ slot in the proposed scheduling scheme. However, the proposed scheduling scheme can schedule frames of different sizes.

The following simulation parameters are used for the system model. All SUs have the same mean arrival rate (λ_v). The maximum number of packet arrivals in slot N_p is chosen according to the experiment. The slot duration T_s is set to $2ms$. The packet size L_p is set to 512 bits. The lowest transmission rate R_1 for any licensed channel is set to 2 Mbps while the control channel rate R_c is varied. The number of channels N_c and the number of nodes V are also varied according to the experiment. The traffic intensity of any channel due to primary user activity is at least 50%. The following values are chosen to simulate the channel quality for each channel: The Nakagami fading factor (m) is equal to 1. The desired PER is equal to 0.01. The average SNR $\bar{\gamma}$ is chosen as 15dB while the doppler spread f_d is chosen as 30Hz. The number of modes C is chosen according to the experiment.

4.5.1 Comparison of Simulation and Analysis

For this experiment, a system model with $V = 3$ nodes and $N_c = 2$ channels is used. The number of modes for each channel is set to $C = 2$. The maximum number of packet arrivals

in a slot is $N_p = 3$ packets. The buffer size of the nodes is set to $Q_v = 20$ packets. The scheduling overhead is $L_{ov} = 296$ bits. The control channel rate is set to $R_c = 300$ kbps. Accordingly, the slot dependent parameter b_n is calculated as 4 and 7 when $n = 1$ and $n > 1$, respectively. The accuracy of the analytical model is evaluated in terms of the performance measures such as the mean queue size of the tagged SU, the mean waiting time of a packet in the tagged SU's queue, the average SU throughput and the blocking probability.

In the analysis, the state distribution converges to the stationary distribution $\mathbf{\Pi}$ in less than 15 state transitions. Fig. 4.4 shows the convergence of the probability $\Pi_{q^1=q'}$ for different values of q' when $N_{slot} = 4$ slots and $\lambda_v = 2$ packets/slot. For the simulation, the queue distribution is obtained by taking the long term average of total number of occurrences of each queue size $q' = \{0, 1, \dots, Q_v\}$ for the tagged SU over a duration of 200,000 slots.

Figure 4.4: Convergence of probability of queue size $q^1 = q'$ (i.e., $\Pi_{q^1=q'}$) for different values of q' .

The mean queue size (expressed in packets) for different mean arrival rates is given in Fig. 4.5. It can be observed that the mean queue size calculated from the analysis matches well with those obtained from the simulation. The analytical and simulation results of the mean waiting time for a packet in the tagged SU's queue (expressed in slots) for different mean arrival rates are shown in Fig. 4.6. Again, the analytical and simulation results match well. It can be observed that as the mean arrival rate increases, the mean waiting time of the packet in a SU's queue also increases.

Figure 4.5: Mean queue size for different mean arrival rates.

Figure 4.6: Mean waiting time for a packet in the queue for different mean arrival rates.

Fig. 4.7 provides the average SU throughput (packets per slot) of the tagged SU for different mean arrival rates λ_v . For the simulation, η is calculated as the ratio of the total number of packet departures for the tagged SU divided by the total simulation time. It can be observed that similar values are obtained from the analysis and the simulation. Fig. 4.8 shows the blocking probability for different mean arrival rates. For the simulation, P_B is obtained as the ratio of the total number of packets dropped by the tagged user (during the simulation) to the total number of arriving packets. It can be observed that the blocking probability obtained from the analytical model accords closely with those obtained from the simulation. It can also be observed that as the mean arrival rate increases, the blocking probability also increases.

Figure 4.7: Average SU throughput for different mean arrival rates.

Figure 4.8: Blocking probability for different mean arrival rates.

4.5.2 Selecting the Optimal Frame Size

The optimal frame size N_{opt} for a given network setting is determined from the analysis by finding the frame size for which the average SU throughput η is the highest. We consider the network setting with $V = 3$ SUs and $N_c = 2$ channels. The mean arrival rate $\lambda_v = 2$ packets/slot. A maximum of $N_p = 3$ packets arrive at a SU's buffer which can support up to $Q_v = 20$ packets. The scheduling overhead is $L_{ov} = 296$ bits. The optimal frame size is found for two different settings of the control channel rate R_c . Fig. 4.9 shows that for the setting $R_c = 300$ kbps, the optimal frame size is $N_{opt} = 3$ slots, and when $R_c = 200$ kbps, the optimal frame size $N_{opt} = 5$ slots. The analytical results are verified by the simulation results. It is observed that N_{opt} increases as R_c decreases. This is because, when R_c is smaller, the time to transmit the scheduling overhead T_{ov} is larger. As a result, more slot times in one frame are needed to compensate for the larger scheduling overhead.

Figure 4.9: Average SU throughput under different frame size N_{slots} for various R_c .

Fig. 4.9 shows that as R_c increases, η also increases. This is because, as R_c increases the time to transmit, the scheduling overhead is reduced, thus resulting in higher average SU throughput. It should be noted that when the channel-user settings are changed, the corresponding optimal frame size may also change. Our analysis of the optimal frame size is applicable for stable operating environment in which the channel-user setting remains stable. For a dynamically changing system, unless we have a scheme to dynamically capture the instantaneous channel-user setting and adapt the frame size accordingly, the system may not be able to operate optimally.

4.5.3 Comparison of the Proposed Scheduling Scheme with Slot-based Scheduling

The performance comparison of the proposed scheduling scheme and the slot-based scheduling scheme are performed under various performance measures, e.g.,

- The average channel throughput, $\bar{T}$
- Blocking probability
- Savings in the scheduling overhead over a frame

For this experiment, the number of channels N_c is chosen in the range of $[2, 5]$ and the number of SUs, V is set to 8. The number of modes for a channel is chosen as $C = 5$. The frame size N_{slot} is chosen in the range of $[1, 20]$. The total simulation time is 200,000 slots. The values for b_n for different channel settings in the network are given in the Table 4.1.

Table 4.1: Slot dependent parameter b_n for different channel settings in the network

N_c	L_{ov} bits	$b_n, n = 1$	$b_n, n > 1$
2	736	4	7
3	1072	3	7
4	1408	2	7
5	1744	1	7

4.5.3.1 Average Channel Throughput $\bar{T}$

The average channel throughput denoted by $\bar{T}$ is expressed as follows

$$\bar{T} = \frac{\sum_{u=1}^{N_c} T_u}{N_c} \quad (4.33)$$

where T_u is the actual throughput for channel u ($u \in \{1, 2, \dots, N_c\}$), which is the ratio between the total number of packet departures (from all SUs) over the channel and the total number of slots during the simulation. In other words, T_u represents the aggregated throughput of all SUs on channel u . T_u is expressed in packets per slot.

A heavy traffic load scenario is considered where the transmission queue (or buffer) of every SU is saturated most of the time. We use packet arrival rate $\lambda_v = 6.64$ packets per slot which is considered to be sufficiently large compared with the transmission rate per slot. The buffer size Q_v for all SUs is set to 100 packets.

Figure 4.10: Average channel throughput $\bar{T}$ under different frame sizes N_{slots} for various network settings.

Fig. 4.10 compares the average channel throughput of the proposed scheduling scheme

with various frame size with that of the slot-by-slot scheduling (i.e., $N_{slot} = 1$). It can be observed that the proposed scheme outperforms the slot-by-slot scheduling.

Fig. 4.10 also shows that $\bar{T}$ decreases for large frame sizes. This is due to the following reasons.

- The estimation of transition probabilities for the PU activity and channel mode is more accurate for a small frame size than that of a large frame size. Consequently, the calculated expected number of packet departures over a frame $T_F(u, v)$ is also more accurate for a small frame size than that of a large frame size.
- Some of the channel resources may be wasted when a large frame size is used. For example, for a network with $N_c = 2$ channels and frame size of $N_{slot} = 10$ slots, the channel capacity (based on c_n and b_n) has the range of $[0, 268]$ packets per frame whereas the number of packets that can be transmitted by a SU (based on a_n and q_n) has a range of $[0, 163]$ packets. Consequently, when the SUs cannot fully utilize the channel capacity, the throughput drops.
- When the frame size is large, even the user with maximum T_F may be experiencing bad channel quality in some slots, which causes lower throughput.

4.5.3.2 Blocking Probability

A light traffic load scenario is considered. The network setting is chosen with $V = 8$ users and $N_c = 2$ channels. For presentation simplicity, the mean of P_B calculated for each SU is considered here. The mean of P_B is denoted by $\bar{P}_B$. The average packet arrival rate is set to $\lambda_v = 2$ packets per slot with the maximum of $N_p = 3$ packets arriving at a SU's buffer. The buffer size is varied between $Q_v = 100$ to $Q_v = 200$ packets.

Fig. 4.11 shows the effect of frame size on the blocking probability. Fig. 4.11 shows that as the buffer size increases, the blocking probability decreases. It can be observed that the slot-by-slot case has a higher probability of blocking than that of the frame-by-frame case (up to $N_{slots} = 10$ slots). However, it can be observed that for large frame sizes the blocking

probability increases. This is because the scheduling scheme allocates N_c SUs during each frame and when the frame size is large the waiting time for the unallocated SUs increases causing more packet drops.

Figure 4.11: Blocking probability under different frame sizes N_{slot} for $V = 8$ users, $N_c = 2$ channels and varying Q_v .

4.5.3.3 Savings in Scheduling Overhead

Compared to the slot-based scheduling schemes, the proposed scheduling scheme schedules on a frame-by-frame basis. In other words, the slot-based scheduling schemes incurs N times the scheduling overhead per frame compared to the proposed scheduling scheme. Therefore, it can be observed that the scheduling overheads over a frame can be reduced by $(1 - 1/N_{slot})\%$ in the proposed scheduling scheme. For example, consider the network with $N_c = 5$ channels and $V = 8$ SUs. In both schemes, a scheduling overhead of $L_{ov} = 1,744$ bits is involved at each scheduling instance. However, using a frame size of $N_{slots} = 6$ slots (which leads to the best performance for the case of $N_c = 5$ channels and $V = 8$ SUs, as

shown in Fig. 4.10) the scheduling overhead over a frame can be reduced by 80% using the proposed scheduling scheme.

4.6 Summary

A novel queue-aware frame-based opportunistic spectrum scheduling scheme for centralized CRN was proposed. As compared with the slot-based scheduling schemes, the proposed scheduling scheme achieves smaller scheduling overhead and higher SU throughput. The analytical results and simulation results match well, thus accuracy of the design is validated. An optimal frame size for a given system setting was also derived from the analytical model and verified through simulation.

Chapter 5

Dynamic Spectrum Access schemes for Prioritized SU Traffic

This chapter presents the design and analysis for various DSA schemes which were proposed as part of our research to handle prioritized SU traffic (i.e., SU traffic containing data transmissions with different quality of service (QoS) requirements) under various CRN scenarios (e.g., centralized and distributed CRN, CRN with and without handoff buffer cases). For simplicity, the SU calls are prioritized into two classes¹, denoted as SU_1 and SU_2 . The calls belonging to the SU_1 priority class have higher priority over those belonging to the SU_2 priority class.

5.1 Research Highlights

The motivation for this work is attributed to various factors which are presented below.

- (1) None of the existing DSA schemes in the literature (references [9, 22, 24, 25, 39–48, 52–57, 59, 60] in Chapter 2) have ever considered SU prioritization. To date, only a few works [51, 72–74] have taken prioritized SU traffic into consideration. In these works, the SUs priorities are used for packet transmission during a slot. However, the SU priority does not affect spectrum handoff. On the other hand, in our work, SUs priorities play an important role during spectrum handoff (e.g., a high priority SU gets spectrum handoff before a low priority SU, in a two priority class case). Our

¹Please note that the proposed analytical models are not restricted to the two priority class case and can be extended to the case of more than two priority classes.

DSA schemes are particularly helpful when multiple SU transmissions (ongoing or those on hold) require spectrum handoff simultaneously.

- (2) An important contribution in this direction is that we provide the performance analysis for the CRN under prioritized SU traffic scenario. It is worth noting that such a performance analysis is not available in the literature. The performance is analyzed using separate analytical models for each proposed DSA scheme based on the continuous time Markov Chains (CTMC).

The layout of this chapter is as follows. First, the system model and the various proposed DSA schemes are presented. Following this, analytical models are presented under each DSA scheme and various performance measures are derived. For performance measures, metrics such as the blocking probability, probability of forced termination, throughput and mean handoff delay are derived. Thereafter, the performances of the various proposed DSA schemes are analyzed and verified with simulation results.

5.2 System Model

Figure 5.1: System model.

The system (Fig. 5.1) is composed of N_c licensed channels, each of which is further divided into N_{sc} sub-channels (this system model was also considered in [44, 45, 47, 48]). The PU and SU transmissions are treated as calls which use dedicated sub-channels. A PU call requires one channel whereas an SU call requires one sub-channel. A PU call

assignment depends on the number of ongoing PU calls (denoted by k) in the system (i.e., a PU call is admitted if $k < N_c$). A PU call assignment is oblivious of the ongoing SU calls. Therefore, an ongoing SU call interfering the new PU call should be handed-off to another vacant sub-channel. We assume a fully connected CRN, i.e., all SUs observe the same channel status (i.e., PU/SU activity). Perfect sensing is assumed to detect the PU activity. A common control channel is assumed to exist for the coordination among the SUs.

The proposed DSA schemes treat the CRN as a call admission control system. The PU and SU transmissions are treated as calls. For the centralized CRN, four novel DSA schemes are presented. These centralized DSA schemes are denoted as DSA-C1, DSA-C2, DSA-C3 and DSA-C4. For the distributed CRN, two novel DSA schemes denoted as DSA-D1 and DSA-D2 are presented. Each proposed DSA schemes has a different spectrum handoff mechanism for the SUs. The schemes DSA-C1, DSA-C2, DSA-D1 will be referred as *DSA schemes without buffer*. In these schemes, the SUs which experience spectrum handoff failures are dropped. The schemes DSA-C3, DSA-C4 and DSA-D2 will be referred as *DSA schemes with buffer* because when a SU experiences unsuccessful spectrum handoff, it is placed in a handoff buffer till further transmission opportunities are available.

Under all the proposed DSA schemes, a number of sub-channels are reserved for high priority secondary user (SU_1), i.e., when the total number of idle sub-channels is less than ζ which is the number of sub-channels reserved for SU_1 , the new SU_2 call will be rejected. In this way, we provide higher priority for the SU_1 calls over the SU_2 calls².

Call arrivals occur independently as poisson processes with mean arrival rates λ_p , λ_1 and λ_2 for PU, SU_1 and SU_2 , respectively. The service times independently follow exponential distributions with mean service rates of μ_p , μ_1 and μ_2 for PU, SU_1 and SU_2 calls, respectively. During a call (PU or SU), the data traffic is disassembled into packets and the packet transmissions are slotted. We assume the slot duration (T_c) to be very small compared to a call duration. We assume the packet delivery mechanism exists for each call

²Note that the PUs are not affected by such sub-channel reservation.

(e.g., modulation and packet acknowledgement). The call admission and spectrum handoff are determined by the DSA schemes which are explained as follows.

5.2.1 DSA schemes for Centralized CRN

For the centralized CRN, we assume that a central controller performs spectrum sensing, sub-channel assignment and spectrum handoff for the SU calls. The central controller tracks all the ongoing calls (i.e., PU, SU_1 and SU_2 calls). Let Y denote the total number of busy sub-channels at a given time instant. New SU_1 , SU_2 and PU calls are dropped when $Y = N_c N_{sc}$, $Y = N_c N_{sc} - \zeta$ and $k = N_c$, respectively. When a PU starts transmission on a given channel, the central controller performs handoff for any ongoing SU calls associated with the sub-channels corresponding to this channel. These displaced SU calls can belong to different priorities (SU_1 and SU_2), and the central controller does spectrum handoff for displaced SU_1 calls followed by spectrum handoff for displaced SU_2 calls (i.e., due to the priority). Our centralized DSA schemes only differ in their spectrum handoff mechanism (explained below).

5.2.1.1 Spectrum Handoff Mechanism for DSA-C1

Spectrum handoff for DSA-C1 is done according to the following steps.

- (1) Randomly assign a displaced SU_1 call to a unique idle sub-channel (if available).
- (2) For the remaining displaced SU_1 calls, terminate the required number of ongoing SU_2 calls and assign each resulting idle sub-channel to a displaced SU_1 call.
- (3) If there are no ongoing SU_2 calls, terminate the remaining displaced SU_1 calls.
- (4) After handoff for all the displaced SU_1 calls, if there are still idle sub-channels, randomly assign a displaced SU_2 call to a unique idle sub-channel.
- (5) When there is no idle sub-channel, terminate the remaining displaced SU_2 calls.

5.2.1.2 Spectrum Handoff Mechanism for DSA-C2

The spectrum handoff in DSA-C2 is the same as DSA-C1 except for steps (2) and (3). No ongoing SU_2 calls are terminated for the sake of SU_1 handoff. When there is no idle sub-channel, the displaced SU_1 calls are terminated.

5.2.1.3 Spectrum Handoff Mechanism for DSA-C3

For DSA-C3, spectrum handoff is done according to the following steps.

- (1) Randomly assign a displaced SU_1 call to a unique idle sub-channel (if available).
- (2) For the remaining displaced SU_1 calls, place the required number of ongoing SU_2 calls on hold and assign each resulting idle sub-channel to a displaced SU_1 call.
- (3) If there are no ongoing SU_2 calls, place the remaining displaced SU_1 calls on hold.
- (4) After handoff for all the displaced SU_1 calls, if there are still idle sub-channels, randomly assign a displaced SU_2 call to a unique idle sub-channel.
- (5) When there is no idle sub-channel, place the remaining displaced SU_2 calls on hold.

When an ongoing SU call is placed on hold, new SU calls (i.e., SU_1 and SU_2) are not accepted. When an ongoing call ($SU_1/SU_2/PU$) completes service, the SU calls put on hold are reassigned to the resulting idle sub-channels according to their priority. These reassigned SU calls resume transmissions for their remaining service times. The act of the central controller placing a SU call on hold and reassigning an idle sub-channel can be viewed as putting the displaced SU call in a virtual buffer till transmission opportunity is available.

5.2.1.4 Spectrum Handoff Mechanism for DSA-C4

For DSA-C4, spectrum handoff is the same as DSA-C3 except for steps (2) and (3). No ongoing SU_2 calls are put on hold for the sake of SU_1 handoff, when there is no idle sub-channel the displaced SU_1 calls are put on hold.

5.2.2 DSA schemes for Distributed CRN

In a distributed CRN, the SU calls independently perform spectrum sensing, sub-channel selection and spectrum handoff. During an SU call, the sender and destination exchange control messages on the common control channel (e.g., handoff messages). A new SU call performs full spectrum sensing at the beginning of the slot which occurs after its arrival time. Then, the new SU call randomly picks an idle sub-channel for transmission. New SU calls are dropped depending on the number of detected busy sub-channels (e.g., a new SU_2 call is dropped if it sensed less than ζ idle sub-channels). An ongoing SU call senses its own sub-channel in every slot to avoid interference to the PU. If a PU call is detected, the ongoing SU call must perform spectrum handoff. The spectrum handoff mechanisms of the distributed DSA schemes are discussed below.

5.2.2.1 Spectrum Handoff Mechanism for DSA-D1

Having detected the PU call, at the commencement of the next slot, a displaced SU (i.e., sender) determines the idle sub-channels through full spectrum sensing. If idle sub-channels are found, the displaced SU randomly selects only one idle sub-channel and claims it using a *contention process* because there could be other displaced SU calls also choosing the same sub-channel. As the displaced SU calls can have different priorities, we adopt the idea of contention resolution from the EDCF (enhanced distributed coordination function) protocol in IEEE 802.11e network [75]. Note that unlike in the IEEE 802.11e network [75] we allow only the winner of the contention process (explained later) to stay on the sub-channel. In other words, at most one displaced SU gets successful handoff on a chosen idle sub-channel. The maximum duration of the contention process is equal to one slot period.

During the contention process, the displaced SUs use an arbitration inter-frame space (AIFS) and contention window (CW) depending on their priority class (similar to [75]). A displaced SU_1 call uses shorter AIFS and CW than those of a displaced SU_2 call, as SU_1 calls have higher priority than SU_2 calls. We denote the AIFS corresponding to the SU_1 calls and SU_2 calls as $AIFS_1$ and $AIFS_2$, respectively. The CW corresponding to the

SU_1 calls and SU_2 calls are denoted as $[1, W_1]$ and $[1, W_2]$, respectively. Each displaced SU senses its chosen idle sub-channel for a duration equal to $AIFS_x + backoff_x$ depending on its priority $x \in \{1, 2\}$, where $backoff_x$ is the time spent for backoff based on the random integer number chosen from the corresponding CW. The handoff for a displaced SU is treated as successful if it chose the lowest $AIFS_x + backoff_x$ among all displaced SUs contending for the same idle sub-channel. An SU call which achieves successful handoff resumes its transmission for the remaining service time. A displaced SU call experiences spectrum handoff failure under two scenarios: 1) the chosen sub-channel becomes busy before its $AIFS_x + backoff_x$ expires (i.e., other SU access the sub-channel first or a collision occurs), 2) if idle sub-channels are not available. The displaced SU calls experiencing spectrum handoff failure are dropped (i.e., treated as force terminated by the incoming PU call).

5.2.2.2 Spectrum Handoff Mechanism for DSA-D2

Unlike the DSA-D1 scheme, in the DSA-D2 scheme, displaced SU calls experiencing spectrum handoff failures are not terminated, but allowed to retry in the subsequent slots till they get successful handoff. Therefore, in the subsequent slot, such displaced SU calls perform spectrum sensing and repeat the contention process. The displaced SU calls can be viewed as being in a virtual buffer and contending for the idle sub-channels to get successful spectrum handoff.

Additionally, in the DSA-D2 scheme, the displaced SU calls are given higher priority over the new SU arrivals. A new SU call (SU_1/SU_2) must sense all idle sub-channels at the beginning of the slot to ascertain any ongoing contention process. Only when spectrum handoff SUs are not present (i.e., no contention process is detected), the new SU call will be admitted. This is accomplished by letting the handoff SUs transmit a jamming signal at the beginning of the slot on their selected idle sub-channels before the commencement of the contention process.

5.3 Performance Analysis

5.3.1 Analytical Models for Centralized CRN

In this section, we develop analytical models for the centralized CRN under each DSA scheme using a separate continuous time Markov chain (CTMC). A CTMC can be described by its state transition characteristics.

5.3.1.1 Centralized DSA schemes without Buffer

The state of the CTMC for the centralized DSA schemes without buffer (DSA-C1 and DSA-C2) is denoted as $[i, j, k]$, where $i \in \{0, 1, \dots, N_c N_{sc}\}$, $j \in \{0, 1, \dots, N_c N_{sc} - \zeta\}$ and $k \in \{0, 1, \dots, N_c\}$ represent the number of ongoing SU_1 , SU_2 , and PU calls in the system, respectively. For a valid state $[i, j, k]$, $Y (= i + j + k N_{sc})$ should not exceed MN .

Let l (where $l \in \{0, 1, \dots, N_{sc}\}$) and m (where $m \in \{0, 1, \dots, N_{sc}\}$) denote the number of SU_1 and SU_2 calls displaced by an incoming PU call when the state is $[i, j, k]$. l and m should satisfy the following conditions:

$$\begin{aligned} l &\leq i, \quad m \leq j \quad \text{and} \quad l + m \in \{0, 1, \dots, N_{sc}\} \\ h_{sc} &= N_c N_{sc} - Y - (N_{sc} - (l + m)) \quad \text{and} \quad h_{sc} \geq 0 \\ r &= (l + m) - \min(h_{sc}, l + m) . \end{aligned} \tag{5.1}$$

In the Eq. (5.1), the first condition gives the maximum number of SUs that can be displaced upon a PU arrival whereas the second condition gives the number of sub-channels available for handoff (denoted as h_{sc}) for the displaced SU calls. Accordingly, the total number of unsuccessful handoff calls (denoted as r) is calculated.

The evolution of the state $[i, j, k]$ of the CTMC is presented under three cases of Y :

- (1) $Y \leq N_{sc}(N_c - 1)$: The system has at least N_{sc} idle sub-channels.
- (2) $N_{sc}(N_c - 1) < Y < N_c N_{sc}$: The system has idle sub-channel in the range $\{1, \dots, N_{sc} - 1\}$.
- (3) $Y = N_c N_{sc}$: The system has no idle sub-channel.

For the case $Y \leq N_{sc}(N_c - 1)$, all displaced SU_1 and SU_2 calls get successful handoff (i.e. $r = 0$) when displaced by a PU arrival. In other words, no calls are terminated (under DSA-C1 and DSA-C2) due to an incoming PU call. For the case $N_{sc}(N_c - 1) < Y < N_c N_{sc}$, $r = \{1, \dots, N_{sc} - 1\}$ number of SU calls experience unsuccessful handoff whereas, when $Y = N_c N_{sc}$, $r = N_{sc}$ SU calls experience unsuccessful handoff. The exact number of terminated SU_1 and SU_2 calls under each centralized DSA policy are explained later. Let l' and m' denote the number of terminated SU_1 and SU_2 calls respectively. Thus, $l' + m' = r$.

5.3.1.1.1 State Transitions under DSA-C1 scheme Under the DSA-C1 scheme, the transitions for the state $[i, j, k]$ for different cases of Y are explained using Fig. 5.2. State transitions from/to the state $[i, j, k]$ occur due to any of the six possible events, namely PU arrival, SU_1 arrival, SU_2 arrival, PU departure, SU_1 departure and SU_2 departure. Each state transition is represented by its corresponding rate. Taking as an example, a SU_2 arrival in state $[i, j - 1, k]$ causes transition to state $[i, j, k]$ with a rate $\delta_1 \cdot \lambda_2$, where $\delta_1 = \mathbf{1}_{(i+(j-1)+kN_{sc} < N_c N_{sc} - \zeta)}$ specifies the condition for sub-channel reservation (i.e., a new SU_2 call is accepted by the system only when the condition $i + (j - 1) + kN_{sc} < N_c N_{sc} - \zeta$ holds), where $\mathbf{1}_{(\cdot)}$ is the indicator function which returns the value 1 when the condition given inside the parenthesis is true and returns 0 otherwise. In Fig. 5.2, $\delta_2 = \mathbf{1}_{(i+j+kN_{sc} < N_c N_{sc} - \zeta)}$ (condition for sub-channel reservation in state $[i, j, k]$) and $\delta_3 = \mathbf{1}_{(i+j+(k-1)N_{sc} \leq N_{sc}(N_c - 1))}$.

The transition rate $\gamma_{l', m'}^{i, j, k}$ from state $[i, j, k]$ to state $[i - l', j - m', k + 1]$ is calculated as follows.

- (1) Let $\mathcal{N}_{l', m'}$ denote the set containing the valid combinations of (l', m') .
- (2) For every valid pair of (l', m') in the set $\mathcal{N}_{l', m'}$, find $R_{l', m'}$ which represents the number of valid combinations of l , m , h_{sc} and $r = l' + m'$. Valid l , m , h_{sc} and r for given $[i, j, k]$ found using Eq. (5.1).
- (3) Then, the transition rate $\gamma_{l', m'}^{i, j, k}$ is given by

$$\gamma_{l', m'}^{i, j, k} = \frac{R_{l', m'}}{\sum_{(l', m') \in \mathcal{N}_{l', m'}} R_{l', m'}} \lambda_p \quad (5.2)$$

Figure 5.2: State transitions for centralized DSA schemes without buffer.

Taking into consideration the handoff mechanism explained in Section 5.2.1.1, the variables l' and m' in Eq. (5.2) are given by $l' = r - \min(j, r)$ and $m' = \min(j, r)$. The value of r is determined by Eq. (5.1).

For the cases $Y \leq N_{sc}(N_c - 1)$ and $N_{sc}(N_c - 1) < Y < N_c N_{sc}$, all the transitions shown in Fig. 5.2 are valid. For the case $Y = N_c N_{sc}$, the transitions between the states $[i, j, k]$ and $[i + 1, j, k]$, and $[i, j, k]$ and $[i, j + 1, k]$ are not considered as the system is full (i.e. idle sub-channels not available). Apart from the state transitions shown in Fig. 5.2, when $Y = N_c N_{sc}$ few transitions occur from the states in which the number of ongoing calls is greater than $N_{sc}(N_c - 1)$ to the state $[i, j, k]$ due to a PU arrival. These transitions are described below.

- Transition from state $[i, j + j', k - 1]$ to state $[i, j, k]$ occurs with rate $\gamma_{0,j'}^{i,j+j',k-1}$, where $j > 0$ and $0 < j' < N_{sc}$. The rate $\gamma_{0,j'}^{i,j+j',k-1}$ is calculated using Eq. (5.2) corresponding to the state $[i, j + j', k - 1]$. During this state transition, j' SU_2 calls are terminated.

- Transition from state $[i + (r' - j'), j + j', k - 1]$ to state $[i, j, k]$ occurs with rate $\gamma_{r'-j',j'}^{i+(r'-j'),j+j',k-1}$, where $j = 0$, $0 < r' < N_{sc}$ and $0 \leq j' < r'$. The rate $\gamma_{r'-j',j'}^{i+(r'-j'),j+j',k-1}$ is calculated using Eq. (5.2) corresponding to the state $[i + (r' - j'), j + j', k - 1]$. During this state transition, $r' - j'$ SU_1 calls and j' SU_2 calls are terminated.

5.3.1.1.2 State Transitions under DSA-C2 scheme Fig. 5.2 also represents the state transitions under the DSA-C2 scheme for the various cases of Y . According to the handoff mechanism under DSA-C2 scheme, the number of terminated SU_1 and SU_2 calls are given as $l' = l - \min(h_{sc}, l)$ and $m' = \max(0, m - \max(0, h_{sc} - l))$ respectively. All state transitions shown in Fig. 5.2 occur similar to the DSA-C1 scheme. Apart from the state transitions given in Fig. 5.2, transitions occur from states $[i + i', j + j', k - 1]$ (in which $i + j + (k - 1)N_{sc} = N_{sc}(N_c - 1)$) to the state $[i, j, k]$ when $Y = N_c N_{sc}$, where $0 \leq i', j' < N_{sc}$ and $0 < i' + j' < N_{sc}$.

5.3.1.2 Centralized DSA schemes with Buffer

The state of the CTMC for the centralized DSA schemes with buffer (DSA-C3 and DSA-C4) is denoted as $[i, j, k, l_q, m_q]$, where $i \in \{0, 1, \dots, N_c N_{sc}\}$, $j \in \{0, 1, \dots, N_c N_{sc} - \zeta\}$ and $k \in \{0, 1, \dots, N_c\}$ represent the number of ongoing SU_1 , SU_2 , and PU calls in the system, respectively. The variables $l_q \in \{0, 1, \dots, N_c N_{sc}\}$ and $m_q \in \{0, 1, \dots, N_c N_{sc} - \zeta\}$ represent the number of SU_1 and SU_2 calls waiting in the buffer due to unsuccessful handoff. Here also, $Y = i + j + k N_{sc}$. For a valid state $[i, j, k, l_q, m_q]$, the following conditions must hold:

$$\begin{aligned}
 Y &\leq N_c N_{sc} \\
 i + j + l_q + m_q &\leq N_c N_{sc}, \quad \text{and} \\
 j + m_q &\leq N_c N_{sc} - \zeta
 \end{aligned} \tag{5.3}$$

As in Section 5.3.1.1, the same three cases of Y are considered to explain the state transitions. The state transitions are also classified according to the buffer status (empty and non-empty).

5.3.1.2.1 State Transitions under DSA-C3 scheme Fig. 5.3 shows the transitions of state $[i, j, k, l_q, m_q]$ for the DSA-C3 scheme when the buffer is empty (i.e. $l_q + m_q = 0$). When $l_q + m_q = 0$, except for the transitions between the states $[i, j, k, 0, 0]$ and $[i - l', j - m', k + 1, l', m']$ for $Y = N_c N_{sc}$, state transitions under DSA-C3 scheme and the DSA-C1 scheme (refer Fig. 5.2) are the same for all cases of Y . Here $l' = r - \min(j, r)$ and $m' = m - \min(j, r)$. During a PU arrival when $Y = N_c N_{sc}$, l' SU_1 calls and m' SU_2 calls are placed in the buffer, hence $[i, j, k, 0, 0]$ changes to $[i - l', j - m', k + 1, l', m']$ with rate $\gamma_{l', m'}^{i, j, k}$. As $\gamma_{l', m'}^{i, j, k}$ is independent of buffer status it is calculated using Eq. (5.2). A PU departure in state $[i - l', j - m', k + 1, l', m']$ reassigns all the SU_1 and SU_2 calls in the buffer, hence state changes to $[i, j, k, 0, 0]$.

Fig. 5.4 shows the state transitions under the DSA-C3 scheme when the buffer is non-empty (i.e. $l_q + m_q > 0$) which occurs only for the case $Y = N_c N_{sc}$. A new SU_1 or SU_2 arrival cannot be accommodated when the buffer is non-empty. A PU arrival adds $l' + m' = N_{sc}$ SUs to the buffer. SU calls in the buffer are reassigned sub-channels following a departure event (PU/ SU_1 / SU_2) according to their priority (refer Section 5.2.1.3). For the symbols $\hat{l}$, $\hat{m}$, l'' and m'' shown in Fig. 5.4, when a SU_1 (or SU_2) departs $\hat{l} = \min(1, l_q)$ and $\hat{m} = m_q - (1 - \min(1, l_q))$. When a PU departs $l'' = \min(N_{sc}, l_q)$ and $m'' = m_q - (N_{sc} - \min(N_{sc}, l_q))$. When transition occurs from state $[i, j + j', k - 1, 0, 0]$, $l_q = 0$ and $m_q = j'$ where $j > 0$ and $0 < j' < N_{sc}$. Similarly, for the transition occurring from state $[i + (r' - j'), j + j', k - 1, 0, 0]$, $l_q = r' - j'$ and $m_q = j'$ where $j = 0$, $0 < r' < N_{sc}$ and $0 \leq j' < r'$. In the Fig. 5.4, $\delta_4 = \mathbf{1}_{(N_{sc}(N_c - 1) \leq i + (j + j') + (k - 1)N_{sc} < N_c N_{sc})}$ and $\delta_5 = \mathbf{1}_{(N_{sc}(N_c - 1) \leq (i + (r' - j')) + (j + j') + (k - 1)N_{sc} < N_c N_{sc})}$.

5.3.1.2.2 State Transitions under DSA-C4 scheme Fig. 5.3 and Fig. 5.4 also represent the state transitions under the DSA-C4 scheme. However, for the DSA-C4 scheme $l' = l - \min(h_{sc}, l)$ and $m' = \max(0, m - \max(0, h_{sc} - l))$. The order of reassignment of SU calls from the buffer is same under both the DSA-C3 and DSA-C4 schemes.

 Figure 5.3: State transitions for centralized DSA schemes with buffer, $l_q = 0$ and $m_q = 0$.

 Figure 5.4: State transitions for centralized DSA schemes with buffer, $l_q + m_q > 0$.

5.3.2 Analytical Models for Distributed CRN

In this section, we develop analytical models for the distributed DSA schemes (DSA-D1 and DSA-D2) using separate CTMCs.

5.3.2.1 DSA-D1 scheme

Similar to the DSA-C1 and DSA-C2, the state of the CTMC under the DSA-D1 scheme (i.e., without buffer case) is denoted as $[i, j, k]$ where i, j and k have the same meaning as defined in Section 5.3.1.1.

The state transitions for the state $[i, j, k]$ under the DSA-D1 scheme are shown in Fig. 5.5, where $l' \in \{0, \dots, N_{sc}\}$, $m' \in \{0, \dots, N_{sc}\}$ and $l' + m' \in \{s, \dots, N_{sc}\}$ (r determined by Eq. (5.1)). Similar to Section 5.3.1.1, the state transitions can be explained on the basis of Y (some transitions are ignored depending on Y). For the cases $Y \leq N_{sc}(N_c - 1)$ and $N_{sc}(N_c - 1) < Y < N_c N_{sc}$, all the state transitions shown in Fig. 5.5 occur. For the case $Y = N_{sc} N_c$, the transitions between the states $[i, j, k]$ and $[i + 1, j, k]$ do not occur, and the transitions between the states $[i, j, k]$ and $[i, j + 1, k]$ also do not occur.

Figure 5.5: State transitions for the DSA-D1 scheme.

The transition rate $\xi_{l',m'}^{i,j,k}$ from state $[i, j, k]$ to state $[i - l', j - m', k + 1]$ in Fig. 5.5 is calculated as $P_d(l', m'|i, j, k) \cdot \lambda_p$, where $P_d(l', m'|i, j, k)$ denotes the probability that l' SU_1 and m' SU_2 calls are dropped following a PU arrival. $P_d(l', m'|i, j, k)$ is expressed as

$$P_d(l', m'|i, j, k) = \sum_{\forall l, m} P_d(l', m'|l, m, i, j, k) \cdot P(l, m|i, j, k) . \quad (5.4)$$

In the above equation, $P(l, m|i, j, k)$ is calculated as

$$P(l, m|i, j, k) = \frac{\binom{N_{sc}}{l, m} \cdot \binom{(N_c - k - 1)N_{sc}}{i - l, j - m}}{\binom{(N_c - k)N_{sc}}{i, j}} \quad (5.5)$$

where l, m, h_{sc} and r are determined by Eq. (5.1) for state $\mathbf{z}$. The notation $\binom{c}{a, b}$ denotes the number of ways of arranging a SU_1 calls, b SU_2 calls in c ($\geq a + b$) sub-channels without repetition.

As $l' + m' = r$ depends only on l, m and h_{sc} corresponding to state $[i, j, k]$, $P_d(l', m'|l, m, i, j, k)$ can be rewritten as $P_d(l', m'|l, m, h_{sc})$. Accordingly, Eq. (5.4) is modified as

$$P_d(l', m'|i, j, k) = \sum_{\forall l, m} P_d(l', m'|l, m, h_{sc}) \cdot P(l, m|i, j, k) . \quad (5.6)$$

$P_d(l', m'|l, m, h_{sc})$ in Eq. (5.6) is equivalent to the probability $P(l_s, m_s \text{ handoff successfully} | l, m, h_{sc})$ (or simply denoted as $P_s(l_s, m_s | l, m, h_{sc})$, where $l_s = l - l'$ and $m_s = m - m'$ denote the number SU_1 and SU_2 calls experiencing successful handoff. The probability $P_s(l_s = e, m_s = f | l, m, h_{sc})$ (where $0 < e + f \leq (l + m) - r$) is calculated recursively as follows.

$$\begin{aligned} P_s(l_s = e, m_s = f | l, m, h_{sc}) &= \sum_{g=0}^l \sum_{h=0}^m P_{sc}(l, g) P_{sc}(m, h) \\ &\cdot \left(P_s(l_s = 1, m_s = 0 | g, h, 1) \cdot P_s(l_s = e - 1, m_s = f | l - g, m - h, h_{sc} - 1) \right. \\ &+ P_s(l_s = 0, m_s = 1 | g, h, 1) \cdot P_s(l_s = e, m_s = f - 1 | l - g, m - h, h_{sc} - 1) \\ &\left. + P_s(l_s = 0, m_s = 0 | g, h, 1) \cdot P_s(l_s = e, m_s = f | l - g, m - h, h_{sc} - 1) \right) \end{aligned} \quad (5.7)$$

where $P_{sc}(l, g) = \binom{l}{g} (1/h_{sc})^g (1 - 1/h_{sc})^{l-g}$ denotes probability that g out of l SU_1 calls contend for the same sub-channel. Similarly, $P_{sc}(m, h) = \binom{m}{h} (1/h_{sc})^h (1 - 1/h_{sc})^{m-h}$ in Eq.

(5.7) denotes probability that h out of m SU_2 calls contend for the same sub-channel. The meaning of Eq. (5.7) is explained as follows. Consider one of the h_{sc} available sub-channels to be tagged. The left-hand side of Eq. (5.7) denotes the probability that e out of l SU_1 calls and f out of m SU_2 calls achieve successful handoff, given l SU_1 calls and m SU_2 calls contend for r sub-channels. The right-hand side of Eq. (5.7) denotes the total probability that at least one successful handoff (i.e. either one SU_1 or one SU_2 or none) occurs in the tagged sub-channel in which exactly g out of l SU_1 calls and h out of m SU_2 calls contend, while the remaining successful handoffs (i.e. excluding the successful handoff happening in the tagged sub-channel) occur in the remaining $h_{sc} - 1$ sub-channels which are contended by the remaining $l - g$ SU_1 calls and $m - h$ SU_2 calls.

In the Eq. (5.7), $P_s(l_s = 1, m_s = 0|g, h, 1)$ is calculated as

$$P_s(l_s = 1, m_s = 0|g, h, 1) = \begin{cases} 0 & \text{if } g = 0, h \geq 0 \\ \sum_{w_1=1}^{\Delta} \binom{g}{1} \frac{1}{W_1} \left(\frac{W_1-w_1}{W_1}\right)^{g-1} + \\ \sum_{w_1=\Delta+1}^{W_1} \binom{g}{1} \frac{1}{W_1} \left(\frac{W_1-w_1}{W_1}\right)^{g-1} \left(\frac{W_2-(w_1-\Delta)}{W_2}\right)^h & \text{if } g > 0, g + h > 0 \end{cases} \quad (5.8)$$

where $\Delta = AIFS_2 - AIFS_1$. Also in Eq. (5.7), $P_s(l_s = 0, m_s = 1|g, h, 1)$ is calculated as

$$P_s(l_s = 0, m_s = 1|g, h, 1) = \begin{cases} 0 & \text{if } g \geq 0, h = 0 \\ \sum_{w_2=1}^{W_2} \binom{h}{1} \frac{1}{W_2} \left(\frac{W_2-w_2}{W_2}\right)^{h-1} & \text{if } g = 0, h \geq 1 \\ \sum_{w_2=1}^{W_1-(\Delta+1)} \binom{h}{1} \left(\frac{W_1-(w_2+\Delta)}{W_1}\right)^g \frac{1}{W_2} \left(\frac{W_2-w_2}{W_2}\right)^{h-1} & \text{if } g > 0, h > 0 \end{cases} \quad (5.9)$$

Note, $P_s(l_s, m_s|l, m, h_{sc}) = 0$ for $h_{sc} \leq 0$, $l_s > l$ and $m_s > m$. The Eqs. (5.8) and (5.9) can be explained in similar fashion. For example, Eq. (5.8) gives the probability that one displaced SU_1 call gets successful handoff while g SU_1 calls and h SU_2 calls contend for the same sub-channel. In Eq. (5.8), the summations give total probability that the tagged SU_1 call gets successful handoff for choosing its backoff value $w_1 \in [1, W_1]$ while the other $g - 1$ SU_1 calls and h SU_2 calls (contending for the same sub-channel) choose their backoff values higher than w_1 .

5.3.2.2 DSA-D2 scheme

Similar to DSA-C3 and DSA-C4 schemes, the state of the CTMC under the DSA-D2 scheme is denoted as $[i, j, k, l_q, m_q]$ where i, j, k, l_q and m_q have the same meaning as defined in

Section 5.3.1.2. The state transitions are shown in Fig. 5.6. The state transitions are explained based on the buffer status (empty and non-empty).

When the buffer is empty (i.e. $l_q + m_q = 0$), all state transitions except that to the state $[i + l_s, j + m_s, k, l_q - l_s, m_q - m_s]$ are valid. The admission of a SU (SU_1 or SU_2) call is done based on $Y (= i + j + kN_{sc})$, similar to other DSA schemes. When the buffer is non-empty (i.e. $l_q + m_q > 0$), all the state transitions except those corresponding to SU arrivals are valid.

In Fig. 5.6, the rate $P_{l,m}^{i,j,k,l_q,m_q}$ is equal to $P(l, m|i, j, k) \cdot \lambda_p$, where $P(l, m|i, j, k)$ is calculated using Eq. (5.5). The rate $\xi_{l_s, m_s}^{i,j,k,l_q,m_q}$ is given by $P(l_s, m_s|l_q, m_q, h'_{sc}) \cdot \lambda_p$, where $P(l_s, m_s|l_q, m_q, h'_{sc})$ is also calculated using Eq. (5.7), by replacing l, m and h_{sc} in $P_s(l_s, m_s|l, m, h_{sc})$ with l_q, m_q and $h'_{sc} = N_{sc}N_c - Y$.

Figure 5.6: State transitions for the DSA-D2 scheme.

5.3.3 Performance Measures

The performance measures for each DSA scheme are expressed using the steady state probability distribution of its corresponding CTMC. We derive performance measures (for both SU_1 and SU_2 calls) such as blocking probability, probability of forced termination, throughput and mean handoff delay.

Let $\Pi_{\mathbf{z}}$ denote the steady state distribution of a CTMC whose state is denoted as $\mathbf{z}$. For the DSA-C1, DSA-C2 and DSA-D1 schemes $\mathbf{z} = [i, j, k]$ whereas for the DSA-C3, DSA-C4 and DSA-D2 schemes $\mathbf{z} = [i, j, k, l_q, m_q]$. In order to simplify the presentation, we use the same notations under all DSA schemes. For any DSA scheme, the corresponding steady state probability distribution $\Pi_{\mathbf{z}}$ is obtained by finding the corresponding transition rate matrix $\mathbf{Q}$ and applying the Gauss-Seidel algorithm [76]. Each row of $\mathbf{Q}$ represents the transitions with respect to a specific state $\mathbf{z}$, as a balance equation. For example, referring to Fig. 5.3, one balance equation with respect to the state $[i, j, k, 0, 0]$ under the DSA-C3 scheme is expressed as

$$\begin{aligned} & [k\mu_p + \delta_2\lambda_2 + \lambda_1 + \gamma_{l',m'}^{i,j,k} + i\mu_1 + j\mu_2] \cdot \Pi_{[i,j,k,0,0]} = \\ & \lambda_p \Pi_{[i,j,k-1,0,0]} + (j+1)\mu_2 \Pi_{[i,j+1,k,0,0]} + (i+1)\mu_1 \Pi_{[i+1,j,k,0,0]} + \\ & (k+1)\mu_p \Pi_{[i,j,k+1,0,0]} + \lambda_1 \Pi_{[i-1,j,k,0,0]} + \delta_1\lambda_2 \Pi_{[i,j-1,k,0,0]} \end{aligned} \quad (5.10)$$

where $\Pi_{[i,j,k,0,0]}$ represents the steady state probability for state $[i, j, k, 0, 0]$ under the DSA-C3 scheme.

5.3.3.1 Blocking Probability

The blocking probability represents the probability with which an incoming SU call (SU_1 or SU_2) is not permitted into the system. The blocking probability of SU_1 calls denoted as P_{B1} is expressed as

$$P_{B1} = \sum_{\substack{\forall \mathbf{z} \\ Y = N_{sc}N_c}} \Pi_{\mathbf{z}}. \quad (5.11)$$

The blocking probability of SU_2 calls denoted as P_{B2} is expressed as

$$P_{B2} = \sum_{\substack{\forall \mathbf{z} \\ Y \geq N_{sc}N_c - \zeta}} \Pi_{\mathbf{z}}. \quad (5.12)$$

5.3.3.2 Probability of Forced Termination

The probability of forced termination represents the probability with which an ongoing SU call (SU_1 or SU_2) is terminated by an incoming PU call. Note that forced termination occurs only with DSA-C1, DSA-C2 and DSA-D1 schemes (i.e., without buffer cases). The probability of forced termination for the SU_1 calls denoted as P_{F1} , is expressed as

$$P_{F1} = \sum_{\substack{\forall \mathbf{z} \\ i \geq l', j \geq m'}} \frac{l' \cdot Q_{\mathbf{z}, \mathbf{z}'} \cdot \Pi_{\mathbf{z}}}{\lambda_1(1 - P_{B1})} \quad (5.13)$$

where $Q_{\mathbf{z}, \mathbf{z}'}$ denotes the transition rate from state $\mathbf{z} = [i, j, k]$ to state $\mathbf{z}' = [i-l', j-m', k+1]$. In Eq. (5.13), the numerator denotes the probability that l' SU_1 calls are terminated in state $\mathbf{z}$ whereas the denominator denotes the effective rate with which a new SU_1 call is assigned a sub-channel.

Similarly, the probability of forced termination for the SU_2 calls denoted as P_{F2} , is expressed as

$$P_{F2} = \sum_{\substack{\forall \mathbf{z} \\ i \geq l', j \geq m'}} \frac{m' \cdot Q_{\mathbf{z}, \mathbf{z}'} \cdot \Pi_{\mathbf{z}}}{\lambda_2(1 - P_{B2})}. \quad (5.14)$$

In Eqs. (5.13) and (5.14), l' and m' depend on the DSA scheme. For the schemes DSA-C3, DSA-C4 and DSA-D2 (i.e., with buffer cases), $P_{F1} = 0$ and $P_{F2} = 0$.

5.3.3.3 Throughput

The throughput under a given SU priority class is expressed as the mean number of ongoing calls in the system having that priority. Let η_1 and η_2 denote the throughput for SU_1 and SU_2 calls respectively. The throughput η_1 is given by $\eta_1 = \lambda_1(1 - P_{B1})(1 - P_{F1})$. The throughput η_2 is given by $\eta_2 = \lambda_2(1 - P_{B2})(1 - P_{F2})$.

5.3.3.4 Mean Handoff Delay

The mean handoff delay for SU_1 (SU_2) calls is defined as the mean time taken by the SU_1 (SU_2) calls to perform successful handoff. Let HD_1 and HD_2 denote the handoff delays for SU_1 and SU_2 calls, respectively. For the DSA-C3, DSA-C4 and DSA-D2 schemes (i.e.,

with buffer cases), the mean handoff delays corresponding to SU_1 and SU_2 calls is given as follows:

$$HD_1 = \frac{\bar{l}'}{\bar{l}} \cdot \frac{\sum_{\mathbf{z}} l_q \Pi_{\mathbf{z}}}{\sum_{l', m', \mathbf{z}} l' \cdot Q_{\mathbf{z}, \mathbf{z}'} \cdot \Pi_{\mathbf{z}}} \quad (5.15)$$

$$HD_2 = \frac{\bar{m}'}{\bar{m}} \cdot \frac{\sum_{\mathbf{z}} m_q \Pi_{\mathbf{z}}}{\sum_{l', m', \mathbf{z}} m' \cdot Q_{\mathbf{z}, \mathbf{z}'} \cdot \Pi_{\mathbf{z}}} \quad (5.16)$$

where $Q_{\mathbf{z}, \mathbf{z}'}$ represents the transition rate from state $\mathbf{z} = [i, j, k, l_q, m_q]$ to state $\mathbf{z}' = [i - l', j - m', k + 1, l_q + l', m_q + m']$. In the above equations, $\bar{l}$ and $\bar{m}$ denote the mean number of SU_1 and SU_2 calls displaced during a PU call arrival, respectively. They are calculated as $\bar{l} = \sum_{l, m, \mathbf{z}} l \cdot P(l, m | \mathbf{z}) \Pi_{\mathbf{z}}$ and $\bar{m} = \sum_{l, m, \mathbf{z}} m \cdot P(l, m | \mathbf{z}) \Pi_{\mathbf{z}}$. $\bar{l}'$ and $\bar{m}'$ denote the mean number of SU_1 and SU_2 calls experiencing unsuccessful handoff during a PU call arrival. They are calculated as $\bar{l}' = \sum_{l', m', l, m, \mathbf{z}} l' \cdot P(l', m' | l, m) P(l, m | \mathbf{z})$ and $\bar{m}' = \sum_{l', m', l, m, \mathbf{z}} m' \cdot P(l', m' | l, m) P(l, m | \mathbf{z})$, where $P(l', m' | l, m)$ is equal to 1 if l' and m' are valid for the given l and m corresponding to the state $\mathbf{z}$, otherwise $P(l', m' | l, m) = 0$. Here, $P(l, m | \mathbf{z})$ (or $P(l, m | i, j, k, l_q, m_q)$) is equivalent to $P(l, m | i, j, k)$ as l and m are independent of l_q and m_q . Therefore, $P(l, m | i, j, k, l_q, m_q)$ is also calculated using Eq. (5.5).

For the DSA-C1, DSA-C2 and DSA-D1 schemes (i.e., without buffer cases) as the unsuccessful handoff calls are dropped immediately, $HD_1 = 0$ and $HD_2 = 0$.

5.4 Results and Discussion

In this section, we evaluate the performance of the CRN under each proposed DSA scheme using the performance measures derived in the Section 5.3. The accuracy of the analysis is verified through simulations. The simulations were performed using MATLAB. In our experiments we use the following parameter settings, $N_c = 3$, $N_{sc} = 4$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_p \in \{0.4, \dots, 1\}$, $\mu_1 = 0.5$, $\mu_2 = 0.5$, $\mu_p = 1.8$ and $\zeta = 1$. For convenience, the arrival rates and service rates are expressed in calls/second. For the DSA-D1 and DSA-D2 schemes, $AIFS_1 = 1$, $AIFS_2 = 2$, $W_1 = 2$ and $W_2 = 3$. The slot duration T_c is equal to $0.02s$. The symbols (a) and (s) in the figures indicate analytical and simulation results respectively.

5.4.1 Throughput

Fig. 5.7 and Fig. 5.8 show the effect of PU arrival rate (λ_p) on throughput of the SU_1 and SU_2 calls under the different proposed DSA schemes, respectively. The throughput corresponding to the DSA-C4 scheme is same as that corresponding to the DSA-C3 scheme, hence it is not shown. It can be seen from Fig. 5.7 and Fig. 5.8 that as λ_p increases, the throughput of the SU calls (η_1 and η_2) decreases under all the DSA schemes. This is because the number of busy sub-channels increases with an increase in λ_p , causing higher blocking probabilities for the SU calls and subsequently lower throughput. It can also be seen that the DSA schemes using handoff buffer give higher throughput (η_1 and η_2) compared to those without buffer. This is because forced termination of SU calls is prevented in the DSA schemes with buffer (i.e. DSA-C3, DSA-C4 and DSA-D2). The throughput of the DSA-D2 scheme is less than those of the centralized DSA schemes which use buffer, because the handoff calls wait longer time in the buffer due to contention process, resulting in a longer service time for SU calls.

Figure 5.7: Effect of PU arrival rate on SU_1 throughput, under all DSA schemes.

Figure 5.8: Effect of PU arrival rate on SU_2 throughput, under all DSA schemes.

It is observed that the DSA-D1 scheme gives the lowest values of η_1 and η_2 among all the DSA schemes. This is because some SU_1 and SU_2 calls are terminated due to collision during handoff. It is also observed that the DSA-C1 scheme gives higher value of η_1 than the DSA-C2 scheme. Further, the DSA-C2 scheme has higher value of η_2 than the DSA-C1 scheme. The reason for both these behaviors is the same, i.e. some SU_2 calls are terminated by the displaced SU_1 calls during handoff under the DSA-C1 scheme but no such termination occurs under the DSA-C2 scheme. This fact is confirmed by the figures Fig. 5.9 and Fig. 5.10 which show the effect of PU arrival rate on the probability of forced termination for the SU_1 and SU_2 calls respectively. It can be seen that the P_{F1} for the DSA-C1 scheme is lower than that of the DSA-C2 scheme, and P_{F2} for DSA-C2 scheme is lower than that of the DSA-C1 scheme.

Fig. 5.9 shows that the distributed case (DSA-D1) has the highest P_{F1} . Similarly, Fig. 5.10 shows the distributed case having the highest P_{F2} . Further, it is intuitive that an increase in λ_p causes an increase in both P_{F1} and P_{F2} , as shown in Fig. 5.9 and Fig. 5.10.

Figure 5.9: Effect of PU arrival rate on forced termination probability P_{F1} , under DSA schemes without buffer.

Figure 5.10: Effect of PU arrival rate on forced termination probability P_{F2} , under DSA schemes without buffer.

The analysis and simulation results match well for the throughput (η_1 and η_2) and probability of forced termination (P_{F1} and P_{F2}), under all the DSA schemes.

5.4.2 Mean Handoff Delay

Fig. 5.11 shows the effect of the PU arrival rate (λ_p) on the mean handoff delay of SU_1 calls (HD_1) under the centralized and distributed schemes using handoff buffer. It can be seen that as λ_p increases D_1 also increases. This is because the channels mostly carry PU traffic as λ_p increases. Fig. 5.11 shows that the DSA-C3 scheme has lowest value of HD_1 . This is because under this scheme, the SU_1 calls get immediate handoff as long as some ongoing SU_2 calls are present (i.e. in such scenario handoff delay is zero), hence the mean handoff delay (HD_1) is smaller. Fig. 5.11 shows the DSA-D2 scheme having higher HD_1 than the centralized DSA schemes. This is because the contention and collisions during handoff lead to longer handoff times.

Fig. 5.12 shows the effect of the PU arrival rate (λ_p) on the mean handoff delay of SU_2 calls (HD_2) under the centralized and distributed schemes which use handoff buffer. It can be seen that as λ_p increases HD_2 also increases. The reason is the same as that given for HD_1 . Fig. 5.12 shows the DSA-C3 scheme having higher HD_2 compared to the DSA-C4 scheme. This is because under DSA-C3 scheme, the number of displaced SU_2 calls is more than that in the DSA-C4 scheme (as some SU_2 are also displaced due to SU_1 handoff). Fig. 5.11 also shows that the DSA-D2 scheme gives the highest value for HD_2 . This reason is the same as that explained for HD_1 . Again, the simulation results and analytical results match well for both HD_1 and HD_2 under all the DSA schemes with buffer.

Figure 5.11: Effect of PU arrival rate on the mean handoff delay of SU_1 calls, under DSA schemes with buffer.

Figure 5.12: Effect of PU arrival rate on the mean handoff delay of SU_2 calls, under DSA schemes with buffer.

5.4.3 Optimal Sub-channel Reservation

As mentioned in Section 5.2, some sub-channels (ζ) are reserved for the SU_1 call arrivals. Using the sub-channel reservation, we block some low priority SU calls to improve the performance of the high priority SU calls. Based on the given parameter settings and blocking probability requirement of the SU_1 calls, an optimal number of sub-channels (ζ) to be reserved under the DSA schemes can be determined from our analysis. Here, optimal number means the minimum number of sub-channels to be reserved in order to guarantee that the blocking probability requirement of the SU_1 calls is satisfied. For instance, consider the following parameter setting: $\lambda_1 = 1.8$, $\lambda_2 = 1.8$, $\lambda_p = 0.06$, $\mu_1 = 0.8$, $\mu_2 = 0.3$, $\mu_p = 0.09$ and a desired blocking probability of 5% for the SU_1 calls under the DSA-C1 scheme. From the analysis, we found that the optimal value of ζ is 4 for DSA-C1 scheme. The analysis results can be verified from simulation. Fig. 5.13 shows the simulation results for the above parameter settings under different values of ζ with DSA-C1 scheme. Fig. 5.13 shows that the desired blocking probability of 5% for the SU_1 calls is satisfied when $\zeta = 4$ sub-channels. In the same manner the optimal ζ can be found for the other DSA schemes.

Figure 5.13: Simulation: Optimal sub-channel reservation under DSA-C1 scheme.

5.4.4 Summary of Performance Analysis

From our analysis, we found that the DSA schemes with buffer are particularly useful in scenarios where the rate of forced termination by the PUs is very high. For example, for the parameter setting $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_p = 1.2$, $\mu_1 = 0.5$, $\mu_2 = 0.5$, $\mu_p = 1.8$ and $\zeta = 1$, the forced termination is more than 25% in the DSA schemes without buffer, whereas this termination is totally prevented by the DSA schemes with buffer. It is also observed that all the centralized DSA schemes give better performance compared to their counterparts (i.e., distributed DSA schemes). However, considering the cost of the infrastructure (e.g., central controller, data base for queued calls) needed for the centralized DSA schemes, the distributed DSA schemes are easier to construct.

5.5 Summary

This work adds great value to the current research in CRN as we designed various DSA schemes which are QoS specific (i.e., manage the SU traffic according to the service class priority). This chapter also provided an in-depth performance analysis for various DSA schemes under various CRN architectures and handoff scenarios. A wide range of performance measures like blocking probability, probability of forced termination, throughput, mean handoff delay and optimal sub-channel reservation based on SU priority class were derived and each DSA policy was accordingly analyzed. The analytical results were further verified using simulation.

Chapter 6

Conclusions and Future Work

6.1 Summary of the Contributions

The purpose of this dissertation is to add value to the current research in cognitive radio networks (CRN) through our original contributions. Three important contributions were made in the course of our research in the following areas:

- (1) Reducing the spectrum sensing overhead
- (2) Reducing the scheduling overhead and improving throughput in the centralized CRN
- (3) Call admission and spectrum handoff based on priority of the SUs in various CRN architectures

6.1.1 Reducing the Spectrum Sensing Overhead

We addressed the problem of sensing overhead incurred by the spectrum sensing mechanism. This work was presented in Chapter 3 of this dissertation. The problem posed by the sensing overhead was explained in the introduction chapter. Two channel status prediction schemes were proposed and analyzed under various PU traffic scenarios (i.e., stationary and non-stationary). One of the channel status prediction schemes was designed using the multilayer perceptron (MLP) neural network whereas the other was designed using hidden Markov models (HMM). These channel status prediction schemes provided an advantage over the existing schemes as they do not require a priori knowledge of the statistics of

the PUs' channel usage. Besides reducing the sensing energy by more than 50%, the two prediction schemes also demonstrate that the spectrum utilization could be improved by more than 68%. The accuracy of the sensing schemes was analyzed based on performance measures such as the probability of wrongly predicting the busy state of the channel and the probability of wrongly predicting the idle state of the channel.

6.1.2 Reducing the Scheduling Overhead in Centralized CRN

We addressed the problem of scheduling overhead incurred by the DSA scheme in a centralized CRN. This work presented in Chapter 4. A slotted transmission scenario was considered. It was explained that the existing DSA schemes (referred as slot-based scheduling schemes) were affected by the scheduling overhead and consequently suffered a loss of throughput. None of the existing schemes have addressed the issue of scheduling overhead to the date. We also explained how the lack of queue-awareness results in lower spectrum utilization in the existing scheduling schemes. We proposed a queue-aware frame-based opportunistic spectrum scheduling scheme which not only achieves significantly higher throughput compared to the slot-based scheduling schemes but also greatly reduces the scheduling overhead. We also determined the selection of optimal frame size to achieve the desired results in the proposed scheduling scheme.

6.1.3 Call Admission and Spectrum Handoff for Prioritized SU Traffic

We focused on designing call admission and spectrum handoff for the SUs in the CRN based on the priority of the SUs' in the SU traffic. Our findings for this work were presented in Chapter 5. We highlighted the fact that to the date, the QoS requirements of the SUs hardly received much attention in the design of the DSA schemes, especially priority of SUs during spectrum handoff has never been considered. This work not only addressed the design of the DSA schemes for the prioritized SU traffic but also presented performance analysis which is not found in the existing literature. Various CRN scenarios (such as centralized and distributed cases, with and without buffer cases) were considered in this

work. Various performance measures were derived for all the proposed DSA schemes. The analysis was verified using simulations.

6.2 Future Research

Besides the above contributions, this dissertation also provided an insight into the existing research in the MAC layer of the CRN. Through this dissertation we also aspire to offer scope for future research in the design of CRN. Some of the research problems are presented below.

6.2.1 CRN for Real PU Scenarios

In most existing works (including those expressed in this dissertation), the licensed user system is assumed to be composed of channels having the same bandwidth, signal-to-noise ratio (SNR) characteristics and PU traffic characteristics. Further, most works assume the PU traffic distribution to be stationary and known a priori. However, in reality this may not be the case. In fact, it is difficult to obtain the statistics of the PUs' channel usage in a real scenario. For example, in [12] the US public safety bands were analyzed for deployment of CRN. It was observed that though the channels were sporadically used by the PUs, most of the channels could be sensed busy due to strong adjacent channel interference. Though a SU usage of such channels might not cause harmful interference to the PUs, the SU may not choose to operate in such channels because of high probability of false alarm. Hence, it will be interesting to design reliable sensing and access mechanisms when the PU channels have strong adjacent channel interference. Another design issue in such a scenario is that most SUs would compete for channels with less adjacent channel interference which may lead to saturation of these data channels and poor spectrum utilization.

6.2.2 Research Challenges for Centralized CRN

The centralized CRN appears as a simple solution for coordinating the SUs access in the licensed spectrum. They also offer scope for designing large networks. In fact, the IEEE

802.22 network [9] is envisioned to offer broadband internet access to large user densities spanning areas of the order of several square-kilometers. However, there are several critical design issues that can be considered for future research. Most existing works assume a low bandwidth common control channel to exist for the exchange of control information between the SUs and the central controller. However, such common control channel may not be available. Nevertheless, a few works assume that a common control channel can be dynamically chosen by the SUs [24, 53].

The most imminent problem in these scenarios is the control channel saturation problem which occurs especially when the number of SUs in the system increases. In such a scenario, the control information (e.g. bandwidth requests) also increases resulting in the saturation on the common control channel. For example, consider a centralized scheme [38] which proposed that the SUs can negotiate the bandwidth and price of the bandwidth with a central controller. Depending on the changes in the availability of the bandwidth (and/or in the pricing of the bandwidth) the bandwidth requests from the SUs may rapidly increase causing control channel saturation. Hence, DSA schemes which consider the dynamic changes in the SU traffic load (e.g., both on control channel and on the data channels) can be investigated.

6.2.3 Research Challenges for Distributed CRN

The distributed CRN does not involve infrastructure. As there is little coordination, the SUs rely upon themselves to gain knowledge of the characteristics of the spectrum usage. For example, the SUs may have to perform full spectrum sensing thereby spending their sensing energy, and also wasting effective transmission time. When the aforementioned scenarios (e.g. time varying channel load) are added, the complexity of the distributed CRN design grows. Further, it will be a challenge to achieve fairness in the allocation and access of the licensed spectrum in a distributed scenario. Hence, it is a challenge to construct low cost and low complexity distributed CRN for the aforementioned scenarios.

Publications

- T. V. Krishna and A. Das, “A survey on MAC protocols in OSA networks,” *Comput. Netw.*, vol. 53, no. 9, pp. 1377–1394, june, 2009.
- V. K. Tumuluru, P. Wang and D. Niyato, “Channel status prediction for cognitive radio networks,” *Wireless Communications and Mobile Computing*, doi: 10.1002/wcm.1017.
- V. K. Tumuluru, P. Wang and D. Niyato, “A novel spectrum scheduling scheme for multi-channel cognitive radio network and performance analysis,” *IEEE Trans. on Vehicular Technology*, vol. 50, no. 4, May 2011.
- V. K. Tumuluru, P. Wang and D. Niyato, “Performance analysis of cognitive radio spectrum access with prioritized traffic,” *IEEE Trans. on Vehicular Technology*, *submitted*.
- V. K. Tumuluru, P. Wang and D. Niyato, “A neural network based spectrum prediction scheme for cognitive radio,” *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–5, may, 2010.
- V. K. Tumuluru, P. Wang and D. Niyato, “An opportunistic spectrum scheduling scheme for multi-channel cognitive radio networks,” *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pp. 1–5, sept. 2010.
- V. K. Tumuluru, P. Wang and D. Niyato, “Performance analysis of cognitive radio spectrum access with prioritized traffic,” *Communications (ICC), 2011 IEEE International Conference on*, *to be presented*.

Appendix A

List of Abbreviations

Abbreviation	Description
ACK	Acknowledgement
AIFS	Arbitration Inter-Frame Space
AMC	Adaptive Modulation and Coding
BP	Back-Propagation
BWA	Baum Welch Algorithm
CPOMDP	Constrained Partially Observable Markov Decision Process
CRN	Cognitive Radio Network
CS	Central Scheduler
CTMC	Continuous Time Markov Chain
CW	Contention Window
DCF	Distributed Coordination Function
DIFS	Distributed Inter-Frame Space
DSA	Dynamic Spectrum Access
DTMC	Discrete Time Markov Chain
EDCF	Enhanced Distributed Coordination Function
FCC	Federal Communications Commission
FSMC	Finite State Markov Chain
HMM	Hidden Markov Model
ILP	Integer Linear Program
MAC	Medium Access Control
MC	Markov Chain
MLP	Multilayer Perceptron
PER	Packet Error Rate
POMDP	Partially Observable Markov Decision Process
PU	Primary User
QoS	Quality of Service
RC	Rendezvous Channel
SINR	Signal to Interference and Noise Ratio
SNR	Signal to Noise Ratio
SU	Secondary User

Appendix B

List of Symbols

Symbol	Description
O	Number of layers in the MLP predictor.
p_g^o	Output of the neuron g in the o^{th} layer of the MLP predictor.
ν_g^o	Weighted sum of the inputs to the neuron g in the o^{th} layer of the MLP.
$\omega_{g,f}^o$	Adaptive weight connecting neuron g in the o^{th} layer of the MLP to a neuron f in the $o - 1^{th}$ layer.
τ	Order of the MLP predictor.
L_{mlp}	Length of the sensing history of a licensed channel during MLP training.
$\mathbf{s}_{mlp}^u$	Sensing history of channel u used in the MLP predictor training.
$\mathbf{s}_{mlp}$	Sensing history of an arbitrary channel used in the MLP predictor training.
n	Slot index.
s_n^u	State of PU activity on channel u in slot n .
s_n	State of PU activity on an arbitrary channel in slot n .
$\mathbf{s}_{mlp}^\tau$	Sensing data needed for one step prediction by the MLP predictor.
p_1^O	MLP predictor output.
$\hat{s}_{n+1}$	Predicted state of an arbitrary channel for slot $n + 1$.
s_{n+1}	Actual state of an arbitrary channel in slot $n + 1$.

Symbol	Description
ϵ_n	MLP error during training.
E	Mean square error criterion.
ω_n	General notation for arbitrary weight $\omega_{g,f}^o$ in slot n .
$\Delta\omega_n$	Update for weight ω_n in slot n .
μ'	Learning rate for MLP predictor.
η'	Momentum term for MLP predictor.
$\delta_g^o(n)$	Local gradient function for neuron g in o^{th} layer of the MLP predictor.
$\delta_h^{o+1}(n)$	Local gradient function for neuron h in $o + 1^{th}$ layer of the MLP predictor.
ω_{new}	New value of weight ω_n in slot n during training.
ω_{old}	Old value of weight ω_n in slot n during training.
N_S	Number of states in HMM predictor.
$\mathcal{S}$	Set of states in the HMM predictor.
S'_g	The g^{th} state in the set $\mathcal{S}$ of the HMM predictor.
Q_n	Variable representing the state of the HMM predictor in slot n .
E	Number of symbols that can be emitted by a state of the HMM predictor.
$\mathcal{P}$	Set of symbols that can be emitted by a state in the HMM predictor.
P_e	e^{th} symbol in the set $\mathcal{P}$.
κ	Length of the sensing data needed for one step prediction by the HMM.
$\mathbf{s}$	Sensing data needed for one step prediction by the HMM.
$A_{g',h'}$	Probability of transition from state $S_{g'}$ in slot n to state $S_{h'}$ in slot $n + 1$ of the HMM predictor.
$B_{h'}(P_e)$	Probability of emitting the e^{th} symbol from set $\mathcal{P}$ when in state $S_{h'}$ of the HMM predictor.
π	Probability distribution of the states in HMM predictor.
$\pi_{g'}$	Steady state probability of state $S_{g'}$ of the HMM predictor.

Symbol	Description
	Parameter vector for HMM predictor.
A	Transition probability matrix of the HMM predictor.
B	Emission probability matrix of the HMM predictor.
$s_{\kappa+1}$	Actual state of the PU activity on a channel in slot $\kappa + 1$.
$\hat{s}_{\kappa+1}$	Predicted state PU activity on the channel in slot $\kappa + 1$ by the HMM predictor.
$\alpha_n(g')$	Probability of reaching state $S_{g'}$ in slot n after observing the given sensing history from slot 1 to slot n .
$\beta_n(g')$	Probability of reaching state $S_{g'}$ in slot n after observing the given sensing history from slot n to slot κ .
$\chi_n(g', h')$	Probability of being in state $S_{g'}$ in slot n and in state $S_{h'}$ in slot $n + 1$ having observed the sensing history up to slot n .
$\sigma_n(g')$	Probability of being in state $S_{g'}$ in slot n having observed the sensing history up to slot n .
c_t	Scaling coefficient used in HMM predictor training.
ρ	Traffic intensity of the channel due to PU activity.
t_{serv}	Mean ON time of the PU activity on a channel.
t_{inter}	Mean inter-arrival time between the PU activity on a channel.
$P_p^e(Busy)$	Probability of wrongly predicting the busy state of a channel.
$P_p^e(Overall)$	Probability of wrongly predicting the any state of a channel.
$P_p^e(Idle)$	Probability of wrongly predicting the idle state of a channel.
SUT_{imp}	Percentage improvement in the spectrum utilization.
SE_{red}	Percentage reduction in the sensing energy.
N_c	Number of channels in the licensed spectrum.
CR_{sense}	A secondary user which uses sensing scheme only.
$CR_{predict}$	A secondary user which uses prediction based sensing scheme.
SUT_{sense}	Spectrum utilization measured for a CR_{sense} device.

Symbol	Description
$SUT_{predict}$	Spectrum utilization measured for a $CR_{predict}$ device.
I_{sense}	Number of idle channels sensed by a CR_{sense} device during the simulation time.
$I_{predict}$	Number of idle channels sensed by a $CR_{predict}$ device during the simulation time
$B_{predict}$	Number of busy channels predicted by a $CR_{predict}$ device during the simulation time.
L_o, L_{o+1}	Number of neurons in the layers o and $o + 1$ of the MLP respectively.
V	Number of secondary users in the CRN.
N_{slot}	Number of slots per frame.
Q_v	Transmission queue size of a secondary user.
λ_v	Packet arrival rate of a secondary user.
N_p	Maximum number of packets that can arrive in a SU's queue in a slot.
A_p, n_p	Number of packets arriving in a SU queue in slot n .
$\bar{\gamma}$	Average SNR used in the calculation of the channel mode.
(u, v)	Channel-user pair corresponding to channel u and SU v .
$\gamma^{u,v}$	Instantaneous SNR measured at the destination SU by the SU v on channel u .
$c_n^{u,v}$	Channel mode corresponding to channel-user pair (u, v) in slot n .
$c^{u,v}$	Channel mode corresponding to channel-user pair (u, v) for an arbitrary slot.
$T_F(u, v)$	Expected number of packet departures over frame F for channel-user pair (u, v) .
q_n^v	Queue status of SU v at the beginning of slot n .
$\mathbf{x}_n^{u,v}$	State observed by SU v corresponding to channel u only.
s_n	PU activity in slot n observed on an arbitrary channel by a tagged SU.
c_n	Mode corresponding to an arbitrary channel observed by a tagged SU in slot n .
q_n	Queue status of a tagged SU at the beginning of slot n .

Symbol	Description
$\mathbf{x}_n$	State of DTMC for a tagged SU corresponding to an arbitrary channel.
$T_n(u, v)$	Expected number of packet departures in slot n for channel-user pair (u, v) .
T_n	Expected number of packet departures in slot n for a tagged SU.
Π_n	State distribution corresponding to a tagged SU for any channel.
$\Pi_{(s_n=s', c_n=c', q_n=q')}$	State probability corresponding to a tagged SU in state $\mathbf{x}_n$
$\mathbf{P}(\mathbf{x}_{n+1} \mathbf{x}_n)$	Transition probability matrix for the state $\mathbf{x}_n$ corresponding to a tagged SU.
d_n	Number of packet departures in slot n for a tagged SU on the given channel.
b_n	Slot dependent parameter.
T_F	Expected number of packet departures over frame F for channel u with respect to the tagged SU.
T_p	Effective transmission time for data packets in the first slot of a frame.
T_s	Slot duration.
T_{ov}	Time taken by scheduling overhead in a frame.
R_1	Transmission rate of any data channel for the mode c_1
R_c	Transmission rate of the control channel.
L_p	Data packet size in bits.
L_{ov}	Scheduling overhead size in bits.
L_{ov}^{up}	Size of uplink scheduling overhead in bits.
L_{ov}^{down}	Size of downlink scheduling overhead in bits.
u_{index}	Size of a channel index expressed in bits.
v_{index}	Size of a SU index expressed in bits.
F	Frame index.
L_{T_F}	Number of bits to represent the expected number of packet departures over a frame.

Symbol	Description
$\mathcal{CS}_{u,v}$	Set containing expected number of packet departures over a frame corresponding to all channel-user pairs.
u^*, v^*	Indexes of assigned channel-user pair.
$\mathbf{s}_F$	Vector containing PU activity of all channels at the beginning of frame F .
s_F^u	PU activity on channel u at the beginning of frame F .
$\mathbf{c}_F^v$	Vector containing modes corresponding to all channels observed by SU v at the beginning of frame F .
q_F^v	queue status observed by SU v at the beginning of frame F .
$\mathbf{y}_F^v$	State of the DTMC corresponding to SU v and its observations on all channels at the beginning of frame F .
$\mathbf{y}_F^1$	State of the DTMC corresponding to a tagged SU and its observation on all channels at the beginning of frame F .
Π_F	Probability distribution of state $\mathbf{y}_F^1$.
M	One of the possible sequences of channel-user assignments done by the central scheduler.
m_t	t^{th} channel-user pair assigned by the central scheduler.
$M_{u,v}$	Maximum number of channel-user assignments.
K	Number of channel-user pairs sharing the highest value of T_F during the scheduling.
Π_{q^1}	Queue distribution of a tagged SU
η	Average SU throughput, expressed in packets/slot.
η_F	Average SU throughput, expressed in packets/frame.
P_B	Blocking probability for the packet arrivals for a tagged SU
$\bar{q}$	Mean queue size of a tagged SU.
$\bar{W}$	Mean waiting time of packets in a tagged SU's queue.
λ'_v	Effective arrival rate of packets entering a tagged SU's queue.
T_u	Actual throughput of channel u based on packet departures from all SUs.
$\bar{T}$	Average channel throughput, expressed in packets/slots.

Symbol	Description
SU_1	High priority SU.
SU_2	Low priority SU.
N_{sc}	Number of sub-channels per licensed channel.
i	Number of ongoing high priority SU calls.
j	Number of ongoing low priority SU calls.
k	Number of ongoing PU calls.
ζ	Number of sub-channels reserved for high priority SU calls.
DSA-C1, DSA-C2	Centralized DSA policies without buffer
DSA-D1	Distributed DSA policy without buffer.
DSA-C3, DSA-C4	Centralized DSA policies with buffer.
DSA-D2	Distributed DSA policy with buffer.
$[i, j, k]$	State of the system for DSA policies without buffer.
l and m	Number of displaced high priority and low priority SU calls during a PU arrival.
h_{sc}	Number of sub-channels available for handoff.
r	Total number of displaced SU calls during a PU arrival.
l' and m'	Number of unsuccessful handoff calls among high priority and low priority SU calls.
l_q and m_q	Number of high priority and low priority SU calls in the buffer.
$[i, j, k, l_q, m_q]$	State of the system for DSA policies with buffer.
Y	Number of occupied sub-channels corresponding to the state of the CTMC used for any DSA policy.
$\mathbf{1}(\cdot)$	Indicator function.
$\mathcal{N}_{l', m'}$	Set containing valid (l', m') corresponding to state of the CTMC used for a given DSA policy.
$R_{l', m'}$	Number of valid combinations of l , m , h_{sc} and r corresponding to a (l', m') pair in set $\mathcal{N}_{l', m'}$.
$\gamma_{l', m'}^{i, j, k}$	Rate of transition from state $[i, j, k]$ to state $[i - l', j - m', k + 1]$ in centralized DSA policies without buffer.

Symbol	Description
λ_p	Mean arrival rate of PU calls.
$\gamma_{0,j'}^{i,j+j',k-1}$	Rate of transition from state $[i, j + j', k]$ to state $[i, j, k]$ in centralized DSA policies without buffer.
$\gamma_{r'-j',j'}^{i+(r'-j'),j+j',k-1}$	Rate of transition from state $[i + (r' - j'), j + j', k - 1]$ to state $[i, j, k]$ in centralized DSA policies without buffer.
$AIFS_1$	AIFS corresponding to high priority SU calls.
$AIFS_2$	AIFS corresponding to low priority SU calls.
W_1	Contention window corresponding to high priority SU calls.
W_2	Contention window corresponding to low priority SU calls.
$backoff_x$	Time spent in backoff based on a random integer chosen from corresponding CW of SU calls $x \in \{1, 2\}$.
$\xi_{l',m'}^{i,j,k}$	Rate of transition from state $[i, j, k]$ to state $[i - l', j - m', k + 1]$ in distributed DSA policy without buffer.
Δ	Difference between the AIFS of high priority and low priority SU calls.
$\mathbf{z}$	State of the CTMC used for the DSA policies.
$\Pi_{\mathbf{z}}$	State distribution for the CTMC used for the DSA policies.
P_{B1}	Blocking probability for high priority SU calls.
P_{B2}	Blocking probability for low priority SU calls.
P_{F1}	Probability of forced termination for high priority SU calls.
P_{F2}	Probability of forced termination for low priority SU calls.
η_1	Throughput for high priority SU calls.
η_2	Throughput for low priority SU calls.
HD_1	Mean handoff delay for high priority SU calls.
HD_2	Mean handoff delay for low priority SU calls.

Bibliography

- [1] W. Lehr and L. W. McKnight, “Wireless internet access: 3G vs. WiFi?,” *Telecommunications Policy*, 2003.
- [2] “Future directions in cognitive radio network research,” *NSF Workshop Report*, march 2009.
- [3] NTIA, “U.S. frequency allocations.” <http://www.ntia.doc.gov/osmhome/allochrt.pdf>.
- [4] “Spectrum policy task force,” Tech. Rep. 02-135, Federal Communications Commission, nov. 2002.
- [5] S. J. Kim, E. C. Kim, S. Park, and J. Y. Kim, “Dynamic spectrum allocation with variable bandwidth for cognitive radio systems,” in *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, pp. 106–109, sept. 2009.
- [6] J. Mitola III and G. Maguire Jr., “Cognitive radio: making software radios more personal,” *Personal Communications, IEEE*, vol. 6, pp. 13–18, aug. 1999.
- [7] “Evaluation of performance of prototype TV-band white space devices phase II,” Tech. Rep. 08-TR-1005, Federal Communications Commission, oct. 2008.
- [8] F. Seelig, “A description of the august 2006 xg demonstrations at fort a.p. hill,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 1–12, april 2007.
- [9] C. Cordeiro, K. Challapali, D. Birru, and S. N. Shankar, “IEEE 802.22: An introduction to the first wireless standard based on cognitive radios,” *Journal of Communications*, vol. 1, no. 1, pp. 38–47, 2006.
- [10] T. Kamakaris, M. Buddhikot, and R. Iyer, “A case for coordinated dynamic spectrum access in cellular networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 289–298, nov. 2005.

- [11] M. McHenry, E. Livsics, T. Nguyen, and N. Majumdar, “XG dynamic spectrum sharing field test results,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 676–684, april 2007.
- [12] S. Jones, E. Jung, X. Liu, N. Merheb, and I.-J. Wang, “Characterization of spectrum activities in the u.s. public safety band for opportunistic spectrum access,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 137–146, april 2007.
- [13] S. Haykin, “Cognitive radio: Brain-empowered wireless communications,” *Selected Areas in Communications, IEEE Journal on*, 2005.
- [14] N. Shah, T. Kamakaris, U. Tureli, and M. Buddhikot, “Wideband spectrum probe for distributed measurements in cellular band,”
- [15] Y. Hur, J. Park, W. Woo, K. Lim, C.-H. Lee, H. Kim, and J. Laskar, “A wideband analog multi-resolution spectrum sensing (MRSS) technique for cognitive radio (CR) systems,” in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, p. 4 pp., 2006.
- [16] Z. Tian and G. B. Giannakis, “A wavelet approach to wideband spectrum sensing for cognitive radios,” in *Cognitive Radio Oriented Wireless Networks and Communications, 2006. 1st International Conference on*, pp. 1–5, june 2006.
- [17] W. Zhang, R. Mallik, and K. Letaief, “Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks,” *Wireless Communications, IEEE Transactions on*, vol. 8, pp. 5761–5766, december 2009.
- [18] T. V. Krishna and A. Das, “A survey on MAC protocols in OSA networks,” *Computer Networks*, vol. 53, no. 9, pp. 1377–1394, 2009.
- [19] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
- [20] J. Jia, Q. Zhang, and X. Shen, “HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management,” *Selected Areas in Communications, IEEE Journal on*, vol. 26, pp. 106–117, jan. 2008.
- [21] Y. Chen, Q. Zhao, and A. Swami, “Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors,” *Information Theory, IEEE Transactions on*, vol. 54, pp. 2053–2071, may 2008.

- [22] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran, “MAC-layer scheduling in cognitive radio based multi-hop wireless networks,” in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, 2006.
- [23] H. Su and X. Zhang, “Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 26, pp. 118–129, jan. 2008.
- [24] J. Zhao, H. Zheng, and G.-H. Yang, “Distributed coordination in dynamic spectrum allocation networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 259–268, nov. 2005.
- [25] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, “Cogmesh: A cluster-based cognitive radio network,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 168–178, april 2007.
- [26] G. Ganesan and Y. Li, “Cooperative spectrum sensing in cognitive radio networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 137–143, nov. 2005.
- [27] H. Jiang, L. Lai, R. Fan, and H. Poor, “Optimal selection of channel sensing order in cognitive radio,” *Wireless Communications, IEEE Transactions on*, vol. 8, pp. 297–307, jan. 2009.
- [28] A. Motamedi and A. Bahai, “Optimal channel selection for spectrum-agile low-power wireless packet switched networks in unlicensed band,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2008, pp. 9:1–9:10, jan. 2008.
- [29] L. Lai, H. El Gamal, H. Jiang, and H. V. Poor, “Cognitive medium access: Exploration, exploitation, and competition,” *Mobile Computing, IEEE Transactions on*, vol. 10, pp. 239–253, feb. 2011.
- [30] Q. Zhao, L. Tong, A. Swami, and Y. Chen, “Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, pp. 589–600, april 2007.
- [31] X. Liu, B. Krishnamachari, and H. Liu, “Channel selection in multi-channel opportunistic spectrum access networks with perfect sensing,” in *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pp. 1–8, april 2010.
- [32] K. P. Murphy, “A survey of pomdp solution techniques,” tech. rep., 2000.

- [33] Z. Wen, T. Luo, W. Xiang, S. Majhi, and Y. Ma, "Autoregressive spectrum hole prediction model for cognitive radio systems," in *Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on*.
- [34] S. Yarkan and H. Arslan, "Binary time series approach to spectrum prediction for cognitive radio," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, 2007.
- [35] I. Akbar and W. Tranter, "Dynamic spectrum allocation in cognitive radio using hidden markov models: Poisson distributed case," in *SoutheastCon, 2007. Proceedings. IEEE*, pp. 196–201, march 2007.
- [36] C.-H. Park, S.-W. Kim, S.-M. Lim, and M.-S. Song, "HMM based channel status predictor for cognitive radio," in *Microwave Conference, 2007. APMC 2007. Asia-Pacific*, pp. 1–4, dec. 2007.
- [37] S. Haykin, *Adaptive filter theory*. Upper Saddle River, NJ, USA: Prentice Hall, 4th ed., 2001.
- [38] M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMnet: new directions in wireless networking using coordinated dynamic spectrum," in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pp. 78–85, june 2005.
- [39] J. Tang, S. Misra, and G. Xue, "Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks," *Comput. Netw.*, vol. 52, pp. 2148–2158, aug. 2008.
- [40] K. Hamdi, W. Zhang, and K. Letaief, "Uplink scheduling with qos provisioning for cognitive radio systems," in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pp. 2592–2596, march 2007.
- [41] D. Gozupek and F. Alagoz, "Throughput and delay optimal scheduling in cognitive radio networks under interference temprature constraints," *Journal of Communications and Networks*, vol. 11, pp. 147–155, april 2009.
- [42] C. Tian and D. Yuan, "A novel multiuser diversity based scheduler with QoS support for cognitive radio networks," in *Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual*, pp. 310–316, may 2009.

- [43] M. Rashid, J. Hossain, E. Hossain, and V. Bhargava, "Opportunistic spectrum access in cognitive radio networks: A queueing analytic model and admission controller design," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 4647–4652, nov. 2007.
- [44] E. Wong and C. Foh, "Analysis of cognitive radio spectrum access with finite user population," *Communications Letters, IEEE*, vol. 13, pp. 294–296, may 2009.
- [45] Y. Zhang, "Dynamic spectrum access in cognitive radio wireless networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, pp. 4927–4932, may 2008.
- [46] S. Tang and B. Mark, "Analysis of opportunistic spectrum sharing with markovian arrivals and phase-type service," *Wireless Communications, IEEE Transactions on*, vol. 8, pp. 3142–3150, june 2009.
- [47] X. Zhu, L. Shen, and T.-S. Yum, "Analysis of cognitive radio spectrum access with optimal channel reservation," *Communications Letters, IEEE*, vol. 11, pp. 304–306, april 2007.
- [48] S. Kannappa and M. Saquib, "Performance analysis of a cognitive network with dynamic spectrum assignment to secondary users," in *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–5, may 2010.
- [49] J. Perez-Romero, O. Salient, R. Agustí, and L. Giupponi, "A novel on-demand cognitive pilot channel enabling dynamic spectrum allocation," in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 46–54, april 2007.
- [50] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking, MobiCom '04*, pp. 216–230, 2004.
- [51] "Standard ECMA-392: MAC and PHY for operation in TV white space." <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-392.pdf>, dec. 2009.
- [52] L. Ma, X. Han, and C.-C. Shen, "Dynamic open spectrum sharing mac protocol for wireless ad hoc networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, nov. 2005.

- [53] C. Cordeiro and K. Challapali, “C-MAC: A cognitive MAC protocol for multi-channel wireless networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 147–157, april 2007.
- [54] L. Ma, C.-C. Shen, and B. Ryu, “Single-radio adaptive channel algorithm for spectrum agile wireless ad hoc networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pp. 547–558, april 2007.
- [55] A. T. Hoang, D. Wong, and Y.-C. Liang, “Design and analysis for an 802.11-based cognitive radio network,” in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pp. 1–6, april 2009.
- [56] A. Benslimane, A. Ali, A. Kobbane, and T. Taleb, “A new opportunistic MAC layer protocol for cognitive IEEE 802.11-based wireless networks,” in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pp. 2181–2185, sept. 2009.
- [57] K. J. Kim, J. S. Park, Y. H. Bae, and B. D. Choi, “Performance analysis of a slotted multi-channel MAC protocols for cognitive radio networks,” in *Proceedings of the 5th International Conference on Queueing Theory and Network Applications, QTNA '10*, pp. 148–155, 2010.
- [58] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, pp. 535–547, mar. 2000.
- [59] H. Su and X. Zhang, “Design and analysis of a multi-channel cognitive mac protocol for dynamic access spectrum networks,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1–7, nov. 2008.
- [60] A. Chia-Chun Hsu, D. Weit, and C.-C. Kuo, “A cognitive mac protocol using statistical channel allocation for wireless ad-hoc networks,” in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pp. 105–110, march 2007.
- [61] G. Brire and M. L. Chaudhry, “Computational analysis of single-server bulk-service queues, M/GY/1,” *Advances in Applied Probability*, vol. 21, no. 1, pp. 207–225, 1989.
- [62] Q. Zhao, S. Geirhofer, L. Tong, and B. Sadler, “Opportunistic spectrum access via periodic channel sensing,” *Signal Processing, IEEE Transactions on*, vol. 56, pp. 785–796, feb. 2008.

- [63] S. Shetty, M. Song, C. Xin, and E. Park, “A learning-based multiuser opportunistic spectrum access approach in unslotted primary networks,” in *INFOCOM 2009, IEEE*, pp. 2966–2970, april 2009.
- [64] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 1998.
- [65] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, feb. 1989.
- [66] K. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *Neural Networks, IEEE Transactions on*, vol. 1, pp. 4–27, mar. 1990.
- [67] P. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, pp. 1550–1560, oct. 1990.
- [68] C. Bystroff and A. Krogh, “Hidden markov models for prediction of protein features,” in *Protein Structure Prediction*, vol. 413, pp. 173–198, 2008.
- [69] C. Borgelt and R. Kruse, “Speeding up fuzzy clustering with neural network techniques,” in *Fuzzy Systems, 2003. FUZZ ’03. The 12th IEEE International Conference on*, vol. 2, pp. 852–856, may 2003.
- [70] Q. Liu, S. Zhou, and G. Giannakis, “Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design,” *Wireless Communications, IEEE Transactions on*, vol. 4, pp. 1142–1153, may 2005.
- [71] L. Kleinrock, *Theory, Volume 1, Queueing Systems*. Wiley-Interscience, 1975.
- [72] U. Wiggins, R. Kannan, V. Chakravarthy, and A. Vasilakos, “Data-centric prioritization in a cognitive radio network: A quality-of-service based design and integration,” in *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pp. 1–11, oct. 2008.
- [73] C. Ghosh, S. Chen, D. Agrawal, and A. Wyglinski, “Priority-based spectrum allocation for cognitive radio networks employing nc-ofdm transmission,” in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pp. 1–5, oct. 2009.
- [74] L.-C. Wang, A. Chen, and D. S. Wei, “A cognitive MAC protocol for QoS provisioning in ad hoc networks,” *Physical Communication*, vol. 3, no. 2, pp. 105–118, 2010.

- [75] S. Choi, J. del Prado, S. S. N, and S. Mangold, “IEEE 802.11 e contention-based channel access (EDCF) performance evaluation,” in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 2, pp. 1151–1156, may 2003.
- [76] W. J. Stewart, *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.