Context Awareness and Intelligence in Cognitive Radio Networks: Design and Applications

by

Kok-Lim Yau

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Network Engineering.

Victoria University of Wellington
2010

Abstract

CR technology, which is the next-generation wireless communication system, improves the utilization of the overall radio spectrum through dynamic adaptation to local spectrum availability. In CR networks, unlicensed or Secondary Users (SUs) may operate in underutilized spectrum (called white spaces) owned by the licensed or Primary Users (PUs) conditional upon PUs encountering acceptably low interference levels. Ideally, the PUs are oblivious to the presence of the SUs.

Context awareness enables an SU to sense and observe its operating environment, which is complex and dynamic in nature; while *intelligence* enables the SU to learn knowledge, which can be acquired through observing the consequences of its prior action, about its operating environment so that it carries out the appropriate action to achieve optimum network performance in an efficient manner without following a strict and static predefined set of policies. Traditionally, without the application of intelligence, each wireless host adheres to a strict and static predefined set of policies, which may not be optimum in many kinds of operating environment. With the application of intelligence, the knowledge changes in line with the dynamic operating environment. This thesis investigates the application of an artificial intelligence approach called reinforcement learning to achieve context awareness and intelligence in order to enable the SUs to sense and utilize the *high* quality white spaces.

To date, the research focus of the CR research community has been primarily on the physical layer of the open system interconnection model. The research into the data link layer is still in its infancy, and our research work focusing on this layer has been pioneering in this field and has at-

tacted considerable international interest. There are four major outcomes in this thesis.

Firstly, various types of multi-channel medium access control protocols are reviewed, followed by discussion of their merits and demerits. The purpose is to show the additional functionalities and challenges that each multi-channel medium access control protocol has to offer and address in order to operate in CR networks. Secondly, a novel cross-layer based quality of service architecture called C^2net for CR networks is proposed to provide service prioritization and tackle the issues associated with CR networks. Thirdly, reinforcement learning is applied to pursue context awareness and intelligence in both centralized and distributed CR networks. Analysis and simulation results show that reinforcement learning is a promising mechanism to achieve context awareness and intelligence. Fourthly, the versatile reinforcement learning approach is applied in various schemes for performance enhancement in CR networks.

Acknowledgments

A special acknowledgment is reserved for my thesis supervisors, Dr. Peter Komisarczuk and Dr. Paul Teal, for their help, guidance, suggestions, contributions, encouragements, as well as ensuing this research has kept on course. Peter has constantly challenged me to investigate further and opened doors for me to network with others in this research field; while Paul has scrutinized all my research papers before submitting them for publication in a timely manner. Thanks to Paul for providing his computing resource that has facilitated rapid progress in my research.

I would like to express my gratitude to Dr. Alan Coulsan and Chatu Lokuge from the Industrial Research Limited (IRL), New Zealand for their initial guidance, suggestions and discussions. Back then, I was blissfully unaware of cognitive radio. Alan and Chatu had introduced me this new and exciting research area.

My compliments to Dr. David Grace from the Communications Research Group at the University of York in UK. David, who is a prominent researcher in cognitive radio, was my mentor during a three-week research attachment. David, who is also the Chair of the Worldwide Universities Network (WUN) Cognitive Communications Consortium (COGCOM), had opened doors for Victoria University to foster future collaboration with world renowned researchers in this field.

Great thanks to Chris Perera from the New Zealand Ministry of Economic Development. She had offered advice on this research as a radio spectrum regulator during my PhD proposal presentation, meeting, and

seminar.

Great thanks also go to Prof. Winston Seah, Dr. Marcus Frean and Dr. Qiang Fu from the School of Engineering and Computer Science at Victoria University who have contributed to my research. Winston has constantly provided reviews and advice on my research papers before submitting them for publication in a timely manner. He has also provided research direction during a dry run of my PhD proposal presentation. Marcus was there to help and explain about reinforcement learning. Qiang has provided reviews and advice on my research outcomes.

My deepest thanks go to the New Zealand Government and Education New Zealand to fully fund my PhD program. This work would have been impossible without their generous funding. Thanks to Victoria University Research Fund, the Faculty of Science and the School of Engineering and Computer Science at Victoria University for funding conferences, workshops and seminars attendance, as well as a research attachment.

Last but not least, I would like to express special thanks to my parents, family and friends for their constant support and words of wisdoms and belief that kept me focused and driven throughout my PhD.

Publications Produced

Throughout the PhD candidature, I have produced the following fully-refereed publications:

1. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Enhancing Network Performance in Distributed Cognitive Radio Networks: Single-Agent and Multi-Agent Reinforcement Learning Approach. In *Proceedings of the 35th IEEE Conference on Local Computer Networks (LCN)* (Denver, Colorado, US, October 2010), IEEE.

2. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Achieving efficient and optimal joint action in distributed cognitive radio networks using payoff propagation. In *Proceedings of the International Conference on Communications (ICC)* (Cape Town, South Africa, May 2010), IEEE.

3. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Applications of reinforcement learning to cognitive radio networks. In *Proceedings of the 1st International Workshop on Cognitive Radio Interfaces and Signal Processing (CRISP) at the International Conference on Communications (ICC)* (Cape Town, South Africa, May 2010), IEEE.

4. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Context awareness and intelligence in distributed cognitive radio networks: A reinforcement learning approach. In *Proceedings of the Australian Commu-*

nications Theory Workshop (AusCTW) (Canberra, Australia, February 2010), IEEE.

5. **YAU, K.-L. A.**, KOMISARCZUK , P., AND TEAL, P. D. Quality of Service (QoS) provisioning in cognitive wireless ad hoc networks: Challenges, design approaches, and open issues. In *Quality of Service Architectures for Wireless Networks: Performance Metrics and Management*, S. Adibi, T. Tofigh, S. Parekh, and R. Jain, Eds. Information Science Reference, IGI Global, US, January 2010, ch. 25, pp. 575-594.

6. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Performance analysis of reinforcement learning for achieving context-awareness and intelligence in cognitive radio networks. In *Proceedings of the 9^{th} IEEE International Workshop on Wireless Local Networks (WLN) at the 34^{th} IEEE Conference on Local Computer Networks (LCN)* (Zurich, Switzerland, October 2009), IEEE.

7. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Cognitive radio-based wireless sensor networks: Conceptual design and open issues. In *Proceedings of the 2^{nd} IEEE International Workshop on Wireless and Internet Services (WISe) at the 34^{th} IEEE Conference on Local Computer Networks (LCN)* (Zurich, Switzerland, October 2009), IEEE.

8. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. A Context-aware and intelligent dynamic channel selection scheme for cognitive radio networks. In *Proceedings of the 4^{th} International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)* (Hannover, Germany, June 2009), IEEE.

9. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. A survey on multi-channel Medium Access Control (MAC) protocols: A cognitive radio perspective. In *Proceedings of the 7^{th} New Zealand Computer Science Research Student Conference (NZCSRSC)* (Auckland, New Zealand, April 2009).

10. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. Medium Access Control (MAC) protocols for cognitive radio networks: Recent advances and design considerations. In *Proceedings of the 7th New Zealand Computer Science Research Student Conference (NZCSRSC)* (Auckland, New Zealand, April 2009).

11. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. C^{2}net: A cross-layer Quality of Service (QoS) architecture for cognitive wireless ad hoc networks. In *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC)* (Adelaide, Australia, December 2008), IEEE.

12. **YAU, K.-L. A.**, KOMISARCZUK, P., and TEAL, P. D. On multi-channel MAC protocols in cognitive radio networks. In *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC)* (Adelaide, Australia, December 2008), IEEE.

Contents

Chapter 1

Introduction

Let's start at the very beginning, a very good place to start,
when you read, you begin with ABC,
when you sing, you begin with do-re-me,
when you research into Cognitive Radio, you begin with Context Awareness and Intelligence...[1]

This thesis presents pioneering work in the field of Cognitive Radio (CR) [1] networks including leverage from existing technologies, a Quality of Service (QoS) architecture, and mechanisms to achieve context awareness and intelligence.

Traditional static spectrum allocation policies have been imposed to grant each wireless service exclusive usage of certain spectrum bands, leaving several spectrum bands unlicensed for industrial, scientific and medical purposes. The tremendous growth in ubiquitous low-cost wireless applications that utilize the unlicensed spectrum bands has laid increasing stress on these limited and scarce radio spectrum resources. Studies sponsored by the Federal Communications Commission (FCC) discovered that the current static spectrum allocation has led to overall low

[1] with apologies to Rodgers and Hammerstein

spectrum utilization where up to 70% of the allocated licensed spectrum remains unused (these are called white spaces) at any one time even in a crowded area [2]. The white space is defined by usage time, frequency and maximum transmission power at a particular location. CR [1] is a novel and promising paradigm for the next-generation wireless communication system that enables an unlicensed user to improve utilization of the overall radio spectrum through dynamic adaptation to local spectrum availability in both licensed and unlicensed spectrum.

CR enables each unlicensed or Secondary User (SU) to sense white space and change its transmission and reception parameters, including operating frequency, adaptively in order to opportunistically use the white space in different channels. Ideally, the licensed or Primary Users (PUs) are oblivious to the presence of SUs. The SUs may operate in the white space conditional upon PUs encountering acceptably low interference levels.

We define *context awareness* and *intelligence* as follows:

- Context awareness enables an SU to sense and observe its complex and dynamic operating environment.
- Intelligence enables an SU to acquire knowledge, which can be learned through observing the consequences of its prior action, about its operating environment so that it carries out the right action at the right time to achieve optimum network performance in an efficient manner without adhering to a strict and static predefined set of policies.

The notion of context awareness and intelligence is very closely related to the concept of Cognition Cycle (CC) [3]. The CC is a state machine, which is embodied in each SU, that defines the mechanisms related to achieving context awareness and intelligence including observation, orientation, learning, planning, decision making, and action selection. The CC is the key element in the design of various applications in CR net-

works such as Dynamic Channel Selection (DCS), topology management, congestion control and scheduling. Hence, a good implementation of the context awareness and intelligence mechanism is of paramount importance, and this is the main focus of this thesis. Other focuses include leverage from existing technologies to this new research area, and a novel QoS architecture for CR networks.

In this chapter, we will present our motivation, goals, major contributions and thesis outline.

1.1 Motivation

Cognitive radio technology has brought about a paradigm shift in the way an SU defines its operating policy, which is a set of decision rules that determine how the SU should behave in various scenarios. Traditionally, the policy is hard-coded into the wireless host. For instance, using a fixed lookup table, a wireless host chooses its modulation technique, such as Quadrature Amplitude Modulation (QAM) and Binary Phase Shift Keying (BPSK), according to different levels of Signal-to-Noise Ratio (SNR). In CR networks, an SU must be able to sense and utilize the *high* quality white space in an efficient manner without adhering to a strict and static predefined set of policies. This is because a static policy is less likely to be applicable in all circumstances in a complex and dynamic operating environment. This has inevitably brought the concept of context awareness and intelligence into play.

The main focus of this thesis is to design practical and simple mechanisms to achieve context awareness and intelligence with respect to a particular application in CR networks, namely Dynamic Channel Selection (DCS). DCS provides the strategy for SUs to select a data channel from the available licensed channels for data packet transmission given that the objective is to increase network-wide throughput, and decrease delay for QoS provisioning. Context awareness and intelligence approaches can be

applied in various applications as shown in Figure 1.1. Accordingly, we have initiated a new and important research area in the field of CR networks, namely context awareness and intelligence. Nonetheless, this is a daunting challenge as we expect that the context awareness and intelligence approaches are the universal solution of most problems and open issues in CR networks.

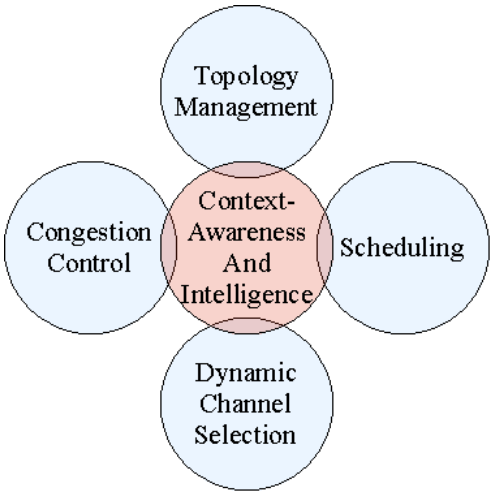


Figure 1.1: The context awareness and intelligence approach is applied in various applications.

This thesis applies Reinforcement Learning (RL) [4], which is an artificial intelligence or machine learning technique, to achieve context awareness and intelligence. The RL, which is a machine learning technique that improves system performance, has the following characteristics:

- *Unsupervised* learning approach. In unsupervised learning, there is no external teacher or critic to oversee the learning process [5]. In other words, an SU learns the knowledge about its operating environment by itself.

- *Online* learning approach. In online learning, an SU learns the knowledge on the fly while carrying out its normal operation, rather

than using empirical data or experimental results from the laboratory.

- *Simple modeling* learning approach. The RL approach models the performance metric(s) of interest and improve it as a whole, rather than modeling the complex and dynamic operating environment. For instance, instead of tackling every single factor that affects network performance such as wireless channel condition and nodal mobilily, RL models the network performance, such as throughput, that covers a wide range of factors that can affect the network performance.

1.2 Goals

CR network is a new emerging research area that presents important challenges. This thesis presents initial work on possible leverage from existing technologies to CR, and a cross-layer QoS architecture for CR networks. The initial work has been instrumental in defining the research direction so that the research into context awareness and intelligence remains relevant and important. For instance, the context awareness and intelligence approach is applicable in the applications proposed in our cross-layer QoS architecture for CR networks. The main goal of this thesis is to design mechanisms to achieve context awareness and intelligence. This thesis investigates the following research questions:

- What are the possible methods of technology leverage from multi-channel Medium Access Control (MAC) protocols to cognitive MAC protocols?

- What is an appropriate QoS architecture for CR networks?

- How are context awareness and intelligence best achieved in centralized CR networks?

- How are context awareness and intelligence best achieved in distributed CR networks?
- How can we apply these context awareness and intelligence approaches to QoS provision for CR networks?

1.3 Major Contributions

This thesis has contributed to pioneering work in the field of CR networks, specifically leverage from existing technologies, a QoS architecture, and mechanisms to achieve context awareness and intelligence. The key contributions are summarized as follows:

1. Possible technology leverages from existing multi-channel MAC to cognitive MAC are proposed. This work is the first attempt to investigate technology leverage for CR networks. The contributions of this work have led to the publication of [6], [7] and [8].
2. A novel cross-layer QoS architecture, along with its challenges and open issues, is proposed. This work is the first attempt to model a QoS architecture as a unified solution for CR networks. The contributions of this work have led to the publication of [9] and [10].
3. Analyses and simulations show that RL is a good approach to achieve context awareness and intelligence, with respect to the application of DCS, in centralized and distributed CR networks. The contributions of this work have led to the publication of [11], [12], [13], [14] and [15].
4. The RL approach is proposed for various applications for the cross-layer QoS architecture to achieve context awareness and intelligence for performance enhancement in CR networks. The contributions of this work have led to the publication of [16] and [17].

1.4 Thesis Outline

The rest of this thesis is structured into the following chapters:

- Chapter 2 provides overviews on traditional spectrum allocation policy, CR networks, cognition cycle as well as current research trends and common assumptions.
- Chapter 3 reviews various types of multi-channel MAC protocols including their operations, merits and demerits in order to present possible technology leverage to CR. The purpose is to introduce the additional functionalities and challenges that each multi-channel MAC protocol has to offer and address in order to function well in cognitive wireless ad hoc networks.
- Chapter 4 provides a novel cross-layer QoS architecture, namely C^2net, for cognitive wireless ad hoc networks. Various cross-layer applications such as DCS, scheduling and congestion control are proposed. Research challenges and open issues in realizing the C^2net architecture are also discussed.
- Chapter 5 presents RL as an approach to achieve context awareness and intelligence in CR networks. Various new features not used in the traditional RL approach are presented.
- Chapter 6 focuses on achieving context awareness and intelligence using the RL approach, with respect to the application of DCS, in centralized CR networks.
- Chapter 7 focuses on achieving context awareness and intelligence using the RL approach, with respect to the application of DCS, in distributed CR networks.
- Chapter 8 shows how to apply the RL approach to model various applications in C^2net in order to enhance performance in CR networks.

- Chapter 9 draws conclusions and discusses future research directions.

Chapters 3 to 8 provide novel contributions; and Chapters 6 and 7 provide major contributions of this thesis, specifically on achieving context awareness and intelligence in CR networks.

Chapter 2

Background

This chapter provides overviews on traditional spectrum allocation policy, CR networks, cognition cycle, as well as current research trends and common assumptions in CR networks.

2.1 Traditional Spectrum Allocation Policy

Traditionally, radio spectrum has been partitioned into ranges of licensed and unlicensed spectrum (or frequency) bands through a static spectrum allocation policy. Some small areas of the spectrum bands, such as the Industrial, Scientific and Medical (ISM) and Unlicensed National Information Infrastructure (UNII), are allocated to unlicensed users who contend among themselves for access to this free resource. Unlicensed users are forbidden to access any of the licensed spectrum bands that have been purchased. Many popular wireless communication systems, including Bluetooth [18], WiFi [19], WiMAX [20], and Zigbee [21], have been operating in unlicensed spectrum bands without incurring any spectrum cost. Other devices such as microwave ovens and cordless phones also operate in those spectrum bands.

2.1.1 An Analogy of Traditional Spectrum Allocation Policy

As an analogy, the static spectrum allocation policy is like a swimming competition where the limited pool (radio spectrum) is divided into many lanes (spectrum bands). Each contestant (spectrum user) is assigned a lane that is used throughout its communication session. The contestant is forbidden to cross over into other lanes or interfere with the other contestants; and the contestant does not generally occupy the whole of the lane. The lanes that represent the unlicensed spectrum bands are typically crowded with many competitors that jostle for space. As the number of unlicensed users increases, it is inevitable that the unlicensed lane becomes more congested. As a consequence, the QoS of the unlicensed users is adversely affected. A scheme that allows use of the contestants' lanes, but without interference to the contestants could alleviate much of the congestion.

2.2 Cognitive Radio Networks

The FCC Spectrum Policy Task Force (2002) discovered that the current static spectrum allocation policy has led to overall low spectrum utilization where up to 70% of the allocated licensed spectrum bands remain unused (these are called white spaces) at any one time even in a crowded area [2]. Hence, the main reason of spectrum scarcity among the unlicensed users is, in fact, because of the static spectrum allocation policy that is inefficient. The white space is defined by usage time, frequency and maximum transmission power at a particular location. Consequently, CR has been proposed so that unlicensed users or SUs are allowed to use the white space of licensed users' or PUs' spectrum bands conditional upon PU encountering acceptably low interference levels.

2.2.1 Two Prominent Characteristics of Cognitive Radio

CR technology enables an SU to change its transmission and reception parameters including operating frequencies. Chapter 1 on page 2 provides the definition of context awareness and intelligence. With respect to DCS, two prominent characteristics of CR are as follows:

- *Context awareness*. Channel sensing capability enables an SU to sense and observe across a wide range of spectrum bands to identify white spaces.

- *Intelligence*. A learning mechanism enables an SU to learn information about the white spaces through observing the consequences of its prior actions; for instance, whether a recent data packet transmission was successful or unsuccessful. This enables the SU to identify white spaces and to allocate data packets opportunistically to *high* quality white spaces at different channels in an efficient manner for performance enhancement without adhering to a strict and static predefined set of policies.

2.2.2 An Analogy of Cognitive Radio

Let's take another analogy. Suppose you are driving to school or work during the peak hours. While driving straight ahead, you find that the lane becomes congested. To arrive on time, you carefully switch to a nearby lane that is less congested, while ensuring that you don't collide with the other road users. The same principle is applicable to CR. If its current licensed or unlicensed spectrum band is fully utilized, an SU switches its operating frequency to another spectrum band without interfering with the PU activity. This occurs when the licensed channel is underutilized or contains white spaces. Through accessing the white spaces in licensed spectrum bands dynamically, the overall spectrum utilization improves. In CR networks, one of the most important tasks is therefore to create a

"friendly" environment for the coexistence between the PUs and the SUs as shown in Figure 2.1.

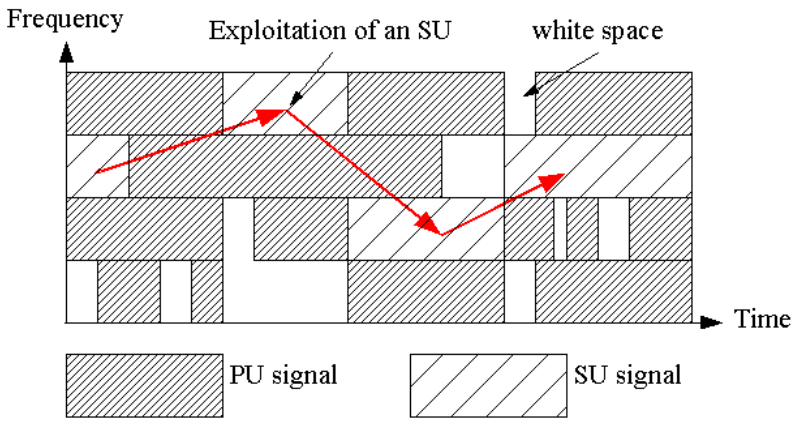


Figure 2.1: An SU exploits white spaces across various channels.

2.2.3 Exploitation of White Space by Secondary Users

In Figure 2.1, the licensed spectrum utilization from the PUs at a particular location is represented by the time and frequency axes. The PUs have higher authority over the licensed spectrum bands. An SU switches its channel across various spectrum bands from time to time in order to utilize the white spaces in the licensed spectrum it is sensing. Since the white space is location dependent, for a successful communication, the white space must be available at both the SU transmitter and receiver. In mobile networks, this is particularly important if the SUs are moving at high speed as from moment to moment each location may have different PU spectrum utilization. However, since the transmission range of the PU is often large, such as transmission for the TV bands, the spectrum utilization of the PU at various locations may have wide geographic uniformity, and thus collaboration in channel sensing for white spaces among the SUs is an effective means to avoid collision with the PU's transmissions.

Not only do the SUs have to search for white spaces, they also need

to use the white spaces efficiently. According to [22], the SUs are expected to operate over a wide range of non-contiguous spectrum bands: 400-800MHz (UHF TV bands) and 3-10GHz. The time scale of the spectrum occupancy varies from milliseconds to hours depending on the activity levels of the PUs.

2.2.4 An Example of Cognitive Radio Networks

An example of emerging standards based CR network is the IEEE 802.22 Wireless Regional Area Network (WRAN) [23], which is a centralized CR network. The IEEE 802.22 working group has been working towards developing CR-based Medium Access Control-Physical (MAC-PHY) air interface for SUs to operate in TV bands. In this approach, the SUs access to licensed spectrum bands is controlled by a centralized Base Station (BS).

2.2.5 A Scenario of Cognitive Radio Network under Consideration

This thesis focuses on centralized and distributed CR networks. The distributed CR network is called Cognitive Wireless Ad hoc Network (CWAN). As an alternative to the infrastructure oriented solution of IEEE 802.22, we consider a cooperative peer-to-peer model such as traditional ad hoc networks in CWAN. The CWAN provides a dynamic mechanism to interconnect SUs through the provision of network relay functions and such networks can be stationary or mobile in nature.

Our primary design focus for centralized and distributed CR networks are around deployment in a complex wireless communication and a broadband access scenario comprised of various heterogeneous stationary and mobile CR hosts or SUs in a densely populated urban or metropolitan area. Consumers may access the CR network using consumer devices, laptops, mobile phones, PDAs, vehicular intelligent transportation systems

and so on, in a single hop or a multihop manner, for example to allow extension of hot spot coverage. Certain unlicensed spectrum bands such as the ISM and UNII bands are highly utilized; however, with CR technology, an SU could search for and utilize unused licensed spectrum bands. This scenario, as shown in Figure 2.2, is may be useful for telecom operators to extend wireless access among subscribers that are outside BS coverage for example.

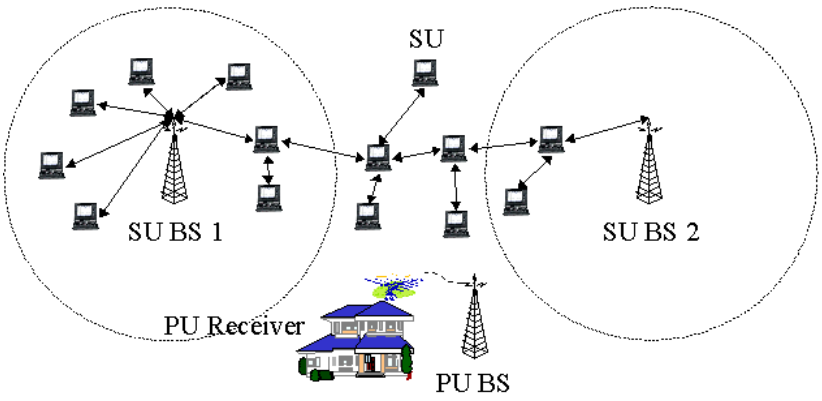


Figure 2.2: CWAN deployment scenario.

2.3 Cognition Cycle

The original cognition cycle, which portrays the notion of context awareness and intelligence in CR networks, is presented in [3]. The CC was first introduced by the Father of CR, J. Mitola III [3]. The adage "practice makes perfect" is the concept that the CC was founded upon. While making a perfect system is a far more difficult endeavor, a CC aims to achieve a system with better performance as time goes by. Although the CC has not been extensively applied in network protocol design, it has great potential for system enhancement.

2.3.1 A Simplified Version of Cognition Cycle

This section describes CC based on the RL [4] approach. A simplified version of CC is shown in Figure 2.3. We model each SU in a CR network as a learning agent or a decision maker. At a particular time instant, the agent observes the state, which is the representation of the operating environment, and the rewards from its operating environment which are a consequence of its previous actions, performs learning, decides, and carries out its action. The operating environment can be internal such as instantaneous queue size, or external, such as the usage of the wireless medium. In general, what an SU does affect its operating environment. The SU's action could affect the operating environment (or state) for better or for worse, or maintain the status quo; and this in turn affects the SU's next course of action. As an example, if an SU fails to transmit well in a channel, it switches to another channel with more white spaces or better transmission properties. Its transmission over the white spaces affects the state by reducing the amount of white spaces in that channel. Hence, at any time instant, the agent aims to improve its reward in the next time instant through carrying out a proper action.

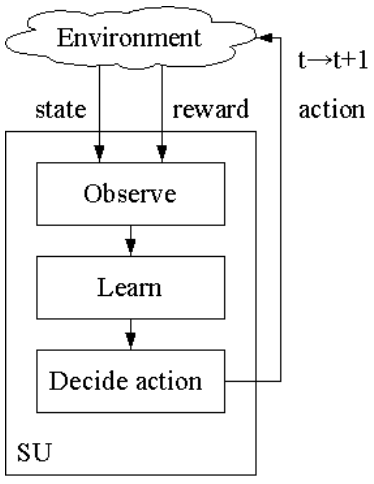


Figure 2.3: A simplified version of CC embedded in each SU.

The most important component in Figure 2.3 is the learning engine that provides knowledge on the operating environment through observing the state and reward. The knowledge or the learning outcome can be shared among the agents in a network by explicit message exchange. As an example, the learning engine could learn the channel conditions such as the PU Utilization Level (PUL) and the Packet Error Rate (PER). Higher levels of PUL in a particular data channel indicate higher levels of PU activity, and hence smaller amount of white spaces. Higher levels of PER indicate higher levels of failed data packet transmission due to uncertain and varying data channel conditions caused by various factors including shadowing, channel selective fading, path loss, PU interference, and others. Various kinds of actions can be carried out by the agent including channel switching, message exchange, backoff, sensing operation and even "cease to act". As time progresses, the agent learns knowledge, which is comprised of the matchings between state, action and reward, in order to carry out the most appropriate action given a particular state.

The representations of the state, reward and action could be optional. For instance, in a single-state or stateless model, the state is not represented and the agent is only adaptative to the rewards.

2.3.2 Two Levels of Cognition Cycle

Two levels of CC are suggested in [24]: *node-level* and *network-level*, as shown in Figure 2.4.

At node-level, each SU runs a CC and makes its own decision in a cooperative or non-cooperative manner. The node-level CC can be used in distributed networks.

Conversely, at network-level, the BS runs a CC and makes its own decision in a multilateral and cooperative manner for the entire network. The network-level CC can be used in centralized networks. An example of the application of a network-level CC is the IEEE 802.22 WRAN [23].

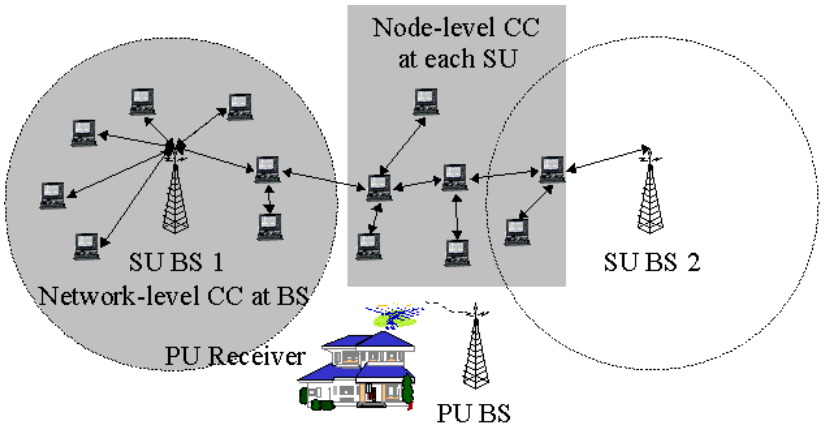


Figure 2.4: Network-level and node-level CC.

In WRAN, each unlicensed Customer-Premises Equipment (CPE) or the SU host is associated with one of the SU BSs. The SU BS coordinates and instructs its CPEs to operate in certain spectrum bands with high quality white spaces for network performance enhancement such as throughput and delay performance.

2.4 Current Trends and Common Assumptions

Cognitive radio is a new research field in wireless communications and networking. At the time this research began, most researches were focusing on the physical layer of the Open System Interconnection (OSI) reference model, and there were little effort to investigate the data link layer. There was not yet a standard available for MAC protocols in static and *distributed* CR networks, and the IEEE 802.22 Working Group was working towards a MAC-PHY air interface standard for *centralized* CR networks. No efforts were made to investigate QoS architecture nor RL as a mechanism to achieve context awareness and intelligence in CR networks. Some of the common assumptions in this research field have been:

- *Static networks* where all the SU hosts are static [25, 26, 27]. This thesis

considers both static and mobile networks.

- *Centralized networks* where each network is comprised of a single SU BS and SU hosts [25, 26]. This thesis considers both centralized and distributed networks.

- *Single collision domain* in distributed CR networks where all the SUs are assumed to be able to hear each other or within communication range of each other [26]. This is a common assumption without which simulation results may be affected by the channel capture effect [28]. The channel capture effect occurs when there is significant unfairness in channel usage. As a consequence, some SUs dominate the usage of the channels with high throughput, while others are starving with low throughput. This thesis considers both single and non-single collision domain while investigating distributed CR networks.

- *Homogeneous channels* where all the available channels across the spectrum bands are assumed to have similar levels of PERs and transmission ranges, though they have different levels of PULs [25, 26, 27]. However, in practice, the SUs are expected to operate over a wide range of non-contiguous frequency bands. *Channel heterogeneity* considers that the properties of the white spaces vary with carrier frequency and time-varying channel condition. In addition, there are many other factors that affect the channel condition such as nodal mobility, neighbour interference and transmission power. Thus, the available white spaces have different levels of PERs and transmission ranges. This thesis considers channel heterogeneity.

- Since the assumption of homogeneous channels is commonplace, the assumption of *identical channel condition, or PER, at all the SUs* is commonplace. This thesis considers both identical and non-identical channel condition at all the SUs. In a scenario with identical channel

conditions, each agent observes a similar level of channel quality for a particular channel. In a scenario with non-identical channel conditions, which is the common case in practice, each agent observes different levels of channel quality for a particular channel.

- *Spectrum pooling* is available at each SU [25, 26, 27]. Spectrum pooling is a new research area in CR networks and this has not been considered in this thesis. Through spectrum pooling, several channels are chosen out of a large pool of candidate channels within a wide range of spectrum bands. Subsequently, each SU chooses one of the chosen channels for data transmission. This thesis adopts this assumption.

Chapter 3

Technology Leverage for Cognitive MAC

This chapter presents technology leverage from multi-channel MAC protocols to cognitive MAC protocols. Firstly, it reviews multi-channel MAC protocols, as well as their merits and demerits. Secondly, it presents cognitive MAC protocols and their functionalities. Thirdly, it presents the operations that multi-channel MAC protocols must deliver and the challenges that must be overcome in order to operate in distributed CR networks or cognitive wireless ad-hoc networks. By providing discussion on possible technology leverage from multi-channel MAC protocols to cognitive MAC protocols, the foundation for further research on the data link layer of the CR networks is established.

3.1 Introduction

For channel access between SUs in a distributed CR network, a cognitive MAC protocol is necessary to coordinate the SUs through channel sensing, selection and access. While research in cognitive MAC is still in its infancy, multi-channel MAC extensions have been realized in IEEE 802.11 to enable all hosts to operate in multiple orthogonal channels simultane-

ously in order to improve network-wide throughput. For instance, IEEE 802.11b/g specifies 3 channels and IEEE 802.11a specifies 12 channels.

Current research in cognitive MAC assumes the availability of a common control channel at all times. This approach has certain hardware requirements that may not be readily available at CR hosts. Hence, other approaches may be necessary.

As shown in Table 3.1, the multi-channel MAC has several functions that can be leveraged by a cognitive MAC due to their similarities in certain aspects, though the CR has an additional requirement to cope with the existence of PUs that have higher authority over the channels. Modifications to existing multi-channel MACs are necessary to cater for the distinguishing features of CR.

Table 3.1: Comparisons of cognitive and multi-channel MAC

Features	Cognitive MAC	Multi-channel MAC
Multi-channel operation	Yes	Yes
Hidden multi-channel problem is addressed	Yes	Yes
Existence of PU	Yes	No

3.2 Chapter Goals

This chapter discusses technology leverage from multi-channel MAC to cognitive MAC to establish a foundation for further research on data link layer protocols for CR networks. It is foreseen that several characteristics of multi-channel MAC have the same effects in distributed CR networks. This chapter addresses the following research questions:

1. What features of multi-channel MACs, as well as their merits and demerits that could be inherited by the cognitive MACs?

2. What are the additional functionalities that multi-channel MACs must offer to qualify as cognitive MACs?

3.3 Overview of Multi-channel MACs

This section focuses on the characteristics that multi-channel and cognitive MAC protocols have in common. Thus, this section assumes non-existence of PUs. To date, a wide range of multi-channel MACs have been designed [29, 30].

3.3.1 Mitigation of Hidden Multi-channel Problem

In general, multi-channel MACs address the channel assignment problem by ensuring that several communication node pairs within two hops of each other avoid choosing the same channel simultaneously for data transmission. This is to mitigate the *hidden multi-channel problem*. In Figure 3.1, ongoing communication from SU1 to SU2 in channel 1 is interrupted when SU3 starts to transmit to SU4 using the same channel 1, resulting in collision at SU2. A cause of this mishap is that SU3 missed the Clear-to-Send (CTS) sent by SU2 because it was engaged in communication with other nodes using another channel. In IEEE 802.11, this problem is overcome by requiring each node to maintain a Channel Usage Table (CUT), which is updated by overhearing Request-to-Send (RTS) and CTS. This keeps track of the channels reserved and utilized, as well as their durations, by an SU's two-hop neighbourhood. The RTS and CTS control messages contain channel reservation information and are sent during a data channel negotiation phase between a communication node pair. A successful negotiation for a data channel is followed by data transmission at the negotiated channel.

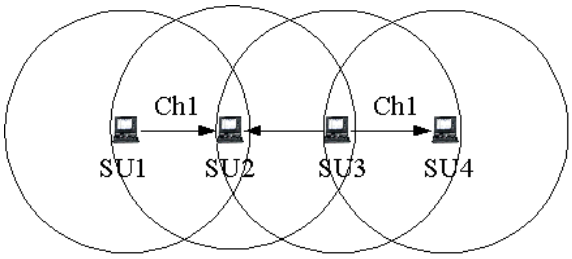


Figure 3.1: Hidden multi-channel problem.

3.3.2 Categories of Multi-channel MACs

Based on the mechanisms of data channel negotiation and channel reservation among a communication node pair, as well as how the CUT is updated at each node, multi-channel MACs are classified into four categories as follows [29, 30]:

- Common Control Channel (CCC)
- Split Phase (SP)
- Common Hopping (CH)
- Default Hopping Sequence (DHS)

Figure 3.2 illustrates the basic operations of different multi-channel MAC protocols. Generally speaking, these MACs are designed to suit nodes with different hardware requirements as shown in Table 3.2. For instance, the CCC approach does not require time synchronization, which is necessary in the other approaches. The next few sections describe the operations, merits and demerits of various multi-channel MACs. A summary of comparison of the multi-channel MACs is available in Table 3.3 at Section 3.3.7.

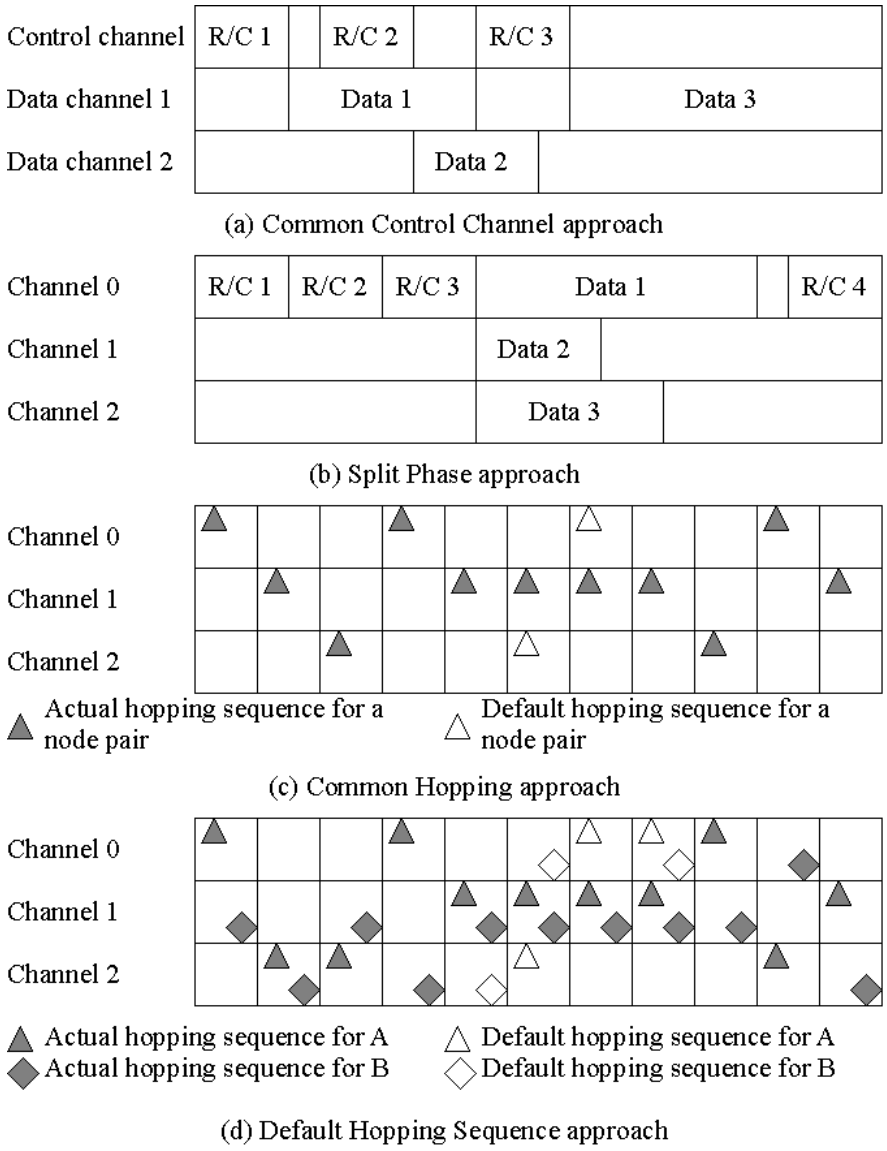


Figure 3.2: Operations of various categories of multi-channel MACs. R/C indicates RTS and CTS control messages handshaking between a communication node pair.

Table 3.2: Hardware requirements for various categories of multi-channel MACs

Hardware requirement	CCC	SP	CH	DHS
Availability of multiple transceivers	Yes	No	No	No
Availability of time synchronization	No	Yes	Yes	Yes
Energy efficiency	No	Yes	No	No

3.3.3 Common Control Channel Approach

3.3.3.1 An Overview of Operation

The common control channel approach applies a single dedicated common control channel for control message exchange and CUT updates at each node. RTS and CTS are sent during data channel negotiation; and ACK is sent after completing a data packet transmission. Data packets are transmitted at any other available data channels, as shown in Figure 3.2(a).

3.3.3.2 Hardware Requirement and Description of Operation

As shown in Table 3.2, the CCC approach does not require time synchronization and does not provide an energy efficient mechanism.

In general, the MAC operation in the CCC approach depends on the number of transceivers at each node. With more than one transceiver, one of them, which is the *control transceiver*, is tuned to the common control channel at all times. Upon successful data channel negotiation, the other transceiver, which is the *data transceiver*, tunes to the negotiated data channel for data transmission. Since the control transceiver is still listening to the common control channel, the node does not miss control messages to update its CUT during data transmission, hence the hidden multi-channel problem is solved. An example of this scheme is [31].

Schemes that use a single transceiver are [32, 33]: during normal operation, the transceiver is tuned to the common control channel; however, if

there is a data packet for transmission, both transmitter and receiver tune to a similar data channel at other frequencies for data transmission, after which both nodes tune back to the common control channel. Therefore, a node may miss several control messages, which lead to an obsolete CUT and so the hidden multi-channel problem arises. In [32], upon completing data transmission and returning to the common control channel, a node waits the duration of Maximum Data Transmission Time (MDTT), which is the maximum time interval for each data packet transmission, before the next data packet transmission starts. The reason for this is that if a particular data channel is busy, it would receive an ACK packet within MDTT for that channel in the common control channel. If no ACK is received, and the data channel is not reserved during the waiting period, the data channel is deemed to be free and available for data packet transmission. In CAM-MAC [33], the transmitter and receiver rely on their idle neighbour nodes to provide channel usage information in a cooperative manner. The idle neighbour nodes, which are listening to the common control channel, have good knowledge of channel usage. Before any data packet transmission, both transmitter and receiver probe their idle neighbour nodes of a selected data channel for its availability. Unless a negative feedback is received from an idle neighbour node, the data channel is deemed to be free.

3.3.3.3 Advantages and Disadvantages

The CCC approach has the disadvantage of saturation in the common control channel [30]. Since all data channel negotiations are conducted at a single common control channel, it is inevitable that congestion can occur, leaving the data channels underutilized as no reservation is made. At the other extreme is when the data channels experience congestion, while the common control channel remains underutilized. Three factors may congest the common control channel, specifically:

- a large number of data channels

- small average data packet size
- a high amount of control overhead

Analysis in [33] shows that a single common control channel is adequate to support a large number of data channels. Substituting the IEEE 802.11 conventional parameters indicates that a single common control channel can support up to 21 data channels even at the very high node density of up to 40 nodes in a single collision domain [33].

Using a single transceiver, the schemes in [32] and CAM-MAC [33] have three disadvantages:

- Lack of support in broadcasting that is important in routing message dissemination such as Route Request (RREQ) and Hello messages.
- Additional delay is incurred while listening for the MDTT interval between consecutive data packet transmissions in [32], as well as extra handshaking and delay in CAM-MAC while probing for channel information at transmitter and receiver sides.
- Assumption of a high density network is applied in CAM-MAC so that there must be idle neighbour nodes with up-to-date channel information.

Based on the discussion above, the CCC approach with multiple transceivers is more suitable for QoS provisioning where delay is critical for time sensitive traffic and broadcasting is necessary.

3.3.4 Split Phase Approach

3.3.4.1 An Overview of Operation

The split phase approach splits all channels into two phases, namely a control phase and a data phase as illustrated in Figure 3.2(b) where control and data packets are sent at different time. During the control phase,

all nodes tune to a common control channel, which is channel 0, for data channel negotiation. Since a communication node pair may not use up a channel for the whole duration of the data phase, multiple communication node pairs may reserve a similar channel. During the data phase, nodes tune to and contend for their negotiated data channels including channel 0 for data packet transmission.

3.3.4.2 Hardware Requirement and Description of Operation

The SP approach uses only one transceiver, and requires time synchronization, as shown in Table 3.2. The SP approach has been applied in energy-efficient MAC protocols such as [34, 35, 36]. In these schemes, the Power Saving Mode (PSM) of IEEE 802.11 standard is used. In PSM, time is divided into beacon intervals, each comprised of an Ad Hoc Traffic Indication Messages (ATIM) window, and a communication window as shown in Figure 3.3. During the ATIM window, all nodes wake up and listen to the common control channel. Nodes with backlogged data packets contend for a channel in the communication window where data packet transmission takes place in the negotiated channel. Nodes that do not engage in communication go back to sleep. Energy efficiency can be further enhanced through adjusting the ATIM window dynamically: with a shorter ATIM window, idle nodes go back to sleep earlier [34, 36].

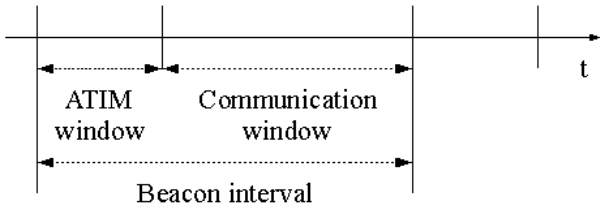


Figure 3.3: Timing in PSM of IEEE 802.11.

3.3.4.3 Advantages and Disadvantages

The SP approach has the advantage of being more energy efficient than other approaches. Since all nodes listen to a common control channel during the control phase, they do not miss control messages, thus this approach does not suffer from the hidden multi-channel problem. It also uses the common control channel for broadcasting purposes. Since the SP approach does not enable concurrent data channel negotiation at different channels, it shares the same problem of the CCC approach that saturation of the common control channel can occur. Three additional disadvantages are:

- Precise time synchronization is required.
- Most channels are wasted during the control phase since all nodes tune to the channel 0 or the common control channel.
- Data channel negotiation can only be performed during the control phase of a beacon interval resulting in longer delay.

To mitigate the second and third disadvantages, control message exchange can be performed at all available channels. Each channel serves as a common control channel for a certain time interval in a sequential and round-robin fashion (called ATIM phase shift mechanism hereafter) so that the control phase is available at all times [37, 34, 35] for data channel negotiation, as shown in Figure 3.4. However, saturation of the control channel remains unsolved.

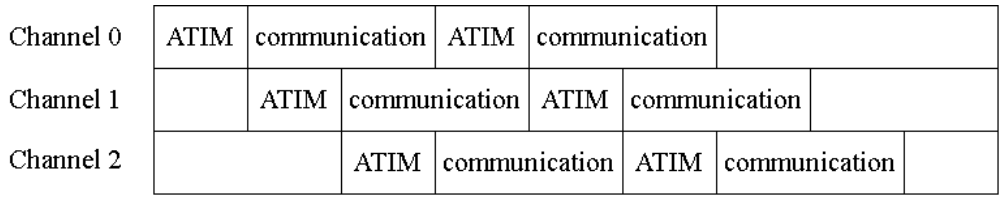


Figure 3.4: ATIM phase shift mechanism in the SP approach.

3.3.5 Common Hopping Approach

3.3.5.1 Hardware Requirement and Description of Operation

The common hopping approach requires every node to hop through all the available channels following a common hopping pattern using only one transceiver, as shown in Figure 3.2(c). Time synchronization is necessary. If a node has data packets to send, it transmits RTS to its receiver that returns a CTS. If the communication node pair agrees on data packet transmission using the channel they are currently in, they stop hopping until data packet transmission completes, while their neighbour nodes continue to hop.

3.3.5.2 Advantages and Disadvantages

In this approach, there is no common control channel and all channels are used for data packet transmission. An advantage is that communication node pairs perform data channel negotiation simultaneously in different channels, hence avoiding saturation in the common control channel. However, there are four disadvantages in this method:

- Hidden multi-channel problems arise when a backlogged node that hops into a new channel may have missed recent RTS/CTS handshaking and starts to transmit RTS.
- Slow channel switches in the current off-the-shelf IEEE 802.11b transceiver that takes about 100-200μs to switch between channels, thus this approach experiences high channel switching delay, and it is highly dependent on hardware performance.
- Lack of support in broadcasting that is important in routing message dissemination such as Route Request (RREQ) and Hello messages.
- Precise time synchronization is required.

3.3.6 Default Hopping Sequence Approach

3.3.6.1 Hardware Requirement and Description of Operation

The DHS and CH approaches have the similar hardware requirements. In the DHS approach, every node determines its default hopping pattern using the seed of a pseudo random generator. The seed, such as the MAC address, is known to a node's neighbour nodes. During normal operation, a node hops and listens to the channel according to its default hopping pattern. If a node wants to send data packets, it determines its receiver node's hopping sequence and hops into its channel accordingly which the receiver is listening to if it is idle. In Figure 3.2(d), node A determines node B's default hopping sequence and hops into its channel for data packet transmission after control message exchange. Both node A and B stop hopping for data packet transmission, after which both of them hop according to their default hopping sequence respectively.

An example of DHS scheme is McMAC [38]. In addition to defining a default hopping sequence, McMAC addresses neighbour node discovery and scheduling. Since there is no common control channel and common hopping pattern, McMAC requires every node to beacon at every channel within a predefined period to enable neighbour node discovery and time synchronization among neighbour nodes. In the scheduling mechanism, a backlogged node transmits its data packet with probability $P_{deviate}$ so that the number of nodes that deviate from their default hopping sequence is controlled if many nodes are backlogged. This helps to balance the number of transmitters and receivers in the network.

3.3.6.2 Advantages and Disadvantages

The DHS and CH approaches have the similar advantages and disadvantages.

3.3.7 Summary on the Merits and Demerits of Multi-channel MACs

A summary of comparison of the merits and demerits of multi-channel MAC protocols is shown in Table 3.3.

Table 3.3: Comparison of various categories of multi-channel MACs

Problem, functions or characteristics	CCC		SP	CH	DHS[a]
	Single transceiver	Multiple transceivers			
Issue on saturation in common control channel	Yes	Yes	Yes	N/A	N/A
Problem on deterioration in hidden multi-channel problem	No	No	No	Yes	Yes
Problem on channel switching delay	No	No	No	Yes	Yes
Support on broadcasting	No	Yes	Yes	No	No

[a]Based on McMAC [38]

3.4 Cognitive MACs and Their Functionalities

This section focuses on the dissimilar characteristics of multi-channel and cognitive MAC protocols, where the presence of PU is a concern.

3.4.1 An Overview of IEEE 802.22

A prominent example of a CR architecture is the IEEE 802.22 WRAN [23], which is currently in the draft process. The IEEE 802.22 adopts a centralized single-hop model and is not suitable for distributed CR networks. In IEEE 802.22, each CPE or SU is associated with an SU BS. To provide a wide coverage, multiple BSs are constructed. Thus, not only does a BS and its CPEs have to detect the presence of incumbent TV or PU signals, but also to coordinate coexistence with overlapping BS and CPEs, or other SUs. IEEE 802.22 is reviewed to provide a list of tasks that a cognitive MAC must provide. In general, the IEEE 802.22 MAC performs three mechanisms:

- Dynamic Spectrum Access (DSA): Access white spaces opportunistically. Detect PU signals across various channels and vacant the channels urgently should PU signals reappear.
- Dynamic Spectrum Sharing (DSS): Coordinate channel sharing among SUs.
- Dynamic spectrum management: Enable data packet transmission across three channels simultaneously through channel bonding.

In DSA and DSS, the BS coordinates the channel sensing procedure among its CPEs in order to detect PU signals in a cooperative manner. This means that the BS determines the channels and times a CPE should sense. Each CPE sends its channel sensing outcome to its BS. With proper channel sensing methodology, the BS has a spectrum occupancy map that covers in-band and out-of-band channels of its entire cell, as well as its neighbouring cells. This sensing method is called distributed sensing. In general, there are two types of measurements: *in-band measurement* measures the channels that the BS and CPEs are using; and *out-of-band measurement* measures the other channels. The in-band measurement is more critical since PU signals must be detected as soon as possible. A two-stage quiet

period mechanism is adopted to perform in-band measurement, as shown in Figure 3.5. The in-band measurement is comprised of *fast-sensing* and *fine-sensing*. Fast-sensing, which uses simple energy-based detection to detect the existence of PU signals, takes approximately 1ms/channel; while fine-sensing, which uses feature-based detection to detect and categorize the signature of PU signals such as wireless microphone, television, and IEEE 802.22 signal, takes approximately 25ms/channel. Fine sensing is carried out if fast sensing detects signals. All BSs, and hence CPEs, are synchronized to perform the in-band measurement simultaneously if they are using the same channels. When all BSs and CPEs keep quiet, any detected signals must be from the PUs. The channel detection time is less than 2s (see Table 3.4) in IEEE 802.22 [23], hence channel sensing must be carried out at least once within this period.

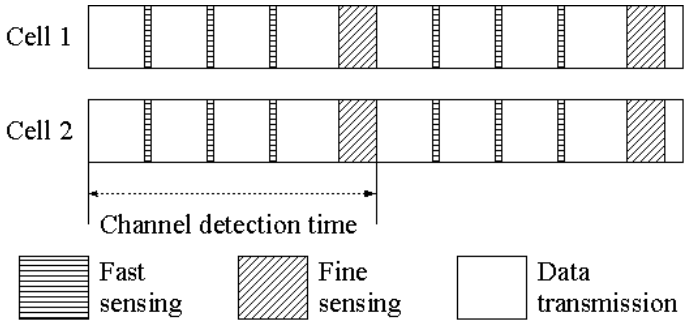


Figure 3.5: Timing for sensing mechanism in IEEE 802.22.

Note that 75ms is required to perform fine sensing on three consecutive channels, which are used simultaneously through channel bonding in dynamic spectrum management. The BS and CPEs switch to backup channels during the fine sensing period. Also, upon detection of PU signal, the BS executes the Incumbent Detection Recovery Protocol (IDRP) [39] that informs its CPEs to use backup channels. The BS and CPEs keep a list of prioritized backup channels that are well maintained through out-of-band measurement. Thus, the BS and CPEs know which channel to switch to when necessary. This means that, even though a CPE misses a beacon re-

lated to channel switching from the BS, IDRP enables a CPE to switch to the most preferred backup channel, providing a gracefully recovery.

Table 3.4 details the Dynamic Frequency Selection (DFS) timing requirements defined in IEEE 802.22 that cover detection, notification and recovery. A BS or CPE must detect the PU signal within CDT for signal strength greater than IDT. During notification and recovery, BS and CPE must cease all transmission within CMT. In addition, CCTT defines the aggregated transmission duration during CMT.

Table 3.4: DFS timing requirements in IEEE 802.22

Parameter	Details	Value for TV broadcasting
Channel Detection Time (CDT)	Time interval that an SU must detect PU signal	$\leq$2s
Channel Move Time (CMT)	Time interval that an SU must vacate channel after detection of PU signal	2s
Channel Closing Transmission Time (CCTT)	Aggregate duration of transmissions during CMT	100ms
Incumbent Detection Threshold (IDT)	PU signal energy above this threshold must be detected	-116dBm (over 6 MHz)

3.4.2 List of Functions for Cognitive MACs

Although IEEE 802.22 is designed for single-hop centralized CR networks, the discussion in the previous section presents an insight into the functions

that a cognitive MAC has to support in distributed CR networks. A list of these CR functions follows:

- *Cooperative sensing*: Cooperative sensing has been proposed to mitigate the effects of fading and shadowing on channel sensing outcomes. Thus, some SUs, called dominant nodes hereafter, have to be elected to perform decision fusion on channel sensing outcomes collected from neighbour SUs or the dominated nodes. Each SU is either a dominant node or a one-hop neighbour to a dominant node and becomes a dominated node. We assume that dominant nodes are elected and readily available in the subsequent discussions in this chapter.

- *Coordination in distributed and cooperative sensing*: Dominant nodes must cooperate with their respective neighbour dominated nodes and with other dominant nodes to perform fast and fine sensing at in-band and out-of-band channels.

- *Notification on PU detection*: Once a PU signal is detected, the MAC should enable a dominated node to inform its dominant node, which performs decision fusion on sensing outcomes. All these functions are done within the timing requirements imposed by the PU, such as the DFS timing requirements in IEEE 802.22.

- *DCS*: Channels are selected for data packet transmission in adaptation to channel availability at an SU.

- *Channel switching*: When a channel is reoccupied by a PU, SU activities have to be switched to a backup channel. The SU transmitter and receiver have to inform each other of channel switching.

- *Compliance with timing requirements*: The SUs have to conform to the CDT, CMT, CCTT and IDT imposed by their PUs to avoid interfering with them.

To the best of our knowledge, none of the existing or proposed cognitive MAC protocols in distributed CR networks perform all of the CR functions.

3.5 Multi-channel MACs in Distributed CR Networks

This section details the operations and challenges that each category of multi-channel MAC protocols has to address in order to operate in distributed CR networks. Multihop data packet transmission is supported in multi-channel MACs such as [32, 37, 35, 33, 36] and these MAC features can be leveraged to facilitate design of cognitive MACs. Generally speaking, cognitive MACs that follow the four categories of multi-channel MAC approaches have the hardware requirements as shown in Table 3.2 and inherit their merits and demerits in Table 3.3. Current research in cognitive MACs assumes the availability of a common control channel at all times [1], and therefore applies the CCC approach. However, without fulfilling the proper hardware requirements, the CCC approach may not be feasible, for instance, lack of multiple transceivers in the CCC approach. In this case, the SP, CH and DHS approaches may be more appropriate. In all the multi-channel MAC approaches, the problems and issues in Table 3.3 should not be neglected.

3.5.1 Common Control Channel Approach

In the CCC approach, the common control channel, which is available at all times, is used as a means of communication for CR functions including cooperative and distributed sensing, notification on PU detection, DCS, and channel switching. The common control channel may be located in one of the following channels:

- Dedicated channel(s) in PU spectrum.
- Dedicated channel(s) in ISM/UNII spectrums.
- Unlicensed Ultra Wide Band (UWB).

As shown in [1], it is infeasible for a CR network to search for a fixed common control channel at PU or licensed spectrum. When PU activity reappears, the SUs must vacate their channel. Thus, a common control channel has to be localized and switchable. In [40, 41], a clustering approach is proposed such that each cluster chooses an available channel for control message exchange so that a global common control channel is not necessary. To countermeasure saturation in the common control channel, effective handshaking has to be designed. Another option is to perform channel bonding at several common control channels. In the data channel, each SU has to sense its channel before transmission.

3.5.2 Split Phase Approach

During the control phase, all SUs tune to a common control channel and perform in a similar way to the CCC approach where the CR functions are performed. Similar to the CCC approach, a common control channel, which has to be localized and switchable, can be located at any of the aforementioned three types of channels in Section 3.5.1. The ATIM phase shift mechanism [37, 34, 35] is difficult to perform in CR networks unless it can be assured that the ATIM window is not overlapping with PU's transmission at all channels. Without the ATIM phase shift mechanism, there are two aforementioned disadvantages as follows:

- Data channels are wasted during the control phase. However, to avoid saturation at the common control channel, the data channels can be used as a common control channel through channel bonding.
- Data channel negotiation can only be performed during the control phase.

In the data channel, each SU has to sense its channel before transmission.

3.5.3 Common Hopping Approach

Whenever an SU newly hops into a channel according to a common hopping pattern in the CH approach, it has to perform channel sensing before any data channel negotiation. Since all SUs are tuned to common channels, message exchange for CR functions is possible. Through distributed sensing coordinated by dominant nodes, advanced sensing of the channel before hopping into the channel is possible. In this case, the common hopping sequence can skip the channels that are already occupied by PUs. Upon detection of PU activity, SUs can inform their dominant node immediately so that channel switching can be performed to hop into the next channel immediately. This means that the duration of an SU in a channel can be dynamic according to the PU activity.

3.5.4 Default Hopping Sequence Approach

Similar to the CH approach, whenever an SU newly hops into a channel, the DHS approach requires each SU to perform channel sensing before any data channel negotiation. Since adjacent SUs may hop into different channels at the same time, cooperative and distributed sensing are difficult among SUs unless they have a common channel for communication, which is not possible using a single transceiver. Suppose PU activity is detected within a channel, in DHS, an SU does not switch to the other channel immediately, but must wait until the next hopping occurrence in order to maintain synchronization among the SUs so that neighbour SUs are able to calculate its hopping pattern accurately, which introduces delay in data packet transmission. Alternatively, a new hopping sequence has to be designed so that an SU skips channels that are already occupied by the PUs, while keeping its neighbour SUs well informed of the channel that it is currently listening to.

3.6 Chapter Summary

This chapter has reviewed various approaches in multi-channel MAC, their merits and demerits. Based on the belief that cognitive MAC protocols for distributed CR networks that apply similar approaches to multi-channel MAC protocols inherit their characteristics, the approach has to be chosen carefully based on its merits, demerits and hardware requirements. The demerit factors remain as open issues in distributed CR networks. Functionalities that a cognitive MAC protocol has to provide, and how these functions can be incorporated into the multichannel MAC protocols are also presented. This chapter has established a foundation for further research in the data link layer of distributed CR networks through the discussion on technology leverage from multi-channel to cognitive MAC protocols. In the coming Chapters 6 to 7, the DCS scheme has been chosen out of the many CR functions as the application under investigation to research into achieving context awareness and intelligence in CR networks.

Chapter 4

C^2net: A Cross-Layer QoS Architecture

This chapter presents a cross-layer QoS architecture called Cross-layer QoS architecture for Cognitive wireless ad hoc NETworks (C^2net), which covers particularly the network and data link layers, based on the Next Steps in Signaling (NSIS) framework [42] from the Internet Engineering Task Force (IETF) [43], as well as its challenges and open issues, as a unified solution for end-to-end QoS provisioning in cognitive wireless ad hoc networks. Firstly, this chapter presents related work on QoS architecture, NSIS framework and several CR regime. The discussion is followed by two novel contributions. Secondly, it presents C^2net. Thirdly, it presents challenges and open issues associated with the cross-layer designs in C^2net posed by the intrinsic complexities of cognitive wireless ad hoc networks to spark new research interests in several unexplored, yet promising areas in this field.

4.1 Introduction

A Cognitive Wireless Ad hoc Networks (CWAN) is a multihop self-organized and dynamic network that applies CR technology for ad hoc

mode wireless communications so that static and mobile nodes within range of each other can communicate in a peer-to-peer and multihop fashion without necessarily involving infrastructure such as a BS. An illustration of CWAN is provided in Figure 2.2 on page 14.

To date, a number of projects have considered the design of QoS architectures for wireless ad hoc networks [44, 45, 46, 47, 48, 49]; but unfortunately none of them can be directly applied to CWAN because CR has an additional requirement to cope with the existence of PUs. A QoS architecture details a framework for the provision of QoS guarantees on an *end-to-end* basis for various traffic types with different priority levels such as video, voice and data. The end-to-end basis means that a source node generates a flow of data packets to its destination node, and it is relayed by intermediate nodes if necessary. Typical QoS parameters that need to be considered include bandwidth, end-to-end delay, packet loss rate and jitter.

In CR networks, research into QoS provisioning in CWAN is still in its infancy and it has been focusing on the following aspects:

- Single-hop static and centralized networks much like the IEEE 802.22 WRAN (see Section 2.4 on page 17 for current research trends). There has been only a perfunctory attempt to provide QoS guarantee based on an end-to-end basis in CWAN [9].

- Physical layer of the OSI reference model. There has been only a perfunctory attempt to improve the data link and network layers [9, 6].

QoS provisioning in CWAN is a daunting challenge for the following reasons:

- The capacity of the wireless channel on which the SUs are operating is apt to change dependent on the PU Utilization Level (PUL). Higher levels of PUL in a particular data channel indicates higher levels of

PU activity, and hence a smaller amount of white space. Specifically, the PUL changes with time.

- The quality of the wireless channel on which the SUs are operating is apt to change dependent on Packet Error Rate (PER). Higher levels of PER indicates higher levels of failed data packet transmission due to uncertain and varying data channel conditions caused by various factors including shadowing, channel selective fading, path loss, PU interference, and others. Specifically, the PER changes with time.

- Nodal mobility. In general, the transmission range using a similar transmission power in wireless channels of different frequencies is different. Specifically, lower channel frequency usually provides larger transmission range.

Ameliorating the effects of low quality of wireless channel and nodal mobility is currently being addressed in traditional wireless ad hoc network solutions. This chapter addresses all the three aforementioned challenges. It presents C^{2}net, which is a cross-layer QoS architecture for CWAN, focusing on the data link and network layers. The main objective of C^{2}net is to provide stable end-to-end QoS assurance to high priority flows, while providing service prioritization to different traffic types. This is realized by a number of distributed features of C^{2}net including topology management, congestion control, scheduling, and DCS.

4.2 Chapter Goals

This chapter discusses C^{2}net to establish a foundation for further research on data link and network layer of CWAN. This chapter addresses the following research questions:

1. What is C^{2}net or the cross-layer QoS architecture for CWAN?

2. What are the cross-layer designs in C²net, their challenges and open issues?

4.3 Related Work

This section reviews QoS architecture, NSIS Framework, and CR regime.

4.3.1 Quality of Service Architecture

Two very early QoS architectures have been proposed for static wired networks, namely Integrated Services (IntServ) [46], and Differentiated Services (DiffServ) [45]. This section reviews some of the key concepts in IntServ and DiffServ architectures, as well as their variants.

4.3.1.1 An Overview of Integrated Services (IntServ) Architecture

The IntServ architecture provides a per-flow granularity in QoS guarantee. This requires every intermediate node of a flow to perform resource reservation and admission control mechanisms. A signaling protocol called Resource Reservation Protocol (RSVP) is used to reserve and maintain resources (or states), such as bandwidth, for each flow at intermediate nodes. The realization of IntServ in wireless networks is questionable because of four disadvantages:

- Scalability concerns as a result of storing state information for each flow at all intermediate nodes.
- The large amount of overhead in RSVP signaling.
- Resource reservation that is difficult to adapt to dynamic topology in wireless ad hoc networks.

- Complex implementation of QoS functions at each intermediate node such as resource reservation (or state information maintenance) and admission control.

4.3.1.2 An Overview of Differentiated Services (DiffServ) Architecture

The DiffServ architecture provides a per-class granularity in QoS guarantee. DiffServ limits complicated QoS functions such as admission control, packet classification and conditioning to the source node. A source node classifies data packets from its various flows according to their QoS requirements based on their respective traffic priority class, marks the DiffServ Codepoint (DSCP) field in the data packet Internet Protocol (IP) header, and conditions the data packets based on a traffic policy. Intermediate nodes that receive the data packet match the DSCP with Per-Hop Behaviour (PHB) and forward the data packet accordingly. The PHB identifies how a data packet should be forwarded according to its priority class. The DiffServ ameliorates the aforementioned four disadvantages of IntServ. However, two disadvantages of DiffServ are:

- Per-class granularity only provides long-term QoS guarantee for each flow.
- There is no QoS signaling to ensure QoS is supported on an end-to-end basis.

4.3.1.3 Variants of QoS Architectures

Based on IntServ and DiffServ frameworks, various QoS architectures for wireless ad hoc networks have been proposed. INSIGNIA [47] adopts the IntServ framework and hence inherits its disadvantages; while SWAN [44] applies the DiffServ model. As DiffServ does not provide end-to-end QoS signaling, a source node in SWAN sends a probing message to its destination node to estimate available resources along its route, such as bottleneck

bandwidth and end-to-end delay. The resource information is required to perform admission control at the source node. FQMM [48] and HQMM [49] apply the hybrid model that embraces both IntServ and DiffServ concepts. The hybrid model provides per-flow granularity to a small amount of high priority flows, while the rest of the flows are treated as per-class granularity. None of these QoS architectures can be adopted in CWAN because of the additional requirement to cope with the existence of PUs.

4.3.2 Next Steps in Signaling Framework

Recently, NSIS framework [42] has been proposed as the end-to-end QoS signaling protocol to supplement the DiffServ model. Using NSIS framework, resource reservation along a route comprised of different QoS models can be made. Hence, the NSIS is particularly suitable for C²net, which is a hybrid QoS model of IntServ and DiffServ.

4.3.2.1 Next Steps in Signaling Framework Components

Architecturally, NSIS is comprised of two components, namely the NSIS Transport Layer Protocol (NTLP) and the NSIS Signaling Layer Protocols (NSLPs) [50]. The NTLP has a messaging component called General Internet Signaling Transport (GIST), which is a successor to RSVP, that uses standard transport layer protocols such as User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Stream Control Transmission Protocol (SCTP), and Datagram Congestion Control Protocol (DCCP) for sending QoS signaling messages. The NSLP provides application-specific functions such as QoS provisioning and security.

4.3.2.2 Quality of Service NSIS Signaling Layer Protocols

This chapter focuses on the QoS NSLP. Four types of signaling messages are defined in QoS NSLP as follows:

- The RESERVE message creates, refreshes, modifies and deletes a flow's resource reservation state information at a node.
- The QUERY message probes available resources along a route, such as bandwidth.
- The RESPONSE message serves as acknowledgment or confirmation of a received QoS NSLP signaling message.
- The NOTIFY message conveys error conditions.

An example of the use of the NSIS signaling scenario for a QUERY message is shown in Figure 4.1. Suppose, node 1 is the source node and node 4 is the destination node. Node 2 and 3 are intermediate nodes in a route that helps to relay data packets of a flow to the destination node. Using its QoS NSLP, node 1 creates a QUERY message, which contains the requested bandwidth information for its flow, to probe bandwidth availability along its route. The GIST encapsulates the QoS NSLP message and transports the signaling message using one of the transport layer protocols until the destination node 4 is reached. Upon receiving the QUERY message, the QoS NSLP of the intermediate node 2 and 3 update their respective available bandwidth in the signaling message. Hence, the key design component of the NSIS framework in a QoS architecture is the QoS NSLP. This component is discussed extensively in Section 4.4.

4.3.3 Cognitive Radio Regime

CR networks can be realized in three different ways [24] as follows:

- *Current regime*. In this regime, SUs are capable of sensing and utilizing white spaces opportunistically at licensed spectrum bands without incurring any cost providing that there is no harmful interference to the PUs. Hence, whenever a PU makes use of its allocated spectrum bands, which has been classified as white space by SUs, the

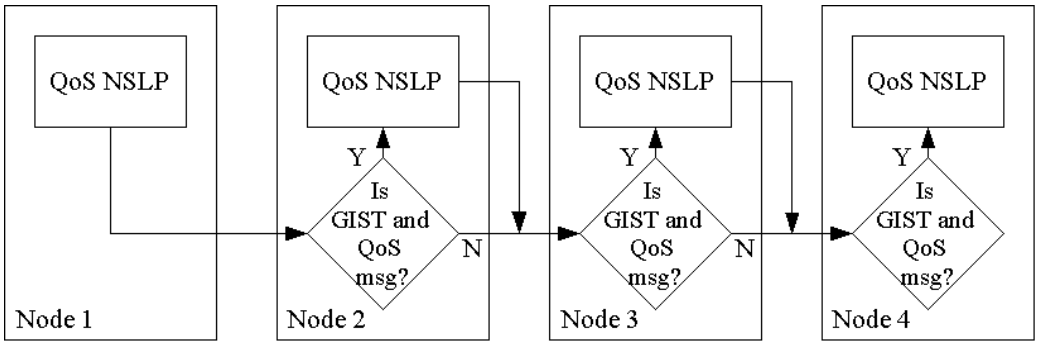


Figure 4.1: NSIS signaling scenario for QUERY message.

SUs must vacate the spectrum bands within the timing requirements imposed by the PU.

- *Common regime*. In this regime, there is equal right for all nodes to spectrum access much like the current unlicensed spectrum bands; hence there is no concept of PU and SU.

- *Market-based regime*. In this regime, spectrum bands is sold as blocks of white spaces by the PU that provides exclusive access to SU purchasers. Hence, the market-based regime provides better a guarantee of white space availability and it is more reliable although it comes at a price. In [24], it is reported that the market-based approach is backed by several prominent regulators such as the FCC, the UK Office for Communication (Ofcom), and the EU Commission Radio Spectrum Policy Group.

4.4 C²net: A Cross-layer QoS Architecture

4.4.1 Quality of Service Model and Cognitive Radio Regime

C²net is a hybrid model of IntServ and DiffServ. In this architecture, a small number of high priority flows, such as voice and video, adopt the IntServ model; while the other flows adopt the DiffServ model. From an economic point of view, consumers prefer to send best-effort flows at the lowest possible price; while high priority flows may incur some charges with occasional packet loss being acceptable as long as the perceived quality is not significantly degraded. Thus, the DiffServ model applies the current regime, while IntServ applies the market-based regime. In the market-based regime, SUs have exclusive access to white spaces in a deterministic manner; hence, the small number of high priority flows achieve better QoS guarantee.

4.4.2 Common Control Channel Approach

The common control channel approach (see Section 3.3.3 on page 26) is adopted. There are two types of channels, namely, the *common control channel* and *data channels*. Both the common control channel and data channels are located in the licensed or unlicensed spectrum bands. Each SU is equipped with two transceivers: the *control transceiver* is tuned to a common control channel at all times; while the *data transceiver* is tuned to one of the data channels for data packet transmission. During normal operation, all SUs are constantly listening to the common control channel. The common control channel is meant for control message exchanges, such as data channel negotiation messages and notification to vacate a data channel upon detection of PU activity. During data channel negotiation, the SU transmitter and SU receiver choose a data channel among all the available data channels for data transmission, after which the data transceiver

is tuned to the negotiated data channel. The SUs constantly explore the data channels in search of high quality white spaces.

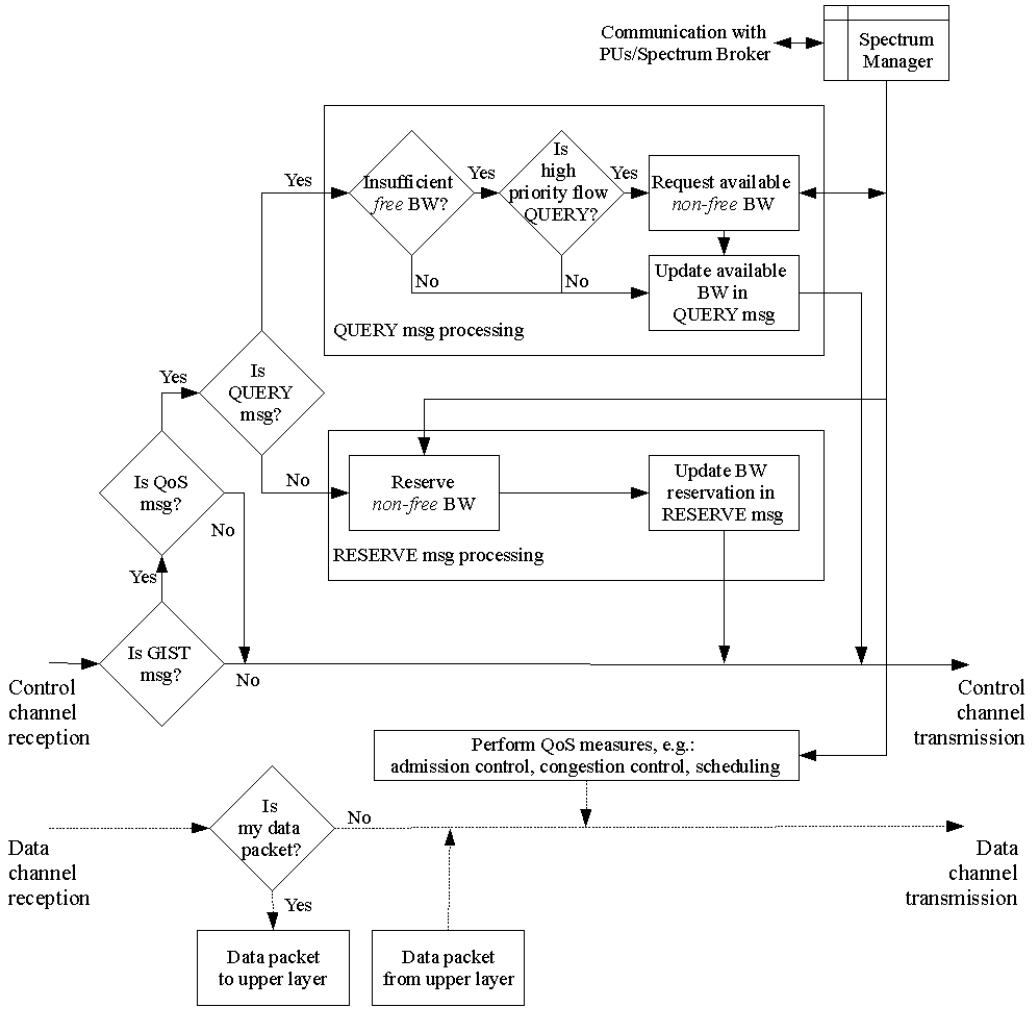


Figure 4.2: Flowchart for QoS elements at each SU in C²net architecture. Solid line indicates control flow; while dotted line indicates data flow.

4.4.3 Quality of Service NSIS Signaling Layer Protocols Operation in the Common Control Channel

The QoS NSLP is the key component of the NSIS framework for QoS provisioning. The flowchart for QoS NSLP in C²net at each intermediate SU node is shown in Figure 4.2, in conjunction with other QoS elements. Procedures at the control channel are related to QoS NSLP, while procedures at the data channel are for all data packets. For brevity, RESPONSE and NOTIFY messages are not shown. Before any QoS signaling is performed, the routing protocol is assumed to have found several routes from the source node to the destination node. Upon receiving control messages on the common control channel, the GIST messages that carry QoS information are processed in the QoS NSLP. The QUERY message processing unit checks for available bandwidth at the SU. Two types of data channels are the *free* unlicensed and licensed channels, as well as *non-free* licensed channels. If the channel availability of the free channels is insufficient and the flow has a high priority level, the SU requests bandwidth from non-free licensed channels through its spectrum manager using a market-based regime. The spectrum manager at the SU determines the amount of bandwidth to be later purchased during the resource reservation process; and communicates with the PUs or a spectrum broker to know about the available bandwidth that could be purchased through spectrum trading. Available bandwidth is updated in the QUERY message, which is then sent to the next hop that implements a similar procedure using the common control channel. The QUERY message is also used for state refreshment, modification and deletion at an SU. For simplicity, only state creation is shown.

In Figure 4.2, the RESERVE message processing unit is implemented for high priority flows only. In this process, the spectrum manager at each SU is requested to purchase and reserve the required white spaces for high priority flows. A description of spectrum trading is proposed by

Buddhikot et al [51].

Whether the reservation is successful is indicated in the RESERVE message which is transmitted from the destination node to its sender node. The state is reserved in a soft manner such that if the QUERY message or data packet from a flow is not received after a certain time interval, the state is withdrawn. In this case, the spectrum manager stops the purchase of white spaces for the flow.

4.4.4 Quality of Service Measures in the Data Channels

On the data channel, QoS measures such as admission control, packet classification, packet marking, rate control, packet shaping and dropping are performed to ensure that the rate and burst profile for each flow is compliant with the Traffic Conditioning Agreement (TCA) as stipulated in the Service Level Agreement (SLA). The purpose is to ensure that the QoS of the high priority flows are not jeopardized. A detailed description of the implementation of the QoS measures is given by Blake [45]. Additionally, if the spectrum manager has reserved white spaces for a high priority flow, its data packets will be forwarded using the reserved resources.

The NSIS framework provides end-to-end QoS signaling and QoS NSLP for QoS provisioning in C²net. However, there are various other factors that affect the end-to-end QoS provisioning at the data link and network layers. A cross-layer approach is adopted to address the issues.

4.5 The Cross-Layer Paradigm

The cross-layer paradigm [52] has overcome the traditional layered approach through joint design of multiple components at various layers of the OSI reference model.

An important question is: "*Why is the cross-layer paradigm potentially important in CWAN?*" In CWAN, an SU has to be aware of its operating envi-

ronment. The DCS scheme, which resides in the data link layer, must sense for white spaces across various data channels and choose a data channel dynamically for data transmission. To enable the functions at the upper layer to be aware of their operating environment, functions such as topology management and congestion control in the network and transport layer respectively must cooperate with the DCS in the lower layer.

Three cross-layer designs are shown in Figure 4.3. *Joint DCS and topology management* in interaction 1 performs channel selection in the presence of dynamics within the new topology, channel condition and PU activity. *Joint DCS and congestion control* in interaction 2 ameliorates local congestion in the CR context. *Joint scheduling and channel condition measurement* decides the next data packet for transmission in interaction 3.

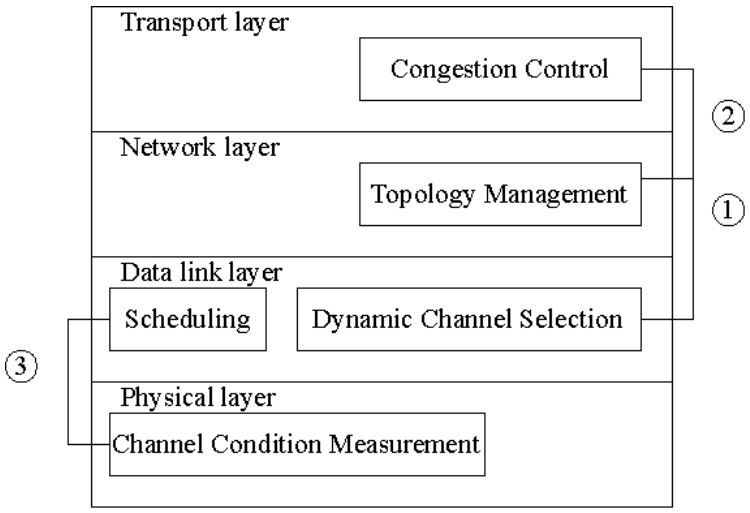


Figure 4.3: Cross-layer framework in C^2net

At the time this chapter was written, little or no effort has been made by the research community to investigate these joint designs in CWAN. The next few subsections discuss these joint designs, their challenges and open issues. In Chapter 8, the RL approach is applied to implement these designs.

4.5.1 Joint Dynamic Channel Selection and Topology Management

4.5.1.1 Objectives

The joint DCS and topology management provides the best strategy to select an available channel among the licensed channels for data transmission from an SU with the objective to reduce the data packet loss of high priority flows for stable end-to-end QoS provisioning, as well as maximizing overall throughput, in the presence of nodal mobility.

4.5.1.2 Descriptions of Operation

For stable, reliable and robust transmissions, some SUs in a neighbourhood that are relatively stable, in terms of mobility characteristics, are selected to form a Dominating Set (DS). Nodal stability is determined using Link Expiration Time (LET), associativity in Hello messages, or both. An SU is considered relatively stable if it is capable of serving as a DS node for the longest time interval compared to its one-hop neighbour SUs. The DS nodes connect among themselves to form a backbone topology, which is connected to the SU BS, while non-DS nodes establish links with DS nodes. Other possible considerations that are relevant to stability, reliability and robustness in DS node selection are energy levels at the SU, signal-to-noise ratio in various channels and so on.

Various clustering algorithms in wireless ad hoc networks utilize the DS concept in order to improve network scalability through reduction of routing overhead [53]. As an added advantage for CWAN, the DS nodes provide a means of coordination for distributed and cooperative sensing (see Section 3.4) in order to mitigate the effects of unreliable channel sensing outcomes without imposing higher sensitivity requirements at each SU. A DS node performs decision fusion on channel sensing outcomes from its neighbour SUs to improve sensing accuracies. The decision fusion is a decision making process where local channel sensing outcomes

at neighbour SUs are combined to reach a more accurate result. In [40], a clustering scheme for CR networks is proposed so that each cluster chooses an available channel for control message exchange, rather than choosing a global common control channel; however, no investigation has been done on data transmission in the clustering scheme.

The licensed data channels have different levels of PULs and PERs. In C^2net, the DCS is performed based on nodal stability, reliability and robustness, as well as backbone connectivity, PUL and PER in each available data channel. The DS nodes, which form a connected backbone, are relatively stable, reliable and robust, and they have higher authority in data channel selection so that data channels with lower PUL and PER are chosen. Non-DS nodes choose the remaining available data channels. In view of the dynamic nature of the network, the backbone and channels must be maintained continuously.

Traditionally, the backbone topology throughout a wireless ad hoc network is formed using the Minimum Dominating Set (MDS) [53] to reduce the number of DS nodes in order to reduce the amount of routing overheads in the entire networks. The routing overheads, such as route request and route reply, are broadcast by nodes to establish and maintain routes throughout the networks. In MDS, only DS nodes are allowed to broadcast the routing overheads. Thus, with reduced number of DS nodes, the routing overheads are reduced. In C^2net, the main purpose is to provide stable data transmission for high priority flows. It forms a Connected Dominating Set (CDS) instead of an MDS. The CDS ensures the connectivity of the DS nodes in the backbone topology. It should be noted that the type of information carried, which is routing overheads in MDS and data packets in CDS, differentiates the backbone functionalities in C^2net from that of traditional schemes. Ensuring connectivity in the backbone topology helps to alleviate congestion and packet loss since the DS nodes have higher authority to select data channels with lower PUL and PER.

4.5.1.3 An Example of the Operation

Consider the snapshot of a dynamic topology in Figure 4.4. Suppose, based on nodal stability, SU1, SU3 and SU4 are relatively stable and become DS nodes. Since SUs are either DS nodes, or direct neighbour to a DS node, it is a valid MDS. However, there is no connectivity between the DS nodes, hence it is a broken backbone topology. As SU2 does not have the higher authority to select data channels with lower PUL and PER for data transmission, it becomes a bottleneck and congestion occurs. Thus, SU2 is chosen as a DS node although it does not have higher stability than SU4. In this case, the DS nodes are connected, and hence form a valid CDS. The connectivity of the backbone topology (SU1-SU2-SU3) is thus maintained. In short, the most stable SU node within a subset to fulfill the connectivity requirement is chosen to become the DS node in backbone topology maintenance.

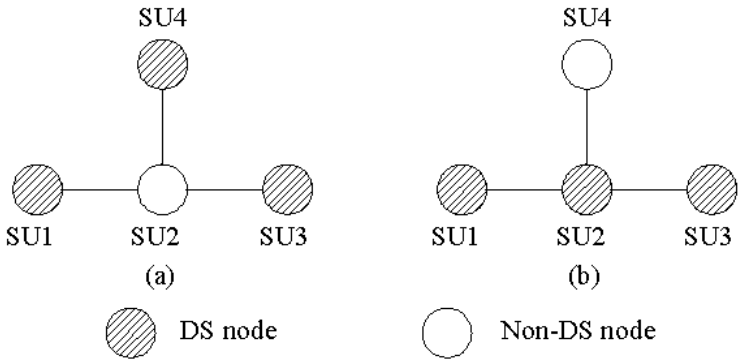


Figure 4.4: A snapshot of mobile SUs. In (a), DS nodes are disconnected, while in (b) they are connected. A solid line indicates a link between SUs.

4.5.1.4 Challenges and Open Issues

The challenges and open issues in this joint design are:

- DCS. The DS nodes must select data channels with lower PUL and PER for data packet transmission.

- Backbone topology construction and maintenance. The DS nodes must form a CDS topology. During DS node selection, the nodes must negotiate among themselves in a distributed manner.

4.5.2 Joint Dynamic Channel Selection and Congestion Control

4.5.2.1 Objectives and Descriptions of Operation

In a single-channel environment, if a node experiences congestion, its neighbour nodes also experience the same congestion. This is not the case in CWAN where multiple data channels exist. Each data channel has different levels of PULs and PERs. Without load-balancing among the data channels, an SU may experience congestion, while its neighbour SUs have more than ample bandwidth. This is a condition that we called channel selective congestion.

The objective is to allocate the available channels according to the traffic load at each SU. In other words, a channel with lower PUL and PER is allocated to an SU with higher traffic load, and vice-versa. This joint design provides load balancing among the channels as a solution to congestion avoidance. An advantage is that congestion can be solved locally at the data link layer, rather than at the transport layer. An SU is able to adapt to the congestion level at various channels.

4.5.2.2 Challenges and Open Issues

The challenges and open issues in this joint design are:

- DCS. Based on their traffic loads, the SUs select their respective data channel with certain levels of PUL and PER for data packet transmission.

- Transport layer monitoring. The source SU of a traffic flow must be put in a wait state or adjust its transmission rate for a certain dura-

tion that depends on how long does it take for the intermediate SU nodes to perform the congestion control mechanism at the data link layer.

4.5.3 Joint Scheduling and Channel Condition Measurement

4.5.3.1 Objectives

Joint scheduling and channel condition measurement provides the best strategy to select the next SU or hop among the neighbour SUs for data transmission from an SU with the objective of reducing the data packet loss of high priority flows for stable end-to-end QoS provisioning, as well as maximizing overall throughput.

4.5.3.2 Descriptions of Operation

The selection of the next SU or hop for data packet transmission by an SU at any time instance is an important event that affects network performance significantly. Consider SU0 with two neighbour SUs and a scheduler with several class-based queues in Figure 4.5. For simplicity, only the highest priority queue is shown. Upon data channel negotiation, SU0 sends to SU1 in channel 1; and to SU2 in channel 2. Each high-priority data packet has a deadline. Suppose earliest deadline first scheduling is applied within the high-priority queue. The Head of Queue (HoQ) data packet is to be sent to SU1. However, SU1 is engaged in communication with another SU, or the channel condition of the link with SU1 is Bad due to high PUL or PER, which leads to several data packet retransmissions. This scenario happens because SUs using different data channels have different levels of contention, PUL and PER levels. The HoQ data packet blocks the next data packet in the queue to be sent to SU2. Eventually, due to expiry of time sensitive data packets, the first and second data packets

are dropped. Extending the simple scenario in Figure 4.5 to a number of class-based queues, say eight, will lead to a complex scheduling design.

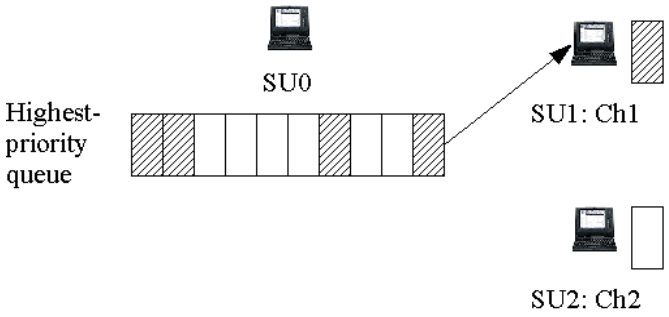


Figure 4.5: HoQ blocking.

Denote channel states by $S_{ch} = \{Good, Bad\}$. The *Good* state indicates data packets will be sent successfully to its neighbour SU; while a *Bad* state indicates failure to do so. Consider the highest priority queue in SU0 with two neighbour SUs in Figure 4.6. Note that the scheduling algorithm is scalable to a large number of priority queues and neighbour SUs. A virtual collision handler determines the next hop for data packet transmission. Only the first data packet for each neighbour SU participates for contention in the virtual collision handler. Suppose the data packet for SU2 wins in the virtual collision handler. The bandwidth request module in Figure 4.6 informs the SU to reserve a sufficient amount of bandwidth at the receiver's data channel for the next hop data transmission. The virtual collision handler is an important component in the scheduler that affects the network performance significantly. Unlike the virtual collision handler in IEEE 802.11e that merely compares the priority level of data packets when more than one data packet complete their respective backoff at the same time [54], it has to consider the deadline of high-priority data packets, the S_{ch} for each neighbour SU, and the PUL and PER of the data channels in order to compute a contention metric to determine the next hop to maximize successful data packet transmission in the shortest time possible.

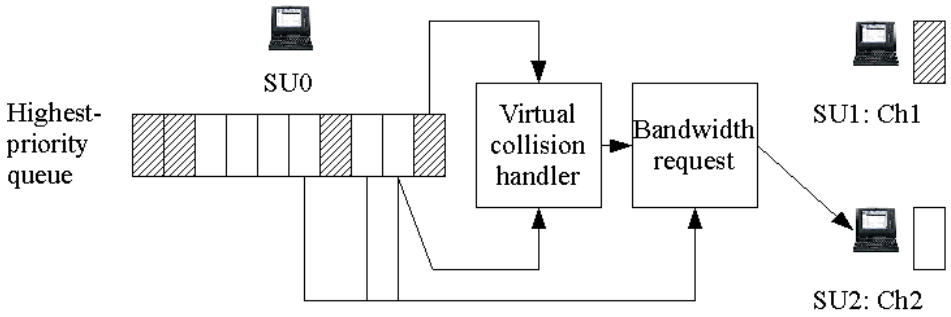


Figure 4.6: Improved scheduling with virtual collision handler.

Figure 4.7 shows the inputs and output of a virtual collision handler. The transmission history component keeps track of successful or unsuccessful data packet transmission to each next hop. The RTS/CTS reservation table keeps track of RTS/CTS information for each available data channel. The link channel table keeps track of the PUL and PER of each data channel to categorize the channels into the *Good* or *Bad* state. A data packet for each next hop has its deadline information extracted into the handler. The virtual collision handler determines the successful next hop.

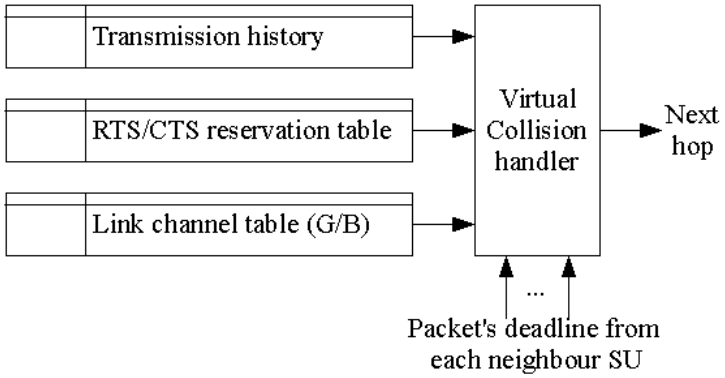


Figure 4.7: A virtual collision handler with its inputs and output.

4.5.3.3 Challenges and Open Issues

The challenges and open issues in this joint design are:

- Virtual collision handler. The SUs determine the winning next hop for data packet transmission, as well as obtaining high quality information from the transmission history, RTS/CTS reservation table and link channel table. Fairness must be achieved among the SUs with high priority data packets.

- State determination. The SUs must infer the *Good* and *Bad* states accurately.

4.5.4 Other Research Challenges

Other research challenges and open issues are:

- *Congestion measurement*. The definition of congestion in a CR context, its metrics, and the mechanism to measure congestion need further clarification.

- *High priority data packet transmission*. To ensure the high priority data packets in all SU queues are sent first, as well as not starving the best-effort data packets, not only does an SU have to ensure the high priority data packets in its queue are sent before best-effort data packets, but also estimate the number of high priority data packets in neighbouring SUs. In the traditional single channel environment, a node can monitor this by reading the DSCP in each data packet it overhears so that it defers its best-effort data packet transmission if it hears higher priority data packets are sent among the neighbour nodes. However, this may not be feasible in a multi-channel environment. Hence, it may be necessary for the SUs to announce the number of high priority data packets in their queues through explicit message exchange.

- *Transport layer protocols*. Traditional transport layer protocols, TCP and UDP, are now being augmented with SCTP and DCCP. None

of these have been designed with CR in mind. However, local congestion control has to be designed to cooperate with the end-to-end congestion control mechanism readily available in both SCTP and DCCP.

4.6 Chapter Summary

A cross-layer QoS architecture called C²net has been proposed for CWAN, which is a multihop self-organized and dynamic CR network. The main objective of C²net is to provide and maintain a stable QoS for high priority flows throughout their connections. C²net is a hybrid model of IntServ and DiffServ that adopts the NSIS framework. The core component for QoS provisioning in the NSIS framework is the QoS NSLP that enables end-to-end QoS signaling protocol for the QoS model embedded in each SU. The IntServ model, which adopts the market-based regime, fulfills the stringent QoS requirements of a flow at reasonable cost by purchasing white spaces from PU if necessary. The DiffServ model, which adopts the current regime, provides services to lower priority flows. Various cross-layer designs as well as their open issues and challenges are discussed. The cross-layer designs are joint DCS and topology management, joint DCS and congestion control, and joint scheduling and channel condition measurement.

Chapter 5

Reinforcement Learning Approach

In wireless networks, context awareness and intelligence are the capabilities that enable each node to observe, learn, and respond to its complex and dynamic operating environment in an efficient manner for network-wide performance enhancement (see Chapter 1, page 2 for a more complete definition). The cognition cycle portrays the notion of context awareness and intelligence in CR networks. This chapter presents reinforcement learning as an approach to achieve context awareness and intelligence in wireless networks including CR networks. The traditional RL approach can be improved to embrace new features that are applicable to wireless networks in order to enhance network-wide performance. The discussion covers the motivation behind this approach, a discussion of the traditional approach, including the important features such as state, action, reward, exploration and exploitation. The chapter then focuses on new features not used in the traditional approach including events, rules and the effects of actions to the operating environment. Finally, this chapter provides a discussion on achieving context awareness and intelligence in CR networks.

5.1 Introduction

5.1.1 Traditional Policy-based Approach

Traditionally, without the application of intelligence, each wireless host applies a policy-based approach and adheres to a strict and static predefined set of policies that is hardcoded, and responds accordingly. A common policy is defined through for example if-then-else conditional statement (see Figure 5.1) or expressed as a state-event-action table. When a node encounters a particular condition (or state) and an event in the operating environment, it performs a corresponding action. A condition such as queue size, is monitored at all times; while an event, such as a call handoff, happens occasionally and it is detected whenever it occurs. A prominent example that applies the policy-based approach is the backoff mechanism in various MAC protocols. The average backoff period is typically doubled on each successive transmission attempt due to failed transmission for a particular data packet. A node determines its backoff period without considering its operating environment such as the number of neighbor nodes and the channel quality.

```
if      (state S1, event E1) then (action A1);
elseif (state S2, event E2) then (action A2);
elseif (state S3, event E3) then (action A3);
...
else    (state Sn, event En) then (action An);
end if;
```

Figure 5.1: The *if-then-else* predefined policy.

5.1.2 Disadvantages of Policy-based Approach

The policy-based approach has a major drawback in that the actions are hardcoded and cannot be changed "on the fly". Specifically, the relation-

ships between the states, events and actions are static.

The wireless communication environment is a complex and dynamic system. For instance, the radio spectrum resources, network topology and nodal availability are uncertain and dynamic factors that affect network performance in a complex manner. Hence, a policy-based system may not be able to cater for all possible states and events encountered throughout its operation, resulting in suboptimal network performance.

5.1.3 Necessity of Intelligence

The drawbacks of the traditional policy-based approach can be overcome by incorporating intelligence into the system. Intelligence enables each node to learn new states, events and actions, as well as matching them so that optimal actions can be approximated and taken. In other words, the policy in Figure 5.1 evolves with time through learning on the fly to achieve an approximation of optimal policy most of the time.

5.1.4 Necessity of Continuous Learning

Continuous learning is necessary so that the policy remains optimal or close to optimal with respect to the ever dynamic operating environment. Specifically, there are three main reasons for continuous learning:

- The operating environment evolves with time such that new state-event pairs may be encountered, and new actions may be discovered, hence the policy must be constantly updated to match the state and event pairs with the optimal or near-optimal actions.

- Network performance brought about by an action may deteriorate with respect to a state-event pair as time goes by, and so rematching may be necessary.

- Most operating environments in wireless networks are dynamic in

nature, e.g. traffic load may follow Poisson process; hence, it may take many trials to learn an efficient policy.

5.1.5 The Reinforcement Learning Approach

In this chapter, we advocate the use of Reinforcement Learning (RL) [4] to achieve context awareness and intelligence. The RL approach is an unsupervised and online machine learning technique that improves network performance using simple modeling (see Chapter 1 on page 4 for more explanation). Instead of tackling every single factor that affects network performance, RL models the network performance, such as throughput, that covers a wide range of factors that can affect the network performance, hence its simple modeling approach. However, more complex implementations of RL are possible to tackle complicated applications. As an example, a RL approach called REINFORCE [55] uses Gaussian distribution to determine its actions.

The RL approach has been applied in a variety of applications such as routing [56] and resource management [57] in wireless networks such as Mobile Ad hoc Networks (MANETs), and recently in CR networks [58, 59, 60, 61, 62, 63, 16, 11, 12, 14, 17, 15].

5.2 Chapter Goal

The chapter discusses RL and addresses the following research questions:

1. What is the appropriate generic RL model to achieve context awareness and intelligence in CR networks?

2. What are the traditional and new features in the RL model to achieve context awareness and intelligence in CR networks?

5.3 Reinforcement Learning

5.3.1 Description of Operation

Q-learning [4] is an on-line algorithm in RL that approximates an optimal policy using only simple modeling. We model each node in the network as a learning agent as shown in Figure 5.2, which is very similar to Figure 2.3 on page 15. Note the additional new feature of "event" and the term "agent" in Figure 5.2.

Section 2.3.1 on page 15 provides a detailed description of the model shown in Figure 5.2. This section provides a brief description of the new feature called "event". The state and event are differentiated in that the *state* is monitored at all times, whereas the *event* happens occasionally and in general is detected whenever it occurs.

At any time instant, the agent carries out a proper action so that the reward should improve in the next time instant. As time progresses, the agent learns to carry out proper actions given a particular state-event pair.

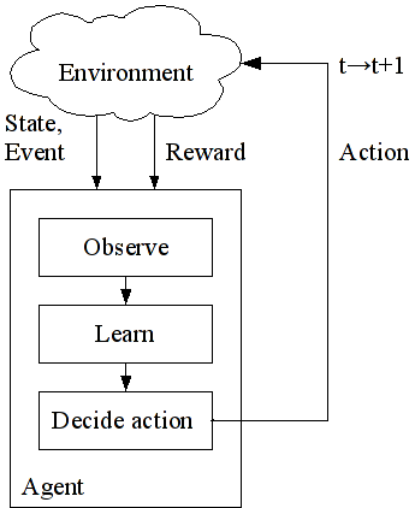


Figure 5.2: Abstract view of an RL agent in its environment.

5.3.2 Q-value Function

In Q-learning, the *learnt action value* or *Q-value*, *Q*(*state*, *event*, *action*) is updated using immediate reward and discounted reward, and maintained in a two-dimensional lookup Q-table with size $|(state, event)|\times|action|$, with $|arg|$ representing the cardinality of *arg*. The immediate reward is the reward received at time t+1 for an action taken at the previous time instant t. For each state-event pair, an appropriate action is rewarded and its Q-value is increased. In contrast, an inappropriate action is punished and the Q-value is decreased. Hence, the Q-value indicates the appropriateness of the selection of an action in a state-event pair. At any time instant, the agent chooses an action with the maximum Q-value so that it receives an optimal or near-optimal reward that enhances its network performance such as throughput. The future return is the estimated discounted rewards it receives in the future. The discounted reward is the estimate of the present value of the expected rewards to be received in the future. The estimation is computed through discounting the expected rewards to the present value.

Denote state by s, event by e, action by a, action set by A, reward by r, learning rate by α and discount factor by γ. The reward can be represented as cost if it is desired to be minimized. At time t+1, the Q-value of a chosen action in a state-event pair at time t is updated as follows:

$$\begin{aligned} Q_{t+1}(s_t, e_t, a_t) &\leftarrow (1-\alpha)Q_t(s_t, e_t, a_t) \\ &+ \alpha(r_{t+1}(s_{t+1}, e_{t+1}) + \gamma \max_{a\in A} Q_t(s_{t+1}, e_{t+1}, a)) \end{aligned} \tag{5.1}$$

where $0\leqslant\alpha\leqslant1$ and $0\leqslant\gamma\leqslant1$. If α = 1, the agent will forget all its previous learnt utilities, giving a single-shot network behaviour. The higher the value of γ, the greater the agent relies on the future return, which is the maximum Q-value in state-event pair at the next time instant. Unless γ=1 where the discounted and immediate rewards share the same weight, the discounted reward always has lower weight compared to the immediate reward.

Changes in the Q-value will lead to changes in agent action. RL searches for an approximation of optimal policy that maximizes its accumulated reward through choosing the action with maximum Q-value. As an example of the usage of discounted reward (or cost in this case), the immediate cost represents the time delay introduced by an upstream node, the discounted cost represents the amount of end-to-end delay from an upstream node (action) to a destination node (state) in a multi-hop routing scheme [56]. The agent chooses an upstream node such that the state-action pair at the upstream node provides the least cost based on (5.1).

5.3.3 Flowchart of the RL Model

Figure 5.3 shows the flowchart of the RL model. At time t, an agent chooses a subset of actions in adherence to a set of rules that exclude actions that violate the network requirements. Next, it chooses an exploitation action, which is the best known action derived from its Q-table, or an exploration action, which is a random action designed to increase knowledge of the operating environment. At the next time instant t+1, it observes the consequences of its previous action including the state, event, and reward; and updates its Q-table and rules accordingly. Further explanation is given in the next few sections. In general, to apply RL, the following representations are necessary: state, event, action and reward; and rules. The representations could be optional, for instance, if the state is not represented, it is called a single-state or stateless model.

5.3.4 Space Representation

All the elements in the operating environment within which a wireless node resides may not be important unless network performance can be improved by addressing them. The state, event, action and reward spaces incorporate the important decision-making factors of a design application. The state characterizes the environmental factors that require

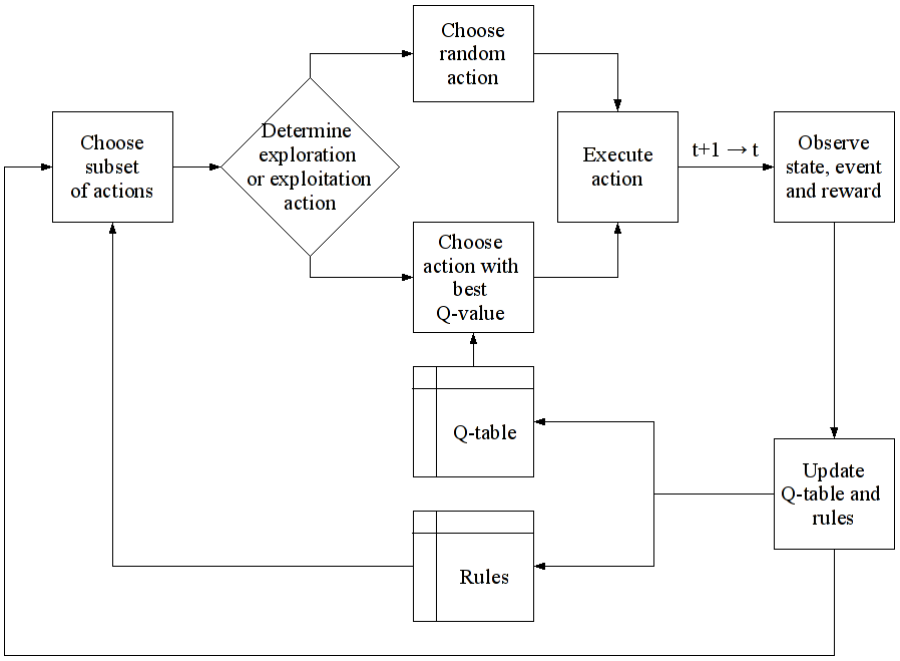


Figure 5.3: Flowchart of the RL model.

constant monitoring; while the event represents the occurrence of events of particular interest that may happen occasionally in the environment.

The variables for the state, event, action and reward can be discrete or continuous. For discrete space, it can be an interval of values segregated into smaller ranges representing different stages or levels in the system, or a counter to keep track of the number of occurrences or simply a boolean representing an occurrence. In a complex scenario, the space can be too large to be stored in memory, therefore to reduce the number of states, two states or events that are close to each other can be merged if the Hamming distance between them is less than a threshold value [4]. The Hamming distance computes the difference between two states such as the values or the number of bits at which two states differ. In some cases, such as the bandwidth provisioning problem, the state, event, action and reward are more appropriately represented as a continuous space. As an example, a RL approach called REINFORCE that uses the Gaussian distribution is

used to generate real-valued actions using the mean and variance of the state, which is updated using reward [55]. However, in Q-learning, it is not possible to represent continuous space in a tabular format. Future research could be pursued for effective approximation-based techniques to achieve continuous space representation.

5.3.5 Exploration and Exploitation

The update of the Q-value in Equation (5.1) does not cater for the actions that are never chosen [4]. Two types of action selections are

- *Exploitation* chooses the best known action, or the greedy action at all times.
- *Exploration* chooses non-optimal actions once in a while in order to improve the estimates of all the Q-values in the Q-table in order that better actions may be discovered.

The balance between exploitation and exploration depends on the accuracy of the Q-value estimation and level of dynamic behaviour in the operating environment. An example of tradeoff methodology is ε-greedy approach [4]. In the ε-greedy approach, an agent chooses the greedy action as its next action with probability 1-ε, and random action with a small probability ε.

5.3.6 Rules

Q-learning must achieve a high level of reward without violating the constraints or rules, which could be imposed by a user requirement or policy. The (*state*,*event*,*action*) entries that violate the rules are marked. Whenever a state and event pair is encountered, the actions that violate the rules are prohibited during exploitation or even exploration.

Three examples of the applications of rules are as follows:

- Several QoS parameters, such as end-to-end delay and packet dropping probability, have to be fulfilled.

- Requirements imposed by the PU, such as the DFS timing requirement (see Table 3.4 on page 36).

- Statistical information is required to perform entry elimination in [57]. The entry elimination identifies and subsequently refrains from executing illegitimate actions. In this case, the agent keeps track of two counters, $C^{M}_{(s,e,a)}$ and $C^{V}_{(s,e,a)}$. The $C^{M}_{(s,e,a)}$ counts the number of times (*state*,*event*,*action*) is found to violate the rules; while $C^{V}_{(s,e,a)}$ counts the number of times the (*state*,*event*,*action*) is visited. An action becomes illegitimate when the ratio of $C^{M}_{(s,e,a)}$ to $C^{V}_{(s,e,a)}$ is greater than a threshold value $T_{(s,e,a)}$.

The definition of the rules is dependent on the applications, and hence it can be static or dynamic. For instance, the $T_{(s,e,a)}$ may be static in order to conform to the QoS requirements imposed by the user, or the requirements imposed by the PU; while it may be dynamically adjusted so that the threshold reduces the number of (*state*,*event*,*action*) entries in order to reduce the number of explorations necessary to approximate the optimal action.

5.3.7 Effects of Actions on the Environment

In wireless networks, the environment dynamics can be affected by the actions of various agents in a shared medium. For instance, if two neighbor nodes access a similar channel in a multi-channel environment, they share the reward or transmission opportunities among themselves. However, this is not always the case as some types of actions (such as channel sensing) do not affect the environment.

There are two types of RL approaches as follows:

- Single-Agent Reinforcement Learning (SARL) [4]. The SARL approach has been applied in this thesis in operating environment with a single agent, such as the base station in a centralized network, so that it learns and takes actions that maximize its own network performance.

- Multi-Agent Reinforcement Learning (MARL) [64]. The MARL approach has been applied in this thesis in operating environment with multiple agents, such as all the SUs in a distributed CR network, so that they learn and take their own respective action as part of the joint action in a cooperative and distributed manner to maximize the network-wide performance. The joint action is the actions taken by all the SUs throughout the entire network.

The SARL has been called RL in most literatures. In this thesis, we refer to SARL and RL as the single-agent approach, and MARL as the multi-agent approach.

The MARL approach is embedded in each agent in the network, and it is more suitable if the agents' actions can affect the environment in distributed networks. To facilitate coordination, the agents share the information related to the rewards among themselves so that each of them can evaluate its own action in a shared environment [65]. For example, a message exchange mechanism is proposed in Section 7.3 so that actions taken by all the agents converge to an optimal or near-optimal network-wide performance, including networks with cyclic topology [14]. Future research could be pursued to investigate coordination among the agents further.

5.4 RL Approach in CR Networks

Chapters 6 and 7 focus on implementing the conceptual cognition cycle using the RL approach. There are two levels of cognition cycle: node-level

and network-level (see Section 2.3 on page 14). This thesis applies the following:

1. SARL to implement the Single-Agent Cognition Cycle (SACC), also known as network-level cognition cycle, in centralized CR networks.

2. MARL to implement the Multi-Agent Cognition Cycle (MACC), also known as nodel-level cognition cycle, in distributed CR networks.

Note that "single-agent" and "multi-agent" are terms commonly found in the field of artificial intelligence and machine learning; while "network-level" and "node-level" are terms commonly found in the field of CR. Chapter 6 discusses SACC. Chapter 7 discusses MACC.

In chapter 4, we have presented a cross-layer QoS architecture, namely C^2net for cognitive wireless ad hoc networks. Using the SACC and MACC models, Chapter 8 presents the RL models for the cross-layer designs to show the usefulness of these models.

5.5 Chapter Summary

This chapter advocates the use of reinforcement learning to achieve context awareness and intelligence in wireless networks, particularly CR networks. In general, context awareness and intelligence enable each agent to observe, learn, and respond to its complex and dynamic operating environment in an efficient manner for network-wide performance enhancement without adhering to a strict and static predefined set of policies. The notion of context awareness and intelligence is very much related to the conceptual cognition cycle in CR networks. This capability is of paramount importance for general functionality and performance enhancement in CR networks. A generic RL model to achieve context awareness and intelligence as well as several new features, which do not exist in traditional RL approaches, including event, rules and effects of actions to environment, are presented. Certainly, there is a great deal of future work

in using the RL model for CR networks. To achieve context awareness and intelligence in CR networks, Chapter 6 discusses SACC for the application in centralized CR networks; while Chapter 7 discusses MACC for the application in distributed CR networks.

Chapter 6

Single-Agent Cognition Cycle

This chapter presents single-agent reinforcement learning for achieving context awareness and intelligence in static and mobile centralized cognitive radio networks through the implementation of the Single-Agent Cognition Cycle (SACC) or the network-layer cognition cycle. Investigation is performed with respect to the DCS scheme. This chapter presents the single-agent reinforcement learning approach rather than the multi-agent reinforcement learning approach. Hence, for simplicity, single-agent reinforcement learning is referred to as reinforcement learning.

Firstly, in the Introduction section, this chapter presents objectives, the RL approach, as well as assumptions and related work. Secondly, it presents related work on the learning mechanism, application of RL, MAC protocols and DCS scheme in the field of CR networks. Thirdly, the chapter presents an RL approach to DCS and proposes several simpler pragmatic DCS mechanisms that are used as a comparison. These mechanisms are Adaptation (Adapt), Window (Win) and Adaptation-Window (AdaptWin). Fourthly, it presents an analytical model for DCS to derive analytical results. Fifthly, it presents simulation experiment, results and discussions. The RL, Adapt, Win and AdaptWin approaches are investigated in detail. This covers three major investigations with respect to DCS as follows:

- The effects of state (see Section 5.3.4 on page 71) on applications that require state representation. In RL, the state emcompasses the condition of the operating environment that are relevant to decision making.
- The effects of various parameters for RL, Adapt, Win and AdaptWin on network performance.
- Comparison of the RL, Adapt, Win and AdaptWin approaches, as well as comparison with analytical results.

The simulation experiment, results and discussions section also discusses the advantages of the RL approach.

6.1 Introduction

6.1.1 Objectives

In static and mobile centralized CR networks, the DCS scheme provides the strategy to select an available licensed data channel for data transmission from an SU BS to a static or mobile SU host. The objective is to maximize overall throughput and minimize delay (in terms of number of channel switchings) in the presence of different levels of PUL and PER in the licensed data channels having different transmission ranges, as well as nodal mobility. The PUL and PER are explained in Section 4.1 on page 44.

6.1.2 The Reinforcement Learning Approach

Reinforcement Learning [4] is here applied to achieve context awareness and intelligence in static and mobile centralized CR networks with respect to DCS, though it can also be applied in topology management, scheduling, congestion control, and other applications (see Chapter 8). The network performance of RL is compared with various simple and pragmatic

learning mechanisms including Adapt, Win, AdaptWin, as well as analytical results. There are several applications that apply RL in CR networks [58, 59, 60, 66, 61, 62, 63]; however, none of them provides comparison with other learning mechanisms and analytical results.

6.1.3 Assumptions and Related Work

To date, research has focused on how an SU exploits and uses the white spaces with the assumption of channel homogeneity and static networks [25, 26, 27]. With channel homogeneity, the available data channels across the spectrum bands have similar levels of PER and transmission range, though they have different levels of PUL. However, our research focuses on the next level of enhancement, which is how an SU exploits and uses *high* quality (or low PER level) white spaces across heterogeneous channels for successful data packet transmission in centralized CR networks with static or mobile SU hosts. In practice, the SUs are expected to operate over a wide range of non-contiguous frequency bands [22], where the time scale of the spectrum occupancy varies from milliseconds to hours. Hence, the RL approach must learn to be responsive to highly dynamic spectrum occupancy. In addition, selected licensed channels must be sufficiently far apart from each other that it is not likely that they are simultaneously suspended by a particular PU. The properties of the white spaces at different frequencies vary with carrier frequency and time-varying channel condition. In addition, there are many other factors that affect the channel condition such as nodal mobility, neighbour interference, and transmission power. Thus, we consider channel heterogeneity where the available white spaces have different PER levels and transmission range. Through context awareness and intelligence, an SU is able to sense white spaces and also to infer their data channel quality so that the successful data packet transmission rate should be high.

A detailed explanation on the common assumptions in the CR research

field is found in Section 2.4 on page 17. In this chapter, our assumptions are as follows:

- Static and mobile networks. Previous schemes [25, 26, 27] assume only static networks.
- Centralized networks as applied in previous schemes [67].
- Channel heterogeneity. Previous schemes [25, 26, 27] assume channel homogeneity.
- Simplified RL model without consideration of events (see Section 5.3.4 on page 71), rules (see Section 5.3.6 on page 73), and effects of actions on the operating environment (see Section 5.3.7 on page 74).

Note that the assumption of a single collision domain, as well as identical or non-identical channel condition at all the SUs (see Section 2.4 on page 17) are applicable in distributed CR networks only and they are ignored in this chapter.

6.2 Chapter Goal

This chapter presents RL as an approach to implement the single-agent cognition cycle. This chapter provides an overview of learning mechanism, as well as related work on CR networks including the application of RL, MAC protocols and DCS in Section 6.3. There are seven new contributions in this chapter with respect to static and mobile centralized CR networks:

- We show how the RL approach and other learning mechanisms including Adapt, Win and AdaptWin can be applied to model the DCS scheme in Section 6.4.

- We show in Section 6.5 how to derive analytical results using Markov chain analysis, specifically for estimating throughput performance, for the DCS scheme.

- We investigate the effects of multiple states in RL on network performance in Section 6.7.

- We investigate the effects of changes in the parameters of RL on network performance in Section 6.8.

- We investigate the effects of changes in the parameters of Adapt, Win and AdaptWin on network performance in Section 6.9.

- We compare RL with Adapt, Win and AdaptWin, as well as analytical results in Section 6.10.

- We discuss the advantages offered by the RL approach compared to the other learning mechanisms in Section 6.11.

The simulation platform, objectives and performance metrics, ordinates, baseline and parameters applicable to all simulations in this chapter are shown in Section 6.6. In addition, we propose solutions for problems associated with RL. The results presented in Section 6.10 show that RL, which has been applied in previous applications including DCS and channel sensing [62, 63, 11], achieves similar network performance to AdaptWin and Win, which provide the highest network performance among the other learning mechanisms studied. We discuss the advantages offered by the RL approach compared to other learning mechanisms in Section 6.11. Finally, Section 6.12 concludes this chapter.

6.3 Related Work

6.3.1 An Overview of the Learning Mechanism

The learning mechanism model is embedded in the SU BS, which is the agent or decision maker. The flowchart of the RL model is shown in Figure 5.3 on page 72. This chapter does not consider the events, rules and effects of actions to the environment. The flowchart of the learning mechanism model under consideration, which is based on the RL model [4], is shown in Figure 6.1.

Two types of action selections are

- *Exploitation* chooses the best known action (aka the greedy action) at all times.
- *Exploration* chooses non-optimal actions once in a while in order to improve the estimates of all the Q-values in the Q-table so that better actions may be discovered. In CR networks, exploration is necessary as most applications require an SU BS to keep track of its operating environment, i.e. out-of-band measurement (see Chapter 3.4.1 on page 34) that requires the SU BS to keep a list of prioritized backup channels in IEEE 802.22 [23]. Therefore, all the learning mechanisms in this chapter perform exploration.

This chapter applies the ε-greedy approach [4] where an agent chooses the greedy action as its next action with probability 1-ε, and random action with a small probability ε.

The two main tasks are

- *Action selection*. During exploitation, the agent observes the operating environment, chooses an exploration or exploitation action, and executes the action.
- *Knowledge update*. The agent observes the consequence of its previous action and reward, and updates its knowledge.

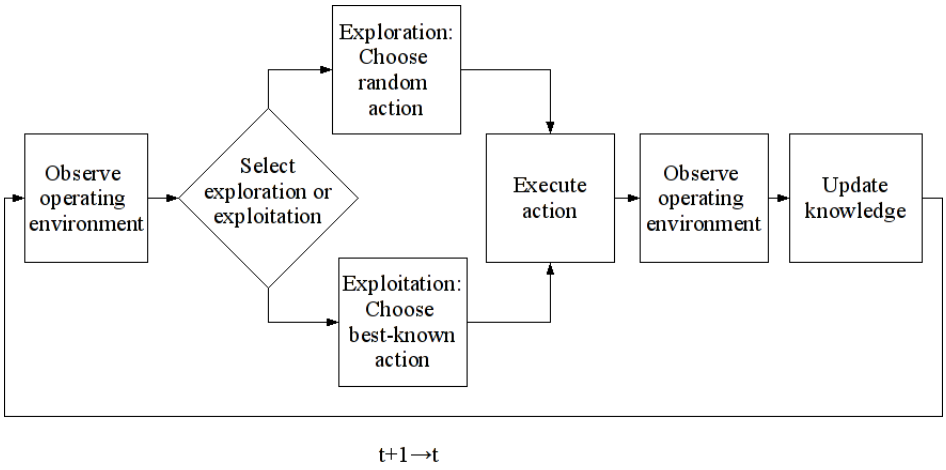


Figure 6.1: Flowchart of the learning mechanism model.

Applying simple and pragmatic learning mechanisms, such as RL, Adapt, Win and AdaptWin, in DCS provides three major advantages:

- A learning mechanism helps an SU BS to adapt to its dynamic and uncertain operating environment.
- A learning mechanism uses a simple modeling approach, thus the complexity involved in modeling the operating environment and channel heterogeneity can be minimized. For instance, an SU B-S that selects a data channel for data transmission does not model the uncertain and varying data channel conditions, which is characterized by various factors including shadowing, channel selective fading, path loss, PU interference and others, that affect the SU performance in a complex manner. Having said that, it is possible to add complexity to the RL model in order to tackle more complex problems.
- Rather than addressing a single factor at a time, a learning mechanism enables an SU BS to observe relevant factors in decision making and achieve a general goal as a whole with regard to the operating environment. For instance, many factors including data channel

conditions, nodal mobility and other unknown reasons may affect throughput performance. The SU BS observes the throughput performance and enhances it as a whole, rather than the need to design various applications to tackle each factor.

6.3.2 Application of Reinforcement Learning in Cognitive Radio Networks

RL has been applied successfully in a number of areas. In [66], a conceptual architecture that applies machine learning technique is suggested to enhance the network performance in CR networks. In this chapter, RL, which is a machine learning technique, is applied to implement the conceptual architecture.

In [58], the tradeoff between exploitation and exploration in an RL approach, namely "multi-armed bandit", is investigated with respect to DCS. In [61], the application of RL to PU signal detection is presented so that the SU can confirm the existence of the PU signal in the future although the PU signal may have deviated from its known signature. The investigations in [58] and [61] use performance metrics in machine learning such as regret and fitness value; while this chapter uses network performance metrics such as throughput and delay.

In [62], the application of RL to DCS in distributed CR networks is presented and the effects of RL parameters on network performance are investigated. The purpose is to reduce call blocking and dropping probabilities. In [63], the application of RL to DCS among a number of BSs is presented. The purpose is to enable each BS to cover a minimum percentage of service area with the highest SINR to support multicast traffic in order to reduce call blocking and dropping probabilities. In [60], the application of RL to detection of white spaces at the SU BS in centralized CR networks is presented. The purpose is to identify channels with the most available white spaces. In [59], the application of RL to DCS in Orthogo-

nal Frequency Division Multiple Access (OFDMA) networks is presented. The purpose is to improve the PU's network performance metrics including spectral efficiency, users' QoS satisfaction, and the amount of licensed spectrum bands to be released to the SUs.

6.3.2.1 New Contributions in Comparison to Related Work

As complements to [58, 59, 60, 66, 61, 62, 63], this chapter provides analytical results, and compares the RL results with that of other learning mechanisms including Adapt, Win and AdaptWin. Previous work considers homogeneous channels and static networks, while this chapter considers heterogeneous channels, and both static and mobile networks. Additionally, the use a Markov chain anlytical model to derive expected network performance for comparison with the RL approach in static and mobile networks is the first of its kind.

6.3.3 Medium Access Control Protocol for Cognitive Radio Networks

In this chapter, the DCS scheme, which is modeled using the RL approach and various learning mechanisms, is applied in a Carrier Sense Multiple Access (CSMA)-based cognitive MAC protocol.

The common control channel approach (see Section 3.3.3 on page 26) is adopted. Each SU is equipped with two transceivers:

- The *Control transceiver* is tuned to a common control channel, which is free from PU activities, and it is used for control message exchange including RTS and CTS that contains channel switching information. The common control channel should also provides the largest transmission range.

- The *Data transceiver* is tuned to one of the available data channels for data packet transmission. The data channels have PU activities.

Hence, two assumptions applied in this chapter are:

- Availability of two transceivers.
- Availability of a common control channel that is free from PU activities.

Our purpose in this chapter is to show the network performance enhancement brought about by the application of RL and various learning mechanisms, and we believe that these assumptions can be relaxed in real applications as described in Section 3.3.3 on page 26. The next two subsections describe the aforementioned two assumptions from the perspective of CR; while section 3.3.3 describes the assumptions from the perspective of multi-channel MAC protocols.

6.3.3.1 Availability of Two Transceivers

Two transceivers are applied in [68, 11, 12]; while a single transceiver is applied in [41, 69, 70]. Using a single half-duplex radio transceiver, each SU cannot transmit and receive simultaneously; however, it can switch its channels dynamically. Using multiple half-duplex radio transceivers, each SU can transmit and receive simultaneously in different channels, so there is network performance enhancement. However, multiple transceivers increase hardware cost. Nevertheless, with the price of transceivers falling dramatically, it is feasible to consider using multiple transceivers at each SU. An example of a cognitive MAC that applies a single transceiver is C-MAC [41]. For neighbour discovery, it requires that each SU listens to and broadcasts information, i.e. its neighbour SUs and the list of data channels that they are listening to, in a common control channel. This is not necessary if two transceivers are used, since the control transceiver is listening to a common control channel at all times, and it can be used for neighbour discovery. Another consideration is the tradeoff between better network performance and higher energy consumption with the increased

number of transceivers. However, the design of the MAC protocol affects this tradeoff since a transceiver can always be made to sleep whenever it is inactive.

6.3.3.2 Availability of a Common Control Channel

Most of the cognitive MAC protocols apply a single common control channel approach including [69, 70, 11, 12]. However, in CR networks, a global common control channel that is free from PUs may be difficult to be found. In [40, 41], clustering schemes for CR networks are proposed such that each cluster chooses an available channel for control message exchange so that a global common control channel is not necessary.

6.3.4 Dynamic Channel Selection

In [25], channel assignment is performed at the granularity of *segments* such that a centralized CR network is segregrated into various segments, which may be affected by different PUs, that use different channels for data transmissions in order to enhance network-wide throughput and delay performance. In [26], channel selection is performed to predict the PU traffic patterns based on history information in order to reduce the number of channel switchings. In [27], channel assignment is driven by a routing protocol so that the link costs caused by the channel switches as a result of PU activities are considered in order to enhance network-wide throughput and delay performance. The investigations in [25, 26, 27] assume homogeneous channels and static networks; while this chapter considers heterogeneous channels such that each data channel may have different levels of PULs, PERs and transmission ranges. This chapter assumes the SU transmits using a fixed transmission power in different data channels; hence the transmission range for each data channel varies. In general, lower channel frequency provides larger transmission range.

Suppose an SU BS communicates with its SU host using channel 1. As

the channel quality or PER deteriorates, the successful data packet transmission rate decreases. The SU BS detects the deterioration in QoS, particularly throughput, and changes to channel 2 that provides better throughput performance. Other factors such as PUL and transmission range may also reduce throughput performance and affect network performance in a complex manner. A data channel with low PUL does not imply a good channel if it has a high PER.

6.3.4.1 Dynamic Channel Selection Scheme under Consideration

In this chapter, learning mechnisms including RL, Adapt, Win and AdaptWin are embedded in the SU BS. The learning mechanisms help the DCS scheme to empirically choose the best possible data channel considering most of the factors that affect the network performance. We assume that the SU BS is always backlogged and it transmits data packets to its SU host. Due to the limited sensing capability at each SU, there are K available data channels. Based on a conventional assumption, the K available data channels for data transmission are provided by the spectrum pooling mechanism (see Section 2.4). The action is to choose a data channel for data transmission from the available data channels set $C=\{c_i=1,2,\ldots,K\}$. Data packet transmission is classified successful when a link-layer acknowledgment is received for the data packet sent, else the transmission is classified unsuccessful. Additionally, if an SU senses PU signals immediately prior to transmission, it is classified unsuccessful.

6.4 Learning Mechanisms as Implementation of SACC

The DCS learning mechanisms determine how an SU BS, which is the agent, chooses its data channel for data transmission. There are two major differences among the four kinds of learning mechanisms, namely RL,

Adapt, Win and AdaptWin, as follows:

- During action selection, "How does the agent choose its best known action during exploitation?"
- During knowledge update, "How does the agent maintain and update its knowledge?"

In the next few subsections, we present the learning mechanisms based on the two aforementioned features.

6.4.1 Reinforcement Learning (RL) Approach

Q-learning [4] (see Section 5.3 on page 69 for theoretical explanation), which is an RL algorithm, is applied to approximate the optimal data channel for data transmission. The SU BS keeps track of the learned action value or Q-value, $Q_t(c_i)$ for all the available data channels C in a Q-table with $|C|$ entries. The Q-value $Q_t(c_i)$, which represents the knowledge, indicates the appropriateness of choosing data channel c_i in the operating environment. In other words, the Q-value estimates the level of local reward for a data channel c_i; hence changes in the Q-value will lead to changes in an SU BS's channel selection. At each attempt to transmit a data packet, the SU BS chooses a data channel c_i and receives a local reward $r_{t+1}(c_{i,t})$ at time t+1.

6.4.1.1 Knowledge Update Procedure

During knowledge update, the Q-value of a chosen data channel $c_{i,t}$ at time t is updated at time t+1 as follows:

$$Q_{t+1}(c_{i,t}) \leftarrow (1-\alpha)Q_t(c_{i,t}) + \alpha r_{t+1}(c_{i,t}) \tag{6.1}$$

where $0 \leq \alpha \leq 1$ is the learning rate, and $r_{t+1}(c_{i,t})$ is the immediate reward, which is the reward received at time t+1 for the data channel selected at

time t. The higher the value of α, the greater the SU BS relies on the immediate reward. For every successful data packet transmission, there is a reward with positive constant value $r_{t+1}(c_{i,t})$=+*RW*, otherwise there is a cost with negative constant value $r_{t+1}(c_{i,t})$=-*CT*. In practice, the value of *RW* and *CT* are based on the amount of revenue and cost that a network operator earns or incurs for each successful or unsuccessful data packet transmission. As time goes by, the agent receives a sequence of rewards from the data packet transmission procedure.

6.4.1.2 Action Selection Procedure

During exploitation, the SU BS chooses an exploitation or greedy action, which is the data channel with the highest Q-value, as follows:

$$c_{i,t} = \underset{c_i \in C}{\operatorname{argmax}}\, Q_t(c_i) \tag{6.2}$$

Two conditions that trigger a channel switch are as follows:

- Switching to a data channel with higher Q-value.
- Exploration.

6.4.1.3 Reinforcement Learning Model for Dynamic Channel Selection Scheme

The RL model for the DCS scheme is shown in Table 6.1. Based on a conventional assumption, the *K* available data channels for data transmission are provided by the spectrum pooling mechanism (see Section 2.4).

6.4.1.4 Derivation of the Reinforcement Learning Model

A similar RL model to this one has been applied in DCS such as [62, 63, 11] although these DCS schemes are applied to enhance different performance

Table 6.1: RL Model (SACC) at SU BS for DCS

	Dynamic Channel Selection Model	
	Description	Representation
Action	Available data channels for data transmission.	C={c_i=1,2,...,K}
Reward	Constant value to be rewarded/incurred for successful/unsuccessful data packet transmission.	$r_{t+1}(c_{i,t})$ = $\begin{cases} +RW, & \text{if successful} \\ -CT, & \text{if otherwise} \end{cases}$

metrics in different kinds of scenarios (see Section 6.3.2 for more details). In the next paragraph, we explain how Equation (6.1) is derived from the original Q-value function in Equation (5.1) on page 70.

The state representation is eliminated. In Section 6.7, the state is represented as a set of an SU BS's neighbour nodes. Since the RL model is embedded in the SU BS, the set of neighbour nodes is comprised of static or mobile SU hosts. In Section 6.8, the set of neighbour nodes is comprised of a *single* static or mobile SU host. Similar trends are observed in Section 6.7 and 6.8. Hence, for simplicity, two SUs are considered in most investigations in this chapter including Section 6.5, 6.8, 6.9 and 6.10, namely an SU BS and an SU host. With a single SU host or state, the state representation is eliminated, which is often called stateless or single-state as explained in Section 2.3 (page 14). This means that after performing a particular action, the SU BS remains in its initial state. In other words, an SU BS does not change its SU host or state during every data packet transmission. As the discounted reward or $\gamma\ max_{a \in A} Q_t(s_{t+1}, a)$ in Equation (5.1) depends on the next state, the γ is set to 0 value because the state never changes when an action is being carried out. Hence, in our RL model, the state and discounted reward are eliminated to give Equation (6.1).

6.4.2 Adaptation (Adapt) Approach

There is no knowledge update in this approach, and the action selection is random during a channel switch.

6.4.2.1 Action Selection Procedure

During exploitation, the SU BS chooses its previous chosen data channel.

Two conditions that trigger a channel switching are as follows:

- The number of consecutive failed data packet transmissions reaches a threshold $n^{\mathbf{Adapt}}$.
- Exploration.

After channel switching, the agent remains in the data channel until either one of these two conditions are encountered.

6.4.3 Window (Win) Approach

In the Win approach, the SU BS keeps track of the probability of successful data packet transmission, P_{S,c_i}^{Win} for all the available data channels C in a Win-table with $|C|$ entries.

6.4.3.1 Knowledge Update Procedure

Denote the number of most recent attempts of data packet transmissions or window size by $n^{\mathbf{Win}}$, and the number of successful data packet transmissions within $n^{\mathbf{Win}}$ using channel c_i by $n_{S,c_i}^{\mathbf{Win}}$.

During knowledge update, the SU BS keeps track of $n_{S,c_i}^{\mathbf{Win}}$ and updates this information in its Win-table.

6.4.3.2 Action Selection Procedure

During exploitation, the SU BS computes the probability of successful data packet transmission using data channel c_i, P^{Win}_{S,c_i}=$n^{\mathbf{Win}}_{S,c_i}/n^{\mathbf{Win}}$, and chooses the data channel with the highest P^{Win}_{S,c_i} as follows:

$$c_{i,t} = \underset{c_i \in C}{\operatorname{argmax}} P^{Win}_{S,c_i} \tag{6.3}$$

Two conditions that trigger a channel switching are as follows:

- Switching to a data channel with higher P^{Win}_{S,c_i}.
- Exploration.

6.4.4 Adaptation-Window (AdaptWin) Approach

AdaptWin incorporates both Adapt and Win approaches.

6.4.4.1 Knowledge Update Procedure

During knowledge update, the agent keeps track of $n^{\mathbf{Win}}_{S,c_i}$ and updates this information in its Win-table.

6.4.4.2 Action Selection Procedure

During exploitation, the SU BS computes P^{Win}_{S,c_i} and chooses the data channel with the highest P^{Win}_{S,c_i} using (6.3).

Two conditions that trigger a channel switching are as follows:

- The number of consecutive failed data packet transmissions reaches a threshold $n^{\mathbf{Adapt}}$.
- Exploration.

A difference between Adapt and AdaptWin in channel switching is that Adapt remains in the exploring channel after exploration; while

AdaptWin chooses the channel with the highest P_{S,c_i}^{Win} using (6.3) after exploration. In AdaptWin, during knowledge update and action selection, AdaptWin follows the Win approach; while the conditions that trigger a channel switching follow the Adapt approach.

6.5 Analytical Model for DCS

In this section, we present analytical models to assess the estimated network performance, specifically, the expected throughput of a DCS scheme in static and mobile centralized CR networks. The analytical models are derived using Markov chain. Since Markov chain is a memoryless analytical tool, it does not apply any learning mechanism. Our purpose is to show whether learning-based RL, Adapt, Win and AdaptWin achieve the estimated throughput offered by non-learning mechanism. In the analytical model, an SU BS chooses the next data channel randomly as long as the data packet transmission is successful; while in the RL, Adapt, Win and AdaptWin model, an SU BS chooses the next best known data channel based on the outcome from the learning mechanisms.

6.5.1 Characteristics of Centralized Cognitive Radio Networks and Assumptions

A graphical representation of our scenario is shown in Figure 6.2, and its characteristics and assumptions are:

- **Primary Users**
  - There are K PUs, PU=[PU_1,. . .,PU_K].
  - Each PU uses one of the K distinctive channels of frequency F=[F_1,. . .,F_K] and broadcasts packets throughout the entire simulation area. The PUs do not change their respective channel,

thus there are K PUs and channel frequencies. The PUs do not use four-way handshaking.

- The PUs are not aware of the presence of the SUs.
- The channel utilization pattern of the PUs follow a Poisson distribution with the mean arrival rate determined according to the PUL level, and among the data channels it follows an independent and identically distributed (i.i.d.) stochastic model.

- **Secondary Users**
  - Each SU node is equipped with two transceivers, namely a *control transceiver* and a *data transceiver*, thus it is capable of accessing two different channels simultaneously.
    - The control transceiver is tuned to a common channel in the ISM band for control message exchange, as well as information broadcast.
    - The data transceiver is tuned to one of the available data channels in the licensed bands for data packet transmission. Thus, the PU activities exist in the data channels only.
  - There are two SUs to model a scenario for SACC: an SU BS and an SU host.
  - The SU BS is always backlogged and transmits data packets to the SU host at every opportunity.
  - The SUs transmit without interfering with the PUs.
  - The learning mechanism model is embedded in the SU BS; while the SU host switches its data channel according to the decision made by the SU BS. The SU host is informed of the changes in the data channel through RTS and CTS control message exchange in the common control channel.
  - The transmission time for a data packet (C) and its header information (H) for the SU is $t_{H+C,SU}$.

- **Channel Characteristics**

  - There are K orthogonal available data channels with similar bandwidth. In this section, the notation for the data channels is C=$\{i$=c_i=1,2,. . .,$K\}$ to indicate data channel c_i in the previous section as data channel i for simplicity. Unless otherwise specified, channel i is referred to data channel i, rather than control channel.

  - Each data channel is characterized by various levels of PUL, L_{c_i}=[L_1,. . .,L_K]. Higher level of PUL in a particular data channel indicates higher level of PU activity. The data packet arrival of PU traffic in data channel i is a Poisson process with mean data packet arrival rate $\lambda_{PU,i}$. Higher values of $\lambda_{PU,i}$ lead to higher PUL in data channel i. According to the superposition property, the merging of multiple Poisson processes with different mean arrival rate $\lambda_{PU,i}$ is equivalent to a single Poisson process with its mean arrival rate $\sum \lambda_{PU,i}$; hence, modeling a single PU in each data channel is sufficient.

  - The PER P_i^E indicates the level of failed data packet transmission due to uncertain and varying data channel conditions caused by various factors including shadowing, channel selective fading, path loss, PU interference, and other factors in channel i.

- **CSMA-based Cognitive MAC Protocol**

  - At the time this thesis is written, there is not yet a standard available for a cognitive MAC protocol. Section 6.3.3 lists related work on cognitive MAC. A CSMA-based cognitive MAC with DCS implementation is presented in this section. Synchronization among the SUs is not necessary. The common control channel approach (see Section 3.3.3 on page 26) is adopted. Each

SU is equipped with two transceivers.

– An illustration of the cognitive MAC protocol is shown in Figure 6.3. Switching delay may be ignored if channel switching is not necessary. Since the most recent spectrum sensing outcome indicates the PU occupancy in a particular data channel, the data channel, which was free, may become busy within a Short Inter-Frame Spacing (SIFS) interval immediately prior to data packet transmission. In this case, the SU BS restarts its data packet transmission cycle with RTS-CTS handshaking, and may reassign its data channel. The RTS and CTS contain channel switching information.

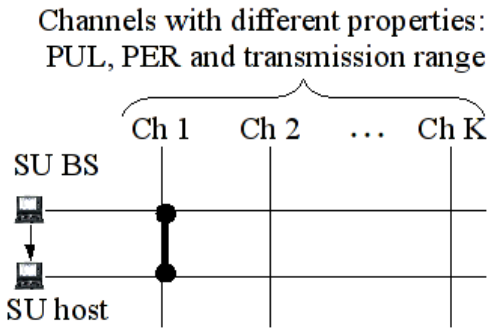


Figure 6.2: Graphical representation of the DCS scheme. The bold line indicates data packet transmission from an SU BS to an SU host over a chosen data channel. The common control channel is not shown.

Next, we provide an analytical model for static networks in Section 6.5.2. Its extension to mobile networks is provided in Section 6.5.3.

6.5.2 Analytical Model for Static Networks

We derive the expected throughput of an SU BS using two Markov chains. In the first Markov chain, as shown in Figure 6.4, we determine the

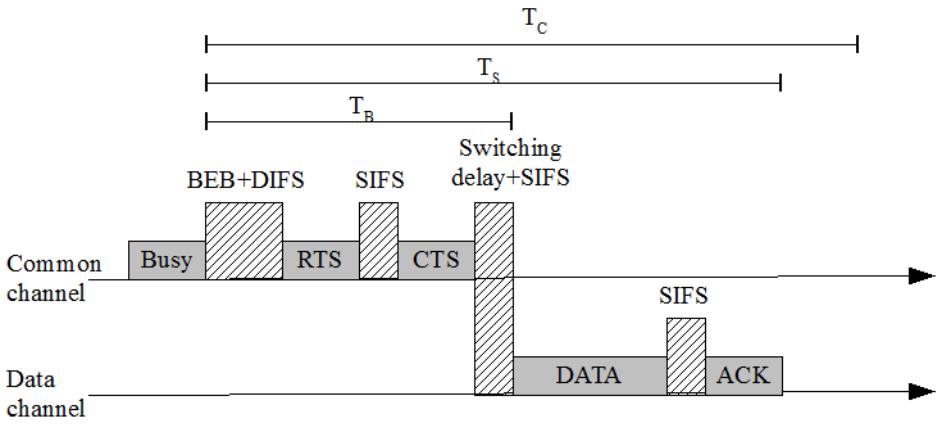


Figure 6.3: Illustration of cognitive MAC protocol.

probability distribution of channel selection $\pi^P(i)$ by the SU BS for data packet transmission over the K available channels. Each state i in the Markov chain represents a channel number that the SU BS can choose for data packet transmission. The Markov matrix, $\mathbf{P}$ is

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \cdots & P_{1K} \\ P_{21} & P_{22} & P_{23} & \cdots & P_{2K} \\ P_{31} & P_{32} & P_{33} & \cdots & P_{3K} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ P_{K1} & P_{K2} & P_{K3} & \cdots & P_{KK} \end{bmatrix} \tag{6.4}$$

With $t_{H+C,SU}$ being the header and data packet transmission time for the SU, the probability of unsuccessful data packet transmission in channel i, P_i, is dependent on its PER P_i^E, and PUL with mean data packet arrival rate $\lambda_{PU,i}$:

$$\begin{aligned} P_i &= 1 - \{(1 - P_i^E)(1 - \textstyle\int_0^{t_{H+C,SU}} \lambda_{PU,i} e^{-\lambda_{PU,i} t} dt)\} \\ &= 1 - \{(1 - P_i^E) e^{-\lambda_{PU,i} t_{H+C,SU}}\} \end{aligned} \tag{6.5}$$

Note that 1-$\int_0^{t_{H+C,SU}} \lambda_{PU,i} e^{-\lambda_{PU,i} t} dt$ represents the probability of no PU arrival within time interval $t_{H+C,SU}$. Suppose the SU BS is transmitting in

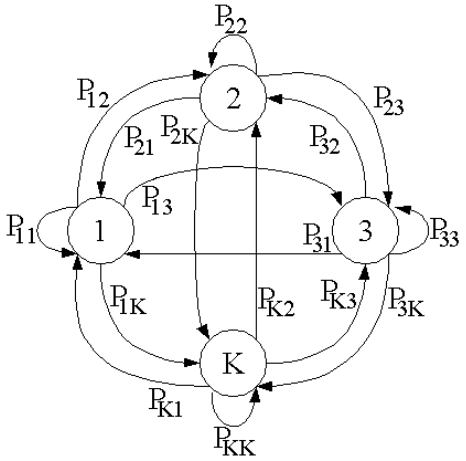


Figure 6.4: Markov chain model of dynamic channel selection.

channel i. There are two situations where it does not switch its channel, or selects the same channel for its next data packet transmission. Firstly, data packet transmission is successful in channel i. Secondly, data packet transmission is unsuccessful in all K channels, thus there is no benefit in switching its channel. Hence, the probability that an SU BS does not switch its channel, P_{ii}, is

$$P_{ii} = (1 - P_i) + \prod_{k=1}^{k=K} P_k \tag{6.6}$$

Next, suppose the SU BS is transmitting in channel i. There are two situations where it switches its channel from i to $j \in \{1,2,\ldots,K\} \backslash i$. Firstly, data packet transmission is unsuccessful in channel i, but only successful in channel j. Secondly, data packet transmission is unsuccessful in channel i, but successful in channel sets m_y. There are many possible channel sets m_y. For instance, in the case of K=4 channels, with i=1 and j=2, there are three sets of channels that provide successful data packet transmission in channel j=2: m_1={2, 3, 4}, m_2={2, 3} and m_3={2, 4}, where each element in the set m_y indicates a channel number. We consider that the probability of choosing channel j is equally divided among the channels in the set of m_y.

If $\mathbf{M}=\{m_y\}$, the probability of the SU BS switches its channel from i to j, P_{ij} can be written as

$$\begin{aligned} P_{ij} &= P_i(1-P_j)(\prod_{k=1,k\neq i,j}^{k=K} P_k) \\ &+ P_i \sum_{m_y \in \mathbf{M}} \frac{\prod_{k=1\ldots K, k\in m_y, k\neq i}(1-P_k) \prod_{k=1\ldots K, k\notin m_y, k\neq i} P_k}{|m_y|} \end{aligned} \tag{6.7}$$

Let the steady state probability for $\mathbf{P}$ be denoted by row vector $\pi^{\mathbf{P}}$, which is comprised of $\pi^P(i)$, and its value is obtained by solving (6.8) and (6.9). The steady state probability $\pi^{\mathbf{P}}$ provides a probability distribution of channel selection over the K available channels by the SU BS for data packet transmission. This means that the higher the probability of $\pi^P(i)$, the more likely it will be that channel i is chosen for data packet transmission by the SU BS. The steady state probability $\pi^{\mathbf{P}}$ is the solution to

$$\pi^{\mathbf{P}}\mathbf{P} = \pi^{\mathbf{P}} \tag{6.8}$$

$$\sum_{i-1}^{K} \pi^P(i) = 1 \tag{6.9}$$

Next, we apply our second Markov chain to estimate the average backoff window stage of the SU BS, $E[N]$, which is used to estimate the backoff duration, t_{BO}, experienced by the SU BS. In general, higher levels of contention at the MAC layer and more frequent unsuccessful data packet transmissions lead to higher levels of backoff stage, and hence longer backoff. The Markov chain is shown in Figure 6.5 (see [71] for more details). The state in the Markov chain represents the level of backoff stage, where the maximum stage level is z. The corresponding contention window size for each stage is $CW_{(i+1)}=2\times CW_i+1$. The contention window size is limited within the range of $CW_{min}=7 \leq CW_{(i+1)} \leq CW_{max}=255$ at all times. Denote the probability of unsuccessful data packet transmission across all the available channels by q, which is computed using $\pi^{\mathbf{P}}$. The

Markov matrix for the backoff process, $\mathbf{Q}$ is

$$\mathbf{Q} = \begin{bmatrix} 1-q & q & 0 & \cdots & 0 \\ 1-q & 0 & q & \cdots & 0 \\ 1-q & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{6.10}$$

with

$$q = \sum_{i=1}^{K} \pi^P(i) P_i \tag{6.11}$$

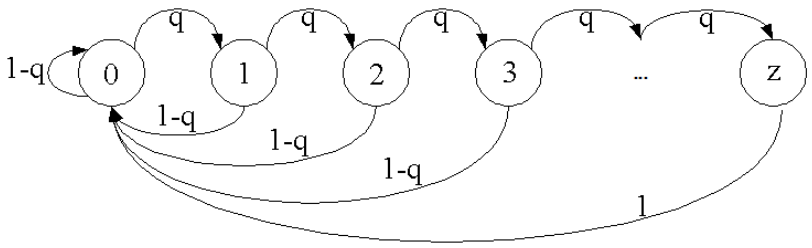


Figure 6.5: Markov chain model of backoff window stage.

The steady state probability for $\mathbf{Q}$ is denoted by vector $\pi^{\mathbf{Q}}$ and its value is obtained by solving (6.12) and (6.13). The steady state probability $\pi^{\mathbf{Q}}$ provides a probability distribution over the backoff stages at the SU BS.

$$\pi^Q(0) = (1-q) \sum_{i=0}^{z-1} \pi^Q(i) + \pi^Q(z) \tag{6.12}$$

$$\pi^Q(i) = q^i[1 - q + q\pi^Q(z)] \tag{6.13}$$

If σ is a backoff slot time, the estimated average backoff window stage $E[N]$ of an SU BS, and its estimated amount of backoff duration, t_{BO} are given by

$$E[N] = \sum_{i=0}^{z} (\frac{CW_i + 1}{2}) \pi^Q(i) \tag{6.14}$$

$$t_{BO} = E[N]\sigma \tag{6.15}$$

Next, we compute the probability of successful and unsuccessful data packet transmission across the available channels, as well as their respective timings. Let the probability of successful data packet transmission be P_S and its respective duration without the switching delay be $T_S = t_{RTS} + t_{CTS} + t_{ACK} + t_{H+C,SU} + t_{DIFS} + 3t_{SIFS} + t_{BO} + 4\delta$ (see Figure 6.3), where δ is the propagation delay and it is assumed to be constant in this thesis. The value of P_S is

$$P_S = \sum_{i=1}^{K} \pi^P(i)[e^{-\lambda_{PU,i} t_{H+C,SU}}(1 - P_i^E)] \tag{6.16}$$

There are two possibilities of timings for unsuccessful data packet transmission. Firstly, with probability P_B, a channel becomes busy, which happens within SIFS duration (see Figure 6.3), immediately before the SU BS attempts to transmit. The duration of this event without the switching delay incurred is $T_B = t_{RTS} + t_{CTS} + t_{DIFS} + 2t_{SIFS} + t_{BO} + 2\delta$. P_B is computed as follows:

$$\begin{aligned} Y &= 1 - \sum_{i=1}^{K} \pi^P(i) \int_0^{t_{SIFS}} \lambda_{PU,i} e^{-\lambda_{PU,i} t} dt \\ &= 1 - \sum_{i=1}^{K} \pi^P(i)(1 - e^{-\lambda_{PU,i} t_{SIFS}}) \end{aligned} \tag{6.17}$$

$$P_B = (1 - P_S)(1 - Y) \tag{6.18}$$

Secondly, with probability P_C, an SU BS fails to receive an ACK packet after data packet transmission due to packet loss or collision with PU transmission. The duration of this event without the switching delay incurred is $T_C = t_{RTS} + t_{CTS} + t_{DIFS} + 2t_{SIFS} + t_{ex} + t_{BO} + 2\delta$, where t_{ex} is the duration of the data packet expiration timer, which is initiated after transmitting a data packet and is reset upon receiving its corresponding ACK packet. The value of P_C is computed as

$$P_C = (1 - P_S)Y \tag{6.19}$$

Next, we compute the probability of the occurrence of channel switching across the K available channels. The channel switching delay, which

is hardware dependent and is assumed to be a constant value, is T_{SW}. Channel switching occurs when the current channel to which the node is listening has an unsuccessful data packet transmission. Also, there is no channel switching when all channels lead to unsuccessful data packet transmission. Therefore, P_{SW} is

$$P_{SW} = \sum_{i=1}^{K} \pi(i)(P_i - \prod_{j=1}^{K} P_j) \tag{6.20}$$

The estimated length of duration for transmitting a data packet or payload incurred with and without channel switching are given by

$$T_1 = P_S(T_S + T_{SW}) + P_B(T_B + T_{SW}) + P_C(T_C + T_{SW}) \tag{6.21}$$

$$T_2 = P_S T_S + P_B T_B + P_C T_C \tag{6.22}$$

The expected system throughput, S in number of packets per second, is then

$$\begin{aligned} S &= \frac{E[payload\ transmitted]}{E[length\ of\ duration\ for\ transmitting\ payload]} \\ &= \frac{P_S}{P_{SW}T_1 + (1-P_{SW})T_2} \end{aligned} \tag{6.23}$$

6.5.3 Analytical Model for Mobile Networks

Consider a static SU BS located at the center of a disk with radius D=$\{d$=d_1, d_2,...,$d_K\}$ that represents different transmission ranges using a fixed transmission power in various channels with d_K being the shortest transmission range, as shown in Figure 6.6. In general, lower transmission carrier frequency provides larger transmission range. Another mobile SU host is moving randomly within the maximum transmission range at distance d_1 from the SU BS. Let region K be the innermost region where the SU BS can choose to use one of the K available channels for data packet transmission if the SU host moves into this region; while region 1 is the outermost region where the SU BS can choose to use one channel only, namely channel 1. The proportion of the areas of the circle in region K, $P_{R,K}$

= $\pi d_K^2/\pi d_1^2$, at the centre and the annuluses at the outer regions, $P_{R,K-x}$ = $\int_{K-x+1}^{K-x} 2\pi d_\rho d(d_\rho)/\pi d_1^2$, provide the proportion of the time duration a mobile SU host spent in the respective regions. For instance, with K=3 and x=1 at region 2 as shown in Figure 6.6, $P_{R,2} = \int_3^2 2\pi d_\rho d(d_\rho)/\pi d_1^2 = \pi(d_2^2\text{-}d_3^2)/\pi d_1^2$. This mobility model is sufficient to demonstrate the effects of channels with different transmission ranges on the expected system throughput although other mobility model can be adopted.

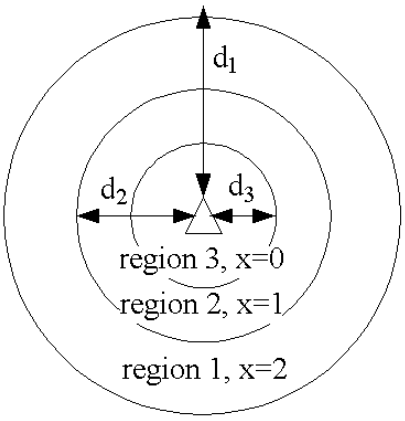
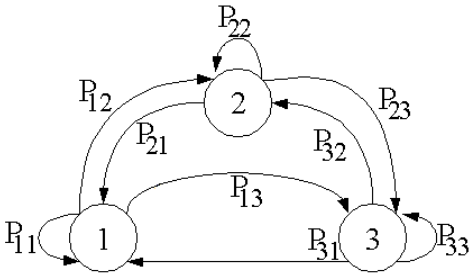


Figure 6.6: An SU BS and its transmission ranges using different channels with K=3.

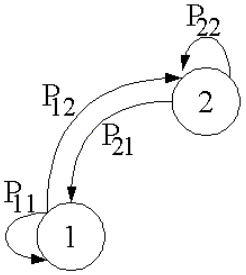
Using an example, the Markov chains at different regions when K=3 are illustrated in Figure 6.7. At the innermost region 3, the sender can choose one of the K=3 channels, so there are K states in the Markov chain in Figure 6.7a; while at the outermost region 1, there is one state only in the Markov chain in Figure 6.7c. This means that P_{ii} is calculated using (6.6) at region K=3; while P_{ii}=1 at region 1. Therefore, Equation (6.6) is rewritten as follows to incorporate all regions:

$$P_{ii,K-x} = (1 - P_i) + \prod_{k=1}^{k=K-x} P_k \tag{6.24}$$

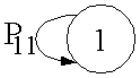
Similarly, P_{ij} is calculated using (6.7) at region K=3; while P_{ij}=0 at region 1. For the set of channels $m_{y,K-x}$, with K=3, i=1 and j=2 as example,



(a) Region 3



(b) Region 2



(c) Region 1

Figure 6.7: Markov chains at difference regions.

then the set $m_{y,K-x}$=$m_{1,3}$={2, 3} at region 3, while $m_{y,1}$=$m_{y,2}$={$\varnothing$}. Equation (6.7) is rewritten as follows to incorporate all regions:

$$\begin{aligned} P_{ij,K-x} &= P_i(1-P_j)(\prod_{k=1,k\neq i,j}^{k=K-x} P_k) \\ &+ P_i \sum_{m_{y,x}\in\mathbf{M}} \frac{\prod_{k=1\ldots K-x,k\in m_{y,x},k\neq i}^{(1-P_k)} \prod_{k=1\ldots K-x,k\notin m_{y,x},k\neq i}^{P_k}}{|m_{y,x}|} \end{aligned} \tag{6.25}$$

The steady state probability for $\mathbf{P_{K-x}}$ at different regions, $\pi^P_{K-x}(i)$, is calculated using (6.26) and (6.27), and these equations are rewritten as

$$\pi^{\mathbf{P}}_{\mathbf{K-x}}\mathbf{P_{K-x}} = \pi^{\mathbf{P}}_{\mathbf{K-x}} \tag{6.26}$$

$$\sum_{i=1}^{K} \pi^P_{K-x}(i) = 1 \tag{6.27}$$

The steady state probabilities for all regions for channel i, $\pi^P(i)$, is calculated as follows

$$\pi^P(i) = \sum_{x=0}^{K-1} P_{R,K-x}\pi^P_{K-x}(i) \tag{6.28}$$

Using the steady state probability vector $\pi^{\mathbf{P}}$ obtained from (6.28), Equation (6.10) to (6.23) are applied to estimate the expected system throughput, S in mobile networks.

6.6 Simulation Setup

This section discusses the simulation scenario, objectives and performance metrics, ordinates, baseline, parameters and organization of the remaining sections relevant to the simulation. This covers the simulation experiments, results and discussions in Section 6.7 to 6.10.

6.6.1 Simulation Scenario

The simulation scenario is provided in Section 6.5.1 and its graphical representation for the DCS scheme is shown in Figure 6.2.

6.6.2 Simulation Platform

We have implemented a CR-enabled environment in the INET framework for OMNeT++ [72]. OMNeT++ provides open-source wireless communication networks simulation package that supports both multi-channel transmission and nodal mobility. Most importantly, it models each component within a wireless host in a modular fashion, hence both data link and physical layers can be easily incorporated into a node.

At the time this simulation platform was developed, other possible platforms were Matlab [73], QualNet [74] and NS2 [75]. Due to the following reasons, these plaforms were not chosen:

- Matlab does not provide network simulation package that incorporates both data link and physical layers within a wireless host.
- Qualnet is a commercial tool that does not provide cognitive radio environment.
- NS2 does not adopt the modular framework, hence it is a complicated task to add additional transceiver as two transceivers, namely control and data transceivers, are required.

The relevant OSI layers in our simulation are data link and physical layers. At the data link layer, the original INET framework simulates the IEEE 802.11 CSMA-based MAC protocol; and at the physical layer, it operates in a single channel environment using a single transceiver in a free-space path loss model. It provides static and mobile, as well as centralized and distributed networks.

There are three main tasks to prepare for the simulation in a CR environment:

- To extend the original INET framework to operate in a multichannel environment. This requires modifications to the transceiver and operating environment.

- To add new transceiver so that each SU has two transceivers, namely control transceiver and data transceiver. This requires modifications to the architecture of the SU, and MAC protocol as the tranceivers must cooperate with each other to transmit control messages and data packets.

- To introduce various learning mechanisms to the MAC protocol.

6.6.3 Simulation Objectives and Performance Metrics

The simulation scenarios consider heterogeneous data channels such that each channel has different levels of PUL, PER and transmission range.

With heterogeneous channels consideration in all simulation scenarios, the goals of the DCS are

- To maximize throughput.

- To minimize number of channel switchings, which causes non-negligible delay for data packet transmission. Additionally, each channel switch also causes energy consumption. Note that, in contrast to the simulation results in Chapter 7 for the MACC approaches, channel switchings for exploration purpose are counted in this chapter.

6.6.4 Simulation Ordinates

Graphs are presented with PUL and PER as ordinate respectively. When PUL is ordinate, each simulation result of mean throughput or mean number of channel switchings is for all possible combinations of PUL. As an example, a PUL of 0.8 for K=3 available data channels may indicate [0.8,0.8,0.8], [0.8,0.7,0.9], and [0.9,0.9,0.6]. In the case of mobile networks, each set of PUL is applied to various data channels with different carrier frequencies, which provides different transmission ranges. As an example,

for [0.8,0.7,0.9], there are six runs in total with different permutation including [0.8,0.7,0.9], [0.8,0.9,0.7], [0.9,0.7,0.8], [0.9,0.8,0.7], [0.7,0.8,0.9], and [0.7,0.9,0.8]. The similar randomness applies to PER among the available data channels.

When network performance is investigated with respect to mean PUL for all data channels, the PER for all data channels is set to 0.1. This investigation shows the effectiveness of the learning mechanisms in choosing a data channel with a low level of PUL for data packet transmission in the presence of low PER for all data channels. When network performance is investigated with respect to mean PER for all data channels, the PUL for all data channels is set to 0.1. This investigation shows the effectiveness of the learning mechanisms in choosing a data channel with a low level of PER for data packet transmission in the presence of low level of PUL for all data channels.

6.6.5 Simulation Baseline

A common simulation baseline is the Random-based DCS. The Random chooses an available data channel in a random manner for every data packet transmission. Hence, it does not apply any learning mechanism.

6.6.6 Simulation Parameters

Table 6.2 shows the simulation parameters that are applicable to all simulation scenarios in Sections 6.7 to 6.10. Additional simulation parameters that are applicable to specific simulation scenarios are shown in seperate tables in Table 6.4 for Section 6.7, Table 6.5 for Section 6.8, and Table 6.6 for Section 6.9.

We here explain some of the simulation parameters of Table 6.2. The characteristics of the PU, SU, channel, and CSMA-based cognitive MAC protocol are discussed in Section 6.5. Each simulation is run for 500s. Each SU has limitation in channel sensing capability, and thus the number of

Table 6.2: Notations and Default Parameter Settings For All Simulation Setup in Section 6.7 to 6.10.

Category	Symbol	Details	Values
Initial ization	K	Number of available data channels	3
	F	Center carrier frequencies of K available data channels	{400MHz, 800MHz, 5.7GHz}
	$P_{c_i}^E$	PER of each available data channel	[0.1,0.9] Default: 0.1
	δ	Propagation delay	1ns
	T	Total simulation time	500s
MAC	t_{SIFS}	SIFS packet duration	10μs
	t_{DIFS}	DIFS packet duration	5μs
	t_{RTS}	RTS packet duration	272μs
	t_{CTS}	CTS packet duration	248μs
	t_{ACK}	ACK packet duration	248μs
	t_{ex}	Data packet expiration timer	5.798ms
	σ	Backoff slot time	20μs
	D	Data rate	2Mbps
Mobile Networks		Mean of speed	20m/s
		Standard deviation of speed	8m/s
SU		Traffic model	Always back-logged
	$t_{H+C,SU}$	Data packet duration	5.44ms
	T_{SW}	Switching delay	100μs
PU		Traffic model	Stochastic channels with Poisson model
	$t_{H+C,PU}$	Data packet duration	5.44ms
	L_{c_i}	PUL of each PU at each available data channel	[0.1,0.9] Default: 0.1

available licensed and orthogonal data channels is limited to K=3 with different carrier frequencies. The SUs transmit using a fixed transmission power at different data channels; hence the transmission range for each data channel varies as shown in Figure 6.6. In mobile networks, the SU host moves in a random direction with its speed following a normal distribution with the given mean and standard deviation; and the SU host changes its direction and speed every second. In static networks, the SU BS could communicate with the SU host using all K=3 data channels; while in mobile network, some of the K=3 data channels may be out of range, however, the SU host must move within the maximum transmission range d_1 from the SU BS in Figure 6.6.

6.6.7 Section Organization

The remainder of this chapter are relevant to simulation experiments, results and discussions, and they are organized as follows:

- Section 6.7 shows the effects of multiple states in RL on network performance.
- Section 6.8 shows the effects of RL parameters on network performance.
- Section 6.9 shows the effects of parameters in Adapt, Win and AdaptWin on network performance.
- Section 6.10 compares RL with Adapt, Win and AdaptWin, as well as the analytical results.

6.7 Effects of Multiple States

6.7.1 Introduction

This section investigates the use of RL for a DCS application that has multiple states that helps SU BS to select heterogeneous data channels opportunistically for data transmission to different SU hosts in static and mobile centralized CR networks. In RL, the state (see Section 5.3.4 on page 71 for more details) encompasses the conditions of the operating environment that are relevant to decision making in an application. Thus, the RL approach in Section 6.4.1 is relaxed with the introduction of an extra SU host; this can be represented using a "state". The DCS scheme selects an available data channel among the licensed channels for data transmission from an SU BS to each SU host. The scenario of the simulation including PU, SU, channel, and CSMA-based cognitive MAC protocol are discussed in Section 6.5.

6.7.1.1 Reinforcement Learning (RL) Approach with State Representation Extension

In this section, the extension to the RL approach in Section 6.4.1 through state representation is discussed. The SU BS keeps track of the learned action value or Q-value, $Q_t(s_t,c_i)$ for all the available data channel C in a Q-table with $|S|\times|C|$ entries. The state $s\in S$ represents the SU hosts associated with the BS. The condition of the state changes with time, for instance, the distance between the SU BS and SU hosts. The Q-value $Q_t(s_t,c_i)$, which represents the knowledge, indicates the appropriateness of choosing data channel c_i by the SU BS to communicate with the SU host s_t in the operating environment. In other words, the Q-value estimates the level of local reward for using a data channel c_i to communicate with SU host s; hence changes in the Q-value will lead to changes in an SU BS's channel selection for each SU host. At each attempt to transmit a data packet to an SU

host s_t, the SU BS chooses a data channel $c_{i,t}$ and receives a local reward $r_{t+1}(c_{i,t})$ at time t+1.

Knowledge Update Procedure During knowledge update, the Q-value of a chosen data channel $c_{i,t}$ for host s_t at time t is updated at time t+1 as follows:

$$Q_{t+1}(s_t, c_{i,t}) \leftarrow (1-\alpha)Q_t(s_t, c_{i,t}) + \alpha r_{t+1}(s_t, c_{i,t}) \tag{6.29}$$

For every successful data packet transmission, there is a reward with positive constant value $r_{t+1}(c_{i,t})$=+RW, otherwise there is a cost with negative constant value $r_{t+1}(c_{i,t})$=-CT. As time goes by, the agent receives a sequence of rewards from the data packet transmission procedure.

Action Selection Procedure During exploitation, the SU BS chooses an exploitation or greedy action, which is the data channel with the highest Q-value, as follows:

$$c_{i,t} = \underset{c_i \in C}{\operatorname{argmax}}\, Q_t(s_t, c_i) \tag{6.30}$$

Two conditions that trigger a channel switch for each state or SU host are similar to the case in Section 6.4.1:

- Switching to a data channel with higher Q-value.
- Exploration.

The RL model with state extension for the DCS scheme is shown in Table 6.3.

6.7.2 Simulation Setup and Parameters

Table 6.2 shows the parameters in the simulation. Table 6.4 shows the addition parameters for the simulation in this section. With N=3, we consider

Table 6.3: RL Model with state extension (SACC) at SU BS for DCS

	Dynamic Channel Selection Model	
	Description	Representation
State	Set of SU hosts associated with the SU BS.	S={s=CR1,CR2,...}
Action	Available data channels for data transmission.	C={c_i=1,2,...,K}
Reward	Constant value to be rewarded/incurred for successful/unsuccessful data packet transmission.	$r_{t+1}(s_t,c_{i,t})$ = $\begin{cases} +RW, & \text{if successful} \\ -CT, & \text{if otherwise} \end{cases}$

a centralized CR network with a single static SU BS, and two static or mobile SU hosts, namely CR1 and CR2, in all scenarios in this section. This is sufficient to show how RL with state representation is applied to DCS. The state represents the SU hosts to which the SU BS wishes to communicate. The condition of the state may change with time, for instance, the distance between the SU BS and SU host changes as the SU host moves.

The parameter values of *RW* and *CT* for RL in Table 6.4 are chosen empirically to achieve the best possible network performance, specifically, throughput. Moderate value of α=0.2 and ε = 0.1 are chosen as the default value for both static and mobile networks. In Section 6.7.3.2 and 6.7.3.3, the effects of α and ε on network performance in static and mobile networks are investigated. The results show that moderate values of α=0.2 and ε=0.1 provide reasonable network performance, hence these values are chosen as the default value for both static and mobile networks.

6.7.3 Simulation Results

Both static and mobile networks are simulated. In the static network, the SU BS perceives similar network performance for using a particular data

Table 6.4: Notations and Default Parameter Settings in Simulation for Investigation into the Effects of Multiple States in Reinforcement Learning on Network Performance

Category	Symbol	Details	Values
Initial ization	N	Number of SU	3 (one SU BS and two SU hosts)
RL		Initial Q-value	1
	α	Learning rate	{0.0125, 0.025, 0.05, 0.1, 0.2, 0.4} Default: 0.2
	ε	Exploration probability	{0.0125, 0.025, 0.05, 0.1, 0.2, 0.4} Default: 0.1
	RW	Reward	15
	CT	Cost	5

channel to transmit to each of the SU hosts. In the mobile network, the SU BS perceives different network performance for using a particular data channel to transmit to the SU hosts because some SU hosts are unreachable for some data channels.

We first compare the RL and Random network performance; followed by investigation into the effects of RL parameters, namely α and ε, on the network performance in static networks, and finally, in mobile networks.

6.7.3.1 Comparison of RL and Random

Figure 6.8 shows the throughput achieved by CR1 and CR2 using the RL and Random scheme for various levels of PUL in static and mobile networks. The RL scheme outperforms the Random scheme for all levels of PUL to provide higher throughput. Both CR1 and CR2 achieve approximately similar individual network performance. Throughput enhancement provided by RL is up to 2.3 and 3.2 times at 0.8 PUL in static and mobile networks respectively. Thus, RL learns well and helps the SU BS

to choose a data channel with low PUL such that the successful data packet transmission rate is high, and so it provides higher throughput. At 0.1 PUL in static networks, throughput enhancement provided by RL is not significant due to the small differences among the Q-values or less differences in the PUL across the available data channels. However, at 0.1 PUL in mobile networks, RL outperforms Random up to 1.7 times because the RL scheme helps the SU BS to choose a data channel with suitable transmission range for data packet transmission to each SU host.

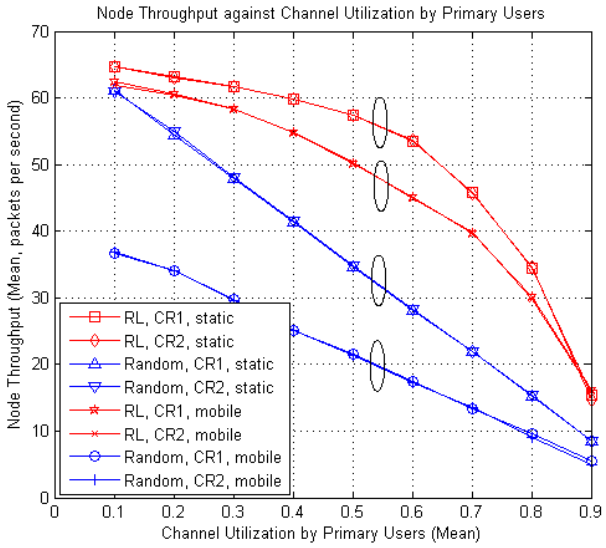


Figure 6.8: The mean throughput at CR1 and CR2 against mean PUL for RL with the state representation extension and for Random in static and mobile networks.

Figure 6.9 shows the number of channel switchings achieved by CR1 and CR2 using the RL and Random scheme for various levels of PUL in static and mobile networks. The RL scheme outperforms the Random scheme for all levels of PUL in providing a lower number of channel switchings. Both CR1 and CR2 achieve approximately similar individual network performance. The RL scheme attains a rather stable number of channel switchings because the ε is kept constant at 0.1 throughout the

simulation; however the number of channel switchings increases at 0.7 PUL in both static and mobile networks. The effect of the number of channel switchings does not affect the throughput significantly due to the low channel switching delay of 100μs; however, this is dependent on the hardware performance in practice that advances as time goes by. As the assumption of a channel switching delay of 100μs is applied in both RL and Random approaches, it is a fair comparison. For Random, the number of channel switchings decreases with PUL, indicating a decreasing number of attempts by the SU BS to transmit data packets. The reason is that failed data packet transmission incurs longer delay while waiting for data packet expiration timer t_{ex} to expire; and this happens more often with increasing PUL. The RL scheme outperforms the Random scheme up to 4.2 times at 0.5 PUL in a static network and up to 4.3 times at 0.3 PUL in a mobile network. For RL, the number of channel switchings is lower in a mobile network compared to a static network. The reason is that, as the SU hosts move further away from the SU BS, the number of channels that fulfill the transmission range requirement decreases, hence the number of channel switchings reduces. For Random, the number of channel switchings is higher for a static network compared to a mobile network, indicating a larger number of attempts to transmit data packets by the SU BS in a static network. RL incurs less delay as the number of channel switchings is smaller.

Figure 6.10 and 6.11 shows the equivalent of Figures 6.8 and 6.9 respectively with linear combination (or sum) of all the local network performance at CR1 and CR2 to provide mean network-wide performance. Only network-wide performance is shown henceforth due to the similarity among the nodal performance at CR1 and CR2.

The next two subsections show the effects of RL parameters including α and ε on network-wide performance with PUL as ordinate.

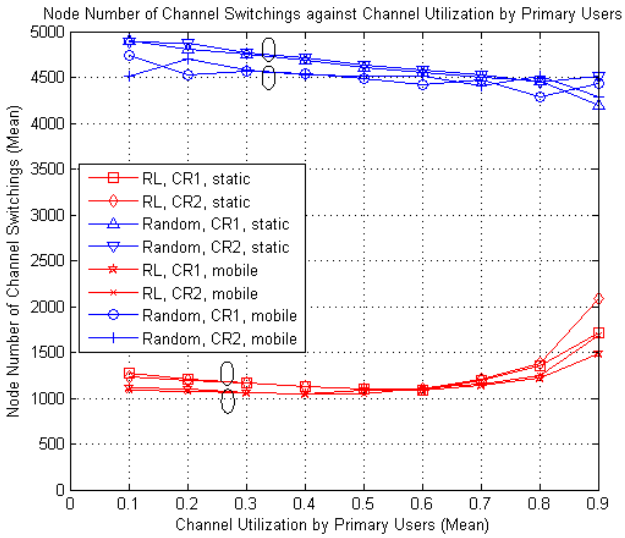


Figure 6.9: The mean number of channel switchings at CR1 and CR2 against mean PUL for RL with the state representation extension and for Random in static and mobile networks.

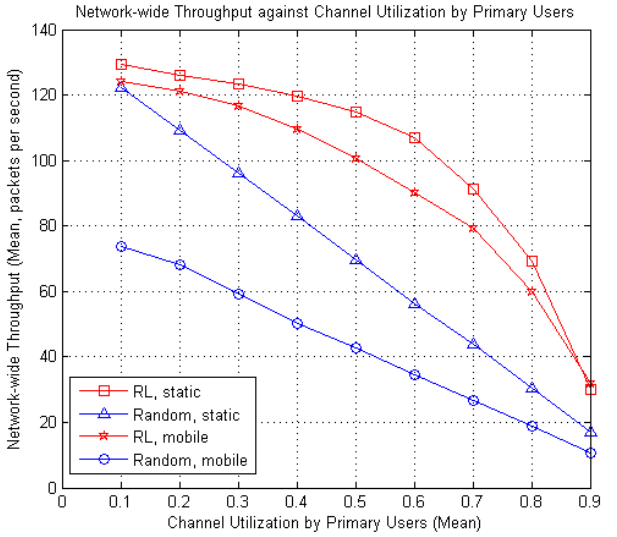


Figure 6.10: The mean network-wide throughput against mean PUL for RL with the state representation extension and for Random in static and mobile networks.

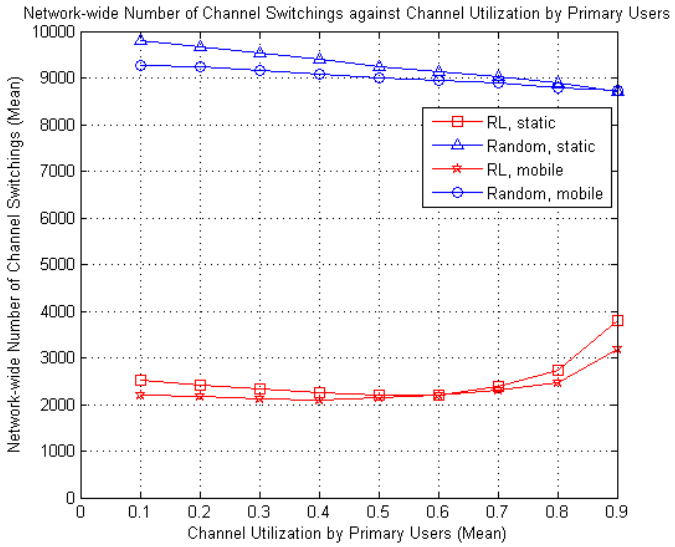
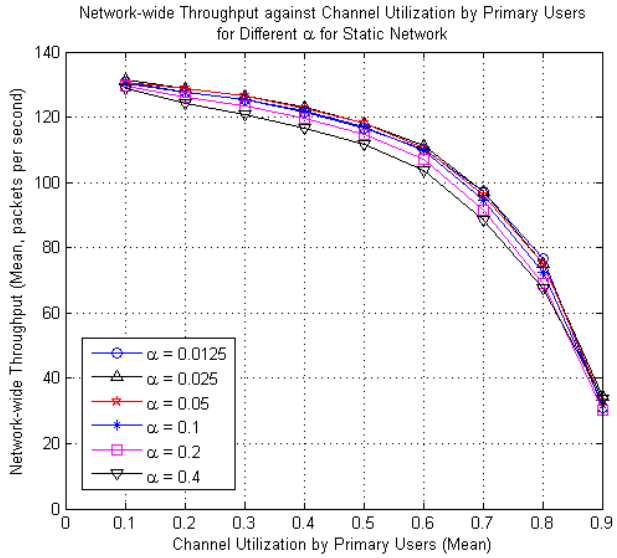


Figure 6.11: The mean network-wide number of channel switchings against mean PUL for RL with the state representation extension and for Random in static and mobile networks.

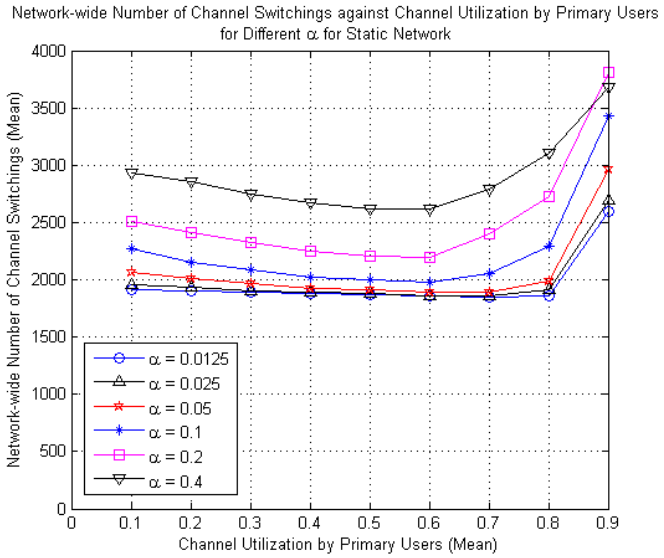
6.7.3.2 Effects of α and ε on Network Performance in Static Network

The throughput and number of channel switchings achieved by RL are investigated for various levels of PUL in static networks. The PER for all data channels is set to 0.1. With PUL as ordinate, Figure 6.12 shows the effect of α, specifically, Figure 6.12a shows the effect of α on throughput; Figure 6.12b shows the effect of α on number of channel switchings. With PUL as ordinate, Figure 6.13 shows the effect of ε, specifically, Figure 6.13a shows the effect of ε on throughput; Figure 6.13b shows the effect of ε on the number of channel switchings.

In Figure 6.12a, it is shown that the value of α does not have a significant effect on throughput. In Figure 6.12b, for each α, the number of channel switchings reaches the lowest value at about 0.6 PUL because the standard deviation between the Q-values is higher at 0.6 PUL. The standard deviation for the PUL is best explained using an example. At 0.2,

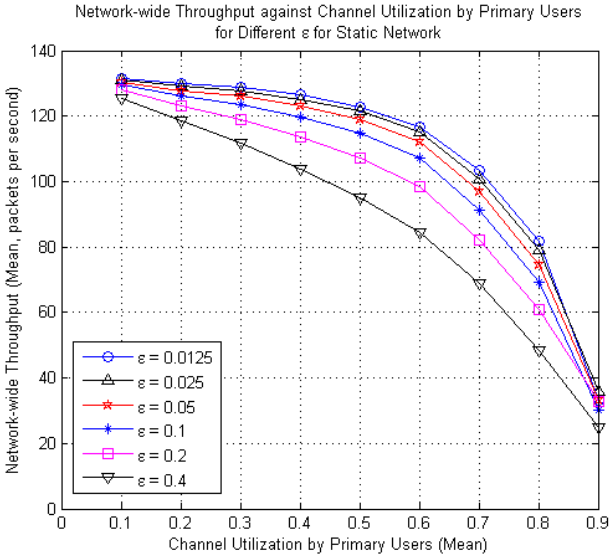


(a) Mean network-wide throughput.

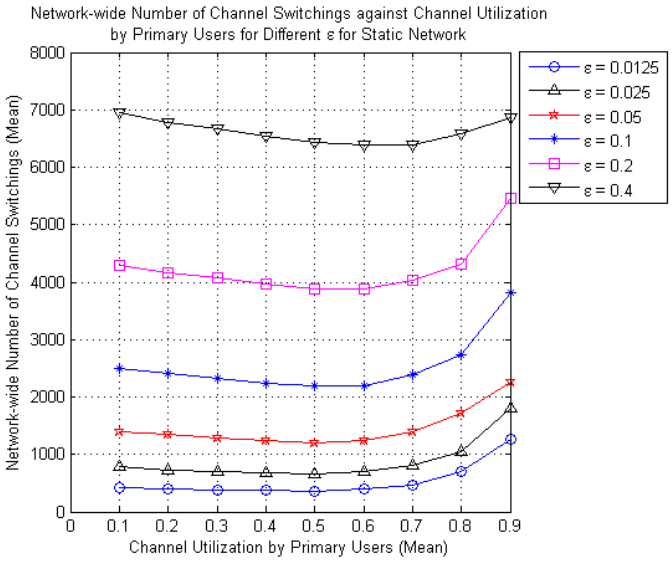


(b) Mean network-wide number of channel switchings.

Figure 6.12: The mean network performance against mean PUL for RL with the state representation extension using different α values in static networks.



(a) Mean network-wide throughput.



(b) Mean network-wide number of channel switchings.

Figure 6.13: The mean network performance against mean PUL for RL with the state representation extension using different ε values in static networks.

the PULs across the three data channels, sorted by increasing standard deviation, are [0.2,0.2,0.2], [0.2,0.3,0.1], [0.3,0,0.3], [0.4,0.1,0.1], [0.2,0,0.4], [0.5,0,0.1], and [0.6,0,0]. At 0.6, higher standard deviation is possible, for instance, [0,0.9,0.9]. Higher standard deviation of PUL leads to more obvious choices of channel selection, for instance, the SU BS chooses channel 1 with no PU activity when the PUL across the data channels is [0,0.9,0.9]. In general, a lower value of α provides a lower number of channel switchings in this scenario.

In Figure 6.13a, the throughput increases as the ε converges to the lowest value or the least exploration. In Figure 6.13b, the number of channel switchings shares the same trend as Figure 6.12b, though the ε results in a larger range in the number of channel switchings. Thus, the ε has greater effect on throughput performance and number of channel switchings compared to α.

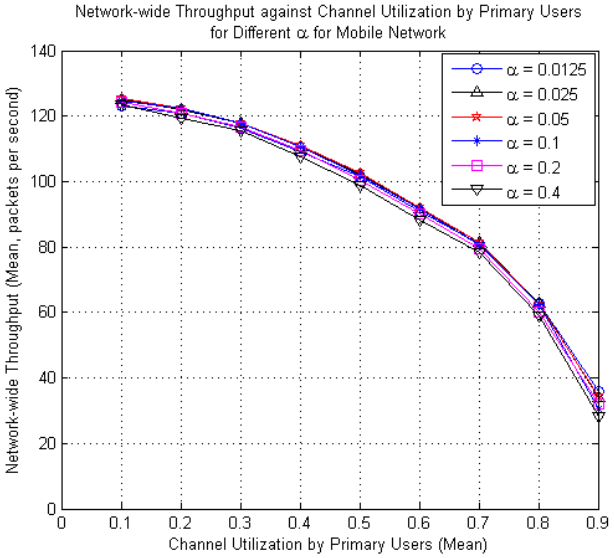
6.7.3.3 Effects of α and ε on Network Performance in Mobile Network

The throughput and number of channel switchings achieved by RL are investigated for various levels of PUL in mobile networks in Figures 6.14 and 6.15. The PER for all data channels is set to 0.1.

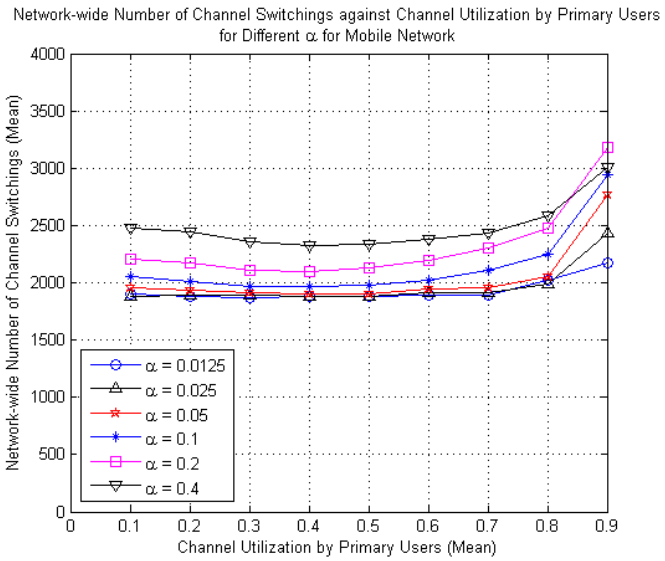
Similar trends are observed in the case of static network in Figures 6.12 and 6.13 although lower throughput and number of channel switchings are observed in mobile networks. This investigation shows that, in addition to PUL and PER, RL helps the SU BS to choose a data channel with suitable transmission range to transmit data packets to different SU hosts because some SU hosts are unreachable using some data channels.

6.7.4 Summary of Research Outcomes

The research outcomes from the investigation on the effects of multiple states in RL on throughput performance and number of channel switchings are summarized as follows:

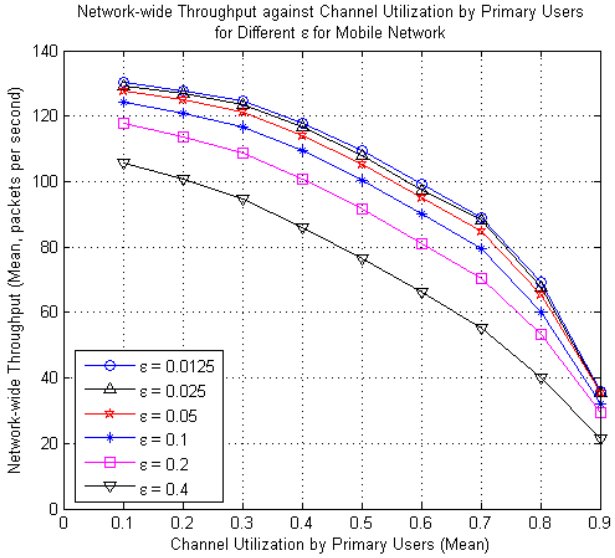


(a) Mean network-wide throughput.

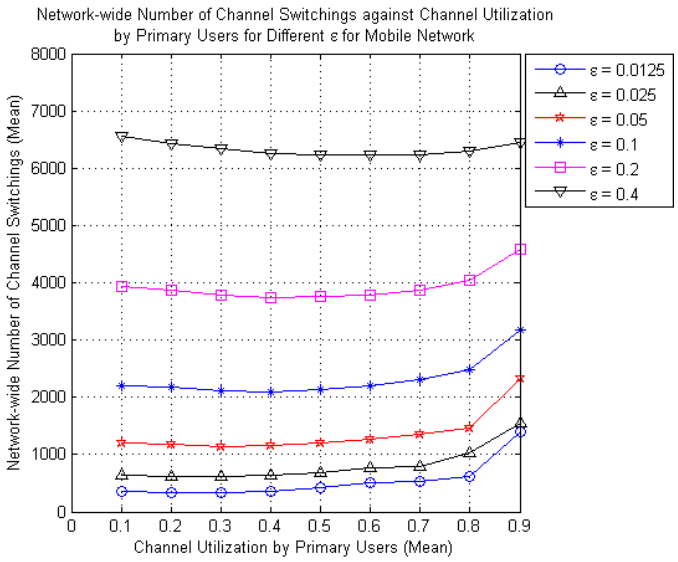


(b) Mean network-wide number of channel switchings.

Figure 6.14: The mean network performance against mean PUL for RL with the state representation extension using different α values in mobile networks.



(a) Mean network-wide throughput.



(b) Mean network-wide number of channel switchings.

Figure 6.15: The mean network performance against mean PUL for RL with the state representation extension using different ε values in mobile networks.

- The RL approach outperforms the Random approach for various levels of PUL in throughput performance and number of channel switchings in static and mobile networks. Hence, the RL approach helps the SU BS to choose a data channel with low PUL and PER, as well as suitable transmission range, in order to maximize the successful data packet transmission rate.

- The states or SU hosts achieve approximately similar individual network performance as expected because the SU BS learns well and might choose different data channel for transmitting data to different SU host.

- The network performance converges to higher throughput and lower number of channel switchings as the values of α and ε decrease to smaller value α=ε=0.0125.

- The exploration probability ε has greater effects on throughput and number of channel switchings when compared to α.

- Similar trends of network performance are observed in static and mobile networks. However, lower throughput and number of channel switchings are observed in mobile networks.

In this section, there are *two* SU hosts, while in Section 6.8, there is a *single* SU host. Similar trends are observed in these two sections. Specifically, the simulation results in Figure 6.10 on the mean network-wide throughput is approximately similar to that in Figure 6.16. Hence, for simplicity, we consider an SU BS and an SU host in the subsequent investigations in Section 6.8 to 6.10.

6.8 Effects of RL Parameters

6.8.1 Introduction

This section investigates the effects of RL parameters on network performance. The simulation results of RL are also compared with that of analytical results, which are derived using Markov chains (see Section 6.5). The RL approach in Section 6.4.1 is applied in this section. We consider a single SU host or state, which is often called stateless or single-state as explained in Section 2.3 on page 14 and Section 5.3 on page 69.

6.8.2 Simulation Setup and Parameters

Table 6.2 shows the general simulation parameters. Table 6.5 shows the addition parameters for the simulation in this section. With N=2, we consider a centralized CR network with a single static SU BS and a single static or mobile SU host in all scenarios in this section. This is sufficient to show the effects of RL parameters on network performance.

The parameter values of *RW* and *CT* for RL in Table 6.5 are chosen empirically to achieve the best possible network performance, specifically, throughput. Moderate value of α=0.2 and ε = 0.1 are chosen as the default value for both static and mobile networks.

6.8.3 Simulation Results

Both static and mobile networks are simulated. The selection of α and ε in RL influences the throughput and the number of channel switchings and the results are shown in Figure 6.16 to 6.19 with respect to PUL, and Figure 6.20 to 6.23 with respect to PER.

Table 6.5: Notations and Default Parameter Settings in Simulation for Investigation into the Effects of Reinforcement Learning Parameters on Network Performance

Category	Symbol	Details	Values
Initial ization	N	Number of SU	2 (one SU BS and one SU host)
RL		Initial Q-value	1
	α	Learning rate	{0.0125, 0.025, 0.05, 0.1, 0.2, 0.4} Default: 0.2
	ε	Exploration probability	{0.0125, 0.025, 0.05, 0.1, 0.2, 0.4} Default: 0.1
	RW	Reward	15
	CT	Cost	5

6.8.3.1 Effects of α and ε on Network Performance in Static and Mobile Networks with respect to Primary User Utilization Level

Figure 6.16 shows that the value of α does not have a significant effect on throughput in static and mobile networks with respect to PUL. Hence, values of α in the range $0.0125 \leqslant \alpha \leqslant 0.4$ enables RL to learn well and help the SU to choose a channel with low PUL and suitable transmission range such that successful data packet transmission rate is high.

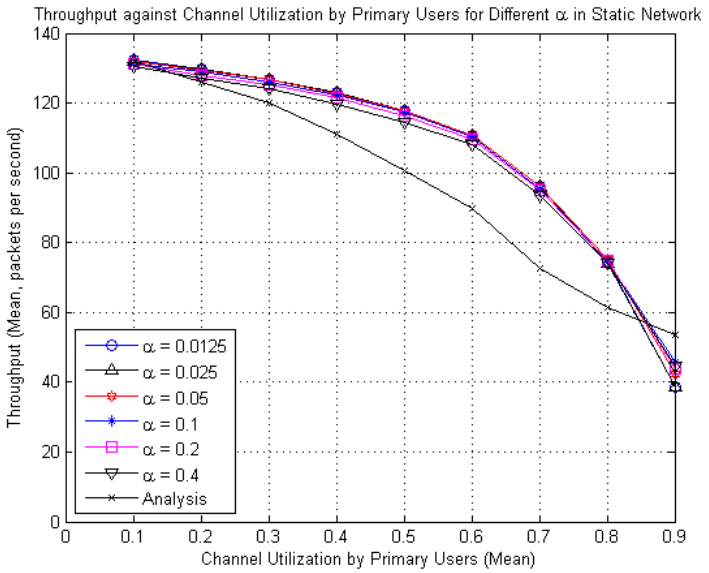
Values of α in the range $0.0125 \leqslant \alpha \leqslant 0.4$ achieves the expected throughput provided by the Analysis. In the analytical model, without using a learning mechanism, the SU BS is expected to provide the best possible throughput in the presence of different levels of PUL. The next channel is chosen randomly as long as the data packet transmission is successful as shown in Equation (6.25). Whenever the SU BS chooses a channel with higher PUL, it tends to switch its channel. Since the SU BS never learns due to the memoryless property of Markov chain, it may choose a channel with high PUL more often, resulting in high number of channel switches

that incur time and hence causing lower throughput performance. In the learning-based RL, the SU BS chooses the next best channel with lower PUL; hence there are lower number of channel switches leading to higher throughput. However, at 0.9 PUL in static networks and 0.1, 0.2 and 0.9 PUL in mobile networks, throughput achieved by RL is lower than Analysis. This is because the Q-values of all the channels become similar at 0.1, 0.2 and 0.9 PUL, and learning is difficult at these PULs in RL.

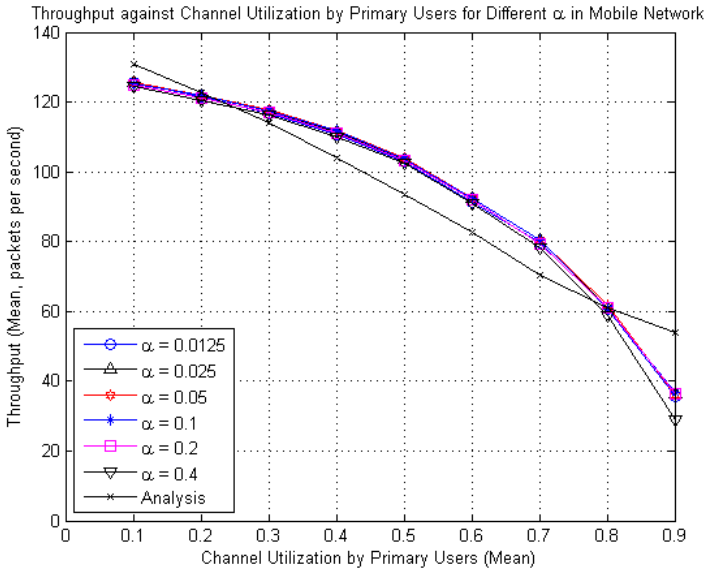
Figure 6.17 shows the effects of α on the number of channel switchings in static and mobile networks with respect to PUL. The smallest value (α=0.0125) provides the lowest number of channel switchings. For each α, the number of channel switchings reaches the lowest value at about 0.5 PUL because of the higher standard deviation of PUL (see Section 6.7.3.2 for explanation on standard deviation of PUL). In Figure 6.17b, the number of channel switchings is variable at 0.9 PUL because the Q-values of all the channels become similar and learning is difficult in RL. In some cases, all the Q-values converge to -*CT* thus no channel switching is performed and this reduces the number of channel switchings; however, in some cases, the Q-values oscillate, and this increases the number of channel switchings because the RL always chooses the greedy action i.e., the channel that has the highest Q-value. To improve the stability, the RL can switch its channel only when the difference of the Q-values among the channels is greater than a certain threshold.

Figure 6.18 shows the effects of ε on throughput in static and mobile networks with respect to PUL. When $\varepsilon \leqslant 0.1$, the throughput is higher than the Analysis for most values of PUL in both static and mobile networks. The throughput increases as the ε converges to the lowest value at $\varepsilon \leqslant 0.0125$ or the least exploration. The effect of ε on the throughput is more significant than is α.

Figure 6.19 shows the effects of ε on the number of channel switchings in static and mobile networks with respect to PUL. Similar trends are observed in the case of α in Figure 6.17; however, the ε results in a larger

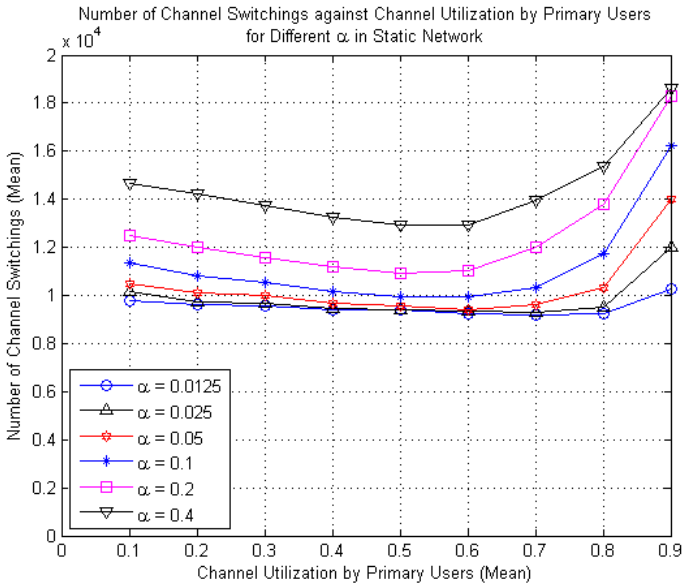


(a) Static network.

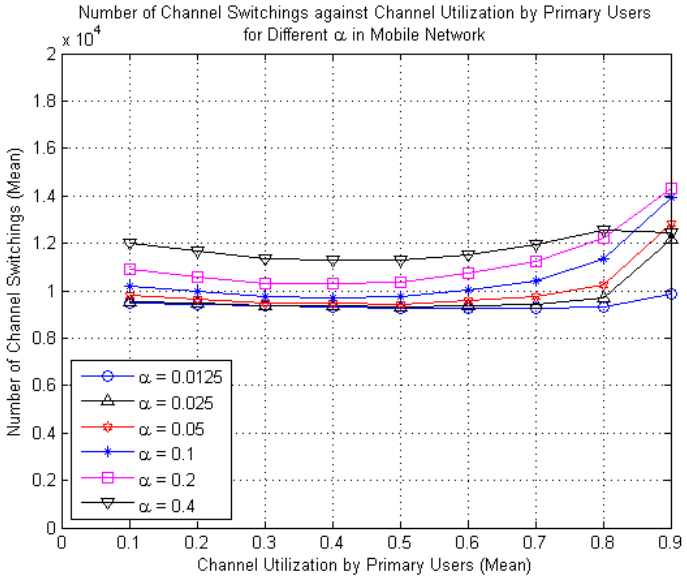


(b) Mobile network.

Figure 6.16: The mean throughput of an SU BS against mean PUL for RL with different α values and for the Analysis. PER for all data channels is set to 0.1. ε is set to 0.1.

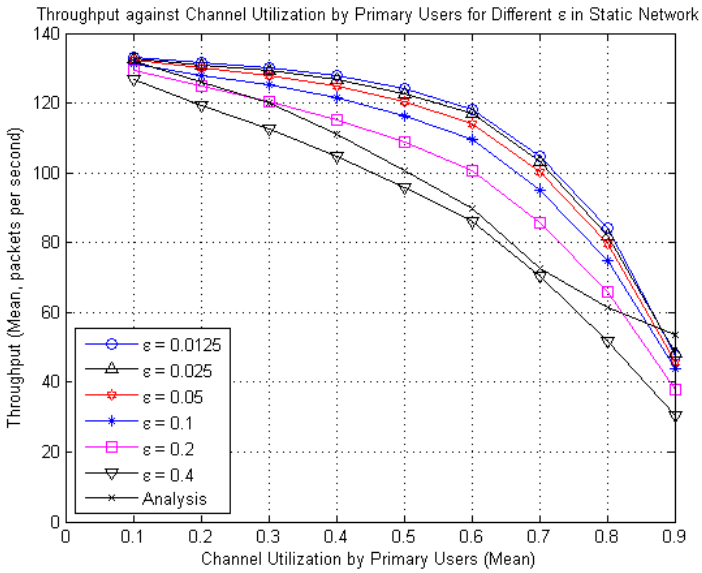


(a) Static network.

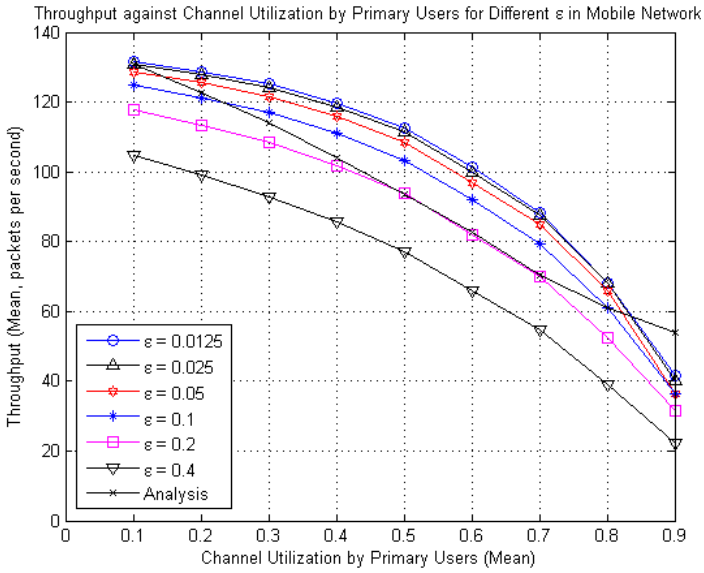


(b) Mobile network.

Figure 6.17: The mean number of channel switchings of an SU BS against mean PUL for RL with different α values. PER for all data channels is set to 0.1. ε is set to 0.1.



(a) Static network.



(b) Mobile network.

Figure 6.18: The mean throughput of an SU BS against mean PUL for RL with different ε values and for the Analysis. PER for all data channels is set to 0.1. α is set to 0.2.

range in the number of channel switchings. Thus, the ε has greater effect on network performance than does α.

6.8.3.2 Effects of α and ε on Network Performance in Static and Mobile Networks with respect to Packet Error Rate

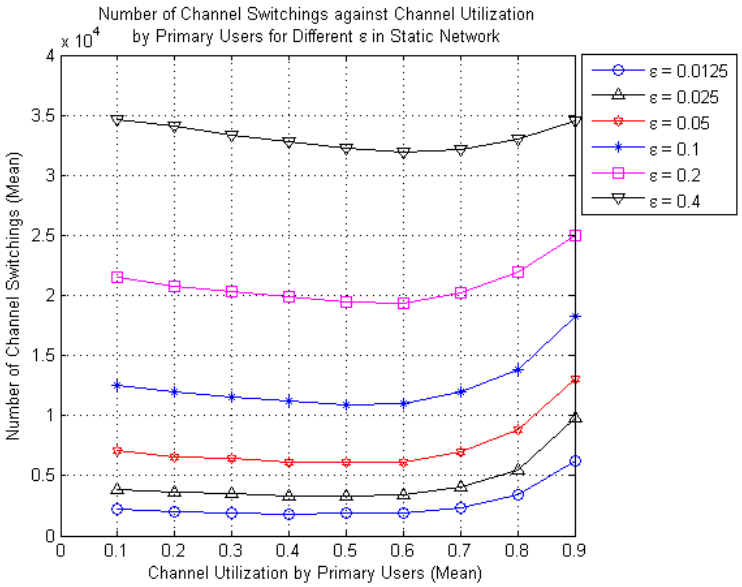
The selection of α and ε in RL influences the throughput and the number of channel switchings and its results with respect to PER are shown in Figure 6.20 to 6.23.

Similar trends are observed for network performance with respect to PUL in Figure 6.16 to 6.19. Figure 6.20 shows that the value of α does not have a significant effect on throughput in static and mobile networks with respect to PER. Figure 6.21 shows that smallest value of α=0.0125 provides the lowest number of channel switchings. Figure 6.22 shows that when $\varepsilon \leqslant 0.1$, the throughput is higher than the Analysis for most values of PER in both static and mobile networks, and the throughput increases as the ε converges to the lowest value at $\varepsilon \leqslant 0.0125$. Figure 6.23 shows that similar trends on the effects of ε on the number of channel switchings are observed in the case of α in Figure 6.21; however, the ε results in a larger range in the number of channel switchings.

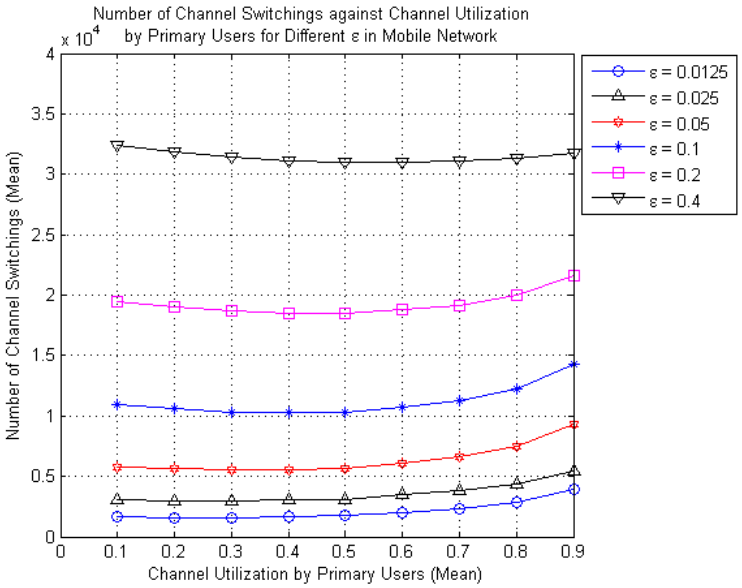
6.8.4 Summary of Research Outcomes

The research outcomes from the investigation on the effects of RL parameters, namely α and ε, on throughput performance and number of channel switchings are summarized as follows:

- The RL approach achieves the expected throughput provided by the analytical results in most of the cases.
- The network performance converges to higher throughput performance and lower number of channel switchings as the values of α and ε decrease.

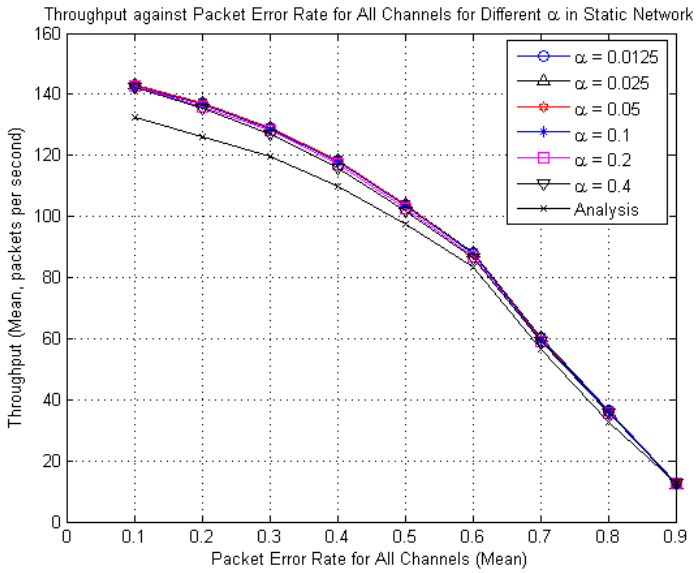


(a) Static network.

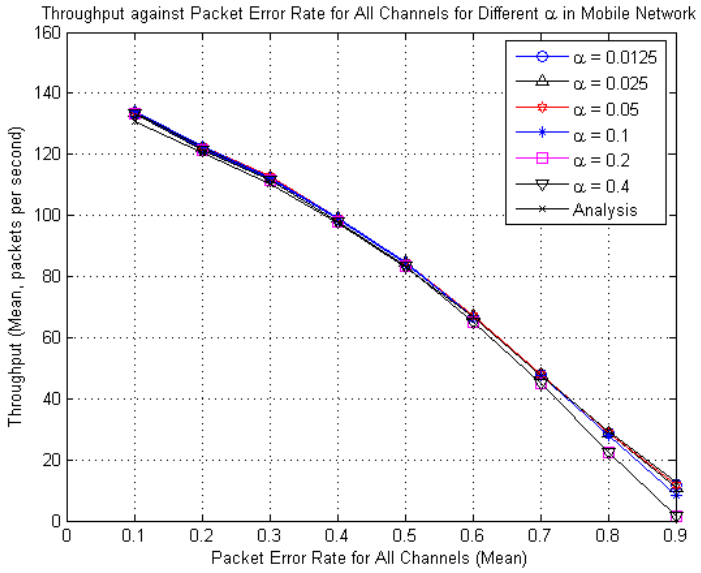


(b) Mobile network.

Figure 6.19: The mean number of channel switchings of an SU BS against mean PUL for RL with different ε values. PER for all data channels is set to 0.1. α is set to 0.2.

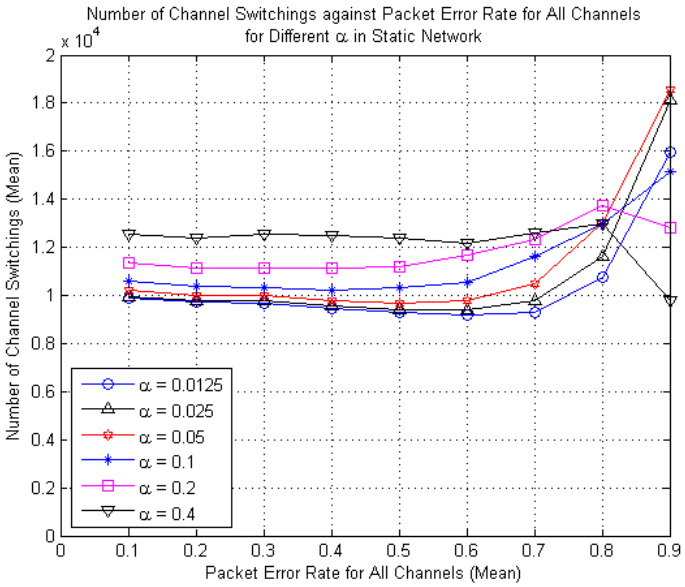


(a) Static network.

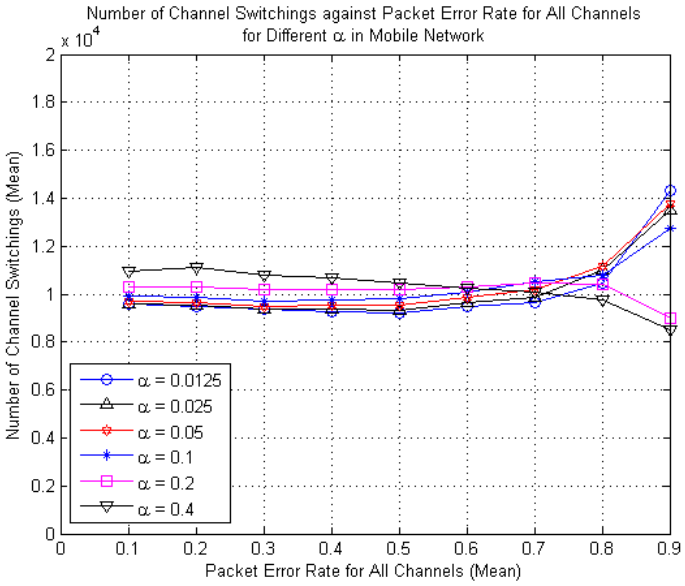


(b) Mobile network.

Figure 6.20: The mean throughput of an SU BS against mean PER for RL with different α values and for the Analysis. PUL for all data channels is set to 0.1. ε is set to 0.1.

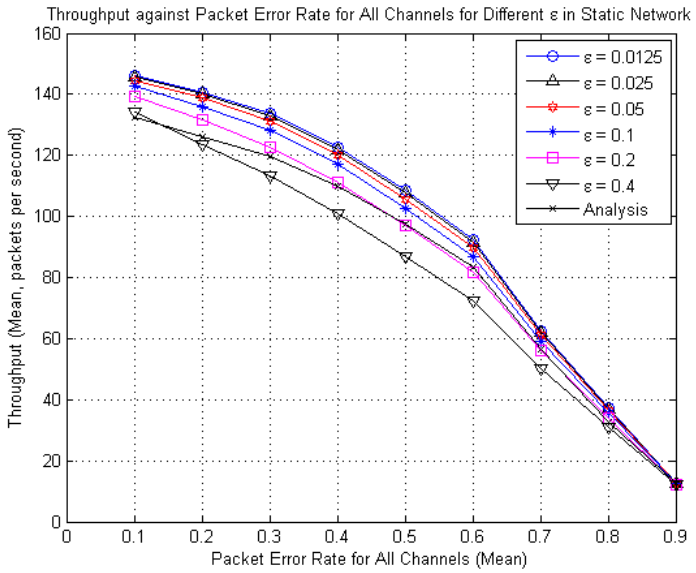


(a) Static network.

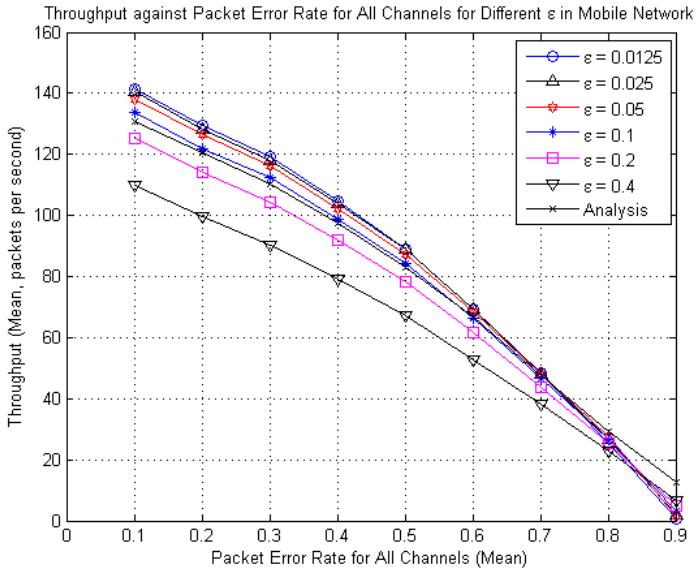


(b) Mobile network.

Figure 6.21: The mean number of channel switchings of an SU BS against mean PER for RL with different α values. PUL for all data channels is set to 0.1. ε is set to 0.1.

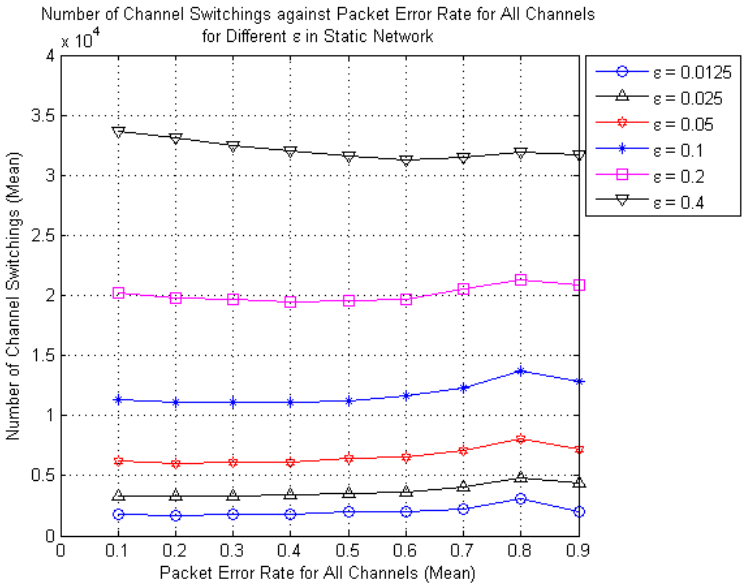


(a) Static network.

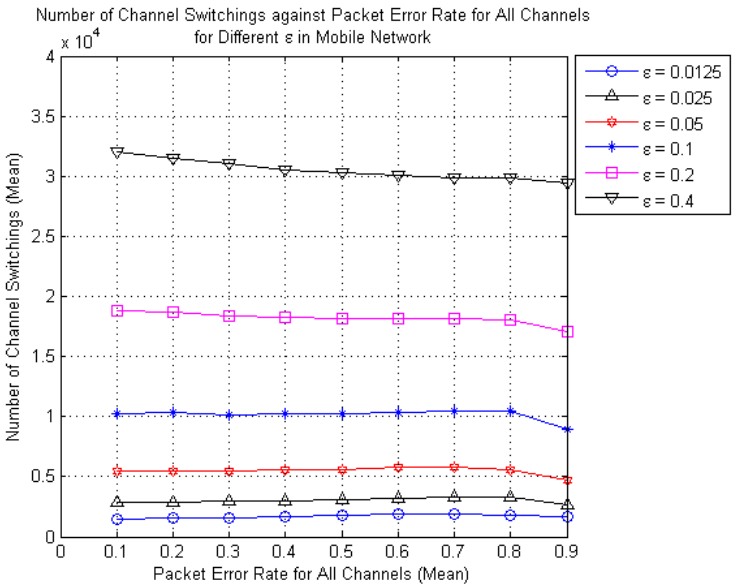


(b) Mobile network.

Figure 6.22: The mean throughput of an SU BS against mean PER for RL with different ε values and for the Analysis. PUL for all data channels is set to 0.1. α is set to 0.2.



(a) Static network.



(b) Mobile network.

Figure 6.23: The mean number of channel switchings of an SU BS against mean PER for RL with different ε values. PUL for all data channels is set to 0.1. α is set to 0.2.

- The ε value has greater effect on the throughput performance and the number of channel switchings than does α.
- The value of ε must be lower than a certain threshold to achieve the expected throughput provided by the analytical results. For instance, with respect to PUL, $\varepsilon \leqslant 0.1$ in static network and at $\varepsilon \leqslant 0.05$ in mobile network.
- Similar trends of network performance are observed in simulations among static and mobile networks, and PUL and PER as ordinates. However, lower throughput and number of channel switchings are observed in mobile networks.

6.9 Effects of Learning Mechanisms Parameters

6.9.1 Introduction

This section investigates the effects of various learning mechanism parameters, specifically $n^{\mathbf{Adapt}}$ in the Adapt, as well as $n^{\mathbf{Win}}$ in Win and AdaptWin, on throughput performance. The purpose is to obtain the learning mechanisms parameters that provide the best possible network performance for comparison with the RL approach in Section 6.10. The learning mechanisms are presented in Section 6.4.2 to 6.4.4. Similar to Section 6.8, we consider a single SU host or state, which is often called stateless or single-state as explained in Section 2.3 on page 14 and Section 5.3 on page 69.

6.9.2 Simulation Setup and Parameters

Table 6.2 shows the parameters for the simulation. Table 6.6 shows the addition parameters in the simulation in this section. With N=2, we consider a centralized CR network with a single static SU BS and a single static or

Table 6.6: Notations and Default Parameter Settings in Simulation for Investigation into the Effects of Adapt, Win and AdaptWin Parameters on Network Performance

Category	Symbol	Details	Values
Initial ization	N	Number of SU	2 (one SU BS and one SU host)
	ε	Exploration probability	0.1
Adapt	$n^{\mathbf{Adapt}}$	Number of consecutive failed data packet transmissions	{1, 2, 4, 8, 16, 32}
Win	$n^{\mathbf{Win}}$	Window size	{1, 2, 4, 8, 16, 32}
AdaptWin	$n^{\mathbf{Adapt}}$		2
	$n^{\mathbf{Win}}$		{1, 2, 4, 8, 16, 32}

mobile SU host in all scenarios in this section. This is sufficient to show the effects of Adapt, Win and AdaptWin parameters on network performance. The scenario of the simulation including PU, SU, channel, and CSMA-based cognitive MAC protocol are discussed in Section 6.5.

6.9.3 Simulation Results

Both static and mobile networks are simulated. The effects of parameter $n^{\mathbf{Adapt}}$ in Adapt are shown in Figures 6.24 and 6.25; of $n^{\mathbf{Win}}$ in Win are shown in Figures 6.26 and 6.27; and of $n^{\mathbf{Win}}$ in AdaptWin are shown in Figures 6.28 and 6.29.

6.9.3.1 Effects of $n^{\mathbf{Adapt}}$ in Adapt on Network Performance

The effects of $n^{\mathbf{Adapt}}$ on throughput with respect to PUL and PER in mobile networks is shown in Figure 6.24 and 6.25 respectively. Figure 6.24 shows that $n^{\mathbf{Adapt}}$=2 provides the highest level of throughput for all lev-

els of PUL. Figure 6.25 shows that $n^{\mathbf{Adapt}}$=1 provides the highest level of throughput from 0.1 to 0.3 PER, followed by $n^{\mathbf{Adapt}}$=2 from 0.4 to 0.6 PER, and followed by $n^{\mathbf{Adapt}}$=4 from 0.7 to 0.9 PER. The effects of $n^{\mathbf{Adapt}}$ on throughput is not significant for various levels of PUL and PER in static networks; so their graphs are not shown. Hence, $n^{\mathbf{Adapt}}$=2 provides the best possible throughput with respect to PUL and PER in most cases in static and mobile networks because it is more adaptive to the changes in the amount of white spaces, or PUL that applies the Poisson process model, and also PER in each data channel.

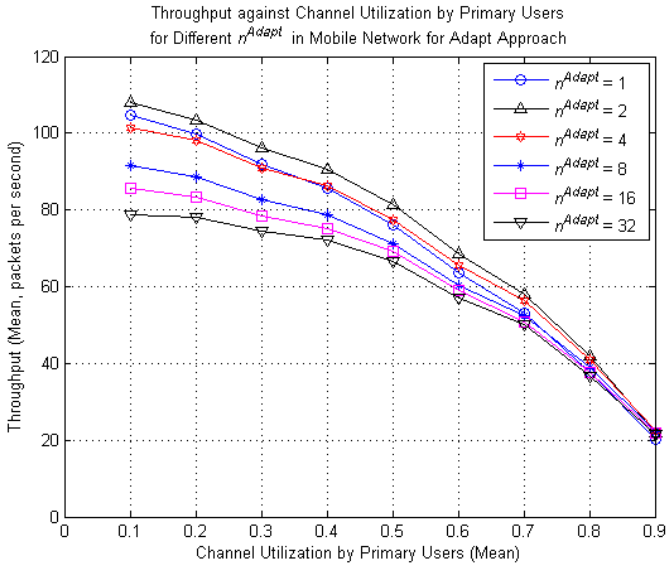


Figure 6.24: The mean throughput of an SU BS against mean PUL for Adapt with different $n^{\mathbf{Adapt}}$ values in mobile networks. PER for all data channels is set to 0.1.

6.9.3.2 Effects of $n^{\mathbf{Win}}$ in Win on Network Performance

The effects of $n^{\mathbf{Win}}$ on throughput with respect to PUL in static and mobile networks are shown in Figure 6.26. Figure 6.26a shows that $n^{\mathbf{Win}}$ = 8 provides the highest level of throughput at 0.1 PUL, and followed by $n^{\mathbf{Win}}$=32

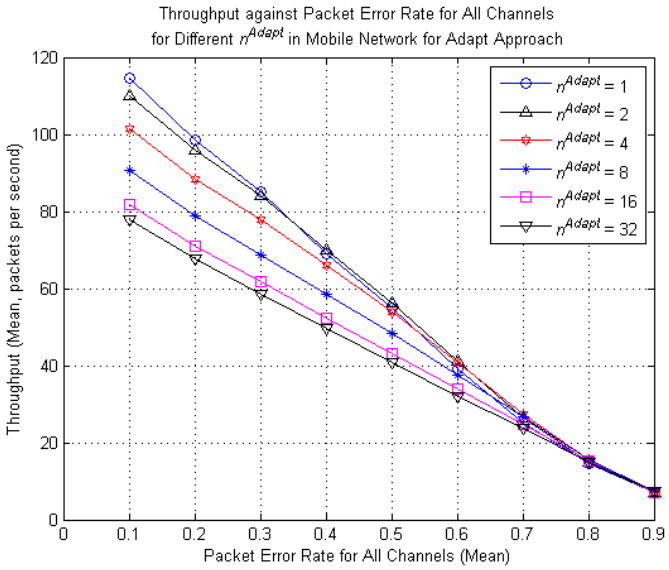
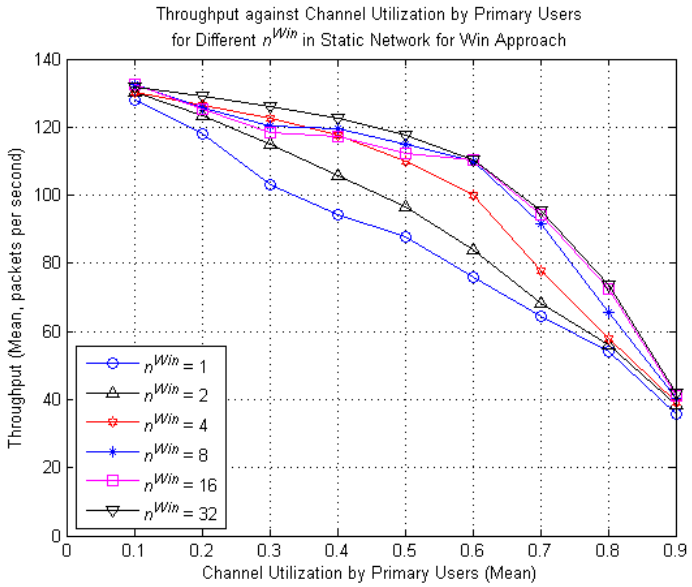


Figure 6.25: The mean throughput of an SU BS against mean PER for Adapt with different $n^{\mathbf{Adapt}}$ values in mobile networks. PUL for all data channels is set to 0.1.

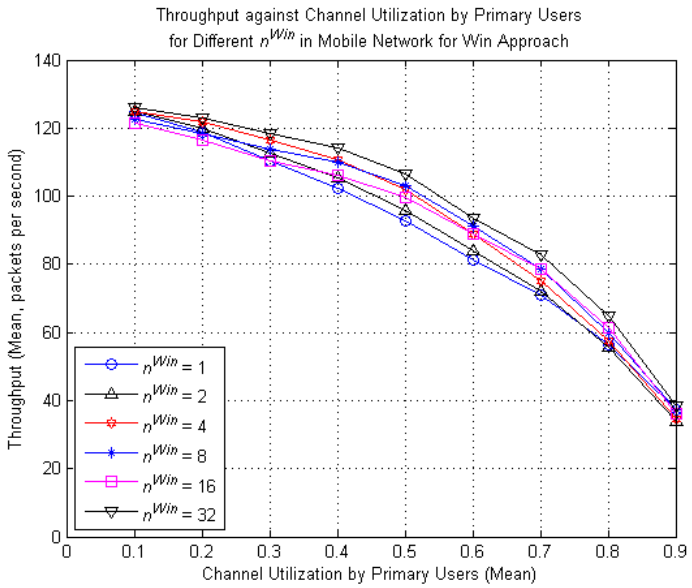
from 0.2 to 0.9 PUL in static networks. Figure 6.26b shows that $n^{\mathbf{Win}}$=32 provides the highest level of throughput for all levels of PUL in mobile networks.

The effects of $n^{\mathbf{Win}}$ on throughput with respect to PER in static and mobile networks are shown in Figure 6.27. Figure 6.27a shows that $n^{\mathbf{Win}}$=8 provides the highest level of throughput at 0.1 PER, and followed by $n^{\mathbf{Win}}$=32 from 0.2 to 0.9 PER in static networks. Figure 6.27b shows that $n^{\mathbf{Win}}$=32 provides the highest level of throughput for all levels of PER in mobile networks.

Hence, window size $n^{\mathbf{Win}}$=32 provides the best possible throughput in most cases because a higher number of most recent attempts of data packet transmissions (or historical information) are applied to compute the probability of successful data packet transmission, P_{S,c_i}^{Win}.

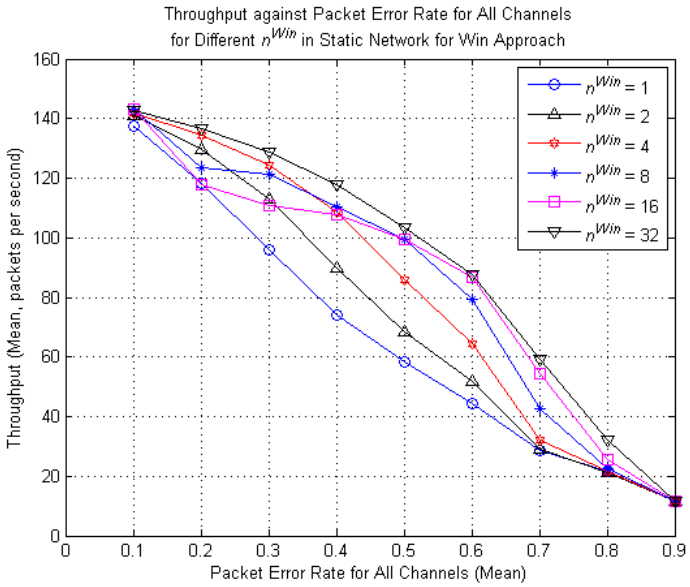


(a) Static network.

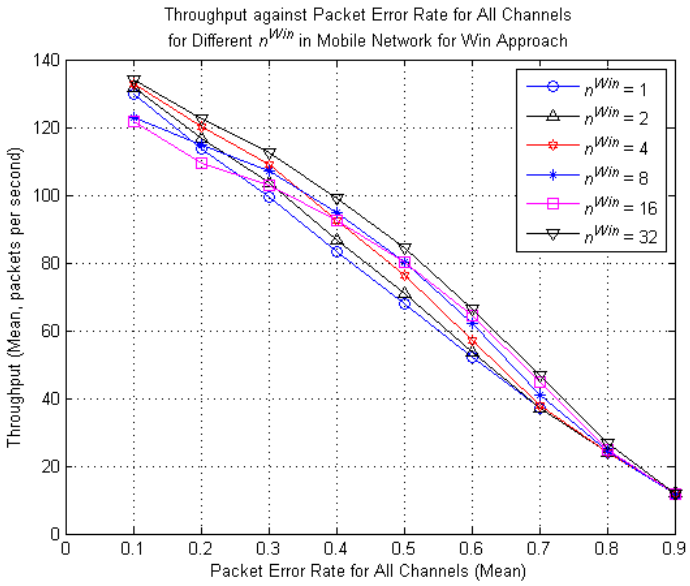


(b) Mobile network.

Figure 6.26: The mean throughput of an SU BS against mean PUL for Win with different $n^{\mathbf{Win}}$ values in static and mobile networks. PER for all data channels is set to 0.1.



(a) Static network.



(b) Mobile network.

Figure 6.27: The mean throughput of an SU BS against mean PER for Win with different $n^{\mathbf{Win}}$ values in static and mobile networks. PUL for all data channels is set to 0.1.

6.9.3.3 Effects of $n^{\mathbf{Win}}$ in AdaptWin on Network Performance

We set $n^{\mathbf{Adapt}}$=2 as it provides the best possible network performance in the Adapt approach. The effects of $n^{\mathbf{Win}}$ on throughput with respect to PUL and PER in static networks is shown in Figure 6.28 and 6.29 respectively. Figure 6.28 shows that $n^{\mathbf{Win}}$=8 provides the highest level of throughput for 0.1 PUL, followed by $n^{\mathbf{Win}}$=32 from 0.2 to 0.8 PUL, and followed by $n^{\mathbf{Win}}$=16 for 0.9 PUL in static networks. Figure 6.29 shows that $n^{\mathbf{Win}}$=8 provides the highest level of throughput for 0.1 PER, followed by $n^{\mathbf{Win}}$=32 from 0.2 to 0.8 PER, and followed by $n^{\mathbf{Win}}$=16 for 0.9 PER. The effects of $n^{\mathbf{Win}}$ on throughput is not significant for various levels of PUL and PER in mobile networks; so their graphs are not shown. Hence, window size $n^{\mathbf{Win}}$=32 provides the best possible throughput in most cases.

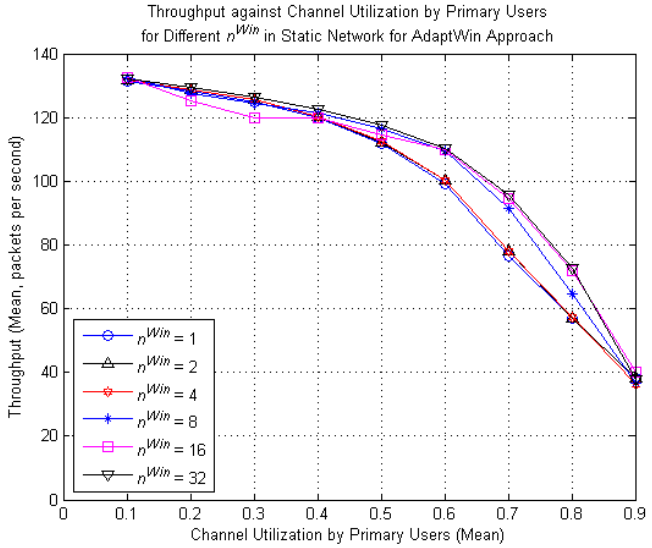


Figure 6.28: The mean throughput of an SU BS against mean PUL for AdaptWin with different $n^{\mathbf{Win}}$ values in static networks. PER for all data channels is set to 0.1. $n^{\mathbf{Adapt}}$ is set to 2.

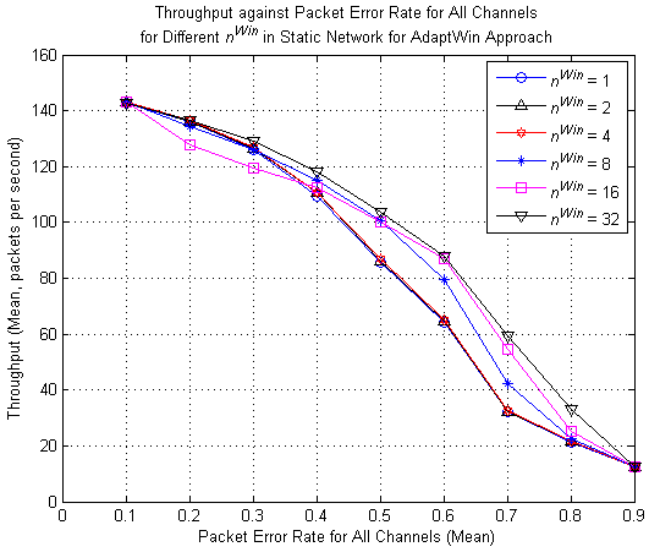


Figure 6.29: The mean throughput of an SU BS against mean PER for AdaptWin with different $n^{\mathbf{Win}}$ values in static networks. PER for all data channels is set to 0.1. $n^{\mathbf{Adapt}}$ is set to 2.

6.9.4 Summary of Research Outcomes

The research outcomes from the investigation on the effects of various learning mechanisms parameters, specifically $n^{\mathbf{Adapt}}$ in Adapt, as well as $n^{\mathbf{Win}}$ in Win and AdaptWin, on throughput performance are summarized. Using the following parameters, the learning mechanisms achieve the best possible throughput performance in static and mobile networks:

- $n^{\mathbf{Adapt}}$ = 2 in the Adapt approach.
- $n^{\mathbf{Win}}$ = 32 in the Win approach.
- $n^{\mathbf{Adapt}}$ = 2 and $n^{\mathbf{Win}}$ = 32 in the AdaptWin approach.

The values of the parameters of $n^{\mathbf{Adapt}}$ in Adapt, as well as $n^{\mathbf{Win}}$ in Win and AdaptWin, that provide the best possible throughput performance are applied for comparison with the RL approach in Section 6.10.

6.10 Comparison of Learning Mechanisms

6.10.1 Introduction

This section compares the network peformance achieved by RL, Adapt, Win, AdaptWin and the analytical results (Analysis). The best possible parameters for the approaches are adopted for comparison. Section 6.8 shows that the best possible throughput is achieved with α=0.0125 for RL in static networks, α=0.05 for RL in mobile networks. Section 6.9 shows that the best possible throughput is achieved with $n^{\mathbf{Adapt}}$=2 for Adapt, $n^{\mathbf{Win}}$=32 for Win, and $n^{\mathbf{Adapt}}$=2 and $n^{\mathbf{Win}}$=32 for AdaptWin in static and mobile networks. Similar to Section 6.8 and 6.9, we consider a single SU host or state, which is often called stateless or single-state as explained in Section 2.3 on page 14 and Section 5.3 on page 69.

6.10.2 Simulation Setup and Parameters

Table 6.2 shows the parameters for the simulation. Table 6.7 shows the addition parameters in the simulation in this section. With N=2, we consider a centralized CR network with a single static SU BS and a single static or mobile SU host in all scenarios in this section. This is sufficient to show the effects of Adapt, Win and AdaptWin parameters on network performance.

6.10.3 Simulation Results

Both static and mobile networks are simulated. We first compare the network performance of RL, Adapt, Win, AdaptWin, Random and Analysis in static and mobile networks with respect to PUL; followed by PER.

6.10.3.1 Comparison of All Learning Mechanisms with respect to PUL

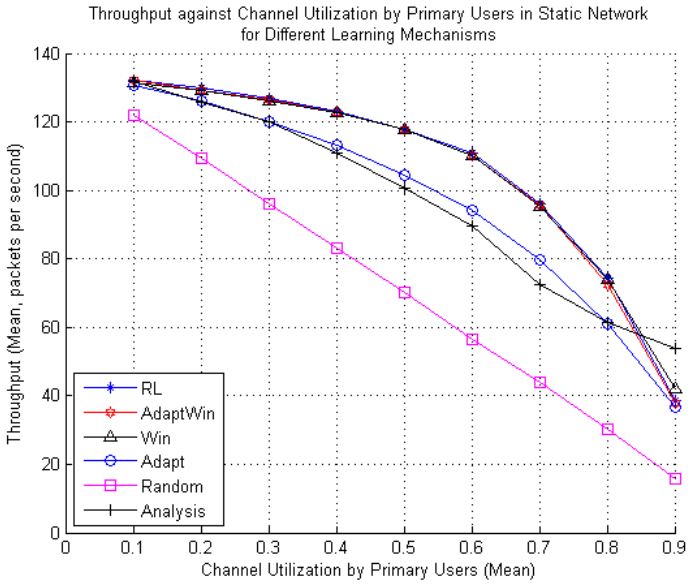
Figure 6.30 shows the throughput achieved by RL, Adapt, Win, AdaptWin, Random and Analysis with respect to PUL in static and mobile network-

Table 6.7: Notations and Default Parameter Settings in Simulation for Comparison of Network Performance Achieved by the RL, Adapt, Win and AdaptWin Approaches

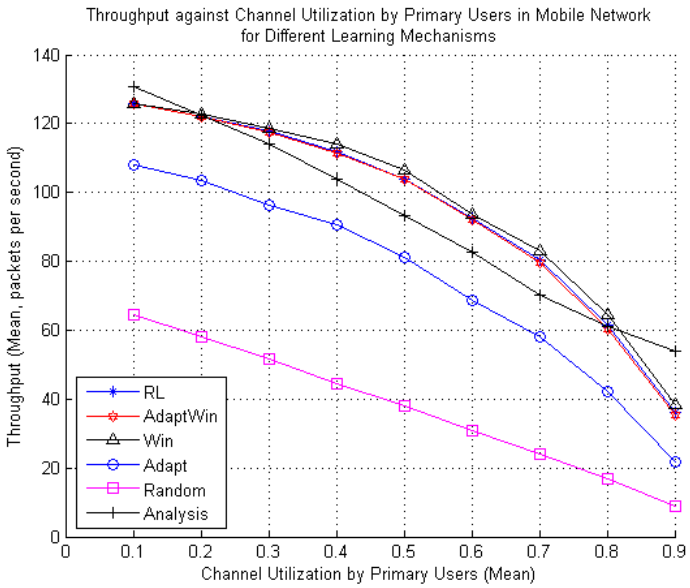
Category	Symbol	Details	Values
Initial ization	N	Number of SU	2 (one SU BS and one SU host)
	ε	Exploration probability	0.1
RL		Initial Q-value	1
	α	Learning rate	α=0.0125 for static networks; α=0.05 for mobile networks
	RW	Reward	15
	CT	Cost	5
Adapt	$n^{\mathbf{Adapt}}$	Number of consecutive failed data packet transmissions	2
Win	$n^{\mathbf{Win}}$	Window size	32
AdaptWin	$n^{\mathbf{Adapt}}$		2
	$n^{\mathbf{Win}}$		32

s. In general, RL, AdaptWin and Win achieve approximately similar individual network performance, which is the highest among the schemes studied, followed by Adapt, and finally Random. In Figure 6.30a, at 0.5 PUL in static networks, the RL, AdaptWin and Win approaches provide approximately 1.7 times, and Adapt provides approximately 1.5 times throughput enhancement in comparison with Random. In Figure 6.30b, at 0.5 PUL in mobile networks, the RL, AdaptWin and Win approaches provide approximately 2.7 times, and Adapt provides approximately 2.1 times throughput enhancement in comparison with Random. At 0.1 PUL in static networks, throughput enhancement provided by all the approaches and Analysis is not significant due to the small differences among the Q-values or less differences in the PUL across the available data channels. However, at 0.1 PUL in mobile networks, RL outperforms Random up to 1.92 times because the RL scheme helps the SU BS to choose a data channel with suitable transmission range for data packet transmission. In comparison with Analysis, RL performs better for all PULs except at 0.1 and 0.9 in static and mobile networks because of the small differences among the Q-values of all data channels. It should be noted that RL, Adapt, Win and AdaptWin choose the next best data channel based on the respective learning mechanisms during channel switching; while in the Analysis, the next data channel is chosen randomly as long as the data packet transmission is successful as shown in Equation (6.7) and (6.25). In short, the RL, Win and AdaptWin approaches learn well and help the SU BS to choose a data channel with low PUL and suitable transmission range such that successful data packet transmission rate is high. They achieve the expected throughput provided by the Analysis. The simulation results for RL and analytical results used to plot Figure 6.30 are analyzed next based on Figure 6.31.

In Figure 6.31, given a mean value of the PUL equal to 0.2, the throughput of the RL, Random and Analysis is investigated for various levels of standard deviation (see Section 6.7.3.2 for explanation on standard devi-



(a) Static network.



(b) Mobile network.

Figure 6.30: The mean throughput of an SU BS against mean PUL for RL, AdaptWin, Win, Adapt, Random and Analysis in static and mobile networks. PER for all data channels is set to 0.1. ε is set to 0.1.

ation of PUL) for the PUL in static and mobile networks. The Random scheme attains a rather stable, slightly decreasing throughput. The low level of throughput for the Random in mobile networks is expected as shown in Figure 6.30. The throughput of both RL and Analysis increases with the standard deviation of PUL. When the standard deviation of the PUL is greater than 0.18 and 0.14 in static and mobile networks respectively, the Analysis provides higher throughput. The reason for the trend is because the higher standard deviation of PUL leads to more obvious choices of channel selection, for instance, the SU BS chooses data channel 2 with no PU activity when the PUL across the data channels is [0.2,0,0.4]. However, in RL, exploration is still performed with ε = 0.1, thus lower throughput is achieved by using RL.

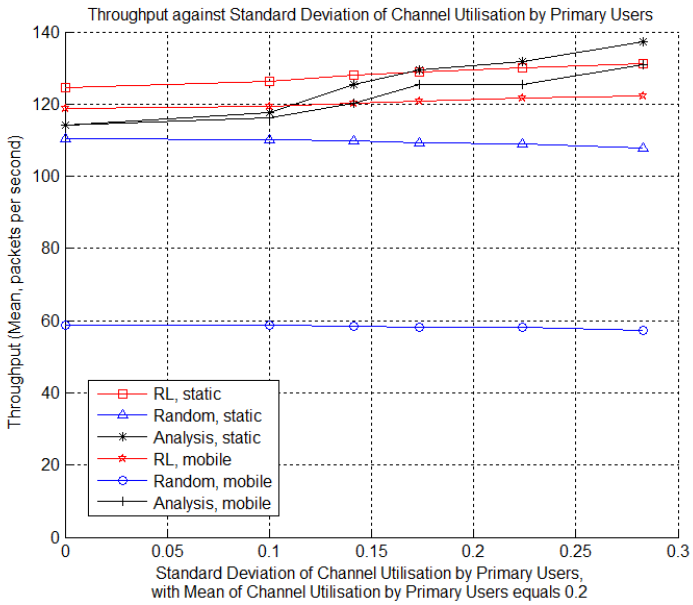


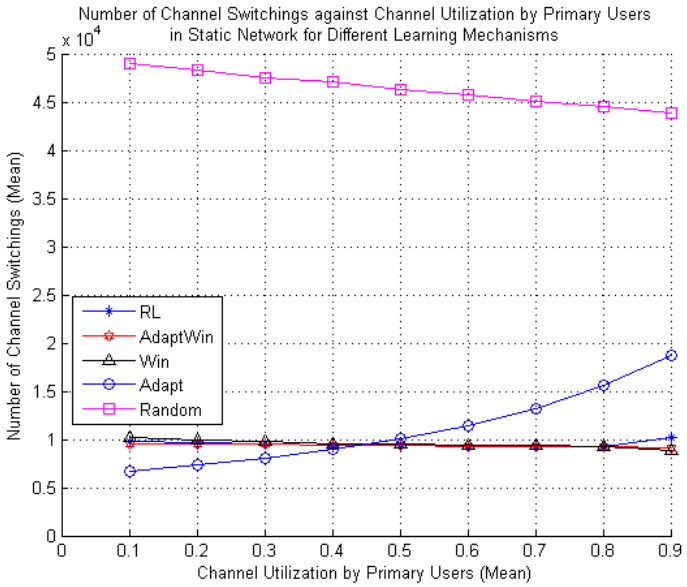
Figure 6.31: The mean throughput of an SU BS against standard deviation of PUL when the mean PUL equals 0.2 for RL, Random and Analysis in static and mobile networks. PER for all data channels is set to 0.1. α is set to 0.2. ε is set to 0.1.

Figure 6.32 shows the number of channel switchings achieved by the

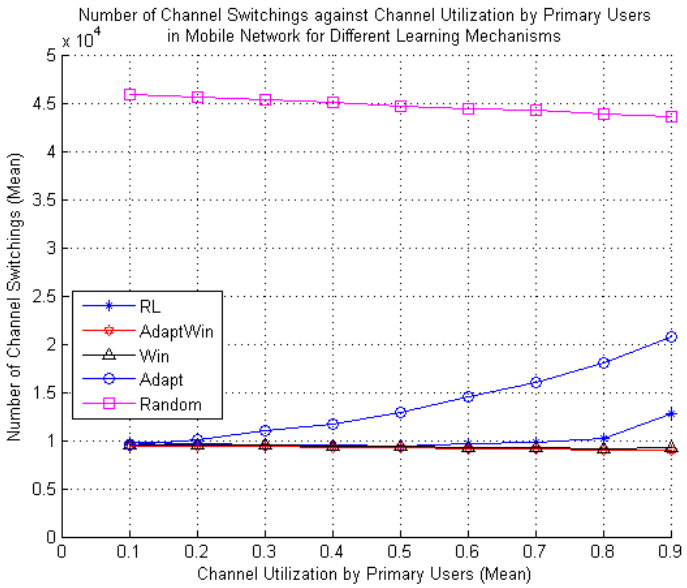
RL, Adapt, Win, AdaptWin and Random with respect to PUL in static and mobile networks. In general, RL, AdaptWin and Win achieve approximately similar individual network performance, which is the highest among the schemes studied, followed by Adapt, and finally Random. In Figure 6.32a, at 0.5 PUL in static networks, the RL, AdaptWin and Win approaches provide approximately 4.9 times, and Adapt provides approximately 4.6 times number of channel switchings reduction in comparison with Random. In Figure 6.32b, at 0.5 PUL in mobile networks, the RL, AdaptWin and Win approaches provide approximately 4.7 times, and Adapt provides approximately 3.5 times number of channel switchings reduction in comparison with Random.

6.10.3.2 Comparison of All Learning Mechanisms with respect to PER

Figure 6.33 shows the throughput achieved by RL, Adapt, Win, AdaptWin, Random and Analysis with respect to PER in static and mobile networks. In general, RL, AdaptWin and Win achieve approximately similar individual network performance, which is the highest among the schemes studied, followed by Adapt, and finally Random. Similar trends are observed for network performance with respect to PUL in Figure 6.30. As shown in Figure 6.33, the RL performs better than the Random in both static and mobile networks with the exception of Random which slightly outperforms RL at 0.9 PER in mobile networks in Figure 6.33b. Our investigation shows that at 0.9 PER, the Q-values of all the data channels converge to -CT. When all the data channels result in poor network performance, the RL approach simply chooses data channel K=3 that provides the shortest transmission range (see Figure 6.6) resulting in transmission failure for all data packet transmission attempts when the SU host moves beyond the transmission range of channel K=3. This issue can be solved by imposing a rule to transmit using a data channel that provides larger transmission range when all Q-values converge to the value of -CT. In comparison with Analysis, the RL, AdaptWin and Win approaches achieve the expected



(a) Static network.



(b) Mobile network.

Figure 6.32: The mean number of channel switchings of an SU BS against mean PUL for RL, AdaptWin, Win, Adapt and Random in static and mobile networks. PER for all data channels is set to 0.1. ε is set to 0.1.

throughput in both static and mobile networks; while Adapt underperforms. In short, RL, AdaptWin and Win learn well and help the SU BS to choose a data channel with low PER and suitable transmission range such that successful data packet transmission rate is high. The simulation results for RL and analytical results used to plot Figure 6.33 are analyzed next based on Figure 6.34.

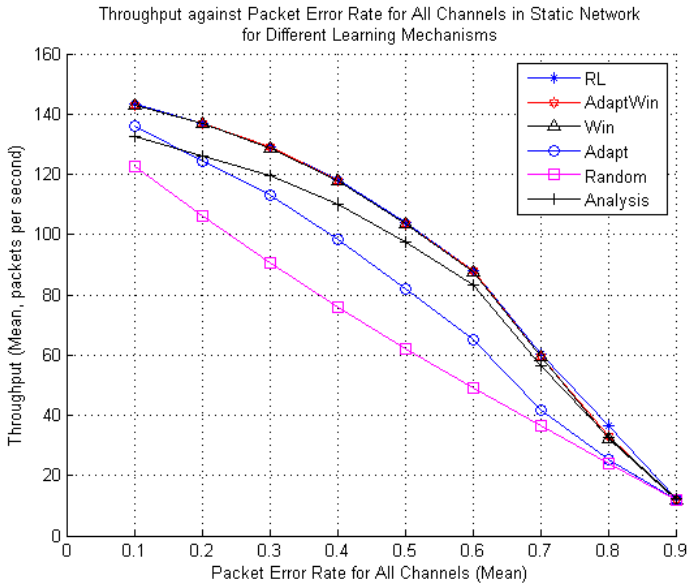
In Figure 6.34, given a mean value of the PER equals to 0.2, the throughput of RL, Random and Analysis is investigated for various levels of standard deviation for the PER (see Section 6.7.3.2 for explanation on standard deviation) in static and mobile networks. Similar trends are observed for network performance with respect to PUL in Figure 6.31.

Figure 6.35 shows the number of channel switchings achieved by RL, Adapt, Win, AdaptWin and Random with respect to PER in static and mobile networks. Similar trends are observed for network performance with respect to PUL in Figure 6.32.

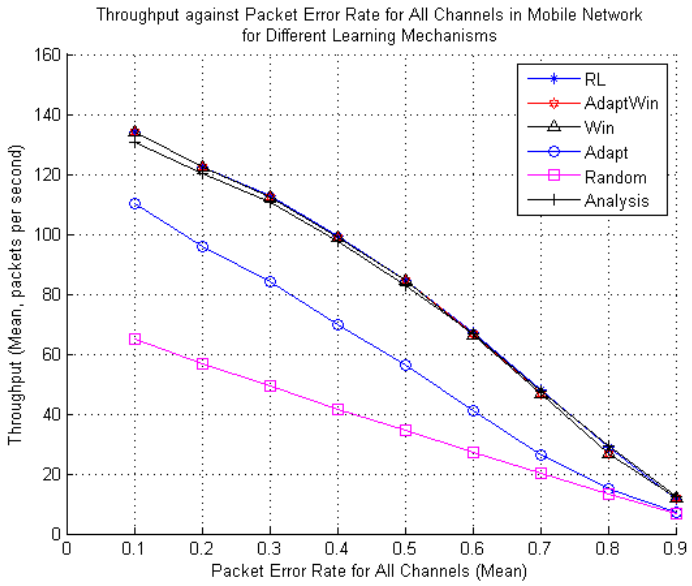
6.10.4 Summary of Research Outcomes

The research outcomes from the investigation on the comparison of RL, Adapt, Win, AdaptWin, Random and Analysis are summarized as follows:

- The RL, AdaptWin and Win approaches achieve approximately similar network performance in most of the cases, which is the highest among the schemes studied, followed by Adapt, and finally Random. In these approaches, an agent receives reward for successful data packet transmissions, and cost for unsuccessful ones. The reward is *RW* in RL, and the probability of $1/n^{\mathbf{Win}}$ in AdaptWin and Win. The cost is *CT* in RL, and the probability of $1/n^{\mathbf{Win}}$ in AdaptWin and Win. The RL approach chooses the channel with the highest Q-value $Q_t(c_i)$, while AdaptWin and Win choose the channel with the highest probablity of successful data packet transmission P_{S,c_i}^{Win}. In constrast, Adapt chooses a channel in a random manner during



(a) Static network.



(b) Mobile network.

Figure 6.33: The mean throughput of an SU BS against mean PER for RL, AdaptWin, Win, Adapt, Random and Analysis in static and mobile networks. PUL for all data channels is set to 0.1. ε is set to 0.1.

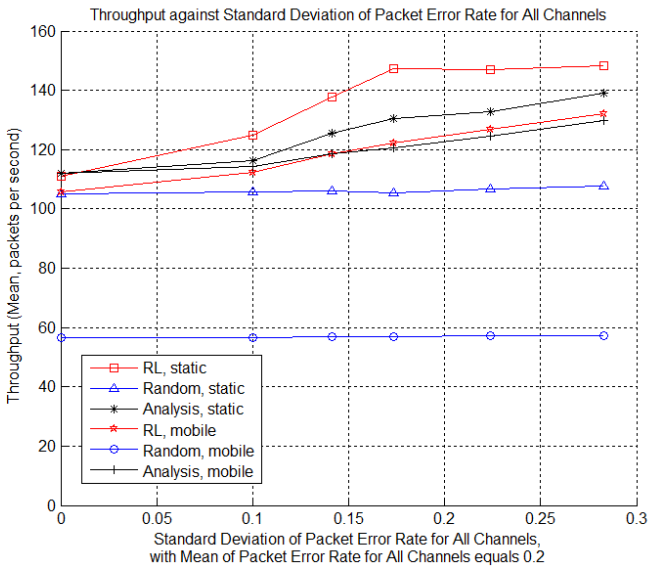
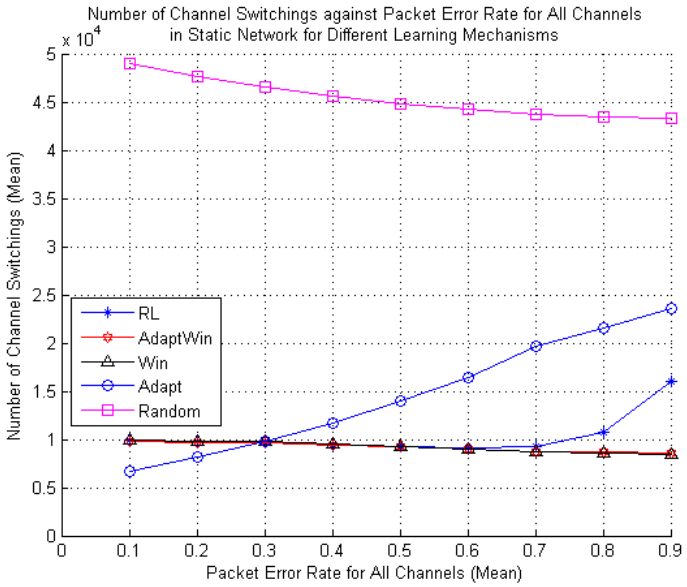
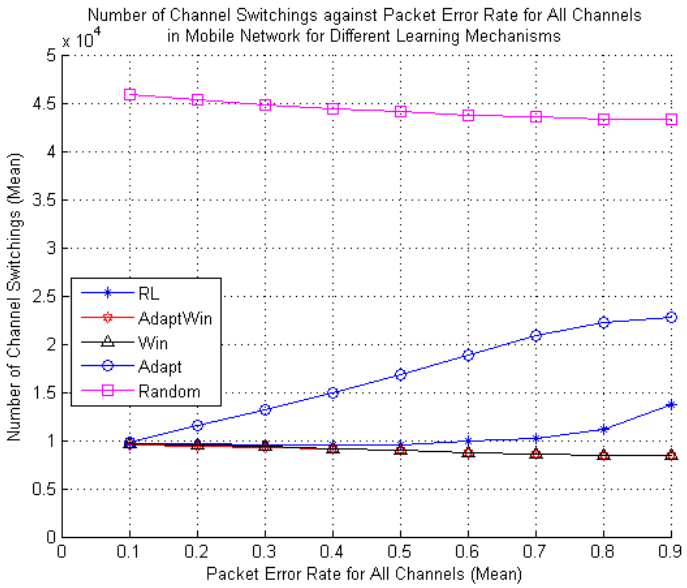


Figure 6.34: The mean throughput of an SU BS against standard deviation of PER when the mean PER equals 0.2 for RL, Random and Analysis in static and mobile networks. PUL for all data channels is set to 0.1. α is set to 0.2. ε is set to 0.1.



(a) Static network.



(b) Mobile network.

Figure 6.35: The mean number of channel switchings of an SU BS against mean PER for RL, AdaptWin, Win, Adapt and Random in static and mobile networks. PUL for all data channels is set to 0.1. ε is set to 0.1.

channel switching, hence its network performance is lower than RL, AdaptWin and Win.

- The RL, AdaptWin and Win approaches achieve the expected network performance provided by the analytical results.

This chapter shows that simple and pragmatic learning mechanims such as AdaptWin and Win achieve similar network performance to the RL approach. Similar RL models have been successully applied in a number of applications [58, 59, 60, 61, 62, 63].

Additionally, there is an open issue in RL: How to learn better when the Q-values among the data channels are close to each other? This is not addressed in previous work.

6.11 Advantages of RL in CR Networks

We have successfully introduced RL and three simple and pragmatic learning mechanisms, namely Adapt, Win, and AdaptWin to implement the single-agent cognition cycle (or network-level cognition cycle as shown in Section 2.3.2 on page 16) in order to achieve context awareness and intelligence in static and mobile centralized CR network. In this chapter, we consider heterogeneous channels, and both static and mobile networks; while previous work considers homogeneous channels and static networks. In Section 6.10, Win and AdaptWin are shown to achieve approximately similar individual network performance, which is the highest among the schemes studied, with the RL approach. The RL, Win and AdaptWin are also shown to achieve the expected throughput obtained from the analytical results. The RL approach outperforms the Adapt approach in all cases. In this section, we discuss two major advantages offered by RL compared to the Win and AdaptWin approaches. This discussion provides an important foundation for future work in this research field. The advantages are as follows:

- Extension of the RL approach in Section 6.4.1 to implement the multi-agent cognition cycle (or node-level cognition cycle) in distributed CR networks.

- Extension of the RL approach in Section 6.4.1 to include state representation, which encompasses the conditions of the operating environment that are relevant to decision making at the SUs in distributed CR networks.

6.11.1 Extension of the Reinforcement Learning Approach to Implement the Multi-Agent Cognition Cycle in Distributed Cognitive Radio Networks

Game-based approach has been the most popular approach for achieving context awareness and intelligence in CR networks. Game-based approach studies the interaction of multiple SUs whose objective is to maximize their individual local rewards. To date, research has been focusing on one-shot or repetitive games, such as the matrix game and potential game [76, 77, 78, 79, 80, 67, 81, 82, 83, 84, 85]. There are several known issues in game-based approach which have been addressed by the multi-agent reinforcement learning approach [64] as follows:

- Mis-coordination [86] where the SUs are not able to converge to an optimal joint action, which is the optimal actions taken by all the SUs throughout the entire network, because of severe negative rewards, and the SUs converge to a safe joint action instead.

- The SUs might converge to a sub-optimal joint action when multiple high performance joint actions exist [86].

- Game-based approach as applied to CR so far requires a complete set of information to compute the Nash equilibrium; hence its extensive and successful usage in centralized CR networks [67, 85].

- Game-based approach assumes that all SUs react rationally as game theorists.

- Game-based approach assumes a single type of utility function throughout the distributed CR network, and hence a homogeneous learning mechanism in all the SUs.

To date, a wide range of game-based applications in CR networks have been developed and shown to attain a certain equilibrium point, notably the Nash equilibrium. On the other hand, several existing MARL approaches have been shown to converge to a fixed optimal point [64]. Although the game-based approach has been successfully applied in CR networks [67, 85], the MARL approach is a good alternative which addresses the aforementioned issues associated with game-based approach. For instance, the MARL supports heterogeneous learning mechanisms in each SU because each SU can represent distinctive performance metrics as local rewards, or Q-values, in a particular distrbuted CR network. In the next chapter on the investigation of the multi-agent reinforcement learning approach, Section 7.3 shows a means of comunication among the SUs, called payoff propagation, that converges to an optimal joint action in a distributed manner including distributed CR networks with cyclic topology; and Section 7.4.6 applies the MARL approach, which encompasses the traditional RL [4] (see Section 6.4.1) and payoff propagation approach, to achieve network-wide performance enhancement through significant reduction in the number of channel switchings.

6.11.2 Extension of the Reinforcement Learning Approach to Include State Representation

In general, a game with states is called a stochastic game [87]; while a game without states is called a matrix game. The state represents the conditions of the operating environment that are relevant to decision making at an

SU such as internal queue size and external channel condition. To date, the matrix game has been widely applied in CR networks and it has been shown to achieve performance enhancement; however, the state representation, which is defined in RL [4] (see Section 5.3.4 on page 71) is ignored in the matrix game. The MARL approach, which solves the stochastic game, provides an alternative solution to counteract the disadvantages posed by the use of matrix game. It achieves optimal joint action in a stochastic game framework [87]. The stochastic game, which is a current and popular research topic [88], is a five-tuple game comprised of agents, states, set of actions available to each agent (or player), transition probability from one state to another, and reward function for each agent. Section 6.7 shows a thorough investigation into the effects of multiple states using the RL approach where the RL model is embedded in SU BS and the state represents a set of SU hosts associated with the SU BS. As similar trends are observed with single and multiple SUs, we choose to assume that there are two SUs comprised of an SU BS and an SU host in this chapter to model a centralized CR network.

6.12 Chapter Summary

In this chapter, RL and other simple and pragmatic learning mechanisms, namely, Adapt, Win and AdaptWin, are applied to DCS to implement the single-agent cognition cycle (or network-level cognition cycle) in order to achieve context awareness and intelligence in static and mobile centralized CR networks. The learning mechanisms differ among themselves in terms of action selection and knowledge update. The Random approach, which chooses an available data channel for data transmission in a uniformly distributed random manner without learning, serves as a baseline. An analytical model based on Markov chain is presented to compute the expected throughput performance of a DCS. This chapter considers channel heterogeneity, and both static and mobile networks are investigated;

while previous work considers channel homogeneity and static networks. In the RL approach, the states or SU hosts achieve approximately uniform individual network performance in the investigation on the effects of multiple states. In RL, the state encompasses the conditions of the operating environment that are relevant to decision making in an application. The effects of RL parameters on network performance were also investigated including the learning rate α and the exploration probability ε. The throughput and number of channel switchings achieve its optimal or near-optimal performance when the α and ε converge to a certain value; and the ε has greater effects on network performance than does α. The RL approach achieves approximately similar network performance with AdaptWin and Win, which provide the highest network performance among the other learning mechanisms studied. The next best network performance is achieved by Adapt, and followed by Random. The RL, Win and AdaptWin approaches achieve the expected throughput obtained from the analytical results; while this is not the case for Adapt, for instance, in a mobile network with different levels of PUL. There are two advantages of RL compared to Win and AdaptWin. Firstly, the extension of current work to achieve multi-agent cognition cycle (or node-level cognition cycle) using multi-agent reinforcement learning in distributed CR networks. Secondly, the extension of current work to include state representation.

Chapter 7

Multi-Agent Cognition Cycle

This chapter presents reinforcement learning, both single-agent and multi-agent approaches, for achieving context awareness and intelligence in static Distributed Cognitive Radio Networks (DCRN)s through the implementation of Multi-Agent Cognition Cycle (MACC) or node-level cognition cycle. Investigation is performed with respect to the DCS scheme. In this chapter, MACC is implemented using the Single-Agent Reinforcement Learning (SARL) and the Multi-Agent Reinforcement Learning (MARL) approaches. Note that, for better clarity, we refer the RL approach in Chapter 6, which is a single-agent approach, as the SARL approach in this chapter.

Firstly, in the Introduction section, this chapter presents objectives, related work, major differences between the SARL and MARL approaches, assumptions and their related work, an overview of distributed learning model, and characteristics of DCRNs. Secondly, it presents an important component in the MARL approach [64], specifically Payoff Propagation (PP). Generally speaking, the PP mechanism provides a means of communication for the SARL [4] approach in Section 6.4.1 on page 91, which is a local learning mechanism. The MARL approach encompasses the SARL approach and the PP mechanism. The application of SARL and MARL approaches to implement the MACC model is presented in two subsections.

Thirdly, it shows the SARL approach in scenario that applies a conventional assumption of identical channel quality (or PER) at all the SUs. Fourthly, it shows the SARL and MARL approaches in scenario that does not apply the assumption of identical channel quality at all the SUs.

7.1 Introduction

A DCRN is a distributed wireless network comprised of a number of SUs that interact with each other in a common operating environment in the absence of fixed network infrastructure or centralized coordinator such as a Base Station or access point.

7.1.1 Objectives

In static DCRNs, the DCS scheme provides the strategy to select an available licensed data channel for data transmission among commmunication node pairs given that the objective is to maximize overall throughput and minimize delay, in terms of number of channel switchings, in the presence of different levels of PUL and PER in the licensed data channels. The PUL and PER are explained in Section 4.1 on page 44. Note that, in contrast to Chapter 6 that considers different transmission ranges for all data channels, this chapter considers similar transmission ranges for all data channels because of the assumptions of static networks and single collision domain as explained later in this chapter. Using the MARL approach, the SUs aim to achieve a joint action, which is the actions taken by all the SUs throughout the entire DCRN, in a distributed manner through learning in order to achieve the objectives.

7.1.2 Related Work

The related work of this chapter is discussed in Section 6.3 on page 84, and the advantages of the SARL and MARL approaches compared to one

of the most popular tool to achieve context awareness and intelligence in CR networks, namely game-based approach, is discussed in Section 6.11.1 on page 160. This thesis is the first attempt to investigate the application of MARL on achieving context awareness and intelligence in CR networks.

7.1.3 Major Differences between SARL and MARL

The major differences between the SARL approach in Section 6.4.1 on page 91 and the MARL approach in this chapter are discussed below and they are summarized in Table 7.1.

- In SARL, the operating environment, such as a centralized CR network, is comprised of a *single* agent or decision maker. The purpose is to achieve *individual network performance enhancement*. Since there is a single agent only, the SARL approach *does not consider the effects of actions to the operating environment* (see Section 5.3.7 on page 74). According to Busoniu et al [88], the SARL approach can be directly applied to the multiagent scenario [89] and thus to DCRNs; however, the SARL approach may not achieve stability, specifically, the agents may change their respective actions frequently, or oscillate between actions, and fail to achieve an optimal joint action. Despite its limitations, the SARL approach has been applied in a significant number of applications and it has been shown to achieve stability in these applications [90, 91, 92, 93].

- In MARL, the operating environment, such as a DCRN, is comprised of *multiple* agents. The purpose is to achieve *network-wide performance enhancement*. Since there are multiple agents, the MARL approach *considers the effects of actions to the operating environment*. The MARL approach achieves stability [88].

Table 7.1: Major Differences between SARL and MARL

Characteristics	SARL	MARL
Number of agent(s) in the operating environment	Single	Multiple
Level of network performance enhancement	Individual	Network-wide
Consideration of the effects of actions on the operating environment	No	Yes
Goal of achieving stability	No	Yes

7.1.4 Assumptions and Their Related Work

A detailed explanation of the common assumptions in the CR research field is found in Section 2.4 on page 17. In this chapter, our assumptions are as follows:

- Static networks as applied in previous schemes [25, 26, 27].
- Distributed networks. Previous schemes [67] consider centralized networks.
- Single collision domain in DCRNs as applied in previous schemes [26]. This assumption is applied in the investigation on the application of the SARL and MARL approaches in DCRNs in Section 7.4. This assumption is not applied in the investigation on the PP mechanism in Section 7.3.
- Channel heterogeneity. Previously proposed schemes [25, 26, 27] assumed channel homogeneity.
- Identical channel condition (or PER) at all the SUs in Section 7.4.5 and non-identical channel condition at all the SUs in Section 7.4.6. Previous schemes [25, 26, 27] assume channel homogeneity.

- Simplified RL model without consideration of events (see Section 5.3.4 on page 71) and rules (see Section 5.3.6 on page 73). However, this chapter considers the effects of actions on the operating environment (see Section 5.3.7 on page 74).

Table 7.2 presents a summary of the assumptions applicable to the three major investigations in this chapter.

Table 7.2: Assumptions on Various Investigations in this Chapter

Assumption	Payoff Propagation (see Section 7.3)	Scenario with identical channel condition (or PER) at all the SUs (see Section 7.4.5)	Scenario with non-identical channel condition (or PER) at all the SUs (see Section 7.4.6)
Single collision domain	No	Yes	Yes
Identical channel condition (or PER) at all the SUs	Not applicable	Yes	No

7.1.5 Distributed Learning Model

Section 6.3.2 on page 86 presents related work on the application of SARL in CR networks. As a complement to [58, 59, 60, 66, 61, 62, 63] which apply the SARL approach only, this chapter applies both SARL and MARL approaches.

As shown in Figure 7.1, we model each SU communication node pair as a learning agent because the SU transmitter and receiver share a single set of learned outcomes or knowledge. At a particular time instant, the agent observes its own local operating environment only due to its limited sensing capability. The agents can improve the global reward in the next time instant through carrying out their respective proper action. The global reward is a linear combination (or sum) of all the local rewards at each agent. The learning engine provides knowledge on the operating environment comprised of multiple agents through observing the consequences of its prior action in the form of local reward. The difference between SARL (see Figure 5.2 on page 69), which is the single-agent approach, and MARL, which is the multi-agent approach, is the additional feature in MARL, namely Payoff Message Exchange (PME) as shown in Figure 7.1. The payoff is computed using local rewards. The PME mechanism provides a payoff message exchange mechanism that helps each agent to communicate and compute its own action as part of the joint action, which is the actions taken by all the SUs throughout the entire DCRN. In other words, the PME is a means of communication for the learning engine embedded in each agent. Note that the SARL approach does not implement the PME mechanism because it is a single-agent approach. As time progresses, the MARL agents learn to carry out the proper actions to maximize the global reward. As an example, the learning engine is used to learn the channel conditions such as PUL and PER. Section 7.4.5 considers that the channel PER at all the agents are *identical*. Section 7.4.6 considers that the channel PER at all the agents are *non-identical* as each agent may observe different uncertain and varying channel conditions caused by various factors including shadowing, channel selective fading, path loss, PU interference, and others. Section 7.4.5 applies the SARL and its enhanced approaches; while Section 7.4.6 applies both the SARL and MARL approaches. SARL maximizes the local rewards; while MARL maximizes the global reward. Based on the application, the reward indicates distinctive performance

metrics such as throughput and successful data packet transmission rate. Thus, maximizing the local and global rewards provides network-wide performance enhancement.

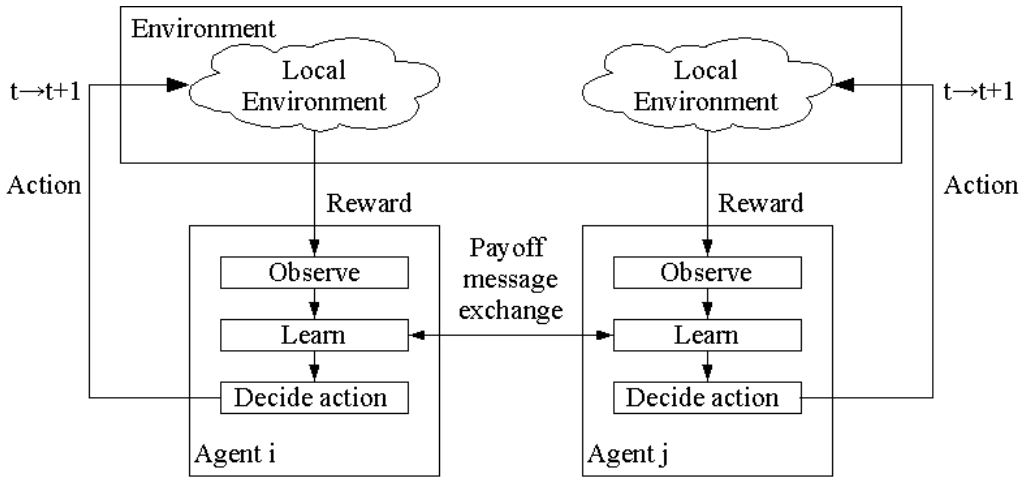


Figure 7.1: Agents (or SU communication node pairs) and their environment.

7.1.6 Characteristics of Distributed Cognitive Radio Networks

We refer to a *single node* as an *SU*; and an *SU communication node pair* as an *agent* henceforth. The single-hop DCRN is illustrated in Figure 7.2, and its characteristics and assumptions are as follows:

- **Primary Users**
  - There are K PUs, PU=[PU_1,. . .,PU_K].
  - Each PU uses one of the K distinctive channels of frequency F=[F_1,. . .,F_K] and broadcasts packets throughout the entire simulation area. The PUs do not change their respective channel, thus there are K PUs and channel frequencies. For instance, PU_1

uses channel frequency F_1, PU_2 uses channel frequency F_2 and so on. The PUs do not switch their channels. The PUs do not use four-way handshaking.

- The PUs are not aware of the presence of the SUs.
- The channel utilization pattern of the PUs follow a Poisson distribution with the mean arrival rate determined according to the PUL level, and among the data channels it follows an independent and identically distributed (i.i.d.) stochastic model.

- **Secondary Users**
  - Each SU node is equipped with two transceivers, namely a *control transceiver* and a *data transceiver*, thus it is capable of accessing two different channels simultaneously.
    - The control transceiver is tuned to a common channel in the ISM band for control message exchange, as well as information broadcast.
    - The data transceiver is tuned to one of the available data channels in the licensed bands for data packet transmission. Thus, the PU activities exist in the data channels only.
  - The SU transmitter of an agent is always backlogged and transmits data packets to its SU receiver at every opportunity.
  - The SARL or MARL model is embedded in the SU transmitter; while the SU receiver switches its data channel according to the decision made by the SU transmitter. The SU receiver is informed of the changes in the data channel through control message exchange in the common control channel.

- **Secondary User Agents**
  - There are V SUs, and hence there are U=$V/2$ agents.

- An agent i is comprised of an SU transmitter T_i and an SU receiver R_i.
- Each agent maintains a single set of knowledge because the SU transmitter and SU receiver must choose a common data channel for data transmission in DCS. The knowledge can be maintained through control message exchange.
- The condition $K \leq U$ is applied so that the agents are competing to use the data channels. In Figure 7.2, K=3$\leq U$=3.
- The agents infer the PUL, PER and contention level in each data channel, and select in a distributed manner a data channel for data transmission.
- An agent i chooses a data channel $c^i_{j,t}$ out of K available data channel for data transmission at time t.

• **Channel Characteristics**

- There are K orthogonal available data channels with similar bandwidth.
- The assumption of a single collision domain is applied in the investigation on the application of the SARL and MARL approaches in DCRNs in Section 7.4. This assumption is not applied in the investigation on the PP mechanism in Section 7.3.
- Each data channel is characterized by various levels of PUL, L_{c_i}=[L_1,. . .,L_K].
- We consider heterogeneous channels and two cases of channel conditions or PER:
  * A scenario with *identical* channel condition (or PER) at all the SUs such that P=[P_1,. . .,P_K] in Section 7.4.5.
  * A scenario with *non-identical* channel condition at all the SUs such that P_i=[$P_{i,1}$, . . .,$P_{i,K}$] in Section 7.4.6. Hence, d-

ifferent agent i may observe different levels of PER using a particular data channel $c^i_{j,t}$.

A data channel with low PUL does not imply a good channel if it has a high level of PER or contention. Figure 7.3, which is a graphical representation of Figure 7.2, illustrates the concept of the DCS scheme. Suppose, agent 1 or T_1-R_1 chooses data channel 1; while agent 2 or T_2-R_2 chooses data channel 2. Data channel K is not chosen because, say, it has high PUL and PER. Agent U chooses data channel 1 because the channel has lower PUL compared to data channel 2. This channel selection provides better network-wide performance.

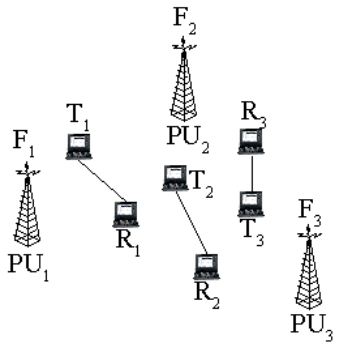


Figure 7.2: Single-hop DCRN. A single node is referred to as an SU, while an SU communication node pair is referred to as an agent. Solid line indicates communication link.

7.2 Chapter Goal

There are three new contributions in this chapter with respect to static DCRNs:

- We show that the PP mechanism achieves an optimal joint action in Section 7.3.

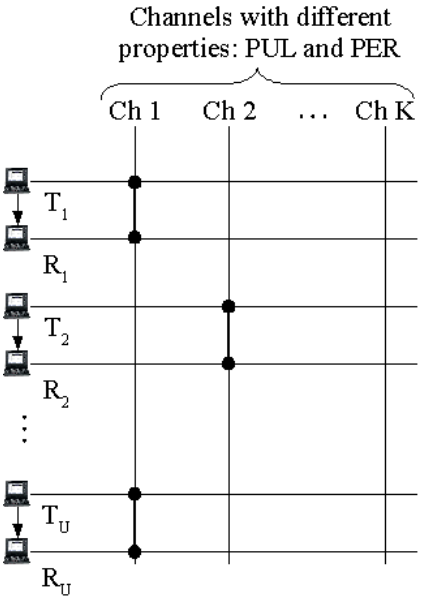


Figure 7.3: Graphical representation of the DCS scheme. Bold line indicates data transmission over a chosen data channel.

- We show that the SARL and MARL approaches achieve a joint action that provides better network-wide performance in DCRNs using scenarios with *identical* channel condition (or PER) at all the agents in Section 7.4.5.
- We show that the SARL and MARL approaches achieve a joint action that provides better network-wide performance in DCRNs using scenarios with *non-identical* channel condition (or PER) at all the agents in Section 7.4.6.

7.3 Payoff Propagation

7.3.1 Introduction

This section presents a novel extended PP mechanism that helps the SUs to achieve an efficient and optimal joint action in a cooperative and dis-

tributed manner through learning in DCRNs. It is suitable to be applied in most applications in DCRN that requires context awareness and intelligence such as DCS, scheduling, and congestion control. Section 7.1.5 provides an overview of the PP mechanism. The PP mechanism is investigated with respect to DCS. The payoff messages are piggybacked on the control messages including RTS and CTS. This section shows that the network-wide performance achieved by the PP mechanism converges to an efficient and optimal joint action in a distributed manner including DCRNs with cyclic topology; and that fast convergence is possible.

The PP mechanism is a cooperative approach where a group of agents cooperate with each other to take an efficient and optimal joint action. The cooperative environment is particularly suitable for multi-hop DCRNs because an agent must cooperate with its next hop neighbour agent that helps it to relay its data packets to its destination.

7.3.1.1 Key Terms

Three key terms are:

- *Joint action* is the set of actions taken by all the agents throughout the entire DCRN.
- *Optimal joint action* is the joint action that provides the ideal and optimal network-wide performance.
- *Efficient joint action* is the joint action that fulfills the requirement on the network-wide performance which in general will be more readily achievable than the network-wide performance provided by the optimal joint action.

The optimal joint action varies with the dynamic and uncertain operating environment; and therefore, attempting to achieve the optimal joint action at most of the times may introduce high cost of overhead, and instability

throughout the entire network. In addition, an agent may have to deviate from the optimal joint action occasionally in order to explore and discover joint actions that provide network-wide performance enhancement. Achieving an efficient joint action is sufficient to provide a network-wide performance guarantee.

7.3.1.2 Performance Metrics

The performance metrics are:

- *Global reward* is a linear combination (or sum) of all the local reward at each agent.
- *Global payoff* is a linear combination (or sum) of all the local payoffs, which are computed using local rewards, generated by each agent, in addition to the local reward at the agent.
- *Convergence time* is the time duration for the PP mechanism to achieve an efficient or optimal joint action.

The PP mechanism optimizes both the global reward and global payoff in order to achieve an efficient or optimal joint action that provides network-wide performance enhancement. Based on the application, such as DCS, the reward and payoff values indicate distinctive network performance metrics such as throughput and successful data packet transmission rate.

7.3.1.3 Main Challenges

There are two main challenges in a multi-agent environment:

- An agent's action is dependent on the other payoff-optimizing agents' actions.
- All agents must converge to an efficient or optimal joint action that provides network-wide performance enhancement.

Generally speaking, from the perspective of each agent, the research question is "How does an agent choose its own action such that the joint action converges to an efficient and optimal joint action?"

7.3.1.4 Assumptions

Section 7.1.4 shows the assumptions applicable in this section, and the additional assumptions are as follows:

- The MARL approach encompasses the SARL approach and the PP mechanism. Hence, a local learning mechanism, such as SARL in Section 6.4.1 on page 91, is available at each agent to provide the Q-values. The Q-values characterize the channel heterogeneity properties for each data channel including PUL and PER. For a particular data channel, the Q-values are different among the agents as each of them observes different levels of PUL and PER. In short, the Q-values are Independent and Identically Distributed (i.i.d.) among the agents and the data channels. Each Q-value has a range of -5$\leq$ $Q_t(a_i,a_{j\in\Gamma(i)})\leq$15, where $\Gamma(i)$ represents all the single-hop neighbour agents of agent i.

- Non-single collision domain. As shown in Figure 7.4, the agents are distributed in a uniform and random manner in a square region. There are two kinds of links as follows:

  - *Communication link* exists within an agent.
  - *Interference link* exists between two neighbouring agents that do not communicate with each other.

  In a single collision domain scenario, the interference link exists among *all* the agents; while in a non-single collision domain scenario, which is considered in this section, the interference link exists among *some* of the agents.

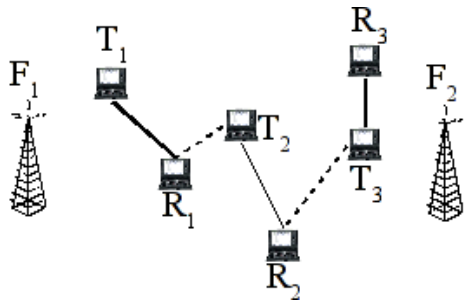


Figure 7.4: Single-hop DCRN without the assumption of single collision domain. Solid line indicates communication link; while dotted line indicates interference link.

7.3.1.5 Contributions

The focus in this section is the PP mechanism. This section shows that the extended PP mechanism addresses the two aforementioned challenges (see Section 7.3.1.3 with respect to DCS. Generally speaking, the DCS scheme provides the channel selection that optimizes the global reward and global payoff in order to achieve an efficient and optimal joint action. The contributions of this section are as follows:

- To show that the extended PP mechanism converges to an efficient and optimal joint action including DCRNs with cyclic topology.
- To show the effects of network density and various essential parameters in the extended PP mechanism on network-wide performance.
- To show that the extended PP mechanism provides fast convergence.
- To show the effects of unstable Q-values provided by the SARL approach on the extended PP mechanism on network-wide performance.

7.3.1.6 Section Organization

The remainder of this section is organized as follows. Section 7.3.2 presents the original PP mechanism. Sections 7.3.3 presents the extended PP mechanism. Section 7.3.4 presents simulation experiments, results and discussions. Section 7.3.5 provides discussion and summary of research outcomes in this section.

7.3.2 Original Payoff Propagation Mechanism

This section first describes the Coordination Graph (CG); followed by the local reward, and finally the original PP mechanism.

7.3.2.1 Coordination Graph

In Figure 7.5, there are U=V/2=4 agents in the DCRN, and each agent is represented by a single node. An interference edge exists between a pair of neighbouring agents that only exchange a small number of control signals with each other. The entire DCRN can be decomposed into smaller and local CGs, which are each a local view of the entire network for each agent. In Figure 7.5, the CG of agent 1 is comprised of agent 1, 2, and 3, hence the representation of the local reward or Q-value $Q_{i,t}(a_{i,t},a_{j\in\Gamma(i),t})$ = $Q_{1,t}(a_{1,t},a_{2,t},a_{3,t})$, where $a_{i,t}\in A$, and A is a set of possible actions. The CG defines collaborative relationships among the agents. A collaborative relationship corresponds to a local payoff message exchange. Each agent runs a local learning mechanism such as the SARL approach independently to update its own Q-values. The approximate global Q-value $Q_t(\mathbf{a_t})$ at time t is a linear combination (or sum) of all the local Q-values for the undertaking action at each agent as follows:

$$Q_t(\mathbf{a_t}) = \sum_{i=1}^{U} Q_{i,t}(a_{i,t}, a_{j\in\Gamma(i),t}) \tag{7.1}$$

Note that Equation (7.1) is not a utility function, which is not defined in the MARL approach [64]. Equation (7.1) shows that the complexity of the global reward optimization can be simplified through maximizing the local rewards. The MARL approach has been shown to converge to an optimal point that maximizes the global reward in a wide range of benchmarking problems from the NIPS 2005 workshop [64].

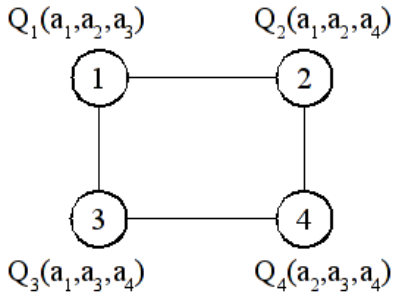


Figure 7.5: A four-agent graph G. Each agent represents an SU communication node pair. The edges are interference edges that exist between the agents that interfere with each other.

7.3.2.2 Local Reward

The Q-value $Q_{i,t}(a_{i,t},a_{j\in\Gamma(i),t})$ is the learned knowledge at agent i and it is maintained in a lookup Q-table with $|A|$ entries. The Q-value represents the local reward that the agent can gain for choosing an action $a_{i,t}\in A=F$, where F is the set of carrier frequencies of the available data channels, in the coexistence of neighbouring agents. As an example, in DCS, the Q-value represents the throughput performance and it is dependent on the local PUL, PER and joint action $\mathbf{a_t}$. The joint action affects the Q-value due to the dependency of the actions among the agents. For example, two neighbour agents that choose a particular action, specifically a data channel, might increase their contention level, and hence reduce their respective Q-values for the action.

7.3.2.3 Payoff Propagation Mechanism

The optimal and efficient joint action $\mathbf{a_t^*}$ is comprised of the channel selections from all the agents such that the global reward and global payoff are optimized. Each agent computes its respective action $a_{i,t}^* \in A$ as part of the optimal joint action using its local Q-value, $Q_{i,t}(a_{i,t}, a_{j\in\Gamma(i),t})$ and its neighbour agents, $Q_{j\in\Gamma(i),t}(a_{j,t}, a_{k\in\Gamma(j),t})$ to achieve the optimal global Q-value $Q_t(\mathbf{a_t^*})$.

We describe the original PP mechanism [64] in this section. Each agent i constantly sends locally optimized reward value or payoff message $\mu_{ij}(a_{j,t})$ to its neighbour agents $j \in \Gamma(i)$ over the edges as shown in Figure 7.6. The payoff $\mu_{ij}(a_{j,t})$ is computed by maximizing, over all the possible actions of agent i, the sum of the locally optimized Q-value $Q_{i,t}(a_{i,t}^*, a_{j\in\Gamma(i),t})$ and all the received payoff messages except that from agent j as follows:

$$\mu_{ij}(a_{j,t}) = \max_{a_i \in A}[Q_{i,t}(a_i, a_{j\in\Gamma(i),t}) + \sum_{k\in\Gamma(i)\backslash j} \mu_{ki}(a_i)] \tag{7.2}$$

where $\Gamma(i)\backslash j$ represents all the neighbour agents of agent i except agent j.

The payoff messages are exchanged among the agents until a fixed optimal point is reached. Before convergence, the payoff messages are an estimation of the fixed optimal point as all incoming messages of an agent are yet to converge. Each agent selects its own optimal action to maximize the *local payoff* as follows:

$$g_{i,t}(a_{i,t}) = \max_{a_i \in A}[Q_{i,t}(a_i, a_{j\in\Gamma(i),t}) + \sum_{j\in\Gamma(i)} \mu_{ji}(a_i)] \tag{7.3}$$

Each agent i determines its optimal action individually as follows:

$$a_{i,t}^* = \underset{a_i \in A}{\operatorname{argmax}} g_{i,t}(a_i) \tag{7.4}$$

The approximate global payoff $g_t(\mathbf{a_t})$ at time t is a linear combination (or sum) of all the local payoffs at each agent as follows:

$$g_t(\mathbf{a_t}) = \sum_{i=1}^{U}[Q_{i,t}(a_{i,t}, a_{j\in\Gamma(i),t}) + \sum_{j\in\Gamma(i)} \mu_{ji}(a_{i,t})] \tag{7.5}$$

Note the difference between the global Q-value, $\sum_i Q_{i,t}$ in (7.1) and the global payoff, $\sum_i [Q_{i,t}+\mu_{ji}]$ in (7.5). The global Q-value is the total local rewards received by all the agents in the network; while the global payoff is the total local rewards and payoff exchanged among the agents in the network. Both Equations (7.1) and (7.5), which are the performance metrics for the PP mechanism, converge to an optimal joint action.

7.3.3 Extended Payoff Propagation Mechanism

In this section, modifications to the original PP mechanism with respect to DCS are presented. The original PP mechanism [64] cannot be applied to the DCS problem for two reasons: failure to converge in a cyclic topology, and its inability to include payoff computation using payoff values from non-interfering agents.

7.3.3.1 Failure to Converge in a Cyclic Topology

Firstly, for a tree-structured graph, the agents would reach a fixed optimal point after a finite number of iterations [64], [94]. For a cyclic topology as shown in Figure 7.6, the original PP mechanism in Section 7.3.2.3 causes the agent to continuously add its own local Q-value in its payoff computation causing the payoff value to increase without bound. Thus, it requires all the agents in the network to occasionally compute the global payoff and update their optimal joint action when the global payoff improves upon the best joint action found so far.

As an example on the effect of the cyclic topology in Figure 7.6, agent 1 calculates $\mu_{12}(a_{2,t})$ using (7.2) and sends it to agent 2. Assume that agent 1, 2, 3, 4, 5, 6, 7 and 8 choose their respective optimal action in a sequential

manner. As time goes by, $\mu_{12}(a_{2,t})$ is computed as follows:

$$\begin{aligned}\mu_{12}(a_{2,t}) &= \max_{a_1 \in A}[Q_{1,t}(a_1, a_{2,t}, a_{8,t}) \\ &+ \mu_{81}(a_{1,t-1}) + \mu_{78}(a_{8,t-2}) + \mu_{67}(a_{7,t-3}) + \mu_{56}(a_{6,t-4}) \\ &+ \mu_{45}(a_{5,t-5}) + \mu_{34}(a_{4,t-6}) + \mu_{23}(a_{3,t-7}) + \mu_{12}(a_{2,t-8}) \\ &+ \ldots + \mu_{78}(a_{8,2}) + \mu_{81}(a_{1,1}) + \mu_{12}(a_{2,0})]\end{aligned} \tag{7.6}$$

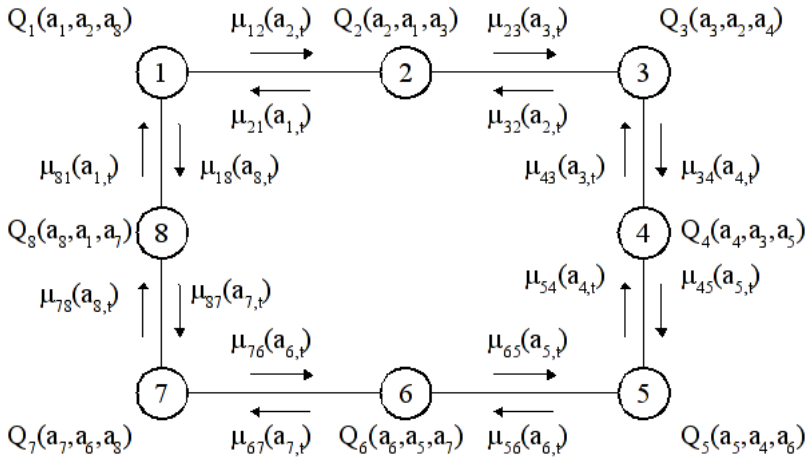


Figure 7.6: Payoff messsage exchanges in a graph with eight agents.

7.3.3.2 Payoff Computation using Payoff Values from Non-Interfering Agents

Secondly, the payoff computation in (7.2) uses payoff values from non-interfering agents. We consider that an agent selects its data channel based on the data channel selections of its two-hop neighbour agents. As time goes by, although the channel selection at agent 1 is not dependent on agent 4, 5 and 6 in Figure 7.6, the $\mu_{12}(a_{2,t})$ in (7.6) is computed using $\mu_{45}(a_{5,t})$, $\mu_{56}(a_{6,t})$ and $\mu_{67}(a_{7,t})$.

7.3.3.3 Locally Confined Payoff Propagation Mechanism

In our modified locally confined PP mechanism, agent i broadcasts the local Q-value of its own current action (or data channel), and its one-hop neighbour agents $j \in \Gamma(i)$. All the Q-values are time-stamped. The payoff message is as follows:

$$\mu_{i\psi}(a_{\psi,t}) = [Q_{k,t}(a_{k,t}, a_{l\in\Gamma(k),t}); T_{Q_{k,t}}], i \in k, j = \Gamma(i) \in k \tag{7.7}$$

Each agent maintains a μ-table with size $N_{n,i} \times |A|$ to keep track of the payoff messages, where $N_{n,i}$ is the number of neighbour agents of an agent i. A neighbour agent $j \in \Gamma(i)$ that receives $\mu_{i\psi}(a_{\psi,t})$ stores each Q-value in the message if it has a more recent time stamp $T_{Q_{k,t}}$ compared to the time stamp of the corresponding Q-value in its μ-table, $T^{'}_{Q_{k,t}}$. The $T^{'}_{Q_{k,t}}$ is also updated. As an example, when agent j receives the $\mu_{i\psi}(a_{\psi,t})$ while it is taking action $a_{j,t}$, the message $\mu_{i\psi}(a_{\psi,t})$=$\mu_{ij}(a_{j,t})$ indicates the local rewards of agent i, and its one-hop neighbour agents while agent j is taking action $a_{j,t}$. The local payoff or Equation (7.3) is rewritten as follows:

$$\begin{aligned} g_{i,t}(a_{i,t}) &= \max_{a_i \in A}[Q_{i,t}(a_i, a_{j\in\Gamma(i),t}) + \sum_{j\in\Gamma(i)} Q_{j,t}(a_{j,t}, a_{k\in\Gamma(j),t}) \\ &+ \sum_{k\in\Gamma(j)\setminus j} Q_{k,t}(a_{k,t}, a_{l\in\Gamma(k),t})] \end{aligned} \tag{7.8}$$

The approximate global payoff or Equation (7.5) is rewritten as follows:

$$\begin{aligned} g_t(\mathbf{a_t}) &= \sum_{i=1}^{U}[Q_{i,t}(a_{i,t}, a_{j\in\Gamma(i),t}) + \sum_{j\in\Gamma(i)} Q_{j,t}(a_{j,t}, a_{k\in\Gamma(j),t}) \\ &+ \sum_{k\in\Gamma(j)\setminus j} Q_{k,t}(a_{k,t}, a_{l\in\Gamma(k),t})] \end{aligned} \tag{7.9}$$

Our modified locally confined PP mechanism addresses the two aforementioned drawbacks in the original PP mechanism while achieving optimal joint action in a distributed manner. The pseudo-code of our modified P-P mechanism is embedded in each agent, and it is shown in Algorithm

Algorithm 1 Pseudo-code of the extended PP algorithm at agent i.

initialize $\mu_{ij} = \mu_{ji} = 0$ for $j \in \Gamma(i)$, $g_{i,t} = 0$
{Tasks: 1. Broadcast payoff message to neighbour agents;
2. Select optimal action}
if (my turn to select an optimal action) **then**
compute and broadcast payoff $\mu_{i\psi}(a_{\psi,t})$ {Refer to (7.7)}
compute $a^*_{i,t}$ {Refer to (7.8) and (7.4)}
return $a^*_{i,t}$
end if
{Task: 3. Receive payoff message}
wait for msg
if (msg = $\mu_{j\psi}(a_{\psi,t})$) **then**
foreach $Q_{k,t}(a_{k,t}, a_{l\in\Gamma(k),t})$
if $(T_{Q_{k,t}} > T'_{Q_{k,t}})\&\&(k \neq i)$ **then**
update $Q_{k,t}(a_{k,t}, a_{l\in\Gamma(k),t})$
update $T'_{Q_{k,t}}$
end if
end foreach
end if

1. The PP mechanism is executed until the agent converges to an optimal local action where the changes of its local Q-values and local payoffs between iterations are insignificant.

The action selection in (7.4) does not cater for the actions that are never chosen. In the ε-greedy approach [4] (see Section 5.3.5 on page 73), an agent performs exploration with small probability ε, and exploitation with probability 1-ε. The ε-greedy approach is applied in this section.

The PP mechanism has been shown to converge to an optimal point in a wide range of benchmarking problems from the NIPS 2005 workshop [64]. The reliability of the PP mechanism for convergence, which is not proven in [64], is shown here.

Proposition 1: *If the entries in the Q-table and μ-table at each agent are stable and fixed, the PP mechanism will converge to an efficient and optimal joint action.*

Proof: Denote the difference between the optimal global Q-value $Q_t(\mathbf{a_t^*})$ and instantaneous global Q-value for exploitation action $Q_t(\mathbf{a_t})$ by δ_t=$Q_t(\mathbf{a_t^*})$-$Q_t(\mathbf{a_t})$. Assume that the Q-values at each agent are stable and fixed. Convergence to an *optimal* joint action $\mathbf{a_t^*}$ happens when δ_t=0; while convergence to an *efficient* joint action happens when $\delta_t \leq \delta^{'}$ where $\delta^{'}$ is the threshold that fulfills the requirement on network-wide performance. With U agents and $|A|$ actions, the number of possible joint actions is $|A|^U$. With exploration probability ε>0, the agent explores all possible joint action $\mathbf{a_t}$. The probability that the optimal joint action being explored is $p^*=1/|A|^U$, thus the probability that δ_t=0 as $t\rightarrow\infty$ equals 1. The probability of an efficient joint action being explored is p. For instance, if 20% of the joint actions fulfill the condition $\delta_t \leq \delta^{'}$, then p=0.2. Suppose, the exploration follows a geometric distribution, then the probability that an efficient joint action could be explored in the n^{th} trial is $f_N(n)=p(1-p)^n$ with n=0,1,2,.... Thus, with p=0.2, the cumulative distribution function that an efficient joint action could be found within n=10 trials is $F_N(10)=1-(1-p)^{(n+1)}$=0.91. At each time step, one of the following events in the set S occurs:

$$S=\begin{cases}\delta_n=0 & f_{N^*}(n)=p^*(1-p^*)^n \\ \delta_n \leq \delta^{'} & f_N(n)=p(1-p)^n \\ \delta_{t=n} \geq \delta_{t<n} & \alpha \\ \delta_n > \delta^{'} & 1-f_{N^*}(n)-f_N(n)-\alpha\end{cases} \tag{7.10}$$

The agents exploit, with probability 1-ε, the best-known joint action that maximizes the local payoff using (7.4). The agent explores with probability ε. As time goes by, the μ-table becomes stable and fixed since the Q-table is stable and fixed, which are the conditions for α=0 so that the probability of $\delta_{t=n} \geq \delta_{t<n}$ is 0. Maximizing the local payoff $g_{i,t}(a_{i,t})$

maximizes the local Q-value at an agent and its neighbour agents as shown in (7.8). Therefore, $\delta_t \rightarrow 0$ as $t \rightarrow \infty$. ■

Proposition 2: *The payoff value in the extended PP mechanism does not increase without bound in a cyclic topology.*
Proof: Consider an eight-agent graph in Figure 7.6. Suppose, agent 1 sends $\mu_{12}(a_{2,t})$, which is comprised of its own local Q-value $Q_{1,t}(a_{1,t},a_{2,t},a_{8,t})$, and its one-hop neighbour agent's local Q-value $Q_{8,t}(a_{8,t},a_{1,t},a_{7,t})$ to agent 2. The $\mu_{12}(a_{2,t})$ becomes extremely large when agent 1 constantly includes its own Q-value $Q_{1,t-n}(a_{1,t-n},a_{2,t-n},a_{8,t-n})$ at time *t-n*, where $n=\{n_1,n_2,\ldots\}$ is time step in the history, into the payoff value. The proposed update (7.7), in contrast to (7.2), does not include $Q_{1,t-n}(a_{1,t-n},a_{2,t-n},a_{8,t-n})$ into the payoff value, and thus it does not increase without bound in a cyclic topology. This explanation can be generalized to all agents in a cyclic topology. ■

7.3.4 Simulation Experiment, Results, and Discussions

Our objective is to enable the SU agents to select their data channel (action) for data transmission such that the channel selection (joint action) by all the agents converges to an efficient or optimal network-wide throughput (global reward).

7.3.4.1 Simulation Setup

This section discusses the simulation platform, objectives and performance metrics, scenario and assumptions, initialization, and parameters.

Simulation Platform. We have implemented a CR-enabled environment in the INET framework for OMNeT++ [72]. More explanations are found in Section 6.6 on page 108.

Simulation Objectives and Performance Metrics. The simulation scenarios consider heterogeneous channels such that each data channel at each agent has different levels of

- Q-value, $Q_i(a_i,a_{j\in\Gamma(i)})$ to indicate different levels of PUL and PER.

With heterogeneous channels consideration in all the simulation scenarios, the goal of the PP mechanism is

- To enable the global payoff value and global Q-value of the PP mechanism to converge to an optimal or efficient joint action. This means that the global payoff, $g_t(\mathbf{a_t})$ converges to a better value as time goes by; while the difference between the optimal global Q-value $Q_t(\mathbf{a_t^*})$ and instantaneous global Q-value for exploitation action $Q_t(\mathbf{a_t})$, (or the δ_t) converges to the value of 0.
- To minimize the convergence time of the PP mechanism.

Simulation Scenario and Assumptions. The simulation scenario is discussed in Section 7.1.6 and the assumptions are discussed in Section 7.1.4 and 7.3.1.4. Figure 7.2 shows the scenario and its graphical representation is shown in Figure 7.3.

Simulation Parameters. Table 7.3 shows the parameters in the simulation.

Simulation Initialization. 500 seconds of time are simulated. There are U=$V/2$=6 agents. Due to the limitation in channel sensing capability, there are K=3 data channels.

Network Topology. The network topologies are connected and there are three levels of network densities d=$\{Low,Medium,High\}$ with cyclic topology. The high density network simulates a single collision domain

scenario where all agents can hear each other; while the medium and low density networks have agents distributed in a uniform and random manner within a region of 300m$\times$300m and 600m$\times$600m respectively.

MARL Parameters. The Q-values characterize the channel heterogeneity properties for each data channel including PUL and PER. For a particular data channel, the Q-values are different among the agents as each of them observes different levels of PUL and PER. In short, the Q-values are Independent and Identically Distributed (i.i.d.) among the agents and the data channels. Higher Q-value indicates better local reward, and hence higher throughput. Each Q-value has a range of -5$\leq Q_i(a_i, a_{j\in\Gamma(i)}) \leq$15; and we consider that an efficient joint action has $\delta_t \leq \delta^{'}$=|15-(-5)|/2=10. So, a single agent that takes a non-optimal action could result in a non-efficient joint action. The Q-values are initialized and fixed throughout the simulation unless otherwise specified.

Payoff Message Exchange. The payoff is piggybacked on the RTS and CTS control messages, which are transmitted every 6.5ms on average, though it can be more often. However, too often control message exchange unnecessarily increases control overhead. The payoffs are transmitted using the control transceiver, hence neighbouring agents can detect the RTS and CTS messages. Each agent explores with a default probability of ε=0.1 when it transmits an RTS.

7.3.4.2 Simulation Results and Discussions

Simulation results are presented in three subsections. Firstly, we show that the PP mechanism converges to an optimal and efficient joint action. Secondly, we investigate the PP mechanism convergence time. Thirdly, we investigate the effects of unstable Q-values on δ_t.

Table 7.3: Notations and Default Parameter Settings in Simulation for Investigation into the Payoff Propagation Mechanism

Category	Symbol	Details	Values
Initialization	U	Number of SU agents	6
	K	Number of available data channels	3
	δ	Propagation delay	1ns
	T	Total simulation time	500s
		Simulation region size	{600m×600m, 300m×300m, 10m×10m}
MAC		Average time interval for RTS and CTS control message broadcast	6.5ms
PP	$Q(a_i, a_{j\in\Gamma(i)})$	Q-value	[-5,15]
	ε	Exploration probability	{0.02,0.1,0.5} Default: 0.1
	$\delta^{'}$	Difference between $Q_t(\mathbf{a_t^*})$ and $Q_t(\mathbf{a_t})$	10

Convergence to Optimal and Efficient Joint Action. Figure 7.7 shows that the global payoff, $g_t(\mathbf{a_t})$, which is calculated using (7.9), of the high, medium and low density networks increases and converges to a fixed point within 4s as the time advances, and becomes stable henceforth. The global payoff fluctuates occasionally due to exploration. The high density network has the highest level of global payoff because it is dependent on the number of neighbour agents as shown in (7.9). The global payoff does not grow without bound as shown in Proposition 2.

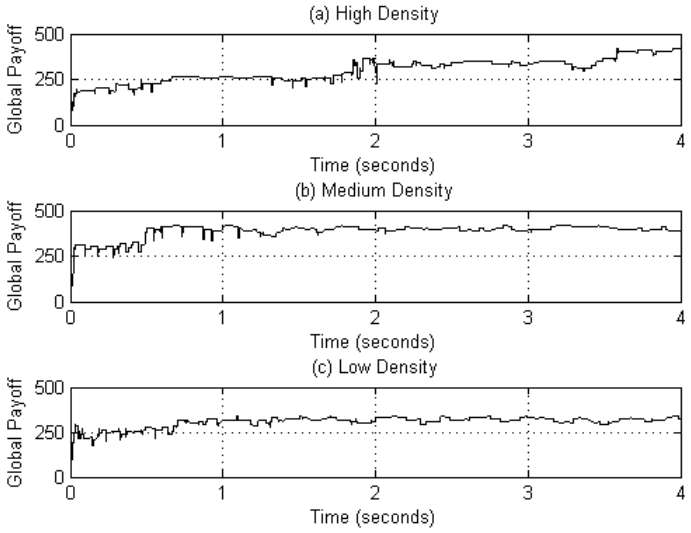


Figure 7.7: Global payoff for high, medium and low density networks.

According to Proposition 1, $\delta_t \rightarrow 0$ as time goes by, and maximizing the local payoff $g_{i,t}(a_{i,t})$ in (7.8) maximizes the local Q-value. Figure 7.8 shows that the δ_t for the high, medium and low density networks decreases to approximately zero value. This indicates the convergence to an optimal point. Note that in the δ_t computation, the Q-values of the exploration actions are replaced by the respective exploitation actions in order to provide smooth results. As discussed, according to Proposition 1, δ_t could increase at times because the μ-table is yet to become stable. As time goes by, the

μ-table becomes stable where α=0 such that the probability of $\delta_{t=n} \geq \delta_{t<n}$ is 0, hence no increment of δ_t is observed henceforth.

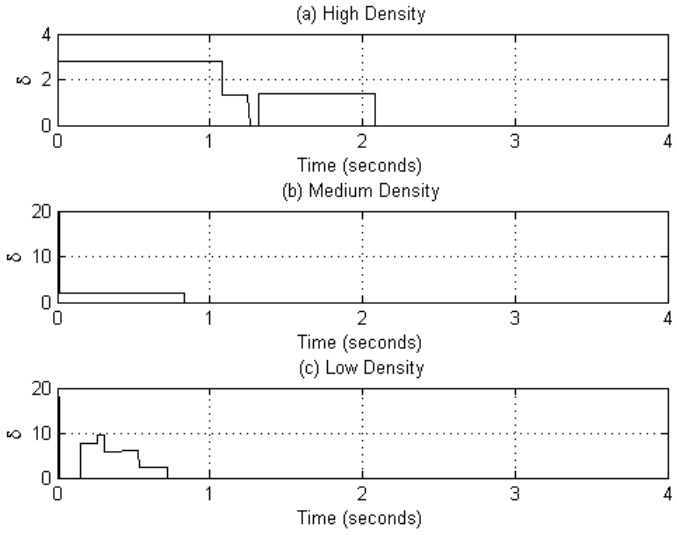


Figure 7.8: The δ_t for high, medium and low density networks.

Convergence Time. Next, simulations were performed on 20 different connected topologies for medium and low density networks respectively. For high density networks, which adopt the single collision domain assumption, simulations were performed using different random seeds. Figure 7.9 shows that all the simulation runs converge within a certain time range. On convergence to an optimal joint action for high density networks, 26% of the runs converge within 1s, and 50% within 1-2s; and the average convergence time is approximately 1.73s for high density, 1.51s for medium density, and 1.7s for low density networks. On convergence to an efficient joint action for high density networks, 74% of the runs converge within 1s, and 18% within 1-2s; and the average convergence time is approximately 0.8s for high density, 0.69s for medium density, and 0.88s for low density networks.

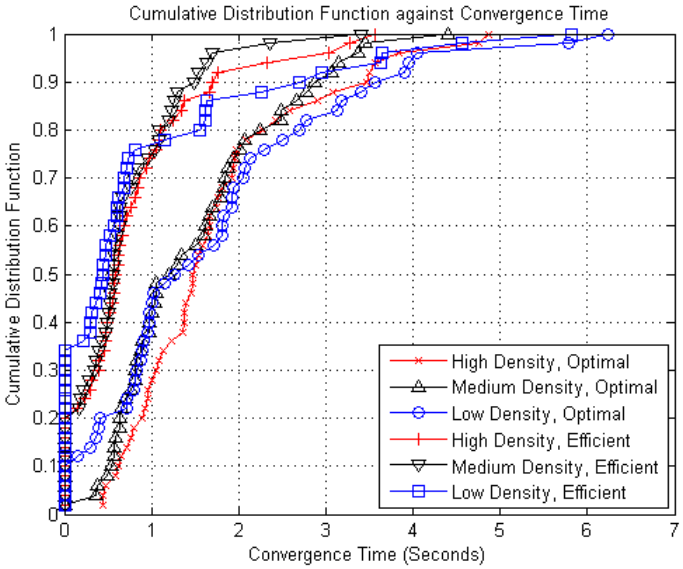


Figure 7.9: Cumulative distribution function of convergence time for high, medium and low density networks.

Figure 7.10 shows that, for high density networks, the convergence time is shorter on average with ε=0.1. The convergence time increases when ε=0.5 because of instability introduced by excessive exploration, and ε=0.02 because of low exploration. The average convergence time to an optimal joint action is 1.73s for ε=0.1, 2.87s for ε=0.5, and 3.98s for ε=0.02. The average convergence time to an efficient joint action is 0.8s for ε=0.1, 1.86s for ε=0.5, and 1.67s for ε=0.02. Hence, faster convergence is possible through the adjustment of ε. It should be noted that, in practice, the agents are expected to adapt to the dynamic operating environment even when they are already taking efficient and optimal joint action, and have knowledge of the operating environment; and hence, convergence must be rapid.

Effects of Unstable Q-values. Next, we examine the effects of unstable Q-values on δ_t in a high density network. Suppose, with every interval

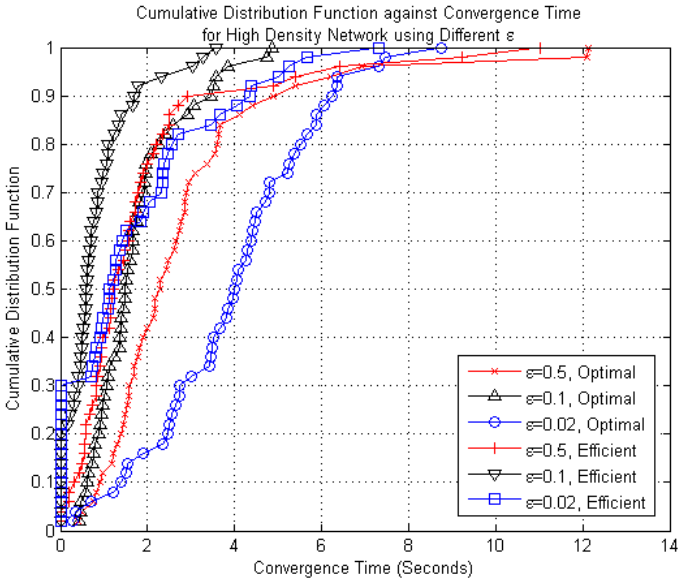


Figure 7.10: Cumulative distribution function of convergence time for high density network using different ε.

t_c of time, with probability 0.5, an agent changes its Q-value for any of its channels. Figure 7.11 shows that when t_c=4s, which is greater than the average convergence time of 1.73s, δ_t converges to 0 value. When t_c=2s, δ_t fails to converge within time 4-6s. When t_c=1s, δ_t fails to converge more frequently. The results indicate that stable Q-values provided by the local learning mechanism is the key factor for convergence as stated in Proposition 1.

7.3.5 Summary of Research Outcomes

The research outcomes from the investigation on the PP mechanism using the performance metrics of global payoff, $g_t(\mathbf{a_t})$; the difference between the optimal global Q-value $Q_t(\mathbf{a_t^*})$ and instantaneous global Q-value for exploitation action $Q_t(\mathbf{a_t})$, δ_t; and convergence time are summarized in this section. This section does not assume a single collision domain, which is

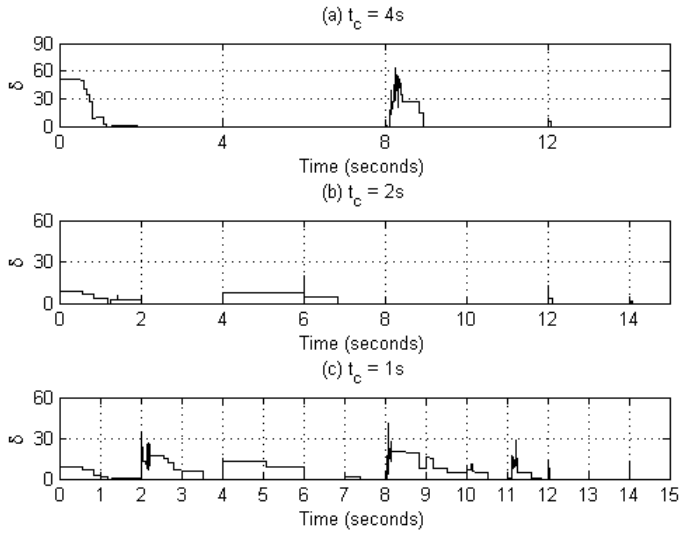


Figure 7.11: The δ_t for high density network with respect to t_c second.

commonplace in the CR networks research field to date. This section assumes that a local learning mechanism such as SARL is available at each agent to provide Q-values. The research outcomes are summarized as follows:

- The $g_t(\mathbf{a_t})$ converges to a fixed point, and the δ_t converges to the value of 0. Hence, the PP mechanism converges to an efficient and optimal joint action given that the entries in the Q-table and μ-table at each agent are stable and fixed.

- The $g_t(\mathbf{a_t})$ does not increase without bound in a cyclic topology.

- Low convergence time is possible through the adjustment of the exploration probability, ε.

7.4 Scenarios with Different Channel Conditions

7.4.1 Introduction

This section investigates the network-wide performance provided by the SARL and MARL approaches in scenarios with *identical* and *non-identical* channel conditions.

7.4.1.1 Differences Between Identical and non-Identical Channel Conditions

In scenarios with *identical* channel condition (or PER) at all the agents, the agents experience similar channel quality when using a particular data channel such that P=[P_1,. . .,P_K]. However, in scenario with *non-identical* channel condition at all the agents, each agent may experience different channel quality, or levels of PER, for using a particular data channel such that P_i=[$P_{i,1}$,. . .,$P_{i,K}$].

In both scenarios, from the perspective of each agent, the research question is "How does an agent choose its own data channel for data transmission such that the channel selection of all the agents provides network-wide performance enhancement?" The heterogeneous channels have different levels of PUL and PER. Based on the channel selection by all the agents, the data channels have different levels of contention. The additional challenge brought about by the scenario with non-identical channel condition at all the agents is the necessity to allocate data channels based on the levels of PER at each agent. This is because a data channel with low PER at agent i may not be the case at agent j. Hence, it is necessary to allocate data channels based on the levels of PER at each agent. The MARL approach considers network performance at neighour agents through the exchange of payoff message in order to maximize the global payoff. This improves the network performance at an agent and its neighbour agents.

Hence, we have chosen to investigate scenarios with non-identical channel condition at all the agents.

7.4.1.2 Section Organization

The remainder of this section is organized as follows. Section 7.4.2 presents an RL-based DCS scheme. Section 7.4.3 presents a general CSMA-based cognitive MAC protocol, and the MAC protocols with DCS implementation including Random-based MAC (RMAC), SARL-based MAC (S-MAC), enhanced SARL-based MAC (eSMAC), and MARL-based MAC (MMAC). Section 7.4.4 presents the simulation setup for the subsequent sections. Section 7.4.5 presents simulation results for scenarios with *identical* channel condition. It compares the network-wide performance achieved by the RMAC, SMAC and eSMAC. Section 7.4.6 presents simulation results for scenarios with *non-identical* channel condition. It compares the network-wide performance achieved by the RMAC, SMAC and MMAC.

7.4.2 Reinforcement Learning-based Dynamic Channel Selection

7.4.2.1 Objectives

In static DCRNs, the RL-based DCS scheme provides the strategy to select an available licensed data channel for data transmission from an SU transmitter to a static SU receiver given that the objective is to maximize overall throughput and minimize delay, in terms of number of channel switchings, in the presence of different levels of PUL, PER and channel contention in the licensed data channels. The PUL and PER are explained in Section 4.1 on page 44. Section 6.4.1 presents the RL model for a single-agent environment. This section presents the RL model for a multi-agent environment, where the RL model is applied in each agent i.

7.4.2.2 Reinforcement Learning Model for Dynamic Channel Selection in Multi-Agent Environments

Denote decision epochs by $t \in T = \{1,2,\ldots\}$, a constant epoch duration by t_D, action or channel selection by $c_j^i \in C$, and immediate reward by $r_{t+1}^i(c_{j,t}^i)$, which is the reward received at time $t+1$ for the data channel selected at time t. An agent i keeps track of the Q-value, $Q_t^i(c_j^i)$ within an interval of $[0, Q_{max}]$ for all the available data channels C in a Q-table with $|C|$ entries. The Q-value $Q_t^i(c_j^i)$, which represents the knowledge at agent i, indicates the appropriateness of choosing data channel c_j^i in the operating environment. In other words, the Q-value estimates the level of local reward for a data channel c_j^i; hence changes in the Q-value will lead to changes in an agent's channel selection. At each decision epoch t, agent i chooses a data channel c_j^i and receives a local reward $r_{t+1}^i(c_{j,t}^i)$ at time $t+1$.

Knowledge Update Procedure. During knowledge update, the Q-value of a chosen data channel $c_{j,t}^i$ at time t is updated at time $t+1$. Equation (6.1) is rewritten as follows:

$$Q_{t+1}^i(c_{j,t}^i) \leftarrow (1-\alpha) Q_t^i(c_{j,t}^i) + \alpha r_{t+1}^i(c_{j,t}^i) \tag{7.11}$$

where $0 \leq \alpha \leq 1$ is the learning rate, and $r_{t+1}^i(c_{j,t}^i)$ is the immediate reward. The higher the value of α, the greater the agent relies on the immediate reward. The reward $r_{t+1}^i(c_{j,t}^i) = N_D / t_D$ is the amount of throughput obtained within the recent epoch t, where N_D is the number of data packets successfully transmitted by the SU transmitter T_i within the epoch. Data packet transmission is successful when a link-layer acknowledgment is received for the data packet sent, else the transmission is unsuccessful. Additionally, if an SU senses PU signals immediately prior to transmission, it is considered unsuccessful. As time goes by, the agent receives a sequence of rewards from the data packet transmission procedure.

Action Selection Procedure. During action selection, for SMAC and eS-MAC, the agent chooses an exploitation or greedy action, which is the data channel with the highest Q-value. Equation (6.2) is rewritten as follows:

$$c_{j,t}^{i} = \underset{c_j^i \in C}{\operatorname{argmax}}\, Q_t^i(c_j^i) \tag{7.12}$$

The MMAC chooses its exploitation action using its own approach to be discussed in Section 7.4.3.5. The joint action affects the Q-value due to the dependency of actions among the agents. For example, two neighbour agents that choose a particular data channel may increase their contention level, and hence reduces their respective Q-values for the action.

Reinforcement Learning Model for Dynamic Channel Selection. The RL model in the DCS scheme for each agent is embedded in the SU transmitter as shown in Table 7.4.

Table 7.4: RL Model (MACC) at SU transmitter of Agent *i* for DCS

	Dynamic Channel Selection Model	
	Description	Representation
Action	Available data channels for data transmission.	C=$\{c_j^i$=1,2,...,$K\}$
Reward	Throughput within t_D.	$r_{t+1}^i(c_{j,t}^i) = N_D/t_D$

7.4.2.3 Major Differences Between RL Models for DCS in Single-Agent and Multi-Agent Environments

The major differences between the RL model for the SACC approach in Section 6.4.1 on page 91 and the RL model for the MACC approach in this chapter are as follows:

- The RL model for the SACC approach is embedded in the SU BS of centralized networks. The RL model for the MACC approach is embedded in the SU transmitter of each agent (or a communication node pair) in distributed networks.

- In the SACC approach, the Q-value is updated using (6.1) after every data packet transmission. In the MACC approach, the Q-value is updated using (7.11) at the end of every epoch $t \in T = \{1,2,\ldots\}$, hence the epoch duration is more than a single data packet transmission cycle.

- In the RL model for the SACC approach, the effects of actions to the environment is not considered. However, in the RL model for the MACC approach, the effects of actions to the environment is the contention level; and the use of throughput or $r^i_{t+1}(c^i_{j,t}) = N_D/t_D$ is a good measurement of the contention level. For instance, high levels of $r^i_{t+1}(c^i_{j,t})$ within t_D indicate low levels of contention and vice-versa. This is not possible in the RL model for the SACC approach as the epoch duration is a single data packet transmission cycle.

  Consider a situation where all the SU agents choose a similar data channel with low PUL and PER for data transmission. Using the SACC approach, the Q-value of the chosen data channel for all the SU agents would be high due to successful data packet transmissions. However, using the MACC approach, the Q-value of the chosen data channel for all the SU agents would be low due to unsuccessful data packet transmissions as a result of high contention level or low N_D within t_D epoch duration.

7.4.3 Cognitive MAC Protocols with Dynamic Channel Selection Implementation

7.4.3.1 Carrier Sense Multiple Access-based Cognitive Medium Access Control Protocol

At the time this thesis is written, there is not yet a standard available for a cognitive MAC protocol in DCRN. Section 6.3.3 provides related work on the cognitive MAC. Section 6.5.1 presents a CSMA-based cognitive MAC for centralized CR networks. This section presents a CSMA-based cognitive MAC with DCS implementation for DCRNs. The common control channel approach (see Section 3.3.3 on page 26) is adopted. Each SU is equipped with two transceivers, thus it is capable of accessing two different channels simultaneously (see Section 7.1.6). An illustration of the CSMA-based cognitive MAC protocol is shown in Figure 7.12. Four-way handshaking is performed to transmit the CTRL control message and DATA message in the common channel and the data channel respectively.

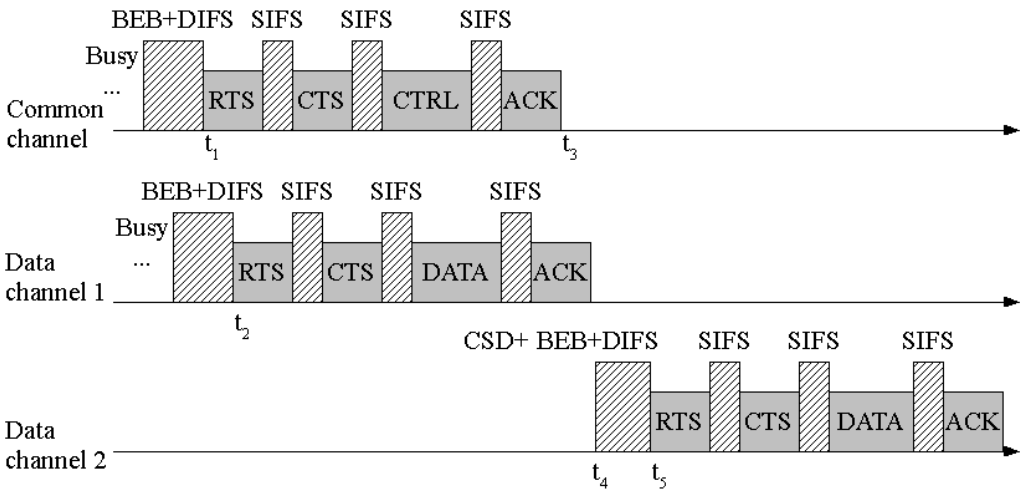


Figure 7.12: Procedure of cognitive MAC for an agent (or an SU communication node pair).

Channel Switching Procedure. For channel switching, the procedure is initiated by an SU transmitter T_i, and an example is shown in Figure 7.12. Agent *i* begins its control transmission cycle at time t_1. Transmitter T_i includes in the CTRL control message to its receiver R_i the channel switching information, including channel switching session ID and the data channel selected by the DCS scheme for data transmission at the data transceiver. At time t_3, T_i receives an Acknowledgement (ACK) message that indicates a successful channel switching; however, since the data transceiver has started its data transmission cycle at $t_2 \leq t_3$, both T_i and R_i switch their data channel at time t_4. The time incurred for the channel switching, initial channel sensing, backoff and Distributed Coordination Function (DCF) Inter-Frame Spacing (DIFS) is $t_5 - t_4$. In the event of failed channel switching, the data transceivers of T_i and R_i would tune to a different data channel, e.g., when the ACK message in the common channel is lost. Consequently, T_i fails to receive CTS for all the RTS it transmits at the data transceiver, and when the maximum backoff stage of 7 is reached, T_i starts another channel switching session.

Restarting Data Transmission Cycle at Data Transceivers. The data transceiver performs channel sensing for at least the SIFS interval immediately prior to all packet transmissions. If it senses a busy channel, it restarts its data transmission cycle to defer its transmission in order to avoid collision with the PU transmission. To improve the throughput performance in a CR environment where transmission is unreliable, the R_i restarts its data transmission cycle whenever it receives an RTS from T_i. Thus, missing CTS or failed data packet transmission does not result in RTS being dropped at the R_i whenever the T_i begins a new data transmission cycle.

Differences between CSMA-based Cognitive Medium Access Control Protocols in Centralized and Distributed Cognitive Radio Networks. There are two differences between the CSMA-based cognitive MAC in

centralized CR networks (see Section 6.5.1 on page 96), and the CSMA-based cognitive MAC in DCRNs in this section. The differences are as follows:

- In this section, four-way handshaking is performed to transmit the CTRL control message and DATA message in the common channel and the data channel respectively. The CTRL control message contains channel switching information, including channel switching session ID and the data channel selected by the DCS scheme for data transmission at the data transceiver. In Section 6.5.1, RTS and CTS control messages are transmitted in the common channel; while DATA and ACK messages are transmited in the data channel. The RTS and CTS messages contain the channel switching information. The purpose of implementing four-way handshaking in the common channel in this chapter is to improve transmission reliability due to the existence of multiple SU agents in the operating environment. Transmission reliability is important because failed transmission of CTRL control message results in failed channel switching at the data transceiver.

- In this section, both control transceiver and data transceiver can operate simultaneously. In Section 6.5.1, the control transceiver transmits RTS and CTS control messages, then the data transceiver transmits DATA and ACK messages.

Next, four types of CSMA-based cognitive MAC protocols based on different methods of DCS, namely RMAC, SMAC, eSMAC and MMAC are presented for later comparison. For each type, the mechanisms of channel switching, DCS, as well as the operation of the control and data transceivers are described.

7.4.3.2 RMAC

In Random-based MAC (RMAC), the DCS scheme chooses a data channel randomly. There are two conditions that trigger data channel switching at agent i. Firstly, an unsuccessful data packet transmission at the data interface when a transmitter T_i fails to receive an ACK message after a data packet transmission. Secondly, an agent must change its data channel at least once every second. This avoids all the agents choosing a particular data channel with low PUL and PER that provides higher occurence of successful data packet transmission at the expense of lower throughput due to a high level of contention. Also, an agent does not switch its data channel within a duration of two data transmission cycles right after a channel switching.

7.4.3.3 SMAC

In SARL-based MAC (SMAC), the DCS scheme applies the SARL approach to choose a data channel. Several functions for SMAC are shown in Algorithm 2, 3 and 4. Each agent divides the time horizon into epochs of duration t_D=$t_{D,SMAC}$ and keeps track of the number N_D of successful data packet transmissions in the past epoch (see Algorithm 2). No synchronisation is required among the agents. At the beginning of each epoch (see Algorithm 3), an agent uses N_D to update its Q-value using (7.11). With probability 1−ε or during exploitation, the agent chooses its data channel in the next epoch using (7.12). During exploitation, in order to improve stability, an agent does not switch its data channel if the difference between the Q-value of its previous exploitation data channel and the current optimal (or near-optimal) data channel using (7.12) is less than a small threshold value of β. With probability ε or during exploration, the agent chooses its data channel in the next epoch randomly; and the agent is not allowed to explore for two consecutive epochs. Although an agent has decided to switch its data channel at the beginning of an epoch, it is on-

Algorithm 2 Function for SMAC, eSMAC and MMAC: Receives ACK for DATA Packet Sent

if (receive ACK for DATA packet sent) **then**

$N_D = N_D + 1$

end if

ly carried out in the midst of an epoch when a new control transmission cycle starts, which is subject to contention among the agents. The control transmission cycle is necessary so that the transmitter can send channel switching information to its receiver for both to use the same data channel for data transmission. Hence, immediately prior to a data channel switching, the transmitter T_i must update the Q-value of its initial data channel which has been learned (see Algorithmn 4). Upon channel switching, it sets N_D=0 and continues to operate in the remaining epoch.

7.4.3.4 eSMAC

According to [64], the SARL approach in SMAC results in instability or oscillations in the presence of multiple agents because each agent switches its data channel from time to time. The SARL approach in enhanced SARL-based MAC (eSMAC) enhances stability through reducing the number of channel switchings at each agent. eSMAC addresses two drawbacks in SMAC that contribute to the instability, and the drawbacks are as follows:

- When several agents undertake exploration at the same time, the Q-values (or the throughput performance) become unstable and they do not portray the exact level of PUL, PER and contention of the data channels. For instance, when two agents explore a particular data channel, the Q-value for the data channel reduces for all agents and does not portray the exact level of contention.

- An agent that explores a particular data channel, and then exploits the other one in the following epoch causes the Q-values of both data

Algorithm 3 Function for sMAC: New Epoch Begins

if (new epoch begin) **then**
  $r^i_{t+1}(c^i_{j,t}) = N_D/t_D$
  $Q^i_{t+1}(c^i_{j,t}) \leftarrow (1-\alpha)Q^i_t(c^i_{j,t}) + \alpha r^i_{t+1}(c^i_{j,t})$ {Refer to (7.11)}
  $R = \text{uniform}(0,1)$ {Generate random number}
  if $R \leq \varepsilon$ **then**
    $c^i_{j,t+1} = \text{uniform}(1,K)$
  else
    $c^i_{temp} = \underset{c^i_j \in C}{\text{argmax}}\, Q^i_t(c^i_j)$ {Refer to (7.12)}
    if $|Q^i_{t+1}(c^i_{temp}) - Q^i_{t+1}(c^i_{j,t})| \leq \beta$ **then**
      $c^i_{j,t+1} = c^i_{j,t}$
    else
      $c^i_{j,t+1} = c^i_{temp}$
    end if
  end if
  return $c^i_{j,t+1}$
end if

Algorithm 4 Function for SMAC and MMAC: Control Transmission Cycle Begins

if (control transmission cycle begin) **then**
  if $c^i_{j,t+1} \neq c^i_{j,t}$ **then**
    {initiate channel switching}
    $r^i_{t+1}(c^i_{j,t}) = N_D/t_D$
    $Q^i_{t+1}(c^i_{j,t}) \leftarrow (1-\alpha)Q^i_t(c^i_{j,t}) + \alpha r^i_{t+1}(c^i_{j,t})$
    $N_D = 0$
  end if
end if

channels in itself and its neighbour agents to fluctuate.

The purpose of eSMAC is to provide stability to the existing SARL approach. The instability is caused by the exploration. Several functions for eSMAC are shown in Algorithm 2, 5, 6 and 7.

To tackle the first drawback, an agent would only explore if its neighbour agents are not exploring (`nbrExplorationBit = false`), and it must announce to its neighbour agents in a CTRL control message when it starts (`pktExplorationBit = true`) and terminates (`pktExplorationBit = false`) its exploration. This is to ensure that there is only a single agent undergoing exploration within a neighbourhood.

To tackle the second drawback, the exploring agent and its neighbour agents must update and store the Q-tables and set N_D=0 during data channel switching in order to learn a new environment whenever the exploration begins. At the end of the exploration, using (7.12), the exploring agent chooses to exploit the data channel being explored or to exploit the other data channel. The agent would have to retrieve its stored Q-table and set N_D=0 if it chooses to exploit the other data channel, otherwise it would maintain its Q-table. The decision is broadcast to the neighbour agents in CTRL control message so that the neighbour agents follow suit to retrieve (`pktRetrieveQtable = true`) or maintain (`pktRetrieveQtable = false`) their Q-tables, and to set N_D=0.

7.4.3.5 MMAC

Similar to the eSMAC, the MARL approach in MARL-based MAC (MMAC) enhances stability through reducing the number of channel switchings at each agent. The MARL approach is a combination of both the SARL in Section 7.4.2 and the extended PP in Section 7.3.3. The SARL approach, which is the learning engine embedded in each agent, provides the local reward, while the PP mechanism provides a means of communication for

Algorithm 5 Function for eSMAC: New Epoch Begins

if (new epoch begin) **then**
  $r^i_{t+1}(c^i_{j,t}) = N_D/t_D$
  $Q^i_{t+1}(c^i_{j,t}) \leftarrow (1-\alpha)Q^i_t(c^i_{j,t}) + \alpha r^i_{t+1}(c^i_{j,t})$ {Refer to (7.11)}
  $R = \text{uniform}(0,1)$ {Generate random number}
  if ($R \leq \varepsilon$) && (*nbrExplorationBit* == false) **then**
    $c^i_{j,t+1} = \text{uniform}(1, K)$
  else
    $c^i_{temp} = \underset{c^i_j \in C}{\text{argmax}}\, Q^i_t(c^i_j)$ {Refer to (7.12)}
    if $|Q^i_{t+1}(c^i_{temp}) - Q^i_{t+1}(c^i_{j,t})| \leq \beta$ **then**
      $c^i_{j,t+1} = c^i_{j,t}$
    else
      $c^i_{j,t+1} = c^i_{temp}$
      if (explore during the previous epoch t) **then**
        retrieveQtable = true
      end if
    end if
  end if
  return $c^i_{j,t+1}$
end if

Algorithm 6 Function for eSMAC: Control Transmission Cycle Begins

```
if (control transmission cycle begin) then
  if c^i_{j,t+1} ≠ c^i_{j,t} then
    {initiate channel switching}
    r^i_{t+1}(c^i_{j,t}) = N_D / t_D
    Q^i_{t+1}(c^i_{j,t}) ← (1 − α)Q^i_t(c^i_{j,t}) + α r^i_{t+1}(c^i_{j,t})
    N_D = 0
    if (explore) then
      store Q-table
      pktExplorationBit = true
    else
      {exploit}
      if retrieveQtable == true then
        pktRetrieveQtable = true
        retrieve Q-table
        retrieveQtable = false
      end if
    end if
  end if
end if
```

Algorithm 7 Function for eSMAC: Receives CTRL Control Message

if (receive CTRL) **then**
  if *pktExplorationBit* == true **then**
    nbrExplorationBit = true
    $c^i_{j,t+1} = c^i_{j,t}$
    $r^i_{t+1}(c^i_{j,t}) = N_D/t_D$
    $Q^i_{t+1}(c^i_{j,t}) \leftarrow (1-\alpha)Q^i_t(c^i_{j,t}) + \alpha r^i_{t+1}(c^i_{j,t})$
    store Q-table
    $N_D = 0$
  else
    nbrExplorationBit = false
    if *pktRetrieveQtable* == true **then**
      retrieve Q-table
      $N_D = 0$
    end if
  end if
end if

the learning engines. Each agent *i* maintains a Q-table with $|A|$ entries; and a μ-table with size $N_{n,i} \times |A|$ to keep track of the payoff messages. The $N_{n,i}$ is the number of neighbour agents of agent *i*.

In MMAC, the DCS scheme applies the MARL approach to choose a data channel. Several functions for MMAC are shown in Algorithms {1, 2, 4, 8}. Each agent divides the time horizon into epochs of duration t_D=$t_{D,MMAC}$. Each agent keeps track of the number N_D of successful data packet transmissions and exchanges payoff messages (7.7) within $t_{D,MMAC}$. Note that, an agent broadcasts its payoff value during exploitation only. The Q-values in the payoff message indicate the performance of each agent during exploitation or the recent exploration if any of the agents is undergoing exploration. No synchronisation is required although the neighbour agents are expected to broadcast at least one payoff message within $t_{D,MMAC}$ to inform the exploring agent of their respective Q-value if any of the agents is undergoing exploration. At the beginning of each epoch, an agent updates its Q-values using N_D and payoff messages received from its neighbour agents. Equation (7.11) is used to update the Q-values, and the payoff message is used to update the stored μ-values. During exploitation, an agent computes the local payoff value for each data channel using (7.8), and approximates and chooses an optimal action with the maximum payoff value using (7.4).

7.4.4 Simulation Setup

This section discusses the simulation platform, objectives and performance metrics, ordinates, baseline, parameters and organization of the remaining sections relevant to the simulation. This covers the simulation experiments, results and discussions in Section 7.4.5 and 7.4.6.

Algorithm 8 Function for MMAC: New Epoch Begins

if (new epoch begin) **then**
  $r^i_{t+1}(c^i_{j,t}) = N_D/t_D$
  $Q^i_{t+1}(c^i_{j,t}) \leftarrow (1-\alpha)Q^i_t(c^i_{j,t}) + \alpha r^i_{t+1}(c^i_{j,t})$ {Refer to (7.11)}
  $R = \text{uniform}(0,1)$ {Generate random number}
  if $R \leq \varepsilon$ **then**
    $c^i_{j,t+1} = \text{uniform}(1, K)$
  else
    $c^i_{temp} = \underset{c^i_j \in C}{\text{argmax}} g_{i,t+1}(c^i_j)$ {Refer to (7.4)}
    if $|g_{i,t+1}(c^i_{temp}) - g_{i,t+1}(c^i_{j,t})| \leq \beta$ **then**
      $c^i_{j,t+1} = c^i_{j,t}$
    else
      $c^i_{j,t+1} = c^i_{temp}$
    end if
  end if
  return $c^i_{j,t+1}$
end if

7.4.4.1 Simulation Platform

We have implemented a CR-enabled environment in the INET framework for OMNeT++ [72]. More explanations are found in Section 6.6 on page 108.

7.4.4.2 Simulation Objectives and Performance Metrics

The simulation scenarios consider heterogeneous data channels such that each channel has different levels of PUL and PER.

Section 7.4.5 investigates identical channel condition (or PER) at all the agents; and Section 7.4.6 investigates non-identical channel condition at all the agents.

With heterogeneous channels consideration in all the simulation scenarios, the goals of the DCS are

- To enable the global Q-value of SMAC, eSMAC and MMAC converges to better Q-value as time goes by.
- To maximize throughput.
- To minimize the number of channel switchings, since switching causes non-negligible delay for data packet transmission. Additionally, each channel switch also causes energy consumption. Note that, in contrast to the simulation results in Chapter 6 for the SACC approaches, channel switchings for exploration purpose are *not* counted in this chapter because all the approaches in the comparison undergo exploration.

In other words, the SUs agents must select their data channels (actions) respectively for data transmission such that the data channel selection (joint action) by all the SU agents converges to network-wide throughput and number of channel switchings (global reward) that provide network-wide performance enhancement.

7.4.4.3 Simulation Ordinates

Graphs are presented with PUL and PER as ordinate. For each value of PUL and PER, the corresponding throughput or number of channel switchings is the average value of 50 runs using different levels of PULs and PERs across the data channels. For instance, a PUL level of 0.2 may indicate the PUL of [0.025,0.248,0.327] or [0.163,0.402,0.035] in the data channels.

7.4.4.4 Simulation Baseline

A common simulation baseline is the RMAC. The RMAC chooses an available data channel for the next data packet transmission in a random manner. Hence, it does not apply any learning mechanism.

7.4.4.5 Simulation Parameters

Table 7.5 shows the simulation parameters that are applicable to simulation scenarios in Section 7.4.5 and 7.4.6. The procedure of the cognitive MAC is shown in Figure 7.12. Additional simulation parameters that are applicable to specific simulation scenarios are shown in seperate tables in Table 7.6 for Section 7.4.5, and Table 7.7 for Section 7.4.6.

We explain some of the simulation parameters in Table 7.5. The characteristics of the PU, SU, SU agent and channel are discussed in Section 7.1.6. Exactly 100 seconds of time are simulated in each run. Each SU has limited in channel sensing capability, and thus the number of available licensed and orthogonal data channels is limited to K=3. There are U=$\{3,6,12\}$ SU agents in a square simulation area of 1000m$\times$1000m. Three levels of network densities are simulated with d=U/K=$\{1,2,4\}$=$\{Low,Medium,High\}$.

Table 7.5: Notations and Default Parameter Settings in Simulation

Category	Symbol	Details	Values
Initial ization	U	Number of SU agents	{3,6,12}
	K	Number of available data channels	3
	δ	Propagation delay	1ns
	T	Total simulation time	100s
SU		Traffic model	Always backlogged
	$t_{H+C,SU}$	Data packet duration	5.44ms
	T_{SW}	Channel switching delay (including initial channel sensing)	2ms
PU		Traffic model	Stochastic channels with Poisson model
	$t_{H+C,PU}$	Data packet duration	5.44ms
		Maximum queue size	5 packets
	L_{c_i}	PUL of each available data channel	[0,0.9] Default: 0.5
SMAC, eSMAC, MMAC	α	Learning rate	{0.05,0.1,0.2,0.4} Default: 0.2
	ε	Exploration probability	{0.05,0.1,0.2,0.4} Default: 0.2
	β	Q-value threshold value	1
		Initial Q-value	1
	Q_{max}	Maximum Q-value	20

7.4.4.6 Section Organization

The remainder of this chapter present simulation experiments, results and discussions, and is organized as follows:

- Section 7.4.5 shows the investigation on the SARL approaches, namely RMAC, SMAC and eSMAC, in scenario with *identical* channel condition at all the agents.

- Section 7.4.6 shows the investigation on the SARL approaches, namely RMAC and SMAC, as well as an MARL approach, namely M-MAC, in scenario with *non-identical* channel condition at all the agents.

7.4.5 Scenario with Identical Channel Condition

This section presents cognitive MAC protocols with three different kinds of DCS implementations based on the SARL approaches, namely RMAC, SMAC and eSMAC. The main focus is the performance enhancement provided by the SARL approaches.

7.4.5.1 Assumptions

Section 7.1.4 shows the assumptions applicable to this section, and the additional assumptions are as follows:

- Identical channel condition (or PER) at all the agents for a particular data channel such that P=[P_1,. . .,P_K]. The differences between identical and non-identical channel condition at all the agents are discussed in Section 7.4.1.1.

- Single collision domain. This means that an interference link exists among all the agents.

7.4.5.2 Contributions

The focus in this section is the application of the SMAC and eSMAC approaches in scenario with identical channel condition. The contributions of this section are as follows:

- To show that the SMAC and eSMAC approaches achieve a joint action that provides better network-wide performance in DCS for D-CRNs.

- To show the effects of network density and various essential parameters in SMAC and eSMAC approaches on network-wide performance.

7.4.5.3 Simulation Experiments, Results, and Discussions

The objectives of this investigation are presented in Section 7.4.4.2.

Simulation Setup and Parameters. The simulation scenario is discussed in Section 7.1.6 and the assumptions are discussed in Section 7.4.5.1. Figure 7.2 shows the scenario and its graphical representation is shown in Figure 7.3. The procedure of the cognitive MAC is shown in Figure 7.12. Simulation parameters are shown in Table 7.5, and additional simulation parameters are shown in Table 7.6.

Some of the important explanations on Table 7.6 are shown below:

Identical Channel Quality (or PER) at all the Agents. Each data channel has a certain level of PER. The level of PER for each data channel is the same for all agents with the default average value of PER across the K data channels being 0. Upon receiving a packet, an SU discards the packet with the PER probability.

Table 7.6: Notations and Default Parameter Settings in Simulation for Investigation into Scenario with Identical Channel Condition

Category	Symbol	Details	Values
Initial ization	$P_{c_i}^E$	PER of each available data channel at agent i	[0,0.3] Default: 0
SU	$t_{CTRL,SU}$	CTRL control message packet duration	5.44ms
SMAC and eSMAC	$t_{D,SMAC}$	Epoch duration	187.14ms

Secondary User. For SU, $t_{DATA,SU}$=$t_{CTRL,SU}$ because the CTRL control message may contain other broadcast information. It also contains information related to channel switching

SMAC and eSMAC. An epoch duration is 30 data transmission cycles, or t_D=$t_{D,SMAC}$=30×(t_{RTS}+t_{CTS}+$t_{DATA,SU}$+t_{ACK}+3t_{SIFS}). The value of $t_{D,SMAC}$ was chosen empirically to provide the best possible network-wide performance.

Simulation Results and Discussions. Firstly, we show that the global Q-value of SMAC and eSMAC stabilizes as time goes by. Secondly, we investigate the effects of network density on network-wide performance. Thirdly, we investigate the effects of α and ε on network-wide performance.

Stabilization of Global Q-value. Figure 7.13 shows that the instantaneous global Q-value for the exploitation data channel for SMAC and eSMAC increases and becomes stable as time goes by in a medium density network. In other words, the agents attain a joint action that provides better network-wide performance. The PUL is L_{c_i}=0.5 with [0.5,0.5,0.5] across the K=3 data channels, and PER is $P_{c_i}^E$=0 with [0,0,0]. The SMAC

and eSMAC parameters are α=0.2 and ε=0.2. With U=6 and Q_{max}=20, the maximum global Q-value is 120. Although L_{c_i}=0.5 for all data channels, due to the Poisson traffic model, the data channels have different levels of PUL at a particular time instant. The SMAC achieves slightly higher global Q-value compared to eSMAC because it can explore the data channels at any time to discover a better data channel; while in eSMAC, an agent can only explore if none of its neighbour agents are doing so.

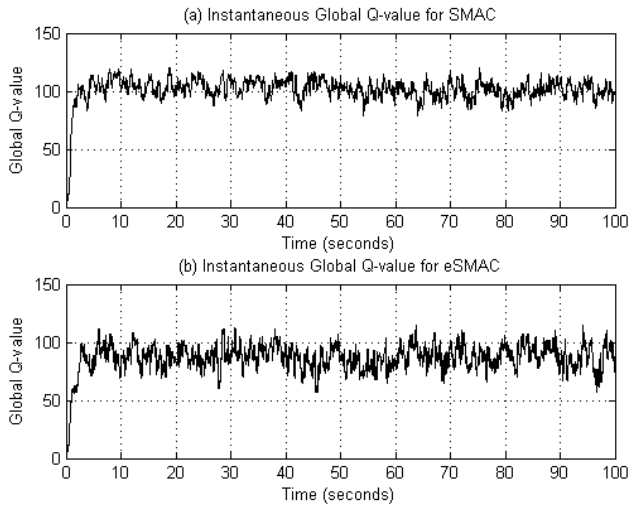


Figure 7.13: Global Q-value for the exploitation data channel for SMAC and eSMAC in a medium density network. PUL L_{c_i}=0.5. PER $P_{c_i}^E$=0.

Effects of Network Density on Network-wide Performance. Figure 7.14 shows the mean throughput for each agent against various levels of mean PUL for RMAC, SMAC and eSMAC in low, medium, and high density networks. The PER is $P_{c_i}^E$=0 across the K=3 data channels. The SMAC and eSMAC parameters are α=0.2 and ε=0.2. The eSMAC achieves the highest amount of throughput, followed by SMAC and RMAC in all types of network densities; and the throughput enhancement offered by the eSMAC and SMAC compared to RMAC reduces as the network density

increases. At PUL L_{c_i}=0.5, the eSMAC outperforms the RMAC by 38%, 14% and 1% in low, medium and high density networks respectively. The throughput achieved by eSMAC is slightly higher than SMAC in all cases. Figure 7.15 shows the equivalent graph with PER as ordinate and PUL is L_{c_i}=0.5 with [0.5,0.5,0.5], and a similar trend is observed. In short, in a high density network or as $d \rightarrow \infty$, the throughput enhancement achieved by SMAC and eSMAC approaches 0. We believe that this happens in most intelligence methods due to the high contention level.

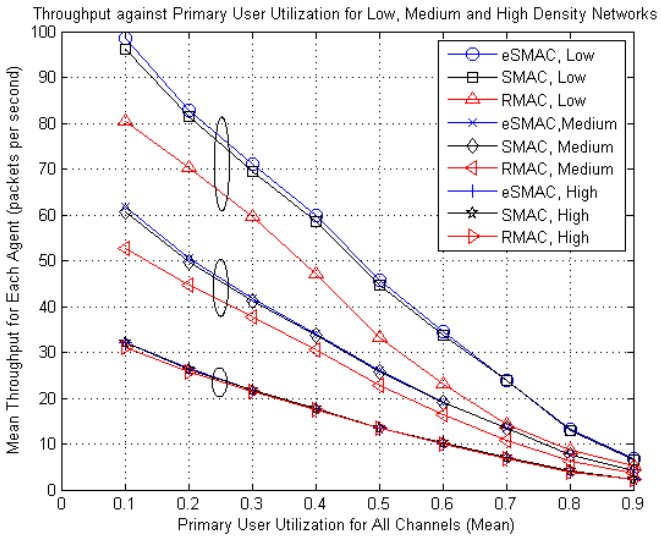


Figure 7.14: The mean throughput for each agent against mean PUL for RMAC, SMAC and eSMAC in low, medium and high density networks. PER $P_{c_i}^E$=0.

Figure 7.16 shows the mean number of exploitation channel switchings for each agent against various levels of mean PUL for SMAC and eSMAC in low, medium and high density networks. The PER is $P_{c_i}^E$=0. The SMAC and eSMAC parameters are α=0.2 and ε=0.2. The eSMAC achieves significantly lower number of channel switchings, hence it provides higher stability. At PUL L_{c_i}=0.5, the number of channel switchings in eSMAC is 22%, 20% and 44% of that in SMAC in low, medium and high density

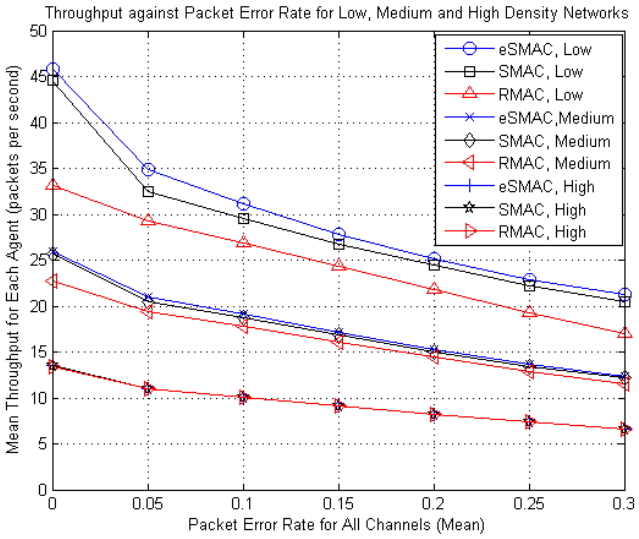


Figure 7.15: The mean throughput for each agent against mean PER for RMAC, SMAC and eSMAC in low, medium and high density networks. PUL L_{c_i}=0.5.

networks respectively. Generally speaking, an agent switches its exploitation data channel because the difference between the Q-values among the data channels is greater than the threshold β=1, and the agent exploits a better data channel. There are two reasons an agent does not switch its data channel. Firstly, all the data channels provide equal levels of network performance, hence an agent exploits the same data channel such as the case at L_{c_i}=0.6 in low density network for eSMAC. Secondly, all the data channels provide very good or very poor network performance, and hence the Q-values approach the Q-value's limit, specifically, $Q_t(a) \rightarrow Q_{max}$ or $Q_t(a) \rightarrow 0$ for $\forall a \in A$. For instance, in high density networks, all the K=3 data channels have high contention level and thus the number of channel switchings is low compared to the medium density networks. Due to the aforementioned reasons, in situation of high density or high PUL, the variance in the number of channel switchings becomes high. Figure 7.17 shows the equivalent graph with PER as ordinate and PUL L_{c_i}=0.5 with

[0.5,0.5,0.5].

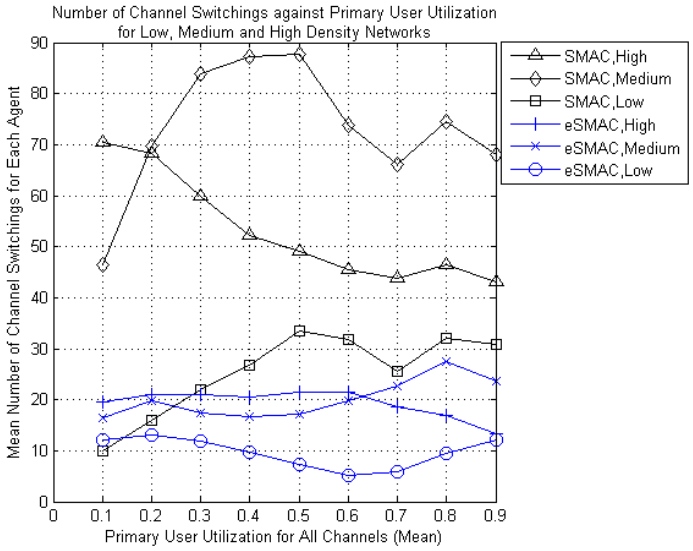


Figure 7.16: The mean number of channel switchings of the exploitation data channel for each agent against mean PUL for SMAC and eSMAC in low, medium and high density networks. PER $P_{c_i}^E$=0.

Effects of α and ε on Network-wide Performance. The effects of α and ε on the throughput performance are shown first, followed by their effects on the number of channel switchings.

Figure 7.18 shows the effects of α on the mean throughput for each agent for various levels of mean PUL for SMAC and eSMAC in medium density networks. The PER is $P_{c_i}^E$=0. The effects of α are insignificant on the mean throughput for each agent. Figure 7.19 shows the equivalent graph with PER as ordinate and PUL L_{c_i}=0.5 with [0.5,0.5,0.5]. A similar experiment is performed to investigate the effects of ε on the mean throughput, with results shown with respect to PUL in Figures 7.20 and with respect to PER in Figure 7.21. The effects of ε on the mean throughput are insignificant.

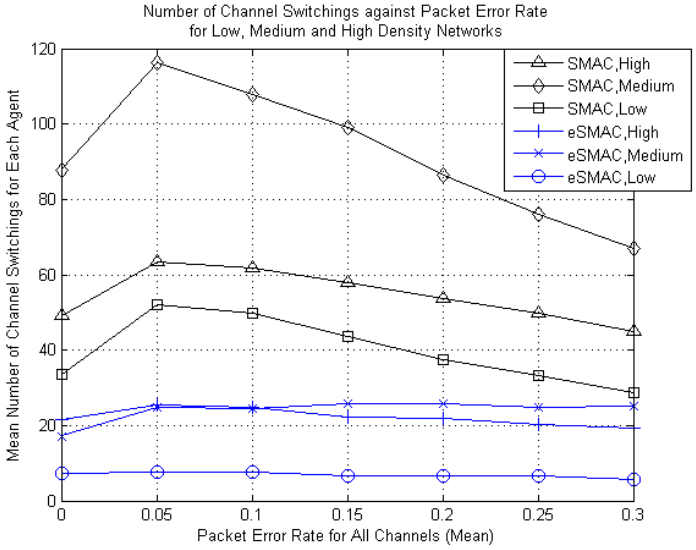


Figure 7.17: The mean number of channel switchings of the exploitation data channel for each agent against mean PER for SMAC and eSMAC in low, medium and high density networks. PUL L_{c_i}=0.5.

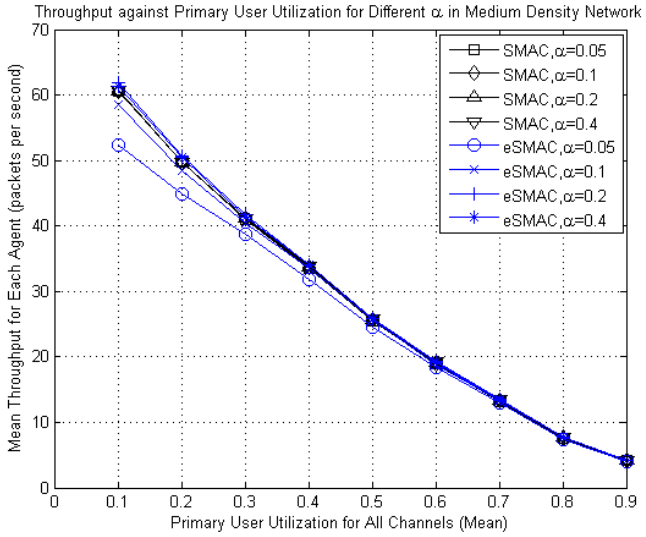


Figure 7.18: The mean throughput for each agent against mean PUL for SMAC and eSMAC with different α values in medium density networks. PER $P_{c_i}^E$=0. ε=0.2.

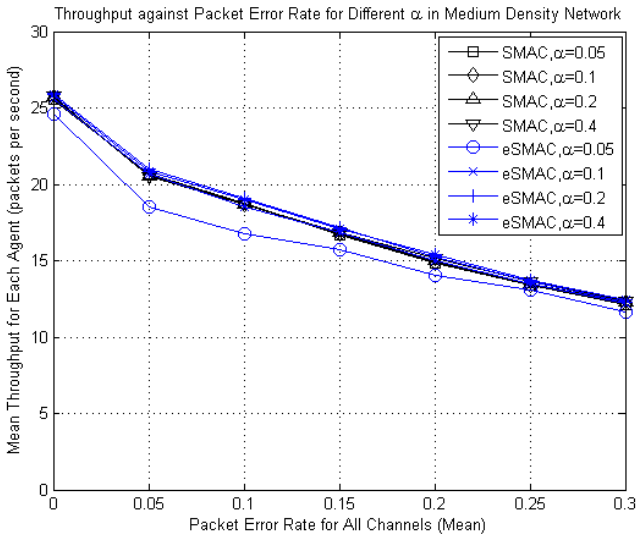


Figure 7.19: The mean throughput for each agent against mean PER for SMAC and eSMAC with different α values in medium density networks. PUL L_{c_i}=0.5. ε=0.2.

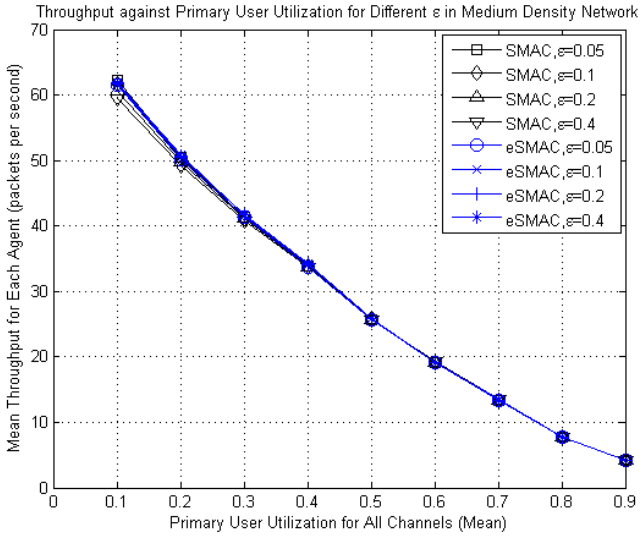


Figure 7.20: The mean throughput for each agent against mean PUL for SMAC and eSMAC with different ε values in medium density networks. PER $P_{c_i}^E$=0. α=0.2.

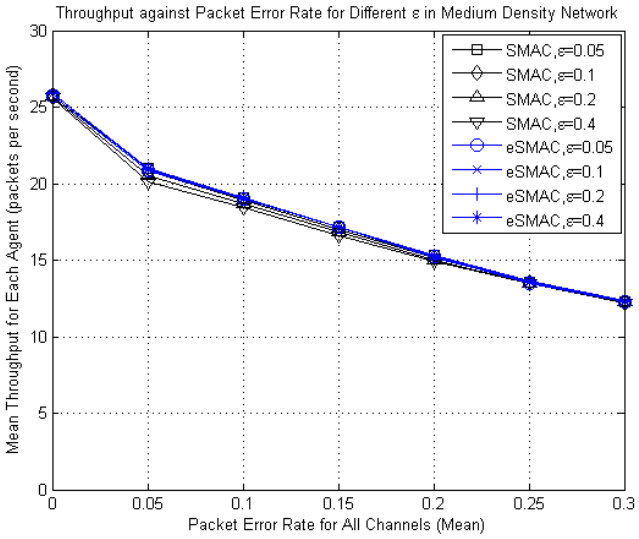


Figure 7.21: The mean throughput for each agent against mean PER for SMAC and eSMAC with different ε values in medium density networks. PUL L_{c_i}=0.5. α=0.2.

Figure 7.22 shows the effects of α on the mean number of channel switchings of the exploitation data channel for each agent against various levels of mean PUL for SMAC and eSMAC in medium density networks. The PER is $P_{c_i}^E$=0. The number of channel switchings increases with α for all the cases. In short, lower values of α provide higher stability; however, in eSMAC, Figure 7.18 shows that throughput performance is better with α=0.2. For instance, there is throughput enhancement of 18% for α=0.2 compared to α=0.05 at L_{c_i}=0.1, but it is just 5.4% at L_{c_i}=0.5. Figure 7.23 shows the equivalent graph with PER as ordinate and PUL L_{c_i}=0.5 with [0.5,0.5,0.5]. A similar experiment is performed to investigate the effects of ε on the mean number of channel switchings, with results shown with respect to PUL in Figures 7.24 and with respect to PER in Figure 7.25, which share similar trends to Figure 7.22 and 7.23 respectively.

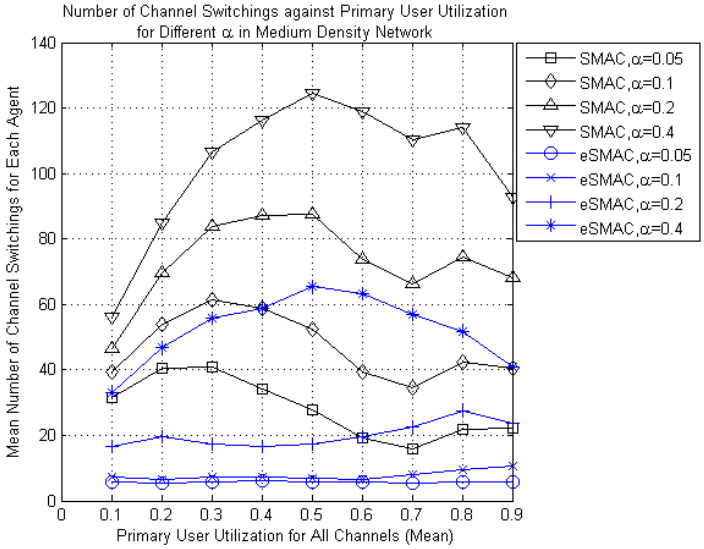


Figure 7.22: The mean number of channel switchings of the exploitation data channel for each agent against mean PUL for SMAC and eSMAC with different α values in medium density networks. PER $P_{c_i}^E$=0. ε=0.2.

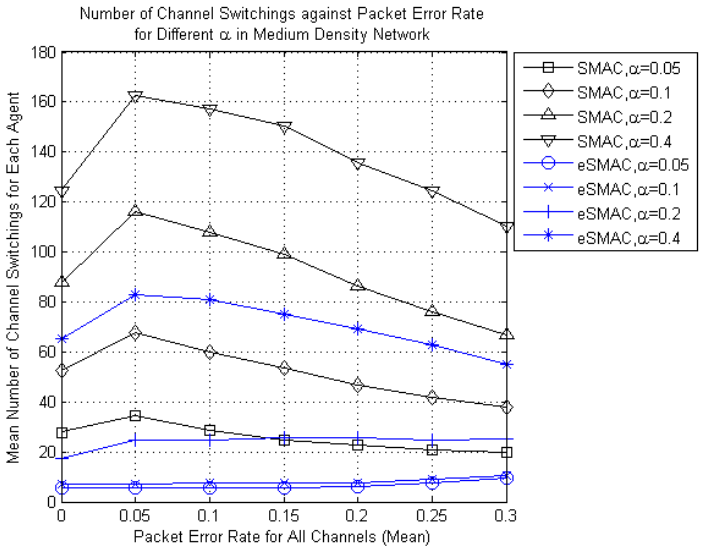


Figure 7.23: The mean number of channel switchings of the exploitation data channel for each agent against mean PER for SMAC and eSMAC with different α values in medium density networks. PUL L_{c_i}=0.5. ε=0.2.

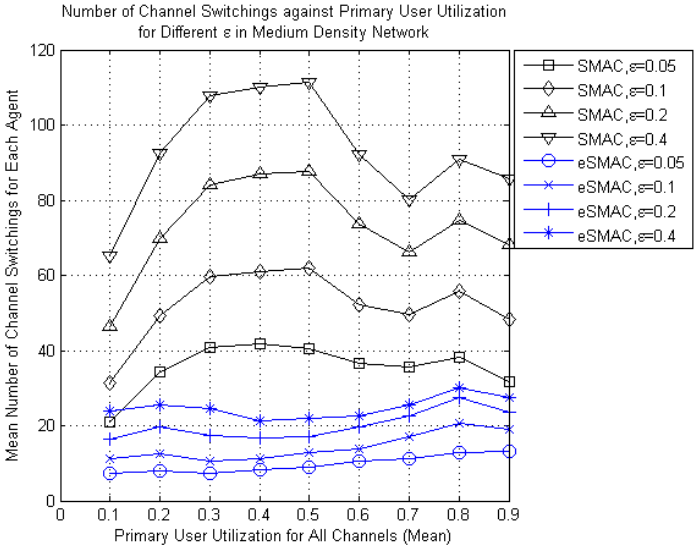


Figure 7.24: The mean number of channel switchings of the exploitation data channel for each agent against mean PUL for SMAC and eSMAC with different ε values in medium density networks. PER $P_{c_i}^E = 0$. α=0.2.

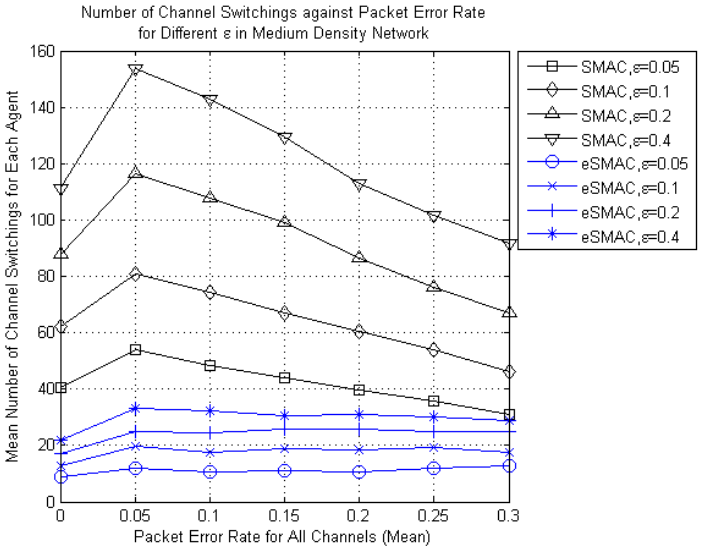


Figure 7.25: The mean number of channel switchings of the exploitation data channel for each agent against mean PER for SMAC and eSMAC with different ε values in medium density networks. PUL $L_{c_i} = 0.5$. α=0.2.

7.4.5.4 Summary of Research Outcomes

The research outcomes from the investigation on the SARL approaches, including SMAC and eSMAC, in order to implement the MACC model in scenarios with identical channel condition using the performance metrics of network-wide throughput and number of channel switchings are summarized as follows:

- The global Q-value for the exploitation data channel for SMAC and eSMAC increases and becomes stable as time goes by.
- The eSMAC approach achieves the highest amount of throughput, followed by SMAC and RMAC in most of the cases with respect to PUL and PER in low, medium and high density networks.
- In high density networks or $d \rightarrow \infty$, the throughput enhancement achieved by SMAC and eSMAC approaches 0.
- The number of channel switchings in eSMAC is significantly lower than that in SMAC.
- In SMAC, lower values of α and ε provide higher throughput and lower number of channel switchings (or better stability).
- In eSMAC, the values of α provide approximately similar level of throughput, however, the throughput decreases if $\alpha < 0.2$. Lower values of α provide lower number of channel switchings. As for ε, lower values provide lower number of channel switchings while achieving approximately similar level of throughput.

7.4.6 Scenario with non-Identical Channel Condition

This section presents cognitive MAC protocols with three different kinds of DCS implementations, namely RMAC, SMAC and MMAC. The SMAC is based on the SARL approach, while the MMAC is based on the MARL

approach. It newly implements the MARL approach, which is a combination of both SARL and the PP mechanism, to further enhance the network-wide performance. The main focus is the performance enhancement provided by the SARL and MARL approaches.

7.4.6.1 Assumptions

Section 7.1.4 shows assumptions applicable to this section, and the additional assumptions are as follows:

- Non-identical channel condition (or PER) at all the agents for a particular data channel such that P_i=[$P_{i,1}$,. . .,$P_{i,K}$]. The differences between identical and non-identical channel condition at all the agents are discussed in Section 7.4.1.1.

- Single collision domain. This means that interference link exists among all the agents.

7.4.6.2 Contributions

The focus in this section is the application of SMAC and MMAC approaches in scenario with non-identical channel condition. The contributions of this section are as follows:

- To show that the SMAC and MMAC approaches achieve a joint action that provides better network-wide performance in DCS.

- To show that the SMAC and MMAC approaches achieve high levels of fairness index.

- To show the effects of network density and various essential parameters in SMAC and MMAC approaches on network-wide performance.

7.4.6.3 Simulation Experiments, Results, and Discussions

The objectives of this investigation are presented in Section 7.4.4.2.

Simulation Setup and Parameters The simulation scenario is discussed in Section 7.1.6 and the assumptions are discussed in Section 7.4.6.1. Figure 7.2 shows the scenario and its graphical representation is shown in Figure 7.3. The procedure of the cognitive MAC is shown in Figure 7.12. General simulation parameters are shown in Table 7.5, and additional simulation parameters are shown in Table 7.7.

Table 7.7: Notations and Default Parameter Settings in Simulation for Investigation into Scenarios with non-Identical Channel Conditions

Category	Symbol	Details	Values
Initial ization	$P_{c_i}^E$	PER of each available data channel at agent *i*	[0,0.3] Default: 0.15
SU	$t_{CTRL,SU}$	CTRL control message packet duration	272μs
SMAC	$t_{D,SMAC}$	Epoch duration	187.14ms
MMAC	$t_{D,MMAC}$		249.52ms

Some of the important explanations on Table 7.7 are shown below:

Non-Identical Channel Quality (or PER) at all the Agents. Each agent observes different levels of PER across different data channels with the default average value of PER across the *K* data channels being 0.15 following a uniform distribution. Upon receiving a packet, an SU discards the packet with the PER probability.

Secondary User. For SU, the CTRL control message is a small packet with $t_{CTRL,SU}$ duration. It contains information related to channel switch-

ing and payoff message. In comparison with Section 7.4.5.3, we consider the CTRL control message does not contain other broadcast information.

SMAC and MMAC. For SMAC, an epoch duration is 30 data transmission cycles, or t_D=$t_{D,SMAC}$=30×(t_{RTS}+t_{CTS}+$t_{DATA,SU}$+t_{ACK}+3t_{SIFS}). For MMAC, an epoch duration is 40 data transmission cycles, or t_D=$t_{D,MMAC}$= 40×(t_{RTS}+t_{CTS}+$t_{DATA,SU}$+t_{ACK}+3t_{SIFS}). The duration $t_{D,MMAC}$ is 25% longer than the $t_{D,SMAC}$ to make a fair comparison whilst allowing payoff message exchange. This is because during each exploration in MMAC, an agent waits and receives payoff messages from its neighbour agents. The value of $t_{D,SMAC}$ and $t_{D,MMAC}$ were chosen empirically to enhance network-wide performance.

Performance Metrics. The performance metrics are discussed in Section 7.4.4.2. An additional performance metric, namely Jain's fairness index, is applied to evaluate the fairness among the throughput achieved by each agent in the entire DCRN. Denote the throughput achieved by agent i by x_i, the Jain's fairness index [95] is as follows:

$$f(x_1, x_2, \ldots, x_u) = \frac{(\sum_{i=1}^{u} x_i)^2}{(u \sum_{i=1}^{u} x_i^2)} \tag{7.13}$$

where $0 \leq f(x_1,x_2,\ldots,x_u) \leq 1$, and $f(x_1,x_2,\ldots,x_u)=1$ when all agents achieve the same level of throughput.

Simulation Results and Discussions Firstly, we show that the global Q-value of SMAC and MMAC stabilizes as time goes by. Secondly, we investigate the effects of network density on network-wide performance. Thirdly, we investigate the fairness index of SMAC and MMAC. Fourthly, we investigate the effects of α and ε on network-wide performance.

Stabilization of Global Q-value. Figure 7.26 shows that the instantaneous global Q-value for the exploitation data channel for SMAC and MMAC increases and becomes stable as time goes by in a medium density network. In other words, the agents attain a joint action that provides better network-wide performance. The PUL is L=0.5 with [0.5,0.5,0.5] across the K=3 data channels, and mean PER at agent i is P^E_{i,c_j}=0.15 for every data channel. The SMAC and MMAC parameters are α=0.2 and ε=0.2. With U=6 and Q_{max}=20, the maximum global Q-value is 120. Although L_{c_j}=0.5 for all data channels, due to the Poisson traffic model, the data channels have different levels of PUL at any particular time instant. Although MMAC aims to increase the global Q-value; while SMAC aims to increase the local Q-value, SMAC achieves slightly higher global Q-value compared to MMAC. This is because $t_{D,MMAC} > t_{D,SMAC}$ or, in other words, MMAC is less responsive to the operating environment compared to SMAC.

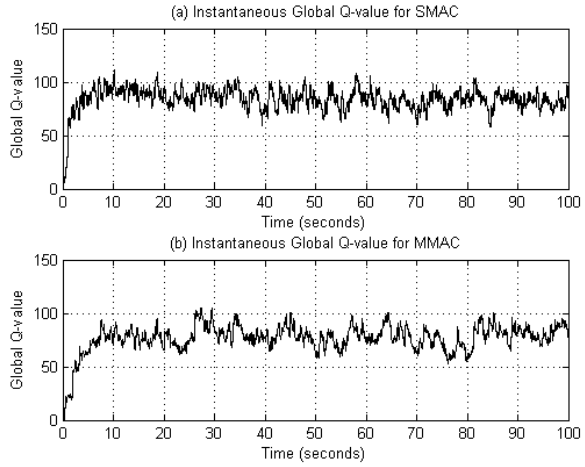


Figure 7.26: Global Q-value for the exploitation data channel for SMAC and MMAC in a medium density network. PUL L_{c_j}=0.5. The mean PER P^E_{i,c_j}=0.15.

Effects of Network Density on Network-wide Performance. Figure 7.27 shows the mean throughput for each agent against various levels of mean PUL for RMAC, SMAC, and MMAC in low, medium and high density networks. The mean PER at agent i is P^E_{i,c_j}=0.15 for every data channel. The SMAC and MMAC parameters are α=0.2 and ε=0.2. The SMAC and MMAC achieve approximately similar throughput, followed by RMAC in all types of network densities; and the throughput enhancement offered by the MMAC and SMAC compared to RMAC reduces as the network density increases. At PUL L_{c_j}=0.5, the MMAC outperforms the RMAC by 1.77 times, 1.5 times, and 1.2 times in low, medium and high density networks respectively. Figure 7.28 shows the equivalent graph with PER as ordinate and PUL is L_{c_j}=0.5 with [0.5,0.5,0.5], and similar trend is observed. In short, in a high density network or $d \rightarrow \infty$, the throughput enhancement achieved by SMAC and MMAC approaches 0. We believe that this happens in most intelligence methods due to the high contention level.

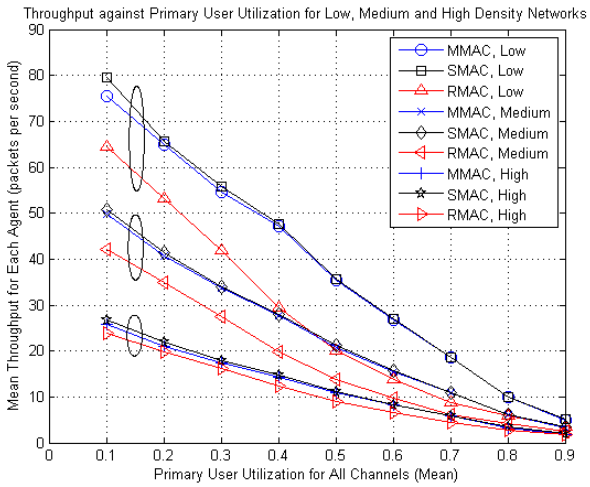


Figure 7.27: The mean throughput for each agent against mean PUL for RMAC, SMAC and MMAC in low, medium and high density networks. The mean PER P^E_{i,c_j}=0.15.

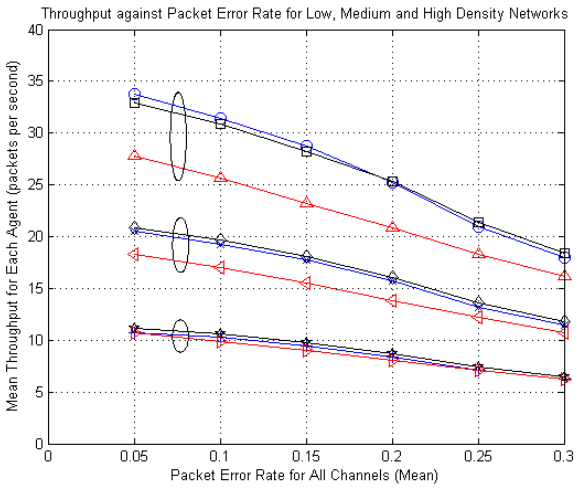


Figure 7.28: The mean throughput for each agent against mean PER for RMAC, SMAC and MMAC in low, medium and high density networks. See legend in Figure 7.27. PUL L_{c_j}=0.5.

Figure 7.29 shows the mean number of channel switchings of the exploitation data channel for each agent against various levels of mean PUL for SMAC and MMAC in low, medium and high density networks. The mean PER at agent *i* is P^E_{i,c_j}=0.15 for every data channel. The SMAC and MMAC parameters are α=0.2 and ε=0.2. The MMAC achieves a significantly lower number of channel switchings, hence it provides higher stability. At PUL L_{c_j}=0.5, the number of channel switchings in SMAC is 10 times, 3.6 times and 2 times of that in MMAC in low, medium and high density networks respectively. Although the duration $t_{D,MMAC}$ is only 25% longer than the $t_{D,SMAC}$, which reduces the number of channel switchings due to longer epoch duration, the MMAC provides a significantly lower number of channel switchings. Generally speaking, an agent switches its exploitation data channel because the difference between the Q-values among the data channels is greater than the threshold β=1, and the agent exploits a better data channel. There are two reasons an agent does not switch its data channel as explained in Section 7.4.6.3. Both SMAC and MMAC

have lower numbers of channel switchings as the PUL increases because $Q_t(a) \rightarrow 0$ for all data channels. The MMAC also increases network stability [64] through reducing the number of channel switching. Figure 7.30 shows the equivalent graph with PER as ordinate and PUL L_{c_j}=0.5 with [0.5,0.5,0.5].

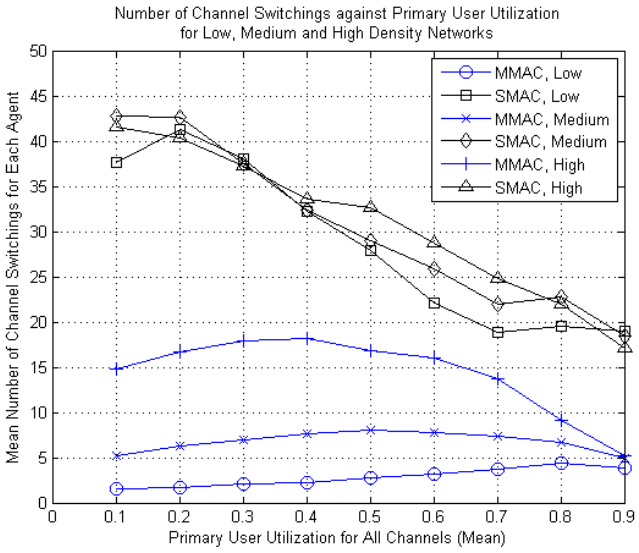


Figure 7.29: The mean number of channel switchings of the exploitation data channel for each agent against mean PUL for SMAC and MMAC in low, medium and high density networks. The mean PER P^E_{i,c_j}=0.15.

Fairness Index of SMAC and MMAC. With respect to PUL in Figure 7.31 and PER in Figure 7.32, RMAC achieves the highest level of fairness index, while SMAC and MMAC achieve approximately similar high levels of fairness index. Figure ? shows the enlarged version of Figure 7.31; while Figure ? shows the enlarged version of Figure 7.32. In Figure 7.31, the mean PER at agent i is P^E_{i,c_j}=0.15 for every data channel; while in Figure 7.32, PUL is L_{c_j}=0.5 with [0.5,0.5,0.5]. In RMAC, all agents choose their respective data channels randomly, hence the Jain's fairness index is close to 1. For SMAC and MMAC, some agents may choose better data channels

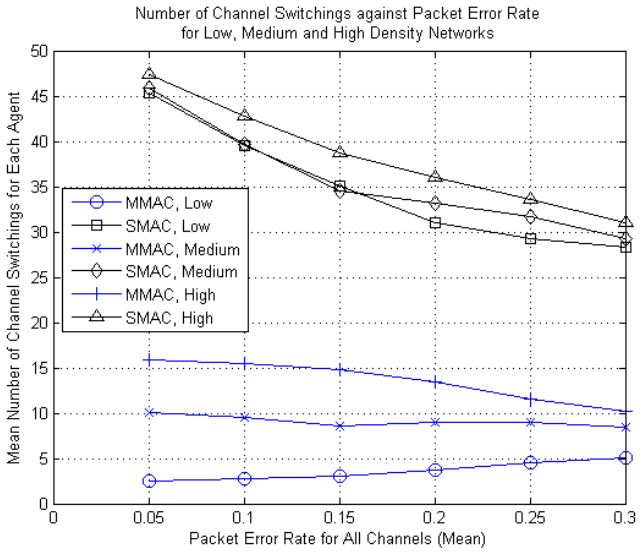


Figure 7.30: The mean number of channel switchings of the exploitation data channel for each agent against mean PER for SMAC and MMAC in low, medium and high density networks. PUL L_{c_j}=0.5.

compared to others, hence the Jain's fairness index is lower than that in RMAC.

Effects of α and ε on Network-wide Performance. Again, similarly to the findings in Section 7.4.5.3 we find that the effects of α and ε on throughput are insignificant in most of the cases, and their graphs are not provided. Figure 7.35 shows the effects of α on the mean number of channel switchings of the exploitation data channel for each agent against various levels of mean PUL for SMAC and MMAC in medium density networks. The mean PER at agent i is P^E_{i,c_j}=0.15 for every data channel. The number of channel switchings increases with α for all cases. In short, lower value of α provides higher stability. Figure 7.36 shows the equivalent graph with PER as ordinate and PUL L_{c_j}=0.5 with [0.5,0.5,0.5]. A similar experiment is performed to investigate the effects of ε on the mean number of channel switchings, and the results are shown in Figures 7.37 and

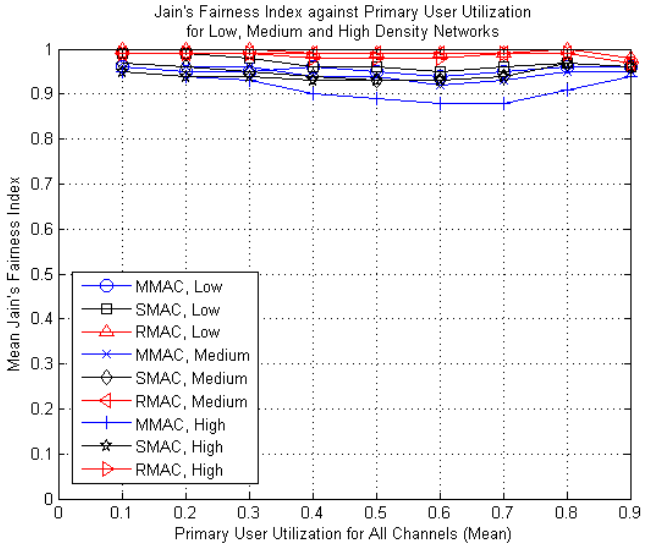


Figure 7.31: The mean Jain's Fairness Index against mean PUL for RMAC, SMAC and MMAC in low, medium and high density networks. The mean PER P^E_{i,c_j}=0.15.

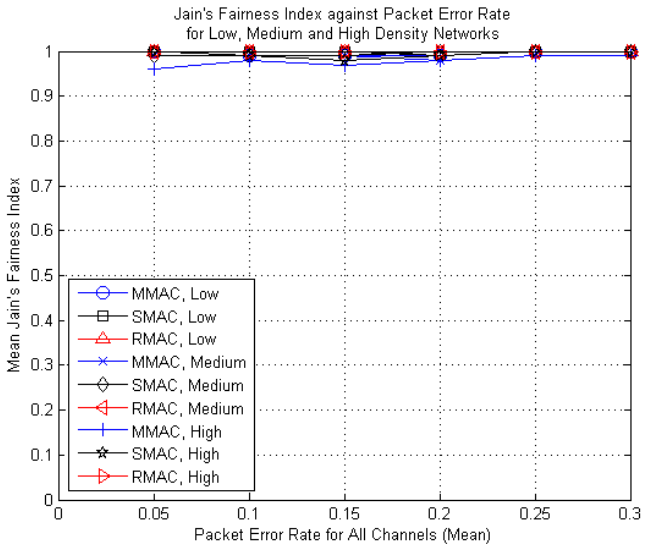


Figure 7.32: The mean Jain's Fairness Index against mean PER for RMAC, SMAC and MMAC in low, medium and high density networks. PUL L_{c_j}=0.5.

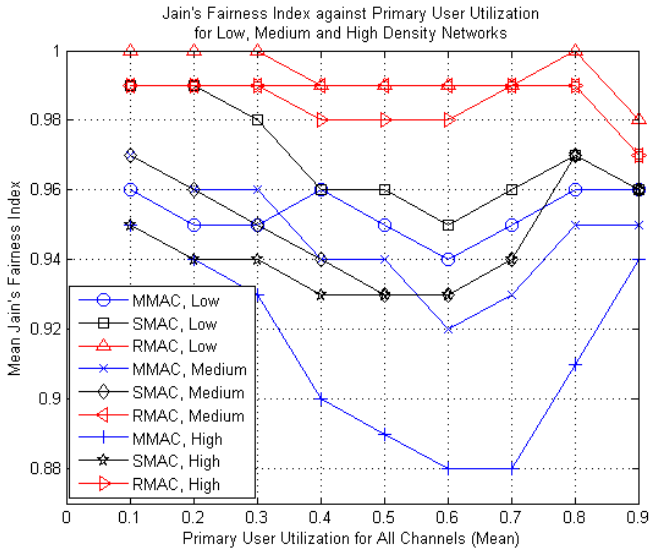


Figure 7.33: Enlarged version of Figure 7.31

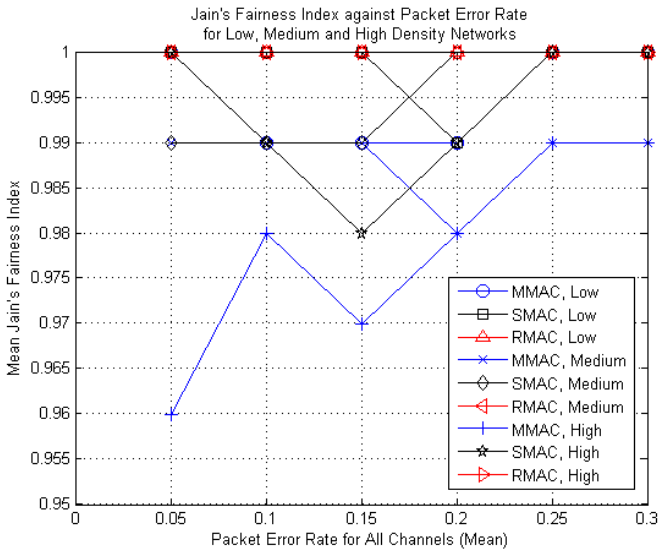


Figure 7.34: Enlarged version of Figure 7.32

7.38, which shares similar trends to Figure 7.35 and 7.36 respectively.

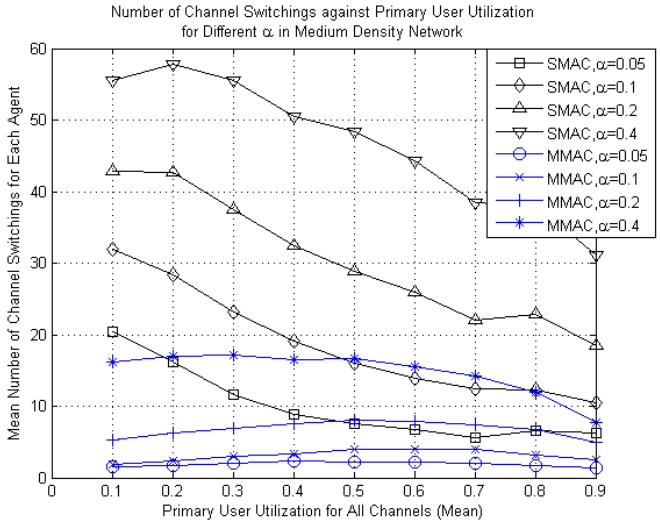


Figure 7.35: The mean number of channel switchings of the exploitation data channel for each agent against mean PUL for SMAC and MMAC with different α values in medium density networks. The mean PER P^E_{i,c_j}=0.15. ε=0.2.

7.4.6.4 Summary of Research Outcomes

The research outcomes from the investigation on the SARL and MARL approaches, including SMAC and MMAC, in order to implement the MACC model in scenarios with non-identical channel condition using the performance metrics of network-wide throughput and number of channel switchings are summarized in this section. This section assumes non-identical channel condition (or PER) at all the agents for a particular data channel such that P_i=[$P_{i,1}$,. . .,$P_{i,K}$]. This section also assumes a single collision domain. The research outcomes are summarized as follows:

- The global Q-value for the exploitation data channel for SMAC and MMAC increases and becomes stable as time goes by.

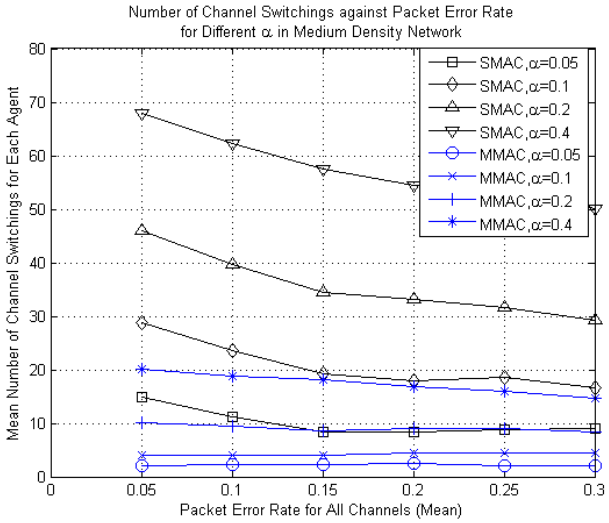


Figure 7.36: The mean number of channel switchings of the exploitation data channel for each agent against mean PER for SMAC and MMAC with different α values in medium density networks. PUL L_{c_j}=0.5. ε=0.2

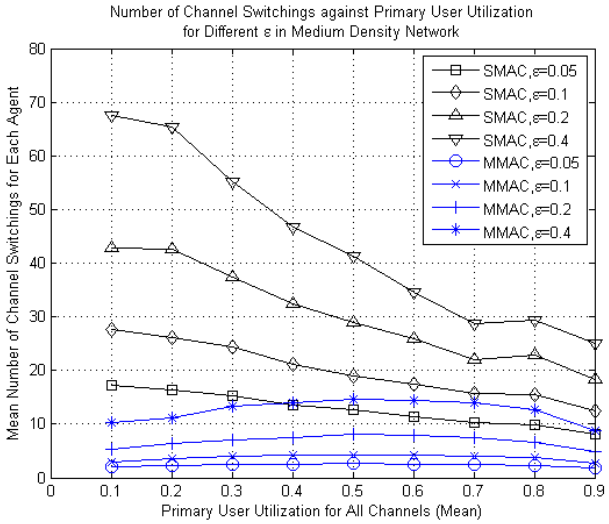


Figure 7.37: The mean number of channel switchings of the exploitation data channel for each agent against mean PUL for SMAC and MMAC with different ε values in medium density networks. The mean PER P^E_{i,c_j}=0.15. α=0.2.

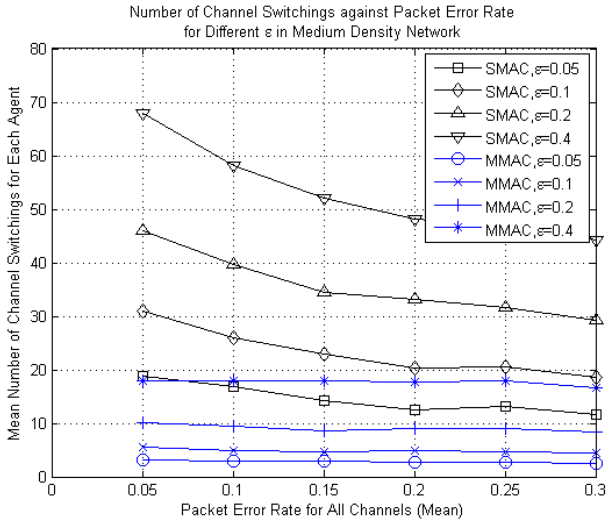


Figure 7.38: The mean number of channel switchings of the exploitation data channel for each agent against mean PER for SMAC and MMAC with different ε values in medium density networks. PUL L_{c_j}=0.5. α=0.2.

- The SMAC and MMAC approaches achieve approximately similar level of throughput, followed by RMAC in most of the cases in low, medium and high density networks.
- In high density networks as $d \rightarrow \infty$, the throughput enhancement achieved by SMAC and MMAC approaches 0.
- The number of channel switchings in MMAC is significantly lower than that in SMAC.
- The SMAC and MMAC approaches achieve an approximately similar high level of Jain's fairness index.
- Lower value of α and ε provides higher throughput and lower number of channel switchings (or better stability) in SMAC and MMAC.

The MARL approach considers network performance at neighbour agents through the exchange of payoff messages in order to maximize the glob-

al payoff. This improves the network-wide performance, in terms of throughput and number of channel switchings, at an agent and its neighbour agents.

7.5 Chapter Summary

In this chapter, the SARL approaches, namely, SMAC and eSMAC, as well as the MARL approach, namely, MMAC, are applied with respect to DCS to implement multi-agent cognition cycle (or node-level cognition cycle) in order to achieve context awareness and intelligence in static distributed CR networks. The MARL approach encompasses the SARL approach and the PP mechanism. The SMAC and eSMAC approaches differ in the exploration procedure: there is only a single agent undergoes exploration among neighbourhood agents in eSMAC in order to improve stability or to reduce number of channel switchings. The Random approach, which chooses an available data channel for data transmission in a uniformly distributed random manner without learning, serves as a baseline. This chapter considers channel heterogeneity and static networks; while previous work considers channel homogeneity and static networks. The major investigations were: 1) Payoff Propagation (PP) mechanism; 2) Scenario with *identical* channel condition (or PER) at all the agents for a particular data channel such that P=[P_1,. . .,P_K]; and 3) Scenario with *non-identical* channel condition (or PER) at all the agents for a particular data channel such that P=[$P_{i,1}$,. . .,$P_{i,K}$].

The PP mechanism provides a means of communication for the traditional SARL approach, which is the local learning mechanism. The extended PP mechanism is shown to converge to an efficient and optimal joint action. The work here has shown that the payoff value does not increase without bound in a cyclic topology. Lastly fast convergence is possible through the adjustment of the exploration probability.

Next we investigated the network performance offered by the SARL

approaches in scenario with *identical* channel condition (or PER) at all the agents for a particular data channel such that P=[P_1,. . .,P_K]. Two SARL approaches, namely SMAC and eSMAC, were applied. It was shown that the global Q-value for the exploitation data channel for SMAC and eSMAC increases and becomes stable as time goes by. The eSMAC achieves the highest amount of throughput, followed by SMAC and RMAC in most of the cases in low, medium and high density networks. In high density networks or as $d \rightarrow \infty$, the throughput enhancement achieved by SMAC and eSMAC approaches 0. The number of channel switchings in eSMAC is significantly lower than that in SMAC, hence eSMAC is more stable. In general, lower values of α and ε provides better throughput and stability.

Lastly we investigated network performance offered by the SARL and MARL approaches in scenario with *non-identical* channel condition (or PER) at the agents for a particular data channel such that P_i=[$P_{i,1}$,. . .,$P_{i,K}$]. The SARL and MARL approaches were applied using SMAC and MMAC protocols which showed that the Q-value for the exploitation data channel increases and becomes stable. The SMAC and MMAC achieve approximately similar level of throughput, followed by RMAC in most of the cases in low, medium and high density networks. In high density networks or as $d \rightarrow \infty$, the throughput enhancement achieved by SMAC and MMAC approaches 0. The number of channel switchings in MMAC is significantly lower than that in SMAC, hence MMAC is more stable. Both SMAC and MMAC achieve approximately similar high level of Jain's fairness index. In general, lower values of α and ε provides better throughput and stability.

Chapter 8

Applications of the Cognition Cycle

This chapter presents the RL models, both single-agent and multi-agent approaches, for the applications proposed in C^2net (see Chapter 4 on page 43). By providing discussions on the proposed RL models, we show how the SARL and MARL approaches presented in Chapters 6 and 7 respectively can be applied to design various applications in CR networks. With that, the foundation for futher research on the RL approach in CR networks is established.

8.1 Introduction

The Cognition Cycle (CC) (see Section 2.3 on page 14) is the key element of CR to provide context awareness and intelligence so that each SU is able to observe and carry out an optimal or near-optimal action in its operating environment for network performance enhancement. Context awareness enables an SU to sense and observe its complex and dynamic operating environment. Intelligence enables an SU to learn knowledge, which can be acquired through observing the consequences of its prior action, about its operating environment so that it carries out the right action at the right

time to approximate and achieve optimum network performance in an efficient manner without adhering to a strict and static predefined set of policies. The CC can be applied in various applications in CR networks such as DCS, topology management, congestion control, and scheduling. The RL models for the cross-layer designs (see Section 4.5 on page 54) proposed in C^{2}net are presented in this chapter to warrant further research on RL in CR networks.

8.2 Chapter Goal

Using RL, both single-agent and multi-agent approaches, this chapter addresses the following research question: How should the SARL and MARL approaches be used to model the cross-layer designs proposed in C^{2}net?

8.3 Related Work

There are two types of CC: SACC for centralized CR networks, and MACC for distributed CR networks. Chapter 6 presents SACC, while Chapter 7 presents MACC. The SACC model, which is embedded in a fixed network infrastructure, such as a base station, makes decision in a multilateral and cooperative manner on an optimal or near-optimal action for the entire network. The MACC model, which is embedded in each SU, makes decisions in a cooperative or non-cooperative, and distributed manner as part of the efficient and optimal joint action for the entire network.

8.4 RL Models for Cross-Layer Designs in C^{2}net

Chapter 4 in Section 4.5 on page 54 introduced various cross-layer designs in CR networks to realize the C^{2}net architecture. C^{2}net is a cross-layer Quality of Service (QoS) architecture proposed in this thesis based on the

Next Steps in Signaling framework [42], which is an end-to-end QoS signaling protocol, for the Cognitive Wireless Ad-hoc Networks (CWANs) (see Section 2.2.5 on page 13). A CWAN is a multihop self-organized and dynamic network comprised of static and mobile SUs. Three cross-layer designs are proposed, namely *joint DCS and topology management*, *joint DCS and congestion control*, and *joint scheduling and channel condition measurement*. The applications are formulated using the RL models of SACC and MACC in order to achieve their objectives. Since CWANs are distributed CR networks, the MACC model is chosen, although it could be applied as SACC in centralized CR networks. For each cross-layer design, the state, action and reward representation for the RL model are defined.

8.4.1 Joint Dynamic Channel Selection and Topology Management

8.4.1.1 Overview of the Joint Design

This section describes in detail the joint dynamic channel selection and topology management introduced in Section 4.5.1 on page 56.

Objectives. This joint design provides the best strategy for channel selection from the available licensed data channels for data transmission among the SUs given that the objective is to minimise the end-to-end data packet loss rate and enhance throughput performance for stable QoS provisioning in the presence of nodal mobility.

Descriptions of Operation. There are two levels of heterogeneity: nodal and channel level. In other words, the nodes and channels have a wide range of characteristics that affect the network performance in a complex manner.

For stable, reliable and robust transmissions, some SUs in the network are selected as Dominating Set (DS) SUs to form a backbone topology that

connects the entire network to the base station; while non-DS SUs establish links with the DS SUs. The DS SUs have the following heterogeneity nodal characteristics compared to their neighbour SUs:

- Higher stability and lower mobility.
- Higher residual energy level.
- Better hardware capability.
- Higher willingness in relaying data packets.
- Maintaining connectivity of the backbone topology.

The heterogeneous channels are characterized by their PUL, PER, and transmission range. For instance, a data channel with low PUL is unfavourable if it has high PER. Due to the importance of the DS SUs, they are given higher authority in channel selection. Thus, data channels that provide higher throughput or with more white spaces are allocated to the DS SUs. Non-DS SUs choose the remaining available data channels. As the SUs and channels are dynamic in nature, all observations and information have to be maintained and updated continuously. In short, both DS SUs and channels must possess the favourable characteristics at most of the times. This joint design is comprised of two components: DCS, and topology management consisting of backbone topology construction and maintenance; hence two distinguishing RL models are necessary.

8.4.1.2 Reinforcement Learning Model

In DCS, the RL model follows Table 7.4 on page 200. In topology management, a DS SU i chooses a next-hop DS SU $n_{i,j}$ among its neighbour SUs j. The selection criteria includes the following two considerations:

- The LET (see Section 4.5.1.2 on page 56 for explanation) of the link between the SU i and SU j, $l_{i,j}$.

- The capability of SU *j* to help the SU *i* to relay its data packets.

The RL model for the next-hop DS SU selection using the MACC approach is shown in Table 8.1. The state $s \in S$ includes the set of IDs of the base stations that SU *i* is sending its data packets to; hence DS SU *i* may choose a different next-hop DS SU $n_{i,j}$ for different base stations. The action $a \in A$ is to choose a neighbour SU *j*, with *J* as the cardinality of the SU *i*'s neighbour SUs. For every successful data packet transmission, there is a reward with positive constant value of $+RW$, otherwise there is a cost with negative constant value of $-CT$. The Q-values for all neighbour SUs *j* are updated from time to time during exploration and exploitation using (7.11). SUs that are relatively selfish or incapable of relaying data packets have low levels of Q-value, and thus are not chosen as the next-hop DS SU.

Let the link LET $l_{i,j}$ be upper bounded by L. For stability, rather than using (7.12), the next-hop DS SU $n_{i,j}$, which has the best possible capability to forward data packets at longer LET, is chosen by DS SU *i* using function *f*:

$$n_{i,j} = \underset{a \in A}{\operatorname{argmax}} f(Q_t(s_t, a) \times \max(L, l_{i,j,t})) \tag{8.1}$$

Table 8.1: RL Model (MACC) at Each SU for Topology Management

	Next-hop DS SU Selection Model	
	Description	Representation
State	Set of base stations.	$S=\{s=I_1,I_2,\ldots,I_B\}$
Action	Set of node *i*'s neighbour nodes *j*.	$A=\{a=1,2,\ldots,J\}$
Reward	Constant value to be rewarded/incurred for successful/unsuccessful data packet transmission.	$r_{t+1}(s_t,a_t) = \begin{cases} +RW, & \text{if successful} \\ -CT, & \text{if otherwise} \end{cases}$

8.4.2 Joint Dynamic Channel Selection and Congestion Control

8.4.2.1 Overview of the Joint Design

This section describes in detail the joint dynamic channel selection and congestion control introduced in Section 4.5.2 on page 59.

Objectives and Descriptions of Operation. The objective of this joint design is to allocate the available data channels according to the traffic load at each SU given that each data channel has different levels of PUL and PER. In other words, a data channel with lower PUL and PER is allocated to an SU with higher traffic load, and vice-versa in order to achieve load balancing among the data channels as a solution to congestion avoidance. This solves congestion locally at the data link layer, rather than at the transport layer.

8.4.2.2 Reinforcement Learning Model

The RL model for the congestion control mechanism using the MACC approach is shown in Table 8.2. The purpose of the congestion control mechanism is to ameliorate the packet dropping rate; hence, the RL model identifies which among the possible channel switching options induce packet dropping. Subsequently, the SU refrains from executing these channel switches.

The state includes four-tuple information that is important for the SU to make congestion control decisions. The parameters b, p_d, b_w and b_s are quantized. For instance, for b, the following applies: $b_i < b_{i+1}$ with N_b being the maximum level of the parameter. K is the number of available data channels. In general, a data channel with low PUL and PER has a high amount of *good* white space that improves the throughput and reduces the data packet loss rate. The state keeps track of the amount of required bandwidth, the current data packet dropping probability, the amount of *good*

white space (or bandwidth) in the current data channel, and the amount of *good* white space across all the data channels. The action A is to choose a data channel that SU i could switch to without jeopardising its throughput performance. Based on the information in the current state, an SU makes a channel switching decision to change from a data channel having bandwidth b_w to another data channel having bandwidth b_s. The cost is based on the level of unfulfilled bandwidth requirement such that a good data channel that fulfills the bandwidth requirement receives a cost of 0; otherwise there is a cost of negative value (or $B_{s,k}-b$). The Q-values for all the data channels are updated from time to time during exploration and exploitation using (7.11). An optimal joint action approximated by the MACC approach using (7.12) helps all the SUs in the network to fulfill their respective required bandwidth.

To ensure that the data packet dropping probability is less than a threshold, $p_d \leq p_{d,th}$, the state-action (s,a) pairs that result in $p_d > p_{d,th}$ are marked as inappropriate, hence the action a is not taken whenever the state s is encountered.

8.4.3 Joint Scheduling and Channel Condition Measurement

8.4.3.1 Overview of the Joint Design

This section describes in detail the joint scheduling and channel condition measurement introduced in Section 4.5.3 on page 60.

Objectives and Descriptions of Operation. The objective of this joint design is to ameliorate the effects of head of queue blocking, where the current data packet transmission blocks the next data packets in the queue. This may result in data packet expiration and subsequently many data

Table 8.2: RL Model (MACC) at Each SU for Congestion Control

	Congestion Control Model	
	Description	Representation
State	State S has four tuple information: 1) amount of required bandwidth, b; 2) current packet dropping probability, p_d; 3) amount of *good* white spaces in the current channel, b_w; and 4) amount of *good* white spaces in all the available data channels, B_s.	S=$\{s$=$(b,p_d,b_w,b_s)\}$, b=$\{b_1,b_2,\ldots,b_{N_b}\}$, p_d=$\{p_{d,1},p_{d,2},\ldots,p_{d,N_d}\}$, b_w=$\{b_{w,1},b_{w,2},\ldots,b_{w,N_w}\}$, B_s=$(B_{s,1},B_{s,2},\ldots,B_{s,K})$, $B_{s,k}$=$(b_{s,1},b_{s,2},\ldots,b_{s,N_b})$
Action	Available data channels for data transmission.	A=$\{a$=$1,2,\ldots,K\}$
Cost	Level of unfulfilled bandwidth requirement.	$r_{t+1}(s_t,a_t)$ = $\begin{cases} \frac{B_{s,k}-b}{b}, & \text{if } b-B_{s,k}\geq 0 \\ 0, & \text{if otherwise} \end{cases}$

packets are dropped. Each available data channel for data transmission has different levels of PUL and PER that may lead to several data packet retransmissions.

8.4.3.2 Reinforcement Learning Model

The RL model for the scheduling mechanism using the MACC approach is shown in Table 8.3, and it is discussed briefly because of similarities to the previous discussions. The RL model considers the priority and deadline of the high-priority data packets, as well as the probability of successful data packet transmission from SU i to SU j using channel k, $P_{s,k}^{(i,j)}$ for each neighbour SU to determine the next-hop SU for transmission to maximize successful data packet transmission in the shortest possible time for QoS provisioning. Based on the current state, an SU makes a decision on the next-hop SU to transmit data packets in order to maximize its rewards (or revenue). The SU is positively rewarded only if it has successfully transmitted a data packet to its next-hop SU, else the SU is penalized with a cost of C_s. The Q-values for all the data channels are updated from time to time during exploration and exploitation using (7.11). An efficient and optimal joint action provided by the MACC approach using (7.12) helps all the SUs to maximize their rewards.

Table 8.3: RL Model (MACC) at Each SU for Scheduling

	Next Hop for Data Packet Transmission Selection Model	
	Description	Representation
State	State S has three tuple information: 1) probability of successful data packet transmission $P_s^{(i,j)}$; 2) deadline of the data packet $d^{(i,j)}$; and 3) priority of the data packet $p^{(i,j)}$.	S=$\{s$=$(P_s^{(i,j)},d^{(i,j)}),p^{(i,j)})\}$, $P_s^{(i,j)}$=$(P_{s,1}^{(i,j)},P_{s,2}^{(i,j)},\ldots,P_{s,K}^{(i,j)})$, $P_{s,k}^{(i,j)}$=$(p_{s,1}^{(i,j)},p_{s,2}^{(i,j)},\ldots,p_{s,N_p}^{(i,j)})$, $d^{(i,j)}$=$(d_1^{(i,j)},d_2^{(i,j)},\ldots,d_{N_d}^{(i,j)})$, $p^{(i,j)}$=$(p_1^{(i,j)},p_2^{(i,j)},\ldots,p_{N_p}^{(i,j)})$
Action	Set of node i's neighbour nodes j.	A=$\{a$=1,2,$\ldots,J\}$
Reward	The reward for sending data packets with different deadlines r_d, and levels of priorities r_p. There are n_s successful data packet transmission within a decision epoch. Data packet with higher priority and closer deadline has higher reward. Cost C_s is incurred for each failed data packet transmission.	$r_{t+1}(s_t,a_t)$=$\sum_{i=1}^{n_s}(r_d\times r_p)$-$C_s$, r_d=$\{r_{d,1},r_{d,2},\ldots,r_{d,N_d}\}$, r_p=$\{r_{p,1},r_{p,2},\ldots,r_{p,N_p}\}$

8.5 Chapter Summary

In this chapter, SARL and MARL approaches were proposed to implement the conceptual cognition cycle in CR networks for various novel cross-layer applications. The purpose is to warrant further research on the SAR-L and MARL approaches in CR networks. The MACC models for three joint designs in C^2net are proposed through the definitions of state, action and reward representations in RL. The designs are joint DCS and topology management, joint DCS and congestion control, and joint scheduling and channel condition measurement.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Cognitive Radio is a next-generation wireless communication system that enables secondary users to exploit underutilized licensed spectrum owned by the primary users to improve the utilization of the overall radio spectrum. The PUs are oblivious to the presence of the SUs. The concept of cognition cycle is the key element of CR to achieving context awareness and intelligence. Context awareness enables an SU to sense and observe its complex and dynamic operating environment. Intelligence enables an SU to learn knowledge (which can be acquired through observing the consequences of its prior action) about its operating environment so that it carries out the right action at the right time to achieve an approximation of the optimum network performance in an efficient manner without adhering to a strict and static predefined set of policies. The CC can be applied in various applications in CR networks such as dynamic channel selection, topology management, congestion control and scheduling. This thesis advocates the application of reinforcement learning, both single-agent and multi-agent approaches, to implement the conceptual CC. There are two levels of CC, namely single-agent or network-level cognition cycle for centralized CR networks; and multi-agent or node-level cognition cycle for

distributed CR networks.

Cognitive Radio is a new emerging research field that is lacking investigation into the data link layer for proper operation at both control and data transceivers. This thesis investigates the data link layer without adopting a number of conventional assumptions in this research field. It considers channel heterogeneity, as well as both static and mobile networks; while previous work assumes channel homogeneity and static networks. This thesis presents technology leverage from existing multi-channel protocols to cognitive MAC protocols, a cross-layer QoS architecture called C^2net, and the RL approaches, including SARL and MARL approaches to achieve context awareness and intelligence in static and mobile, centralized and distributed CR networks.

This thesis has achieved its overall goal. With respect to DCS, it provides a detailed understanding of the RL approach through analysis and simulation. The five research questions given in Section 1.2 on page 5 are answered below:

1. **What are the possible methods of technology leverage from multi-channel MAC protocols to cognitive MAC protocols?**
   This thesis reviews various approaches in multi-channel MAC protocols, their merits and demerits. The approaches are common control channel, split phase, common hopping, and default hopping sequence. Based on the belief that cognitive MAC protocols for distributed CR networks that apply similar approaches to multi-channel MAC protocols inherit their characteristics, the approach has to be chosen carefully based on its merits, demerits and hardware requirements. However demerit factors remain as open issues in distributed CR networks. Functionalities that cognitive MAC protocols have to provide, and how these functions can be incorporated into the multichannel MAC protocols are also presented.

2. **What is an appropriate QoS architecture for CR networks?**

This thesis presents a cross-layer QoS architecture called C^2net for cognitive wireless ad-hoc networks, which is a multihop self-organized and dynamic CR network. The main objective of C^2net is to provide and maintain a stable QoS provisioning to high priority flows throughout their connections. C^2net is a hybrid model of IntServ and DiffServ that adopts the NSIS framework. The core component for QoS provisioning in the NSIS framework is the QoS NSLP that provides end-to-end QoS signaling protocol. The IntServ model fulfills the stringent QoS requirements of a flow at reasonable cost by purchasing white spaces from PU if necessary. The DiffServ model provides services for lower priority packets. Various cross-layer designs as well as their open issues and challenges are presented. The cross-layer designs are joint DCS and topology management, joint DCS and congestion control, and joint scheduling and channel condition measurement.

3. **How is context awareness and intelligence best achieved in centralized CR networks?**
   This thesis presents SARL and other simple and pragmatic learning mechanisms, namely, Adapt, Win and AdaptWin, as approaches to achieve single-agent cognition cycle (or network-level cognition cycle), which encompasses the context awareness and intelligence concept in centralized CR networks. The learning mechanisms differ among themselves in terms of action selection and knowledge update. An analytical model based on Markov chain is presented to compute the expected throughput performance of a DCS scheme. The SARL approach achieves similar network performance to AdaptWin and Win, which provide the highest network performance among the other learning mechanisms studied. The SARL, Win and AdaptWin approaches achieve the expected throughput obtained from the analytical results; while this is not the case for Adapt. For SARL, the throughput and number of channel switchings achieve optimal or

near-optimal levels when learning rate α and exploration probability ε are low; and ε has a greater effect on network performance than does α. There are two advantages for SARL compared to Win and AdaptWin. Firstly, the extension of current work to achieve node-level cognition cycle using MARL in distributed CR networks. Secondly, the extension of current work to include state representation, which encompasses the condition of the operating environment that are relevant to decision making at the agent.

4. **How is context awareness and intelligence best achieved in distributed CR networks?**
   This thesis presents SARL and MARL approaches to achieve the multi-agent cognition cycle (or node-level cognition cycle), which encompasses the context awareness and intelligence concept in distributed CR networks. The SARL approaches proposed are SMAC and eSMAC. The eSMAC improves the stability of SMAC through reducing the number of channel switchings. The proposed MARL approach is MMAC, and it encompasses the SARL approach and the PP mechanism.

   The PP mechanism provides a means of communication for the SARL approach. The extended PP mechanism is shown to converge to an efficient and optimal joint action given that the entries in the Q-table and μ-table at each agent (or a communication node pair) are stable and fixed. The payoff value does not increase without bound in a cyclic topology. Fast convergence is possible through the adjustment of the exploration probability ε.

   For scenarios with *identical* channel condition (PER), the eSMAC improves network stability through monitoring the exploration procedure. It is shown that the global Q-value for the exploitation data channel for SMAC and eSMAC increases and becomes stable as time progresses. In high density networks, the throughput enhancement

achieved by SMAC and eSMAC approaches 0. The number of channel switchings in eSMAC is significantly lower than that in SMAC, hence eSMAC is more stable. In SMAC, lower values of α and ε provides better throughput and stability; however, in eSMAC, the values of α provide approximately similar level of throughput, however, the throughput decreases if α<0.2.

For scenarios with *non-identical* channel condition, the MMAC improves network stability. It is shown that the global Q-value for the exploitation data channel for SMAC and MMAC increases and becomes stable as time goes by. In high density networks, the throughput enhancement achieved by SMAC and MMAC approaches 0. The number of channel switchings in MMAC is significantly lower than that in SMAC, hence MMAC is more stable. Lower value of α and ε provides better throughput and stability.

5. **How can we apply these context awareness and intelligence approaches to QoS provisioning for CR networks?**
   This thesis presents how to apply the SACC and MACC to implement the conceptual CC in CR networks for various novel cross-layer applications. The RL models for three joint designs in C^{2}net are proposed through the discussion of the state, action and reward representation of the RL model. The designs are joint DCS and topology management, joint DCS and congestion control, and joint scheduling and channel condition measurement.

9.2 Future Work

This section highlights the most significant directions for future work.

9.2.1 Investigation on Technology Leverage from Multi - Channel to Cognitive Medium Access Control Protocols

Chapter 3 has established a foundation for further research in the data link layer of distributed CR networks through the discussion on technology leverage from multi-channel to cognitive MAC protocols. There are two categories of open issues as follows:

- Open issues associated with multi-channel MAC protocols. For instance, in the common control channel approach, if a single tranceiver is applied, there is lack of support for broadcasting which is important in routing message dissemination such as Route Request and Hello messages.
- Open issues associated with the additional requirement to cope with the existence of PUs that have higher authority over the data channels. This includes incorporating all the necessary CR functions into the multi-channel MAC including dynamic spectrum access, dynamic spectrum sharing, and dynamic spectrum management functions in Section 3.4 on page 33.

9.2.2 Investigation on C^{2}net: A Cross-Layer Quality of Service Architecture for Cognitive Radio Networks

Chapter 4 has established a foundation for further research into the data link and network layer of distributed CR networks through the discussion on C^{2}net, which is a QoS architecture for cognitive wireless adhoc networks. This includes incorporating the CR functions into the NSIS framework. For instance, local congestion control can be designed to cooperate with the end-to-end congestion control mechanism in the NSIS framework. Investigation can also be performed on the cross-layer designs using the RL approach as shown in Chapter 8. The cross-layer designs

discussed were joint DCS and topology management, joint DCS and congestion control, and joint scheduling and channel condition measurement.

9.2.3 Further Investigation on the Reinforcement Learning Model

Chapter 5 has presented a generic RL model to achieve context awareness and intelligence in CR networks. This RL model can be further investigated. This includes the following:

- New features not used in the traditional RL approach are events, rules and the effects of actions to the operating environment. On the events, the PU signal detection can be modeled as an event so that upon its detection, the RL model carries out some required functions. On the rules, the RL model can incorporate the rules imposed by the PUs, such as channel detection time, which is the time interval that an SU must detect PU signal. On the effects of actions to the operating environment, future research could be pursued for coordination among the agents.
- Effective approximation-based techniques to achieve continuous space representation. Currently, Q-learning is a tabular-based approach that may not be scalable to a large number of state-event-action pairs. The continuous space representation improves scalability as the agent does not keep track of each state-event-action pair in its Q-table.

9.2.4 Further Investigation on the Single-Agent Cognition Cycle

Chapter 6 presented the SARL approach to achieve context awareness and intelligence in centralized CR networks. The SARL approach can be further investigated. This includes the following:

- Short-term network performance enhancement. The SARL approaches achieve long-term network performance enhancement, rather than short term. However, throughput and delay performance enhancement may need to be achieved in a short time frame to provide QoS guarantee to high priority data packets.

- State representation and discounted rewards. The SARL model applied in Chapter 6 has ignored the state representation and discounted rewards. Further investigation on their application and performance enhancement can be performed.

- Collaboration between channel sensing and DCS. This enables the distributed sensing mechanism to collect information about the level of PU activity across a wide range of channels, while the DCS scheme applies this information to determine its operating channel.

9.2.5 Further Investigation on Multi-Agent Cognition Cycle

Chapter 7 has presented the SARL and MARL approaches to achieve context awareness and intelligence in distributed CR networks. The SARL and MARL approaches can be further investigated. This includes the following:

- Relaxing the single collision domain assumption, which is applied in Section 7.4.5 and 7.4.6. Single collision domain is a common assumption in the research field of CR networks, and it assumes that all the agents can hear each other. Performance enhancement brought about by the SARL and MARL approaches without this assumption are an interesting topic for future research.

- Short-term fairness. The MARL approach achieves long-term network performance enhancement including fairness, rather than

short term goals. Since there are many SUs competing for the data channels in a multi-agent environment, an MARL model that provides short-term fairness is important to provide QoS guarantee to high priority data packets.

Appendix A

Abbreviations

ACK	Acknowledgement
Adapt	Adaptation
AdaptWin	Adaptation-Window
ATIM	Ad Hoc Traffic Indication Messages
BPSK	Binary Phase Shift Keying
BS	Base Station
CC	Cognition Cycle
CCC	Common Control Channel
CCTT	Channel Closing Transmission Time
C^2net	Cognitive wireless ad hoc NETworks
CDS	Connected Dominating Set
CDT	Channel Detection Time
CH	Channel Hopping
CG	Coordination Graph
CMT	Channel Move Time
CPE	Customer-Premises Equipment

CR	Cognitive Radio
CSMA	Carrier Sense Multiple Access
CTS	Clear-to-Send
CUT	Channel Usage Table
CWAN	Cognitive Wireless Ad-hoc Network
DCCP	Datagram Congestion Control Protocol
DCF	Distributed Coordination Function
DCRN	Distributed Cognitive Radio Network
DCS	Dynamic Channel Selection
DFS	Dynamic Frequency Selection
DHS	Default Hopping Sequence
DiffServ	Differentiated Services
DIFS	DCF InterFrame Spacing
DS	Dominating Set
DSA	Dynamic Spectrum Access
DSCP	DiffServ Codepoint
DSS	Dynamic Spectrum Sensing
DSM	Dynamic Spectrum Management
EDF	Earliest Deadline First
FCC	Federal Communications Commission
GIST	General Internet Signaling Transport
HoQ	Head of Queue
IBM	In-Band Measurement
IDRP	Incumbent Detection Recovery Protocol

IDT	Incumbent Detection Threshold
IETF	Internet Engineering Task Force
i.i.d.	independent and identically distributed
IntServ	Integrated Services
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
LCT	Link Channel Table
LET	Link Expiration Time
MAC	Medium Access Control
MAC-PHY	Medium Access Control-Physical
MACC	Multi-Agent based Cognition Cycle
MANET	Mobile Ad hoc NETworks
MARL	Multi-Agent Reinforcement Learning
MDS	Minimum Dominating Set
MDTT	Maximum Data Transmission Time
MG	Matrix Game
NSIS	Next Steps in Signaling
NSLP	NSIS Signaling Layer Protocol
NTLP	NSIS Transport Layer Protocol
OBM	Out-of-Band Measurement
Ofcom	Office for Communication
OFDMA	Orthogonal Frequency Division Multiple Access
OSI	Open System Interconnection
PER	Packet Error Rate

PG	Potential Game
PHB	Per-Hop Behaviour
PME	Payoff Message Exchange
PP	Payoff Propagation
PSM	Power Saving Mode
PU	Primary User
PUL	Primary User Utilization
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RL	Reinforcement Learning
RL-DCS	Reinforcement Learning-based Dynamic Channel Selection
RREQ	Route Request
RSVP	Resource ReSerVation Protocol
RTS	Request-to-Send
SACC	Single-Agent based Cognition Cycle
SCTP	Stream Control Transmission Protocol
SG	Stochastic Game
SIFS	Short InterFrame Space
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SP	Split Phase
SU	Secondary User
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol

UDP	User Datagram Protocol
UHF	Ultra High Frequency
UNII	Unlicensed National Information Infrastructure
UWB	Unlicensed Ultra Wide Band
Win	Window
WRAN	Wireless Regional Area Network

Bibliography

[1] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty. Next Generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Computer Networks*, 50(13):2127–2159, September 2006.

[2] FCC Spectrum Policy Task Force. Report of the spectrum efficiency working group. Technical Report 02-155, Federal Communications Commission, November 2002.

[3] J. Mitola and G. Q. Maguire. Cognitive radio: making sofware radios more personal. *IEEE Personal Communications*, 6(4):13–18, August 1999.

[4] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. MIT Press, Cambridge, MA, 1998.

[5] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 1999.

[6] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. On multichannel MAC protocols in cognitive radio networks. In *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pages 300–305, Adelaide, Australia, December 2008. IEEE.

[7] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Medium access control (MAC) protocols for cognitive radio networks: Recent advances and design considerations. In *Proceedings of the 7th New Zealand Computer Science Research Student Conference (NZCSRSC)*, Auckland, New Zealand, April 2009.

[8] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. A survey on multi-channel medium access control (MAC) protocols: A cognitive radio perspective. In *Proceedings of the 7th New Zealand Computer Science Research Student Conference (NZCSRSC)*, Auckland, New Zealand, April 2009.

[9] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. C^{2}net: A cross-layer Quality of Service (QoS) architecture for cognitive wireless ad hoc networks. In *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pages 306–311, Adelaide, Australia, December 2008. IEEE.

[10] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Quality of service (QoS) provisioning in cognitive wireless ad hoc networks: Challenges, design approaches, and open issues. In Sasan Adibi, Tom Tofigh, Shyam Parekh, and Raj Jain, editors, *Quality of Service Architectures for Wireless Networks: Performance Metrics and Management*, chapter 25, pages 575–594. Information Science Reference, IGI Global, US, January 2010.

[11] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. A context-aware and intelligent dynamic channel selection scheme for cognitive radio networks. In *Proceedings of the 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 1–6, Hannover, Germany, June 2009. IEEE.

[12] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Performance analysis of reinforcement learning for achieving context

awareness and intelligence in cognitive radio networks. In *Proceedings of the 9th International Workshop on Wireless Local Networks (WLN) at the 34th IEEE Conference on Local Computer Networks (LCN)*, pages 1046–1053, Zurich, Switzerland, October 2009. IEEE.

[13] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Enhancing network performance in distributed cognitive radio networks: Single-agent and multi-agent reinforcement learning approach. In *Proceedings of the 35th IEEE Conference on Local Computer Networks (LCN)*, Denver, Colorado, US, October 2010. IEEE.

[14] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Achieving efficient and optimal joint action in distributed cognitive radio networks using payoff propagation. In *Proceedings of the International Conference on Communications (ICC)*, Cape Town, South Africa, May 2010. IEEE.

[15] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Context-awareness and intelligence in distributed cognitive radio networks: A reinforcement learning approach. In *Proceedings of the 11th Australian Communications Theory Workshop (AusCTW)*, Canberra, Australia, February 2010. IEEE.

[16] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Cognitive radio-based wireless sensor networks: Conceptual design and open issues. In *Proceedings of the 2nd International Workshop on Wireless and Internet Services (WISe) at the 34th IEEE Conference on Local Computer Networks (LCN)*, pages 955–962, Zurich, Switzerland, October 2009. IEEE.

[17] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D. Teal. Applications of reinforcement learning to cognitive radio networks. In *Proceedings of the 1st International Workshop on Cognitive Radio Interfaces*

and Signal Processing (CRISP) at the International Conference on Communications (ICC), Cape Town, South Africa, May 2010. IEEE.

[18] K.V.S.S.S.S. Sairam, N. Gunasekaran, and S.R. Redd. Bluetooth in wireless communication. *IEEE Communications Magazine*, 40(6):90–96, June 2002.

[19] B.P. Crow, I. Widjaja, L.G. Kim, and P.T. Sakai. Ieee 802.11 wireless local area networks. *IEEE Communications Magazine*, 35(9):116–126, September 1997.

[20] C. Eklund, R.B. Marks, K.L. Stanwood, and S. Wang. Ieee standard 802.16: a technical overview of the wirelessmantm air interface for broadband wireless access. *IEEE Communications Magazine*, 40(6):98–107, June 2002.

[21] Chonggang Wang, K. Sohraby, R. Jana, Lusheng Ji, and M. Daneshmand. Voice communications over zigbee networks. *IEEE Communications Magazine*, 46(1):121–127, January 2008.

[22] D. Cabric, S. M. Mishra, and R. W. Brodersen. Implementation issues in spectrum sensing for cognitive radios signals, systems and computers. In *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 772 – 776, Pacific Grove, CA, November 2004. IEEE.

[23] IEEE 802.22 WRAN WG. Wireless regional area networks. `http://www.ieee802.org/22/`.

[24] Linda Doyle and Tim Forde. The wisdom of crowds: cognitive ad hoc networks. In Qusay H. Mahmoud, editor, *Cognitive Networks*, chapter 8, pages 203–221. John Wiley & Sons, July 2007.

[25] Kaigui Bian and Jung-Min Park. Segment-based channel assignment in cognitive radio ad hoc networks. In *Proceedings of the 2nd International Conference on Cognitive Radio Oriented Wireless Networks and*

Communications (CrownCom), pages 327 – 335, Orlando, FL, USA, August 2007. IEEE.

[26] M. Hoyhtya, S. Pollin, and A. Mammela. Performance improvement with predictive channel selection for cognitive radios. In *Proceedings of the 1st International Workshop on Cognitive Radio and Advanced Spectrum Management (CogART)*, pages 1–5, Aalborg, Denmark, May 2008. IEEE.

[27] Chunsheng Xin, Liangping Ma, and Chien-Chung Shen. Path-centric channel assignment in cognitive radio wireless networks. In *Proceedings of the 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pages 313 – 320, Orlando, FL, USA, August 2007. IEEE.

[28] M. Benveniste. Wireless LANs and 'neighborhood capture'. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2148 – 2154. IEEE, September 2002.

[29] Jean Walrand Jeonghoon Mo, Hoi-Sheung Wilson So. Comparison of multi-channel MAC protocols. In *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Montreal, Canada, October 2005. ACM.

[30] J. C.-P. Wang, M. Abolhasan, F. Safaei, and D. Franklin. A survey on control separation techniques in multi-radio multi-channel MAC protocols. In *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT)*, Sydney, Australia, October 2007. IEEE.

[31] Huimin Chen, Zhou Wei, Min Jia, and Xihao Chen. A new multi-channel MAC protocol with power control for ad hoc networks. In

Proceedings of the IET International Conference on Wireless, Mobile and Multimedia Networks, Hangzhou, China, November 2006. IEEE.

[32] Nakjung Choi, Yongho Seok, and Yanghee Choi. Multi-channel MAC protocol for mobile ad hoc networks. In *Proceedings of the 58^{th} IEEE Vehicular Technology Conference (VTC-Fall)*. IEEE, October 2003.

[33] Luo Tie, M. Motani, and V. Srinivasan. CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks. In *Proceedings of the 3^{rd} International Conference on Broadband Communications, Networks and Systems (BROADNETS)*, San Jose, USA, October 2006. IEEE.

[34] Chunghwan Son, Neung-Hyung Lee, Byungseung Kim, and Saewoong Bahk. MAC protocol using asynchronous multi-channels in ad hoc networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Kowloon, Hong Kong, March 2007. IEEE.

[35] P. Tan and Mun Choon Chan. AMCM: Adaptive multi-channel MAC protocol for IEEE 802.11 wireless networks. In *Proceedings of the 3^{rd} International Conference on Broadband Communications, Networks and Systems (BROADNETS)*, San Jose, USA, October 2006. IEEE.

[36] Jingbin Zhang, Gang Zhou, Chengdu Huang, S.H. Son, and J.A. Stankovic. TMMAC: an energy efficient multi-channel MAC protocol for ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Glasgow, UK, June 2007. IEEE.

[37] Cheng-Shien Lin, Meng-Chun Wueng, Ting-Hung Chiu, and Shyh-In Hwang. Concurrent multi-channel transmission (CMCT) MAC protocol in wireless mobile ad hoc networks. In *Proceedings of the 9^{th} International Conference on Advanced Communication Technology*, Gangwon-Do, South Korea, February 2007. IEEE.

[38] Sheung Hoi, W. So, J. Walrand, and Jeonghoon Mo. McMAC: A parallel rendezvous multi-channel MAC protocol. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Kowloon, Hong Kong, March 2007. IEEE.

[39] Carlos Cordeiro, Kiran Challapali, and Monisha Ghosh. Cognitive PHY and MAC layers for dynamic spectrum access and sharing of TV bands. In *Proceedings of the 1st international workshop on technology and policy for accessing spectrum (TAPAS)*, Boston, MA, August 2006. ACM.

[40] Tao Chen, Honggang Zhang, G.M. Maggio, and I. Chlamtac. Topology management in cogmesh: A cluster-based cognitive radio mesh network. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Glasgow, UK, June 2007. IEEE.

[41] C. Cordeiro and K. Challapali. C-MAC: A cognitive MAC protocol for multi-channel wireless networks. In *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 147–157, Dublin, Ireland, April 2007. IEEE.

[42] R. Hancock, G. Karagiannis, J. Loughney, and S. Van den Bosch. Next steps in signaling (NSIS): framework. Technical Report RFC4080, Internet Engineering Task Force, 2005.

[43] Internet Engineering Task Force. IETF. `http://www.ietf.org`.

[44] Gahng-Seop Ahn, Andrew T. Campbell, Andras Veres, and Li-Hsiang Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN). *IEEE Transactions on Mobile Computing*, 1(3):192–207, July 2002.

[45] S. Blake. An architecture for differentiated services. Technical Report RFC2475, Internet Engineering Task Force, 1998.

[46] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. Technical Report RFC1633, Internet Engineering Task Force, 1994.

[47] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 60(4):374–406, April 2000.

[48] Hannan Xiao, W.K.G. Seah, A. Lo, and K.C. Chua. A flexible quality of service model for mobile ad-hoc networks. In *Proceedings of the IEEE 51st Vehicular Technology Conference (VTC-Spring)*, pages 445–449, Tokyo, Japan, May 2000. IEEE.

[49] He Yan and H. Abdel-Wahab. HQMM: A hybrid QoS model for mobile ad-hoc networks. In *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC))*, pages 194–200, Cagliari, Italy, June 2006. IEEE.

[50] X. Fu, H. Schulzrinne, A. Bader, D. Hogrefe, C. Kappler, G. Karagiannis, H. Tshofenig, and S. V. d. Bosch. NSIS: A new extensible IP signaling protocol suite. *IEEE Communications Magazine*, 43(10):133–141, October 2005.

[51] M.M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans J. DIMSUMnet: New directions in wireless networking using coordinated dynamic spectrum. In *Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pages 78–85, Taormina, Italy, June 2005. IEEE.

[52] Q. Zhang and Y.-Q. Zhang. Cross-layer design for QoS support in multihop wireless networks. *Proceedings of the UEEE*, 96(1):64–76, January 2008.

[53] Lichun Bao and J. J. Garcia-Luna-Aceves. Topology management in ad hoc networks. In *Proceedings of the 4th ACM international symposium on mobile ad hoc networking & computing (MobiHoc)*, Annapolis, USA, June 2003. ACM.

[54] Daqing Gu and Jinyun Zhang. QoS enhancement in IEEE 802.11 wireless local area networks. *IEEE Communications Magazine*, 41(6):120–124, June 2003.

[55] T.C.-K. Hui and Chen-Khong Tham. Adaptive provisioning of differentiated services networks based on reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 33(4):492–501, November 2003.

[56] R. Arroyo-Valles, R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro. Q-Probabilistic routing in wireless sensor networks. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, Melbourne, Australia, December 2007. IEEE.

[57] F.R. Yu, V.W.S. Wong, and V.C.M. Leung. A new QoS provisioning method for adaptive multimedia in wireless networks. *IEEE Transactions on Vehicular Technology*, 57(3):1899–1909, May 2008.

[58] A. Alaya-Feki, E. Moulines, and A. LeCornec. Dynamic spectrum access with non-stationary multi-armed bandit. In *Proceedings of the IEEE 9th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 416–420, Recife, Brazil, July 2008. IEEE.

[59] F. Bernardo, R. Agusti, J. Perez-Romero, and O. Sallent. A self-organized spectrum assignment strategy in next generation OFDMA networks providing secondary spectrum access. In *Proceedings of the 3rd IEEE International Conference on Communications (ICC)*, pages 1–5, Dresden, Germany, June 2009. IEEE.

[60] U. Berthold, Fangwen Fu, M. van der Schaar, and F.K. Jondral. Detection of spectral resources in cognitive radios using reinforcement learning. In *Proceedings of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–5, Chicago, USA, October 2008. IEEE.

[61] Y.B. Reddy. Detecting primary signals for efficient utilization of spectrum using Q-learning,. In *Proceedings of the 5th International Conference on Information Technology: New Generations (ITNG)*, pages 360–365, Las Vegas, USA, April 2008. IEEE.

[62] Jiang Tao, D. Grace, and Yiming Liu. Performance of cognitive radio reinforcement spectrum sharing using different weighting factors. In *Proceedings of the 3rd International Conference on Communications and Networking in China (ChinaCom)*, pages 1195–1199, Hangzhou, China, August 2008. IEEE.

[63] Mengfei Yang and D. Grace. Cognitive radio with reinforcement learning applied to heterogeneous multicast terrestrial communication systems. In *Proceedings of the 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 1–6, Hannover, Germany, June 2009. IEEE.

[64] Jelle R. Kok and Nikos Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *The Journal of Machine Learning Research*, 7:1789–1828, December 2006.

[65] M.W.M. Seah, Chen-Khong Tham, V. Srinivasan, and Ai Xin. Achieving coverage through distributed reinforcement learning in wireless sensor networks. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, pages 425–430, Melbourne, Australia, December 2007. IEEE.

[66] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea. Applications of machine learning to cognitive radio networks. *IEEE Wireless Communications*, 14(4):47–52, August 2007.

[67] D. Niyato and E. Hossain. Competitive spectrum sharing in cognitive radio networks: a dynamic game approach. *IEEE Transactions on Wireless Communications*, 7(7):2651–2660, July 2008.

[68] Shu Tao, Shuguang Cui, and M. Krunz. WLC05-3: medium access control for multi-channel parallel transmission in cognitive radio networks. In *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, pages 1–5, San Francisco, USA, November 2006. IEEE.

[69] A. Chia-Chun Hsu, D.S.L. Weit, and C.-C.J. Kuo. A cognitive MAC protocol using statistical channel allocation for wireless ad-hoc networks. In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, pages 105–110, Kowloon, Hong Kong, March 2007. IEEE.

[70] Mansi Thoppian, S. Venkatesan, and Ravi Prakash. CSMA-based MAC protocol for cognitive radio networks. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–8, Espoo, Finland, June 2007. IEEE.

[71] I.N. Vukovic and N. Smavatkul. Delay analysis of different backoff algorithms in IEEE 802.11. In *Proceedings of the IEEE 60th Vehicular Technology Conference (VTC-Fall)*, pages 4553–4557. IEEE, September 2004.

[72] INET. Inet framework for OMNet++/OMNEST. `http://www.omnetpp.org/doc/INET/`.

[73] Matlab. Matrix Laboratory. `http://www.mathworks.com/`.

[74] QualNet. Qualnet Network Simulator. http://www.scalable-networks.com/ .

[75] NS2. Network Simulator 2. http://www.isi.edu/nsnam/ns/.

[76] Xiaowen Gong, Wei Yuan, Wei Liu, Wenqing Cheng, and Shu Wang. A cooperative relay scheme for secondary communication in cognitive radio networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6. IEEE, November 2008.

[77] Zhu Han, C. Pandana, and K.J.R. Liu. Distributive opportunistic spectrum access for cognitive radio using correlated equilibrium and no-regret learning. In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, pages 11–15. IEEE, March 2007.

[78] M. Maskery, V. Krishnamurthy, and Qing Zhao. Decentralized dynamic spectrum access for cognitive radios: cooperative design of a non-cooperative game. *IEEE Transactions on Communications*, 57(2):459–469, February 2009.

[79] Madhusudhan R. Musku, Anthony T. Chronopoulos, Satish Penmatsa, and Dimitrie C. Popescu. A game theoretic approach for medium access of open spectrum in cognitive radios. In *Proceedings of the 2^{nd} International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pages 336–341. IEEE, August 2007.

[80] N. Nie and C. Comaniciu. Adaptive channel allocation spectrum etiquette for cognitive radio networks. In *Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 269–278. IEEE, November 2005.

[81] Hai Ngoc Pham, Jie Xiang, Yan Zhang, and T. Skeie. QoS-aware channel selection in cognitive radio networks: A game-theoretic approach. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–7. IEEE, November 2008.

[82] Hang Qin, Hui Wang, and Huaibei Zhou. A selfish game-theoretic approach for cognitive radio networks with dynamic spectrum sharing. In *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE)*, pages 1105–1109. IEEE, December 2008.

[83] Hsien-Po Shiang and M. van der Schaar. Queuing-based dynamic channel selection for heterogeneous multimedia applications over cognitive radio networks. *IEEE Transactions on Multimedia*, 10(5):896–909, August 2008.

[84] S. Subramani, T. Basar, S. Armour, D. Kaleshi, and Fan Zhong. Non-cooperative equilibrium solutions for spectrum access in distributed cognitive radio networks. In *Proceedings of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–5. IEEE, October 2008.

[85] Ji Zhu and K.J.R. Liu. Dynamic spectrum sharing: A game theoretical overview. *IEEE Communications Magazine*, 45(5):88–94, May 2007.

[86] Spiros Kapetanakis, Daniel Kudenko, and M. J. A. Strens. Reinforcement learning approaches to coordination in cooperative multi-agent systems. In *Adaptive Agents and Multi-Agent Systems*. Springer Berlin/Heidelberg, January 2003.

[87] M. Bowling and M. Veleso. Multiagent learning using a variable learning rate. *ACM Artificial Intelligence*, 136(2):215–250, April 2002.

[88] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):156–172, March 2008.

[89] S Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, pages 426–431, August 1994.

[90] Robert Crites and Andrew Barto. Improving elevator performance using reinforcement learning. *Advances in Neural Information Processing Systems*, 8(1):1017–1023, 1996.

[91] Maja J. Mataric. Reward functions for accelerated learning. In *Proceedings of the 11th International Conference on Machine Learning (ICML)*, pages 426–431, 1994.

[92] Maja J. Mataric. Learning in multi-robot systems. In *Proceedings of the Workshop on Adaption and Learning in Multi-Agent Systems*, pages 152–163, 1995.

[93] Maja J. Mataric. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, October 1997.

[94] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[95] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC Research Report TR-301, 1984.