# THÈSE

Prśentée pour obtenir le grade de Docteur en Sciences de l'Université d'Avignon et des Pays de Vaucluse France

**SPÉCIALITÉ : Informatique**

École Doctorale 536: *Sciences et Agrosciences*

Laboratoire d'Informatique d'Avignon (UPRES No 4128)

# *Optimisation des Systèmes Partiellement Observables dans les Réseaux Sans-fil: Théorie des jeux, Auto-adaptation et Apprentissage*

par

**Oussama Habachi**

**Soutenue publiquement le 28 Septembre devant un jury composé de :**

| | | |
|---|---|---|
| Tijani Chahed | Professeur à Telecom Sud-Paris | Rapporteur |
| Konstantin Avrachenkov | Directeur de recherche à l'INRIA Sophia-antipolis | Rapporteur |
| Mérouane Debbah | Professeur à SUPELEC, Paris | Examinateur |
| Eitan Altman | Directeur de recherche à l'INRIA Sophia-antipolis | Directeur de thèse |
| Yezekael Hayel | Maitre de conférence à l'Université d'Avignon | Co-directeur |
| Rachid El-Azouzi | Maitre de conférence à l'Université d'Avignon | Co-directeur |

*Laboratoire LIA*, Avignon


# THESIS

A thesis submitted in partial fulfillment for the degree of Doctor of Phylosophy

**IN : Computer Science**

Doctoral School 536: *Sciences and Agrosciences*

Laboratory of Informatique of Avignon (UPRES No 4128)

# *Optimization of Partially Observable Systems in Wireless Networks: Game Theory, Self-adaptivity and Learning*

by

**Oussama Habachi**

**Commity :**

| | |
|---|---|
| Tijani Chahed | Professor at Telecom Sud-Paris |
| Konstantin Avrachenkov | Director of research at INRIA Sophia-antipolis |
| Mérouane Debbah | Professor at SUPELEC, Paris |
| Eitan Altman | Director of research at INRIA Sophia-antipolis |
| Yezekael Hayel | Assistant professor at the University of Avignon |
| Rachid El-Azouzi | Assistant professor at the University of Avignon |


*Laboratoire LIA*, Avignon

UNIVERSITY OF AVIGNON

# *Abstract*

Since delay-sensitive and bandwidth-intense multimedia applications have emerged in the Internet, the demand for network resources has seen a steady increase during the last decade. Specifically, wireless networks have become pervasive and highly populated. These motivations are behind the problems considered in this dissertation. The topic of my PhD is about the application of game theory, queueing theory and learning techniques in wireless networks under some QoS constraints, especially in partially observable environments. We consider different layers of the protocol stack. In fact, we study the Opportunistic Spectrum Access (OSA) at the Medium Access Control (MAC) layer through Cognitive Radio (CR) approaches. Thereafter, we focus on the congestion control at the transport layer, and we develop some congestion control mechanisms under the TCP protocol.

The roadmap of the research is as follows. Firstly, we focus on the MAC layer, and we seek for optimal OSA strategies in CR networks. We consider that Secondary Users (SUs) take advantage of opportunities in licensed channels while ensuring a minimum level of QoS. In fact, SUs have the possibility to sense and access licensed channels, or to transmit their packets using a dedicated access (like 3G). Therefore, a SU has two conflicting goals: seeking for opportunities in licensed channels, but spending energy for sensing those channels, or transmitting over the dedicated channel without sensing, but with higher transmission delay. We model the slotted and the non-slotted systems using a queueing framework. Thereafter, we analyze the non-cooperative behavior of SUs, and we prove the existence of a Nash equilibrium (NE) strategy. Moreover, we measure the gap of performance between the centralized and the decentralized systems using the Price of Anarchy (PoA).

Even if the OSA at the MAC layer was deeply investigated in the last decade, the performance of SUs, such as energy consumption or Quality of Service (QoS) guarantee, was somehow ignored. Therefore, we study the OSA taking into account energy consumption and delay. We consider, first, one SU that access opportunistically licensed channels, or transmit its packets through a dedicated channel. Due to the partial spectrum sensing, the state of the spectrum is partially observable. Therefore, we use the Partially Observable Markov Decision Process (POMDP) framework to design an optimal OSA policy for SUs. Specifically, we derive some structural properties of the value function, and we prove that the optimal OSA policy has a threshold structure.

Thereafter, we extend the model to the context of multiple SUs. We study the non-cooperative behavior of SUs and we prove the existence of a NE. Moreover, we highlight a paradox in this situation: more opportunities in the licensed spectrum may lead to worst performances for SUs. Thereafter, we focus on the study of spectrum management issues. In fact, we introduce a spectrum manager to the model, and we analyze the hierarchical game between the network manager and SUs.

Finally, we focus on the transport layer and we study the congestion control for wireless networks under some QoS and Quality of Experience (QoE) constraints. Firstly, we propose a congestion control algorithm that takes into account applications' parameters and multimedia quality. In fact, we consider that network users maximize their expected multimedia quality by choosing the congestion control strategy. Since users ignore the congestion status at bottleneck links, we use a POMDP framework to determine the optimal congestion control strategy. Thereafter, we consider a subjective measure of the multimedia quality, and we propose a QoE-based congestion control algorithm. This algorithm bases on QoE feedbacks from receivers in order to adapt the congestion window size. Note that the proposed algorithms are designed based on some learning methods in order to face the complexity of solving POMDP problems.

Université d'Avignon et des Pays de Vauclus

# *Résumé et organisation de la thèse*

**Mots clès** : Théorie des jeux, Évaluation de performances, Apprentissage.

La dernière décennie a vu l'émergence d'Internet et l'apparition des applications multimédia qui requièrent de plus en plus de bande passante, ainsi que des utilisateurs qui exigent une meilleure Qualité de Service. Dans cette perspective, beaucoup de travaux ont été effectués pour améliorer l'utilisation du spectre sans fil. Le sujet de ma thèse de doctorat porte sur l'application de la théorie des jeux, la théorie des files d'attente et l'apprentissage dans les réseaux sans-fil, en particulier dans des environnements partiellement observables. Nous considérons différentes couches du modèle OSI. En effet, nous étudions l'accès opportuniste au spectre sans fil à la couche MAC en utilisant la technologie des radios cognitifs (CR). Par la suite, nous nous concentrons sur le contrôle de congestion à la couche transport, et nous développons des mécanismes de contrôle de congestion pour le protocole TCP.

Les expériences de la Federal Communication Commission (FCC) révèlent que le spectre sans fil est encore très peu utilisé. Afin d'optimiser son utilisation, la technologie CR a émergée ces dernières années. En effet, cette technologie a permis d'explorer au mieux les opportunités qui existent dans le spectre fréquentiel. Tout d'abord, nous nous concentrons sur la couche MAC, et nous étudions les stratégies d'accès opportuniste au spectre sans fil pour les utilisateurs secondaires (SUs). Nous considérons que les SUs peuvent profiter des opportunités dans les canaux primaires, tout en assurant un niveau minimal de qualité de service. Nous considérons que les SUs peuvent également choisir de transmettre leurs paquets en utilisant un canal dédié (comme la technologie 3G par exemple). Par conséquent, les SUs ont deux objectifs conflictuels: la recherche des opportunités dans les canaux primaires, mais en dépensant de l'énergie pour détecter les canaux non utilisés, ou la transmission à travers le canal dédié, mais avec un délai de transmission plus élevé. Nous modélisons le système en utilisant la théorie des files d'attente et la théorie des jeux, et nous considérons à la fois le modèle slotté et le modèle non-slotté. Par la suite, nous analysons le comportement non-coopératif des SUs, et nous prouvons l'existence d'un équilibre de Nash entre les SUs. En outre, nous proposons une expression analytique du prix de l'anarchie qui mesure l'écart de performance entre le système centralisé et le système décentralisé.

Malgré que l'OSA à la couche MAC a été profondément étudié dans la dernière décennie, les performances des SUs, telles que la consommation d'énergie ou la qualité de service,

ont été en quelque sorte ignorées. Nous étudions l'OSA avec des contraintes d'énergie et de qualité de service. Nous considérons, en premier lieu, que les SUs peuvent accéder aux canaux primaires, ou transmettre à travers un des canaux dédiés. L'état du spectre sans-fil est partiellement observable par les SUs. Par conséquent, nous utilisons les Processus de Décision Markovien dans les environnements Partiellement Observables (POMDP) pour concevoir la stratégie optimale d'OSA pour les SUs. Plus précisément, nous tirons certaines propriétés structurelles de la *value function* et nous prouvons que la politique optimale d'OSA pour les SUs est une politique à seuils.

Par la suite, nous étudions le modèle dans un contexte multi-utilisateurs. Nous analysons le comportement non-coopératif des SUs et nous prouvons l'existence d'un équilibre de Nash. En outre, nous mettons en évidence un paradoxe dans cette situation: plus de disponibilités du spectre sans fil peut engendrer une perte de performance pour les SUs. Par la suite, nous nous concentrons sur l'étude de la gestion du spectre sans-fil. En effet, nous introduisons un gestionnaire de spectre dans le modèle, et nous analysons le jeu hiérarchique entre le gestionnaire du réseau et les SUs. Plus précisément, nous prouvons l'existence d'un équilibre de Stackelberg, une stratégie commune pour les SUs et le gestionnaire du réseau de telle sorte que l'utilisation du spectre soit optimisée.

Enfin, nous nous concentrons sur le contrôle de congestion à la couche transport et nous étudions le contrôle de la congestion pour les réseaux sans fil avec des contraintes de qualité de service et de qualité d'expérience. Nous proposons, en premier lieu, un algorithme de contrôle de congestion qui prend en compte les paramètres des applications et la qualité multimédia. En effet, nous considérons que les utilisateurs maximisent leur qualité multimédia en choisissant une politique de contrôle de congestion. Etant donnée que les utilisateurs ne connaissent pas l'état de congestion aux goulots d'étranglement dans le réseau, nous utilisons un POMDP pour modéliser le contrôle de congestion. Par la suite, nous considérons une mesure subjective de la qualité multimédia, et nous proposons un algorithme basé sur la qualité d'expérience pour contrôler la congestion. En effet, Les utilisateurs adaptent la taille de leur fenêtre de congestion en se basant sur les rétroactions de la qualité d'expérience. Nous utiliserons des modèles d'apprentissage pour concevoir les algorithmes de contrôle de congestion afin de remédier à la complexité des solutions des problèmes POMDP.

# *Acknowledgements*

First and foremost, I am thankful to Allah Subhanahu wa-taala that by his grace and bounty, I am able to write my PhD thesis.

I wish to express my sincere gratitude to the Reviewers, Professor Tijani Chahed, from Telecom Sud-Paris, France and Full Time Researcher, and Konstantin Avratchenkov, Director of Research at INRIA - Sophia Antipolis - France, for their useful reviews and suggestions that really improved the quality of the manuscript. I am grateful to Professor Merouane Debbah, Professor from Supelec to be the part of the Jury selected for my thesis defense.

I am deeply indebted to my supervisors Eitan Altman, Yezekael Hayel and Rachid El-azouzi, whose professional suggestions and guidance helped me in my research work. They have been an immense help in guiding, directing and influencing my work.

I deeply acknowledge Professor Mihaela van der Schaar from UCLA, Los Angeles for inviting me as a visiting student. Her professional suggestions and guidance helped me in my research work.

Finally, I want to thank to my family members for all their love, patience, and encouragement.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **AIMD** | **A**dditive **I**ncrease **M**ultiplicative **D**ecrease |
| **CR** | **C**ognitive **R**adio |
| **DP** | **D**ynamic **P**rogrmming |
| **DSA** | **D**ynamic **S**pectrum **A**ccess |
| **DSS** | **D**ynamic **S**pectrum **S**haring |
| **EIMD** | **E**xponential **I**ncrease **M**ultiplicative **D**ecrease |
| **FCC** | **F**requency **C**ommunications **C**ommission |
| **FTC** | **F**ollow **T**he **C**rowd |
| **FIFO** | **F**isrt **I**n **F**ist **O**ut |
| **GSM** | **G**lobal **S**ystem for **M**obile communication |
| **IIAD** | **I**nverse **I**ncrease **A**dditive **D**ecrease |
| **IP** | **I**neternet **P**rotocol |
| **LIFO** | **L**ast **I**n **F**ist **O**ut |
| **LP** | **L**inear **P**rogramming |
| **MAC** | **M**edium **A**ccess **C**ontrol |
| **MCU** | **M**ultimedia **C**ontroller **U**nit |
| **MDP** | **M**arkov **D**ecision **P**rocess |
| **MOS** | **M**ean **O**pinion **S**core |
| **NAT** | **N**etwork **A**ddress **T**ranslation |
| **NE** | **N**ash **E**quilibrium |
| **PoA** | **P**rice **o**f **A**narchy |
| **POMDP** | **P**artially **O**bservable **M**arkov **D**ecision **P**rocess |
| **POSG** | **P**artially **O**bservable **S**tochastic **G**ame |
| **PS** | **P**rocess **S**haring |
| **PSTN** | **P**ublic **S**witched **T**elephone **N**etwork |

| | |
|---|---|
| **PSNR** | **P**eak **S**ignal to **N**oise **R**atio |
| **PU** | **P**rimary **U**ser |
| **PWLC** | **P**iece**W**ise **L**inear and **C**onvex |
| **QoS** | **Q**uality **o**f **S**ervice |
| **QoE** | **Q**uality **o**f **E**xperience |
| **RF** | **R**adio **F**requency |
| **RTT** | **R**ound **T**rip **T**ime |
| **RTP** | **R**eal-time **T**ransport **P**rotocol |
| **RTCP** | **R**eal-time **T**ransport **C**ontrol **P**rotocol |
| **SIP** | **S**ession **I**nitiation **P**rotocol |
| **SNE** | **S**ymmetric **N**ash **E**quilibrium |
| **SDR** | **S**oftware **D**efined **R**adio |
| **SOS** | **S**pectrum **O**ccupancy **S**tate |
| **SON** | **S**elf **O**rganizing **N**etworks |
| **SQRT** | **SQ**uare **R**oo**T** inversely proportional Increase/Square Root proportional Decrease |
| **SU** | **S**scondary **U**ser |
| **TCP** | **T**ransport **C**ontrol **P**rotocol |
| **TURN** | **T**raversal **U**using **R**elay **N**at |
| **TVWS** | **T**ele**V**ision **W**hite **S**pace |
| **UDP** | **U**ser **D**atagram **P**rotocol |
| **UHF** | **U**ltra **H**igh **F**requency |
| **UMTS** | **U**niversal **M**obile **T**elecommunication **S**ystem |
| **VHF** | **V**ery **H**igh **F**requency |
| **WiFi** | **Wi**reless **Fi**delity |
| **WiMAX** | **W**orldwide **I**nteroerability for **M**icrowave **A**ccess |

# *Dedicated:*

***to my grandfather**, he was here at the beginning but gone at the end.*

***to my nephews Youssef et Omar**, they weren't here at the beginning, with all my love.*

***to my grandmother Khadouja and my aunt Monjeya**, I miss them very much.*

***to my father Mustapha, my mother Warda, my brother Ahmed, my sister Asma and my brother-in-law Baraket**, for all their love, patience, and encouragement, they have been a tremendous encouragement to me over the past three years. I couldnt have done this without them.*

***to my grandmother Mariem, My grandfadher mohamed, my aunts Basma, Rawdha, Zakya, Rachida and Rym, to my uncles Neji, Rachid, Kamel, Abda, Amar, Mokhtar, Abderahmen, Khamis**, for their love and encouragement.*

***to Sheikh Hsan**, for his spiritual guidance and moral support.*

***to Majed, Mohamed, Raiss, Habib, Said, Issam and Refka**, for all their love and encouragement.*

***to Chayma Ouhibi, Nouha Khomsi and Basma Abrougui**, for all their love and encouragement. The last two years were awesome because of your presence.*

***to my best friends Nebli, Khaled, Abidi, Mourad, Aymen, Karim, Gomaa, Salma, Zak and Rym**, while finishing my final manuscript and writing these lines, I miss them very much...*

# Part I

# Introduction

# Chapter 1

# Introduction

The first decade of the new millennium has seen the rise of the number of Internet users. Moreover, there has been a steady increase in requirements and expectations of network services. In fact, a growing number of multimedia applications, ranging from audio and video to sophisticated simulations and virtual reality environments, emerged in the Internet. Indeed, wireless networks become pervasive, highly populated and increasingly complex. Note that wireless communications become a key element in our modern life, such as cellular phones, wireless headset, Satellite TV receiver, etc. Specifically, there has been a dramatic development of the mobile telecommunication industry. In fact, the number of cellular users has already surpassed the number of users subscribing to wired telephone services. Thereby, the demand for wireless spectrum has been growing rapidly, and the spectrum scarcity is becoming a severe problem that we have to face.

We propose, in this dissertation, some applications of game theory, self-adaptivity and learning in wireless networks at different layers of the protocol stack. In fact, we study the network management at the Medium Access Control (MAC) layer, and the congestion control at the transport layer. Then, we focus on the Opportunistic Spectrum Access (OSA) at the MAC layer through the Cognitive Radio (CR) paradigm. Thereafter, we study the congestion control at the transport layer under some QoS and QoE constraints.

Basically, the term cognition (from Latin, cognoscere, "to know") was used in many disciplines to model aspects that are closely related to the concepts of knowledge, intelligence, and learning. Note that the increasing capacity of mobile devices opened doors for introducing smart behaviors and mechanisms in wireless networks. In networking, cognition is mainly motivated by the system complexity and the difficulty to develop simple decision-making elements. Recent years have seen a wide use of the words *cognitive*, *intelligent* or *smart* in different networking and communication contexts. For

example, [1] and [2] mentioned *cognitive radio*, [3] and [4] mentioned *cognitive networks*, and we find *smart radio* in [5], and *smart antennas* in [6]. All these terms are accepted with a justification for where to add cognition to the network. Joseph Mitola III presented the CR idea firstly in a seminar at KTH, The Royal Institute of Technology in Stockholm [1]. The concept of CR comes out of the aim to utilize the radio spectrum more efficiently, and opened new doors for emerging applications. In fact, experiments from the Federal Communication Commission (FCC) reveals that the wireless spectrum is not efficiently utilized (see Figure 1.1).

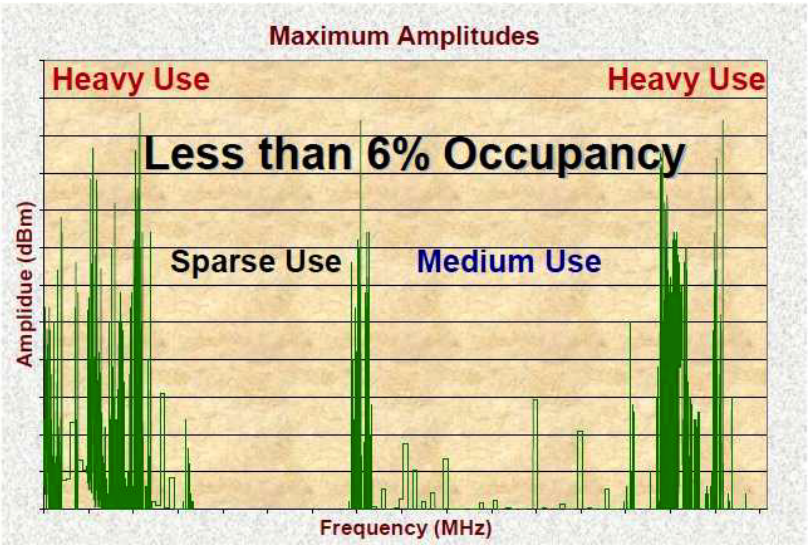


FIGURE 1.1: Opportunistic spectrum access can provide improvements in spectrum utilization (Figure taken from [7]).

In the literature, the legacy spectrum holders are denoted *primary users* (PUs), and the unlicensed users are denoted *secondary users* (SUs). It was mentioned in the FCC report [8] that PUs are unaware of the presence of SUs.

Sharing efficiently network resources has been, usually, handled in a decentralized manner at the transport layer through end-to-end congestion control mechanisms. Note that TCP dominates today's communication protocols at the transport layer, in both wireless and wired networks, due to its simple and efficient solutions for end-to-end flow control, congestion control and error control of data transmission over IP networks (see [9] and [10]). However, despite of the success of TCP, the existing TCP congestion control is considered unsuitable for delay-sensitive, bandwidth-intense, and loss-tolerant multimedia applications, such as real-time audio streaming, and video-conferences (see [9] and [11]). Therefore, multimedia users aim to utilize intelligent congestion control that is aware of the media content and the wireless environment.

## 1.1 Outline

This dissertation is structured into three parts. In the following chapter, we provide some techniques for the design, the analysis and the implementation of QoS-based applications for wireless networks in partially observable environments. Furthermore, we present some applications of game theory, self-adaptivity and learning in wireless networks. In the second part of this dissertation, we focus on the OSA in CR networks. In Chapter 3, we derive an optimal sensing policy for SUs having the possibility to transmit through a dedicated band, or to sense licensed channels. We consider both the slotted and the non-slotted models for PUs. Furthermore, we propose, in Chapter 4, an optimal energy-delay constrained OSA for CR networks. We formulate the problem using a POMDP framework, we derive some structural properties, and we prove the existence of an optimal threshold-based stationary policy. The non-cooperative OSA, in CR networks, is studied in Chapter 5. We model the OSA problem using Partially Observable Stochastic Game (POSG), and prove the existence of a symmetric Nash equilibrium (SNE) between SUs. Moreover, we study the network management in order to improve the spectrum utilization through a Stackelberg game formulation. In the third part of this dissertation, we focus on the self-adaptive congestion control at the transport layer. We present a media-aware congestion control in Chapter 6. Following this chapter, in Chapter 7, we present a QoE-aware congestion control algorithm for conversational services in wireless environments. We conclude this dissertation and give possible directions for future researches in Chapter 8. We provide all the thesis publications in Appendix A.

# Chapter 2

# Techniques for Design and Analysis of QoS-based Models in Partially Observable Environments

Contents



Unlike wired networks, in which the data transmission is isolated from interaction with other transmissions, in wireless networks, the medium is shared between all devices that are in the same transmission range. To overcome the interference between wireless devices, wireless networking technology has become an active research area in the last decade. Wireless networks are increasingly used with the advent of standards such as WiFi, WiMAX, Bluetooth and UMTS. There is no doubt that the next-generation wireless technologies promise higher levels of complexity.

This chapter is devoted to introduce the CR architecture and the congestion control, and to define some basic theoretical concepts, which will be used in the following chapters. The remaining sections of the chapter are structured as follows: In the next section, we present CR networks and their practical implementation. We introduce, in Section 2.2, the congestion control for wireless networks. Section 2.3 provides some insight about the decision theory, and we describe some basics of the queueing theory in Section 2.4. Section 2.5 introduces the game theory, and Section 2.6 introduces learning algorithms. We present some application of game theory, self-adaptivity and learning for wireless networks in Section 2.7. Finally, Section 2.8 concludes the chapter.

## 2.1 CR networks

There is a general agreement that traditional fixed spectrum allocation can be very inefficient, considering that most of the time, bandwidth that was allocated is not used and the corresponding channel is idle, which form *spectrum holes*. Although the unlicensed access to the spectrum achieves better utilization of the spectrum by using spectrum holes, (see Figure 2.1) it introduces new challenges such as: the identification of spectrum holes, the competition between SUs, etc. Note that the design of CR networks involves several disciplines, such as decision theory, queueing analysis and game theory.

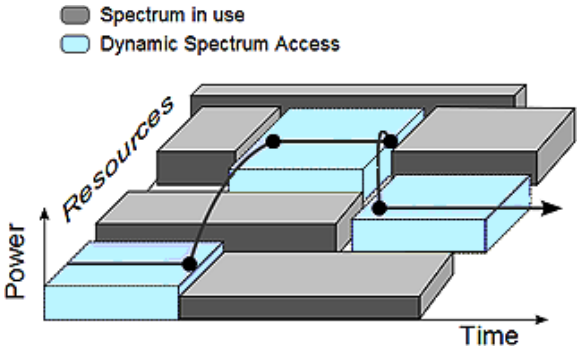


FIGURE 2.1: Wireless spectrum holes.

Furthermore, many studies showed that while some frequency bands in the spectrum are heavily used, other bands are largely unused. Note that most of the available radio spectrum was already allocated to existing wireless systems. Thus, the importance of CR paradigm aroused for allocating valuable wireless resources. The term cognition is described as the faculty of a mobile or a network to adapt its communication parameters (transmission power for mobiles or frequency for a base station) to perturbations of its environment. For instance, Ian F. Akyildiz et al. defined CR in [7] as follows:

"*A "Cognitive Radio" is a radio that can change its transmitter parameters based on interaction with the environment in which it operates*".

A big new challenge in the networking community is how to put *cognition* into networks. A radio system having this capability is called a CR, which generally uses the Software-defined Radio (SDR) technology. In fact, CR users are equipped with an SDR in order to sense and access the licensed spectrum. The SDR is considered to be the key technology that allows mobile devices to implement CR in practice. Both concepts SDR and CR are introduced in order to enhance the efficiency of the spectrum utilization in wireless systems. An SDR is defined as a reconfigurable wireless communication system that tunes dynamically its transmission parameters, such as operating frequency bands, modulation mode and transmission protocol. This adjustability can be achieved by software-controlled signal processing algorithms. The main functions of an SDR are:

- *Multi-band operation*: the ability to transmit over different frequency spectrums (cellular bands, TV bands, etc.).
- *Multi-standard support*: the ability to support different standards (GSM, WiMAX, WiFi, etc.), and different interfaces within the same standard (e.g. 802.11a, 802.11b, 802.11g in the WiFi standard).
- *Multi-service support*: the ability to support multiple types of services (3G, broadband wireless Internet, etc.).
- *Multi-channel support*: the ability to transmit and to receive over multiple frequency bands simultaneously.

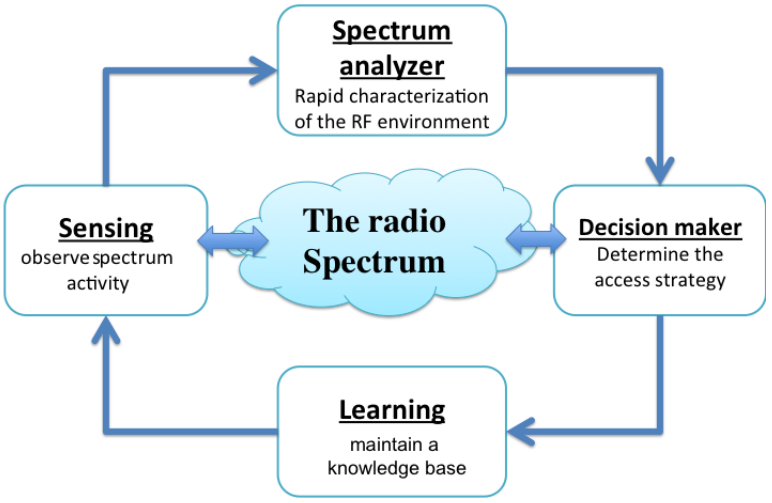


FIGURE 2.2: Components of a CR user.

A CR is aware of its environment, the internal state and predefined objectives, and looks for channel occupancy, modulation, etc., in order to make decision about its behavior.

For instance, a CR user may use SDR, so that it can reconfigure itself in order to optimize its transmission parameters. We illustrate, in Figure 2.2, the architecture of a CR node. The different components of a CR user are defined as follows:

- *An SDR-based wireless transceiver* that observes the activity of the frequency spectrum, and changes dynamically its transmission parameters.
- *A spectrum analyzer* that uses measured signals to analyze the spectrum utilization and ensure that the transmission over the spectrum is not interfered with PUs. Various signal processing techniques can be used in order to infer the spectrum usage information.
- *A decision maker* that defines the spectrum access strategy based on knowledge of the spectrum utilization. The optimal decision depends on the PUs' behavior, as well as the competitive or cooperative behavior of SUs. Different techniques, such as optimization theory, game theory and stochastic optimization, can be used in order to obtain an optimal solution.
- *A learning and knowledge extraction mechanism* that uses information of spectrum usage to understand the RF environment, i.e. the behavior of PUs. CR users maintain a knowledge base in order to adapt their transmission parameters and achieve the desired objective.

The new spectrum-licensing paradigm, initiated by the FCC in [8], promoted the idea of using the CR technology in order to face the spectrum scarcity problem. The new spectrum licensing allows unlicensed users to access the spectrum as long as they do not harm PUs, which can be achieved by spectrum sensing or power control. With the development of the CR technology, Dynamic Spectrum Access (DSA) and OSA become promising approaches that achieve major gains in the efficiency of spectrum utilization, and solving the spectrum scarcity problem. The design of DSA involves academia and industry, as well as spectrum policy makers to deal with both technical consideration and regulatory requirements. Furthermore, the development of DSA requires multidisciplinary knowledge, such as wireless communications, signal processing, optimization, artificial intelligence, decision theory, etc. For example, the competition and the cooperation between SUs accessing the same licensed bands can be modeled using game theory and utility-based techniques.

Game theory seems an ideal mathematical tool for evaluating the performance of communication systems. Since licensed channels have been opened for the unlicensed use, several works have focused on the interaction between SUs. Note that SUs may compete or cooperate with each other when accessing the spectrum. The competitive and the

TABLE 2.1: Standards for CR Aspects

| **Aspects** | **Covering Standard Bodies** |
|---|---|
| *Definition* | IEEE Dyspan, ETSI, ITU-R. |
| *Coexistence* | IEEE 802.19, IEEE Dyspan. |
| *SDR* | IEEE Dyspan, SDR forum, ITU-R, OMG. |
| *Radio Interfaces* | IEEE 802.22, 3GPP. |
| *Heterogeneous Access* | ESTI, IEEE Dyspan. |
| *Spectrum Sensing* | IEEE 802.22, IEEE Dyspan. |

cooperative behavior of SUs was depicted in [12], [13], [14], [15], [16] and [17]. For example, authors of [18] proposed a game theoretic framework to analyze the behavior of cognitive radios for distributed adaptive channel allocation. They defined two different objective functions for the spectrum sharing games, which capture the utility of selfish users and cooperative users, respectively. Based on the utility definition for cooperative users, they showed that the channel allocation problem can be formulated as a potential game, and thus converges to a deterministic channel allocation Nash equilibrium point. The survey paper [19] presented some some application of game theory. The survey outlines research challenges and future directions in game theoretic modeling approach in CR networks.

The potential of CR users has been recently identified by various policy [8] and [20], research [21], standardization [22], [23], and [24], and commercial organizations. The IEEE 1900 Standards Committee on Next Generation Radio and Spectrum Management was established in 2005 and jointly supported by the IEEE Communications Society (ComSoc) and the IEEE Electromagnetic Compatibility (EMC) Society. The concern of IEEE 1900 is to address key standardization issues in the emerging fields of spectrum management and advanced radio system technologies such as CR, SDR, and adaptive radio systems. Tables 2.1 and 2.2 give some standards for the CR technology. The paper [25] and references therein provide an extensive study of standards in the CR field.

The licensed spectrum can be utilized by SUs through either OSA or Dynamic Spectrum Sharing (DSS). In the first approach, SUs access licensed channels only when PUs are not using them. Using the DSS, SUs are allowed to use simultaneously the spectrum with PUs, as long as their transmissions do not cause harmful interferences with PUs.

The main challenge for CR networks is to locate *spectrum holes* and distribute them efficiently. The surveys [7], [26] and [27] provide a summary about recent works and design issues in CR networks.

TABLE 2.2: IEEE Dyspan Working Groups

| | |
|---|---|
| *IEEE 1900.1* | Terminology and concepts for next generation radio systems and spectrum management. |
| *IEEE 1900.2* | Interference and coexistence analysis. |
| *IEEE 1900.3* | Conformance evaluation of SDR software modules. |
| *IEEE 1900.4* | Architectural building blocks enabling network device distributed decision-making in heterogeneous wireless access networks. |
| *IEEE 1900.5* | Policy language and policy architectures for managing CR, and for DSA applications. |
| *IEEE 1900.6* | Spectrum sensing interfaces and data structures for dynamic spectrum access and other advanced radio communication systems. |
| *IEEE P1900.7* | Radio interface for white space dynamic spectrum access radio systems supporting fixed and mobile operation. |
| *IEEE 802.22* | Wireless communication at 54-863 MHz. It has an arrangement related to the identification of PUs and defining power levels so as not to interfere with adjacent bands. It is targeting at using CR techniques to allow sharing of the TV spectrum with broadcast service. |

## 2.2 Congestion control in wireless networks

With the increase of the heterogeneity and the complexity of the Internet, the standard TCP congestion control mechanism becomes inefficient (see [28] and [29] for example). The main reasons, for this inefficiency, is that congestion signals are only indicated by packet loss, and TCP uses fixed Additive Increase Multiplicative Decrease (AIMD) algorithm to adapt the congestion window size. Nevertheless, the window size should be changed according to the network environment and the media content. Note that physical impairments of the wireless transmission medium increase the complexity of designing a media-aware congestion control for wireless environments.

Despite of the success of TCP, the existing congestion control is considered unsuitable for delay-sensitive, bandwidth-intense, and loss-tolerant multimedia applications, such as real-time audio streaming, video-conferences etc. (see [9] and [11]). There are three main reasons for this:

- First, TCP is error-free and trades transmission delay for reliability. In fact, packets may be lost during transport due to network congestion and physical impairments. TCP keeps retransmitting them until they are transmitted successfully, even with a large delay. Note that although multimedia packets are successfully received, they are not decodable if they are received after their respective delay deadlines.

- Secondly, TCP congestion control adopts an AIMD algorithm, which linearly increases its congestion window size per Round-Trip Time (RTT) when there is no packet loss, and multiplicatively decreases the congestion window size when packet loss occurs. This results in a fluctuating TCP throughput over time, which significantly increases the end-to-end packet delay, and leads to worse performances for multimedia applications [11].
- Finally, standard TCP congestion control is based on network performance metrics (namely QoS metrics) and not on a subjective metric of the quality perceived by the user (measured through the QoE). In wireless systems, where the environment has an important impact on the quality of multimedia applications, a QoE-based congestion control for TCP is welcome.

Some variant of TCP was proposed, such as TCP Vegas [30] and FAST TCP [31], using the RTT values for the congestion indication. Note that the RTT usually increase before packet losses occur when the network is congested. FAST TCP is developed at the Netlab, California Institute of Technology and now being commercialized by FastSoft. It is compatible with existing TCP algorithms, requiring modification only to the computer which is sending data.

The key idea of designing a wireless TCP is to distinguish the cause of packet loss [28]. Many schemes are proposed in the literature. For example, TCP Veno [32] estimates the backlogged packet in the buffer of the bottleneck link, as illustrated in Algorithm 1. It determines the optimal throughput the network can accommodate based on the minimal RTT, denoted $BaseRTT$. The difference between the optimal throughput and the actual throughput can be used to derive the amount of backlogged packets in the queue of the bottleneck link. TCP Veno suggests that the loss is said to be random if the number of backlogged data is below a threshold $\beta$, and congestive otherwise.

**Algorithm 1** TCP Veno Algorithm: distinguish the cause of packet loss [32]

when packet loss is detected by fast retransmit:
**if** $(DIFF < \beta)$ **then**
  $ssthresh = cwndloss \times (4/5)$;
  //where $DIFF = (cwnd/BaseRTT - cwnd/RTT) \times BaseRTT$
  //random loss ( due to bit errors ) is most likely to have occurred
**else**
  $ssthresh = cwndloss/2$ ;
  // congestive state is most likely to have occurred,
  //even there occurs random loss at this time
**end if**
when packet loss is detected by retransmit-timeout timer:
ssthresh is set to half of the current window ;
slow start is performed; // performs the same action as in Reno

## 2.3 Decision-making models

Whether we make it consciously or not, every day we make several decisions. Frequently, it is not trivial to make the right decision for some problems. Usually, decisions we take have not only immediate results or outcomes, but impact also our future decisions. Unless we take into account both present and future impact of our decisions, we may not achieve good overall performances. We study, in the following section, a decision model, useful for studying a wide range of multi-stage optimization problems.

### 2.3.1 Markov decision process

We focus, in this section, on the sequential decision model, Markov decision process (MDP), where the decision maker, usually called agent or controller, makes decisions sequentially. We denote by decision epoch, every time the agent has to make a decision. At every decision epoch, the agent observes the state of the system and chooses an action. Choosing an action in a given state has mainly two results: the agent receives a reward, and the system evolves to a possibly different state at the next decision epoch. We formulate an MDP problem as follows:

- *Decision epochs*: Denote by $\mathcal{T}$ the set of decision epochs. If this set is finite, the decision problem is said to be *finite horizon* problem, otherwise it is called an *infinite horizon* problem.
- *States*: At every decision epoch, the system occupies a state $s(t)$. $\mathcal{S}$ denotes the set of all possible states.
- *Actions*: We denote the set of actions for each state $s$ by $\mathcal{A}_s$, and the set of all possible actions is referred to as $\mathcal{A} = \cup_{s\in\mathcal{S}}\mathcal{A}_s$.
- *Immediate reward* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to I\!R$: We denote by $r_t(s, a)$, defined for state $s \in \mathcal{S}$ and action $a \in \mathcal{A}_s$, the real-valued function that assigns, for a given decision epoch $t$, a value as outcome for taking the action $a$ in the state $s$. If $r_t(s, a)$ is positive, it is called reward function. Otherwise, it is called cost function.
- *Transition probabilities* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$: When the agent takes the action $a$ in the state $s$, the system state in the next decision epoch is determined by transition probabilities $p_t(.|s, a)$. We usually assume that $\sum_{j\in\mathcal{S}} p_t(j|s, a) = 1$.
- *Decision rules*: Decision rules are functions $d_t : \mathcal{S} \to \mathcal{A}$, which specify the action choice when the system is in the state $s$ at the decision epoch $t$.

A decision rule is said to be *Markovian* if it depends on previous system states and actions only through the current state of the system, and said to be deterministic if it determines the action to be chosen with certainty. We define, in the following, strategies for agents in our decision problem.

**Definition 2.1.** A policy, contingency plan or strategy specifies the decision rule to be used at every decision epoch. A policy $\pi = (d_1, d_2, \cdots)$ is a sequence of decisions, one for every decision epoch. We denote $\Gamma$ the set of all possible policies.

**Definition 2.2.** We call a stationary policy, a policy that determines the action to be chosen depending on the system state, regardless of decision epochs. A stationary policy has the form $\pi_s = (d, d, \cdots)$, and we denote by $\Gamma_s$ the set of all stationary policies.

The utility function, denoted $U$, represents the satisfaction of the agent. Note that the agent is trying to maximize its utility function if we have considered a reward function in the instantaneous reward, or trying to minimize its utility function if we have considered a cost function in the instantaneous reward. Specifically, there are three types of utility functions: the total expected reward, the average expected reward, and the discounted expected reward, defined as follows:

- The total expected reward: $V = \sum_{t \in \mathcal{T}} r_t(s, a)$.
- The average expected reward: $V = \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t(s, a)$.
- The discounted expected reward: $V = \sum_{t \in \mathcal{T}} \gamma^t r_t(s, a)$, where $\gamma$ is a discount factor.

Note that MDP is not designed to solve decisions problems when the system state is partially observable. Hopefully, decision problems for partially observable environment can be modeled using a Partially Observable Markov Decision Process (POMDP) framework.

### 2.3.2 Partially observable Markov decision process

The POMDP is a very general and powerful framework, extending the application of MDPs to a wider range of problems. Smallwood and Sondik proposed the first exact POMDP algorithm in 1971, [33]. They proposed the value iteration algorithm to solve POMDP problems (see [33], [34] and [35]). Note that a POMDP is an MDP, in which agents are unable to observe the system state. The agent's goal remains to maximize the expected future rewards.

A POMDP can be described as a tuple $\langle \mathcal{T}, \mathcal{S}, \mathcal{A}, R, \mathcal{P}, \Omega, O \rangle$ where:

- $\mathcal{T}, \mathcal{S}, \mathcal{A}, R$ and $\mathcal{P}$ describe an MDP.
- $\Omega$ is a finite set of observations an agent can experience of its world.
- $O : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\Omega)$ is the observation function, which maps actions and states to a probability distribution over possible observations.

As the agent does not directly observe the global state of the system, it infers the global system state based on past observations and actions that can be summarized in a belief vector $\omega(t) = \{\omega_1(t), ..., \omega_{2^N}(t)\}$, where $\omega_j(t)$ is the conditional probability that the system state $s(t) = j$.

Note that a POMDP may be reduced to an MDP over the belief space. Specifically, we define, in the following, an important property of the value function for a POMDP optimization: the Piecewise Linear and Convex (PWLC) property.

It is due to Smallwood and Sondik [34] that the value function $V(\lambda(t))$ is shown to be convex and piecewise linear, as illustrated in Figure 2.3, where $\lambda(t)$ denotes the belief vector at the time slot $t$. In the example illustrated in Figure 2.3, the domain of $V(\lambda(t))$ is partitioned into a finite number of regions. Each region is characterized by a $\Upsilon$-vector. Note that the value function is given by the inner product of $\lambda(t)$ and a vector $\Upsilon_i(t)$, where $\lambda(t)$ is in the region characterized by the vector $\Upsilon_i(t)$. The belief vector is transformed into a possibly different point in the space of belief at the succeeding time slot, depending on actions and observations. Note that the domain of $V(\lambda(t-1))$ is partitioned into tree regions at the time slot $t-1$, and become partitioned into four regions at the time slot $t$. The PWLC property of the value function is the key element for designing an optimal solution for POMDP problems.

**Definition 2.3.** The value function $V(\lambda(t))$, where $\lambda(t)$ is the belief vector, is said to be PWLC if it can be represented by a finite set of $|S|$-dimensional vectors, $\Upsilon = \{\Upsilon_1, \Upsilon_2, \cdots\}$, such that $V(\lambda(t))$ is the inner product of the belief vector and a $\Upsilon$-vector.

We present, in the following section, one of the major approaches of programming that is usually used in order to solve MDP and POMDP problems.

### 2.3.3 Dynamic programming

The Dynamic Programming (DP) techniques transform complex problems, such as MDP and POMDP, into sequences of simpler subproblems. The key idea of the DP is the multi-stage nature of the optimization procedure. Richard Bellman introduced the term *dynamic programming* in 1940s. He refined this concept to the modern meaning in 1953

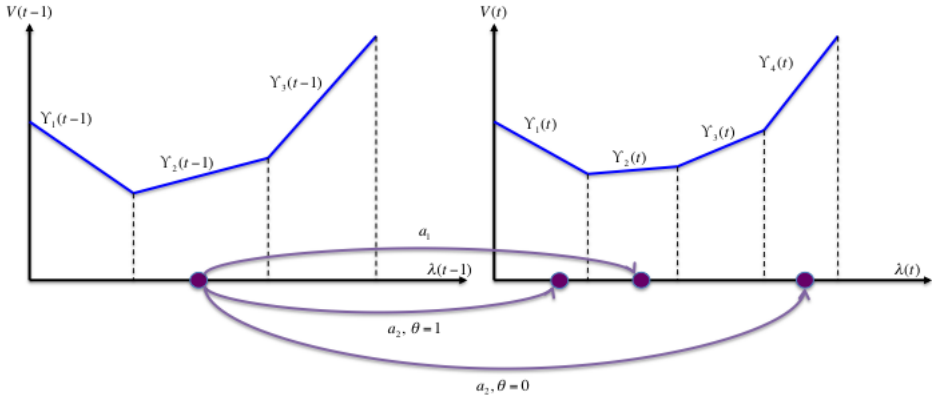


FIGURE 2.3: The structure of the value function at the time slots $t-1$ and $t$.

[36] for decision problems. The optimality of the DP solution results from the following principle of optimality:

**Definition 2.4.** In an optimal sequence of decisions or choices, each subsequence must also be optimal.

MDP and POMDP problems are solved, with the DP, by using Bellman's equations, which are also called DP equations or optimality equations.

**Definition 2.5.** The Bellman's equations are expressed as follows:

- The total expected reward: $V^{\pi}(s) = r(s, \pi(s)) + \sum_{s' \in \mathcal{S}} p(s'|s, \pi(s)) V^{\pi}(s')$.
- The average expected reward: $g_u(s_0) + V^{\pi}(s|s_0) = r(s, \pi(s)) + \sum_{s' \in \mathcal{S}} p(s'|s, \pi(s), s_0) V^{\pi}(s'|s_0)$, where $g_u(s_0)$ is a constant that depends on the initial state $s_0$.
- The discounted expected reward: $V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, \pi(s)) V^{\pi}(s')$, where $\gamma$ is a discount factor.

## 2.4 Queueing analysis

Basically, the queueing theory is the mathematical study of waiting lines or queues. Note that the queueing theory was applied in diverse fields. Specifically, the queueing theory represents an important mathematical tool for computer and network analysis. For example, the queueing analysis may answers the following questions:

- What is the packet delay at routers?
- What is the fraction of packets that will be lost?

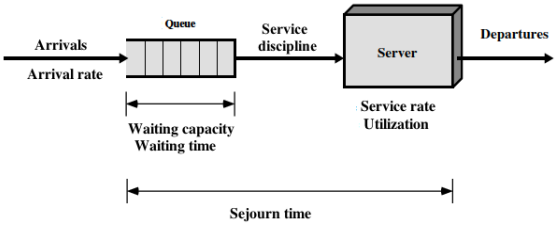


FIGURE 2.4: Single-server queueing model.

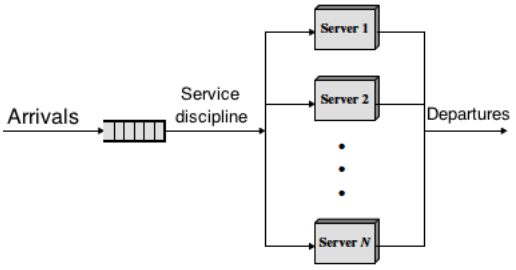


FIGURE 2.5: Multi-server queueing model.

- What is the optimal size of the buffer?

A queueing model can be either single-server (see Figure 2.4) or multi-server (see Figure 2.5), and is characterized by:

- *The arrival process*: it is usually assumed that the arrival times (of packet for example) are independent and have a common distribution. A Poisson process arrival is defined by exponential inter-arrival times.
- *The service times*: they are usually assumed to be independent and identically distributed, and independent of the inter-arrival times.
- *The service discipline*: there are many possibilities for the order in which costumers enter service (FIFO, LIFO, Random, priority, PS, etc.).
- *The service capacity*: the number of servers helping customers.
- *The waiting capacity*: the number of costumers that can be present in the system simultaneously.

Despite of the complexity of the queueing theory, its application for the performance analysis of wireless networks may be remarkably straightforward.

Kendall introduced a shorthand, four-part notation $a/b/c/d$ to characterize these queueing models. The first letter determines the inter-arrival time distribution, the second one determines the service time distribution, the third letter specifies the number of servers, and the last one represents the waiting capacity of the system. For example, the letter G denotes a general distribution, an exponential distribution is denoted by the letter M, and D denotes deterministic distribution. Some examples are M/M/1, M/M/c, M/G/1, M/M/c/K. Moreover, we have the very special PASTA [37] property:

**Definition 2.6.** For $M/\cdot/\cdot$ queueing systems with Poisson arrivals, the PASTA property holds: arriving customers find on average the same situation in the queueing system as

an outside observer looking at the system at an arbitrary point in time. More precisely, arriving customers observes the system in its stationary regime.

The major performance measures, in the analysis of queueing models, are:

- The distribution of the waiting time and the sojourn time of a customer. The sojourn time is the waiting time plus the service time.
- The distribution of the number of customers in the system.
- The distribution of the busy period of the server.

## 2.5 Game theory

### 2.5.1 Overview

In this section, we present some basics of the game theory. The game theory models the behavior of multiple players in interaction. It provides mathematical tools for studying conflicts and cooperation between rational players. Note that rational players are players *wanting more rather than less of a good*. The rationality is widely used as an assumption of the behavior of individuals in micro-economic models, and appears in almost all decision-making models.

There are several applications of game theory. If players know only their local state, the non-cooperative game may be adapted by players. In non-cooperative games, players act individually in order to maximize their own payoff. If players care about the long-term benefits, the repeated game may be employed in order to take into account future rewards. If a group of players cares about mutual benefits, the cooperative game may be employed. In fact, in cooperative games, coalitions of players, having joint actions, are formed in order to maximize a mutual utility. Finally, a stochastic game is a dynamic game with probabilistic transitions, played by one or more players.

We define a game by the following components:

- *A set of players*: $N = \{1, \cdots, n\}$.
- *A set of actions*: $A = \cup_{i \in N} A_i$, where $A_i$ is the set of all possible actions for the player $i$.
- *An utility function*: We define the utility function for player $i$, $u_i : A \rightarrow \mathbb{R}$, the player preference. We denote vector of utility functions for all players by $\mathbf{u} =$

$(u_1, \cdots, u_n) : A \to \mathbb{R}^n$. Note that the utility function represents the desirability of an action for players. An utility function for a given player assigns a number for every possible outcome of the game with the property that higher (or lower) number implies that the outcome is more preferred.

In the following, we define strategies for a player in the game.

**Definition 2.7.** A strategy of a player defines the action the player will select in every distinguishable state of the world. In repeated games, the strategy of a player is a set of decision rule, one for each stage of the game, that specify the action to be chosen.

### 2.5.2 The Nash equilibrium

The most famous property of game theory is the Nash Equilibrium [38]. The NE is an action vector such that there is no individual benefit from unilateral deviation.

**Definition 2.8.** The NE is defined as a set of strategies (one for each player), having the property that there is no increase in the utility of any player if it chooses a different action, given other players' actions. Note that $\mathbf{u}^* = (u_1^*, u_2^*, ..., u_N^*)$, is a NE if:

$$\forall i \in \{1, \ldots, N\}, \quad u_i^* = \arg\max_{u_i} R_i(u_i, \mathbf{u}_{-\mathbf{i}}^*). \tag{2.1}$$

### 2.5.3 Hierarchical game

When there is some hierarchy or priority between players in the game, the latter may be modeled using a hierarchical game. Specifically, players are divided into two sets: leaders and followers. In fact, there are two stages in the game: leaders choose, first, theirs actions, and then followers choose their actions based on observations of leaders' actions. One of the proprieties of such game is the Stackelberg Equilibrium, which is a situation where neither leaders nor followers have incentive to change theirs actions.

**Definition 2.9.** The Stackelberg equilibrium is defined as a couple of strategy profiles $(\mu^*, \mathbf{u}^*)$, where the strategy $\mu^*$ maximizes the utility of the leaders, and $\mathbf{u}^*$ is the best response of followers to leaders' strategies.

Stackelberg game formulations were already proposed in the CR literature (see for example [39] and [40]), as the natural hierarchy between PUs and SUs is very similar to the hierarchy between leaders and followers.

### 2.5.4 Partially observable stochastic games

Partially observable stochastic games [41] can be considered as an extension of stochastic games for partially observable environments [42]. It is also very closely related to the model of an extensive game with imperfect information [43]. Furthermore, POSG can be seen as an extension of a POMDP for the multi-user context [33]. In fact, POSG focus on self-interested users in partially observable environments. A POSG is a tuple $< \mathcal{I}, \mathcal{S}, \{b_0\}, \{\mathcal{A}_i\}, \{\mathcal{O}_i\}, \mathcal{P}, \{\mathcal{R}_i\} >$ defined by:

- $\mathcal{I}$ is a set $\mathcal{I} = \{1, ..., N\}$ of $N$ players.
- $\mathcal{S}$ is a finite set of states.
- $b_0$ represents the initial state distribution.
- $\mathcal{A}_i$ is a finite set of actions for player $i$.
- $\mathcal{O}_i$ is a finite set of observations for player $i$.
- $\mathcal{P}$ is a set of Markovian state transition and observation probabilities, where $\mathcal{P}(s', \mathbf{o}|s, \mathbf{a})$ denote the probability of taking the joint action $\mathbf{a}$ in state $s$ results in a transition to the state $s'$ and the joint observation $\mathbf{o}$.
- $\mathcal{R}_i : S \times A_1 \times \cdots \times A_n \to I\!R$ is a reward function for the player $i$.

## 2.6 Learning

In the architectures of future networks, where mobiles manage their communication parameters autonomously (power, frequency, ...), it is important to study learning algorithms that allow mobiles to use efficiently network opportunities. There are many applications of learning-based algorithms in the literature such as sharing spectrum in cognitive networks, routing protocols in ad hoc networks or as the distribution of traffic between operators. Specifically, MDP problems can be solved by many online reinforcement learning approaches, which can be classified into two categories: model-based approaches (e.g. RTDP [44] and Prioritized Sweeping [45]), and model-free approaches (e.g. Q-learning [46] and SARSA [47]). A model-based learning approach builds empirical models of the state evolution and the resulting reward based on interaction experiences, and applies standard DP algorithms such as value iteration or policy iteration to solve it. In contrast to the model-based approach, a model-free approach directly learns the optimal policy without specifying any model of the state evolution and reward function. There were some friendly debates within the reinforcement learning community

as to whether model-based or model-free could be shown to be clearly superior to the other (see [48] and [49]). However, all these reinforcement learning approaches suffer from the well-known curse of dimensionality problem, meaning that a practical MDP problem involves an enormous state and action spaces, which significantly impacts the complexity and the convergence time to solve the problem.

## 2.7 Some applications of game theory, self-adaptivity and learning in wireless networks

In this dissertation, we focus on the MAC layer, and we study the wireless spectrum management. Specifically, CR has been considered as a promising technology to enhance the radio spectrum efficiency via opportunistic transmission at link level. Note that locating frequencies that are not utilized by PUs, at a given time slot, represents the main challenge in designing CR networks. Moreover, SUs' transmissions depend not only on opportunities available in the licensed spectrum, but also on the competition with each other. Note that if CR users support multimedia applications, such as video streaming, VoIP or online gaming, they must be able to guarantee some QoS requirements. We further focus, in this dissertation, on self-adaptive wireless networks, where CR users are energy-efficient and have some QoS requirements that must be guaranteed.

Furthermore, we focus on the transport layer, and we study the congestion control in wireless networks under some QoS and Quality of Experience (QoE) constraints. Note that network users ignore, generally, the buffers' occupation level, which depends on the throughput of all users transmitting over the network. Specifically, we focus, in this dissertation, on the design of foresighted congestion control mechanisms for wireless networks, which are aware of the media content. We describe, in the following, some applications of game theory, self-adaptivity and learning in wireless networks.

### 2.7.1 Cognitive radio

During this dissertation, we address different layers of the protocol stack. We focus, first, on a low level of the protocol stack, the MAC layer. The first, contribution of this dissertation is an OSA policy for CR networks. In fact, we consider a system composed of several channels, where only one channel is shared between all SUs. Note that SUs have also the aptitude to sense licensed channels, and use one of them if it is idle. Specifically, we consider both the slotted and the non-slotted models, and we study the OSA as a queueing system. Thereafter, we consider that SUs may decide individually whether to sense licensed channels or to use the dedicated band. We prove the existence of a NE

between SUs, and we compare the performance of SUs at the NE with the performance of the global system, managed through a centralized controller, using the price of the anarchy (PoA).

The second contribution is a POMDP-based OSA mechanism for CR networks. In fact, we consider that SUs take into account energy consumption and QoS requirements, which were often ignored in existing OSA solutions. Specifically, we formulate the problem using a POMDP framework with an average reward criterion, and we assume that SUs may decide to use another dedicated medium of communication (such as 3G) in order to transmit their packets. We derive some structural properties of the value function, and we show the existence of optimal OSA policy in the class of threshold strategies.

Moreover, we propose two learning and knowledge extraction mechanisms. Most of researches in the OSA area assume that some information such that statistics about the activity of PUs are priory known by SUs, which may not be realistic in decentralized systems. In practice, CR users base on learning methods to get insight about the Radio Frequency (RF) environment. Specifically, we present two learning-based protocols to estimate licensed channels' dynamics: rate estimator, and transition matrices estimator.

The last, but not the least, contribution at the MAC layer is a non-cooperative OSA for CR networks. In fact, as SUs spend energy for sensing licensed channels, they may choose to be inactive during a given time slot in order to save energy. Then, there exists a tradeoff between large packet delay, due to the presence of PUs and collisions between SUs, and high-energy consumption (spent for sensing and transmitting over licensed channels). We study this problem considering a two levels approach. Firstly, we consider several SUs competing in order to access licensed channels, and we study the NE among these SUs. The NE is obtained by using a Linear Program (LP). We identify a paradox in this CR context: when licensed channels are more occupied by PUs, this may improve the spectrum utilization by SUs. Second, based on this observation, we propose a Stackelberg formulation, where a network manager may increase the occupation of licensed channels in order to improve the average throughput of SUs. We prove the existence of a Stackelberg equilibrium that maximizes the average throughput of SUs.

### 2.7.2 Transport layer

We focus on the transport layer and we highlight the following contributions: The first contribution is a media-aware congestion control mechanism. In fact, we consider several end-to-end users sharing the network. As users ignore the congestion status at bottleneck links, we model the congestion control using a POMDP framework. Moreover, we prove the existence of an optimal stationary policy, and we derive some structural properties

of the value function. Thereafter, we propose a low-complexity learning-based algorithm that can be implemented on mobile devices having a limited computational capacity.

The second contribution, at the transport layer, is a QoE-aware congestion control for conversational services in wireless environments. In fact, standard TCP congestion control is based on network performance metrics (namely QoS metrics) and not on a subjective metric of the quality perceived by the user (measured through the QoE metrics). Therefore, we propose an end-to-end QoE-based congestion control mechanism that maximizes the subjective quality of multimedia through Mean Opinion Score (MOS) feedbacks from receivers.

## 2.8 Conclusion

In this chapter, we have introduced some theoretical concepts that will be useful for the analysis of wireless networks in partially observable environments. We have presented some applications of the game theory, self adaptivity and learning in partially observable environment. Specifically, we study the OSA at the MAC layer in CR networks and the self-adaptive congestion control at the transport layer. Since the static spectrum allocation has been shown not efficient and unable to manage the increasing number of wireless users, a new licensing scheme is being developed allowing the dynamic access to the spectrum in order to improve the spectrum utilization, through CR approaches. Nevertheless, implementing the CR technology introduces new challenges about the management of the wireless spectrum. To achieve such goal, several disciplines can be involved, such as decision theory, queueing analysis and game theory. We study in the following part of this thesis the impact of the OSA mechanisms on the performance of SUs. Specifically, the next chapter focuses on the performance of SUs through a queueing analysis. We consider both the centralized and the decentralized models.

# Part II

# Opportunistic Spectrum Access in Cognitive Radio Networks

# Chapter 3

# Opportunistic Spectrum Access for Cognitive Radio Networks: A Queueing Analysis

**Contents**



## 3.1 Introduction

Since the FCC has proposed, in November 2002, to open the use of many bands that has already been assigned but not sufficiently utilized, CR-based wireless network architectures have been proposed in order to allow SUs to access licensed channels. Indeed, the FCC report reveals that the electromagnetic spectrum has gaps, i.e. frequency bands that are assigned to licensed users, at a particular time and specific geographic location, are not being utilized. Note that locating unused frequencies, accounting for the energy spent in sensing, represents a big challenge for SUs. Moreover, the proposed CR architectures do not guarantee some QoS levels for SUs, which are mainly impacted by the PUs' activity and the interaction between SUs.

The operation model, described in [20], introduces a new set of theoretical problems involving game theory, queueing theory, and decision theory. Specifically, we focus, in

this chapter, on SUs having the faculty to sense licensed bands and access them if idle, or to access a dedicated channel. We are interested in designing an optimal OSA policy for unlicensed users. In the first part of this chapter, we consider slotted communications for PUs and SUs. Indeed, we consider that the system is perfectly synchronized, and we assume that PUs and SUs have the same slot duration. Moreover, we ignore the sensing errors, i.e. the false alarm and missing probability of sensing are zero. Thus, if the SU senses a licensed channel as idle, it is still idle during the whole time slot. Most of previous works in the OSA area for CR networks have already taken these assumptions (see [50], [51], [52], and [53]). In the second part of this chapter, we consider a more realistic scenario, where PUs operate in a non-slotted mode. Due to the agreement between the service provider and PUs, the number of licensed channels should be higher than the number of PUs transmitting simultaneously. We further assume that PUs are able to determine whether there is a free licensed channel or not. As PUs have the highest priority to access their own licensed channels, if all the licensed channels are occupied, a new PU preempts a SU that is using a licensed channel. The rejected SU aborts the transmission and tries to transmit the whole packet at the next time slot. As the access to licensed channels is opportunistic, successful SUs' transmissions are highly dependent of the presence of PUs. Note that the dedicated channel represents a guarantee of a QoS level for SUs.

Lots of recent works dealt with CR technologies and their performances. The survey paper [2] presented some interesting problems for evaluating the performance of CR systems. In [54], authors considered an energy efficient spectrum access policy. Each SU senses the spectrum and selects subcarriers taking into account data rate requirements and maximum power limit. This work is close to ours as authors studied the problem by considering a non-cooperative behavior of SUs. Moreover, they considered energy efficient allocation scheme. Note that authors considered that each SU that has traffic to transmit systematically senses the spectrum and locates the available subcarrier set. In fact, authors decoupled the sensing and the access decisions, and the OSA problem is resumed to a decision about which channel to access from the set of available subcarriers. However, in our model, we consider that SUs may decide to access the dedicated channel without sensing the licensed spectrum.

Authors of [55] proposed an OSA algorithm for SUs composed of two parts: first, a SU decides whether the licensed channel is idle or not. Second, it determines whether this channel is a good opportunity or not. However, authors did not consider the impact of multiple SUs. In fact, they have focused on the model of one SU accessing opportunistically a channel licensed for a PU. In [56], [57] and [58], authors considered the non-cooperative behavior of CR users accessing multiple licensed channels.

Unlike most of previous works in the DSA area, we study decision-making methods and the corresponding equilibrium analysis using the queueing theory. Jagannathan et al. considered in [59] a model similar to ours, where SUs choose either to acquire dedicated spectrum or to use spectrum holes. They considered a pricing model and studied the interaction between SUs as a non-cooperative game. There are several differences between their work and ours. Firstly, they considered that SUs sense systematically the licensed spectrum and make the decision about transmitting over licensed channels or through the dedicated spectrum after the sensing outcome. However we consider that SUs choose the transmission medium before sensing in order to economize the energy spent for sensing when accessing dedicated bands. Secondly, they considered that there is a centralized component that schedule SUs trying to access licensed channels, whereas we consider that SUs are in competition, and collisions occur when several SUs access the same licensed channels. Moreover, authors did not consider the energy spent for sensing licensed channels.

In [60], authors considered a model where there are several channels available to choose from. The transmitter has to probe the channels to learn their quality. Probing many channels may yield one with a good gain but reduces the effective time for transmission within the channel coherence period. The problem is to obtain optimal strategies to decide when to stop probing and start transmitting.

Author of [61] proposed a cross-layer queueing model that considers multiple CR users competing for spectrum opportunities. They considered an infrastructure-based CR system consisting a CR base station and multiple CR users. The base station controls transmissions to/from CR users. In this chapter, we consider an infrastructure-less CR network, where CR users access, solely, licensed channels.

In [62], authors considered a scheduling algorithm that estimates the number of packet which can be transmitted over a frame by each SUs in each licensed channel. In contrast to this work, where a central scheduler performs the spectrum scheduling, we consider that SUs contend to access licensed channels, without the need of a central controller.

Authors of [63] applied the queueing analysis to characterize the relationship between the arrival rate of the cognitive traffic and the queue distribution of CR user. The design of cooperative CR for SUs was depicted, using a queueing analysis, in [64] and [65].

The remainder of this chapter is as follows. In the next section, we present the system model. Section 3.3 focuses on the model where PUs' transmissions are slotted. In Section 3.4, we present the non-slotted model, and we conclude the chapter in Section 3.5.

## 3.2 The system model

In this chapter, we consider a system composed of $K+1$ channels, where PUs are licensed to use $K$ channels, and one dedicated channel is shared between all SUs. PUs (resp. SUs) arrive following a Poisson process with rate $\lambda_p$ (resp. $\lambda_s$). Note that each SU decides whether to sense the licensed channels or not. If it senses the spectrum and finds one free channel, it transmits its packets. We denote by $p$ the probability that a SU senses licensed channels. This probability may be considered as the proportion of SUs that chooses to sense the spectrum. This repartition of SUs can be set by a central controller, or determined individually by SUs in a decentralized manner. Moreover, we consider that SUs are operating via a limited battery, and have to be energy efficient. We assume that there is a cost $\alpha$ for sensing one licensed channel, and if a SU decides to sense, it senses all the $K$ licensed channels. Note that SUs may sense licensed channel and stop sensing once they find a free channel. However, this strategy will increase the collision between SUs. Many works, such as [54], [59] and [66], considered that SUs sense all the licensed channels. Some other works considered periodic sensing (see [67] and [68]), whereas authors of [69] and [70] considered that the SU selects and senses randomly one licensed channel. None of these strategy was shown to be better than the others since it is highly dependent to the studied model. For example, if SU do not care about energy consumption, total sensing is the best strategy. However, if SUs do not care about the transmission delay, sensing one licensed channel (either random or periodic) may be the best strategy. The service rates are denoted by $\mu_p$ (resp. $\mu_s$) for the licensed channels (resp. the dedicated channel), and are supposed to have an exponential distribution. The system model is depicted in Figure 3.1, and is composed of two sub-systems. The first one, namely subsystem $S_1$, represents the secondary subsystem, and the primary subsystem, denoted by $S_2$, is licensed for PUs and open for SUs' opportunistic access.

We give, in the following, some intuitions about the optimal OSA strategy for SUs in our model. Because of the cost of sensing, when the blocking probability in the primary subsystem $S_2$ increases, SU have less incentive to sense the spectrum. In fact, even if SUs do not find a free licensed channel, they also pay the sensing cost. However, if they decide to use the dedicated channel without sensing, they do not pay the sensing cost but transmit their packet with higher delay than using the licensed channels. Moreover, the more there are SUs in the subsystem $S_1$, the higher is the transmission delay for all the SUs using that subsystem. Thus, there is a tradeoff for SUs whether to sense or not licensed channels. Table 3.1 summarizes the parameters of the model.

Obviously, SUs have to deal with the two following performance metrics: the packet delay and the energy spent for transmission. In fact, if the SU senses licensed channels and finds one free channel, it transmits the held packet with a lower delay than transmitting

TABLE 3.1: Description of system parameters

| Parameter | Description |
|---|---|
| $\lambda_p$ | arrival rate of PUs |
| $\lambda_s$ | arrival rate of SUs |
| $\mu_p$ | service rate in a licensed channel |
| $\mu_s$ | service rate for a SU in the dedicated channel |
| $K$ | the number of channels allocated for PUs |
| $p$ | probability of sensing licensed channels |
| $\alpha$ | the cost of sensing one channel for a SU |
| $\rho(p)$ | $\frac{(\lambda_p+p\lambda_s)}{\mu_p}$ |

over the dedicated channel. However, it spends energy for sensing licensed channels. We define the main global metric of the system, which is the average total cost $U_S$, as a composition of the two following parts: the average sojourn time of a SU inside the system and the cost of sensing:

- The average sojourn time, denoted by $T_S$, depends on several parameters: arrival rates of PUs and SUs, service rates, the number of licensed channels and the sensing probability.
- The sensing cost $c_s$ depends on the number of licensed channels, and on the probability of sensing. We assume that this cost is linear with the number of licensed channels, i.e. $c_s(p, K) = \alpha K p$. In fact, the cost of sensing represents the energy spent in sensing. Note that SUs are supposed to sense all the licensed channels.

The average total cost, for a SU that chooses to sense licensed channels with probability $p$, is given by:

$$U_S(p, K) = T_S(p, K) + c_s(p, K) = T_S(p, K) + \alpha p K. \tag{3.1}$$

The average sojourn time $T_S$ of a SU inside the system depends on the decision taken by the SU: to use licensed channels or the dedicated one. We denote by $T_{S_1}$ (resp. $T_{S_2}$) the sojourn time if the SU that decides to transmit over the dedicated channel (resp. licensed channels). We assume that the sensing period is negligible compared to the sojourn time in both subsystems. Thus, the average sojourn time $T_S$ is expressed by:

$$T_S(p, K) = (1 - p)T_{S_1}(p, K) + pT_{S_2}(p, K). \tag{3.2}$$

## 3.3 The slotted model

In this section, we consider that SUs and PUs evolve in a slotted model, and that they have the same time slots' durations. Moreover, we consider a perfect sensing, i.e. the false alarm and the missing probability equal zero. The secondary subsystem $S_1$ is composed of SUs that have not sensed the licensed channels (see Figure 3.1). Note that SUs that sensed licensed channels and do not find a free one are rejected from the system. As the dedicated channel is equally shared between all SUs, the subsystem $S_1$ can be modeled using an M/M/1 queue. In fact, SUs are sharing one dedicated channel during the time slot.

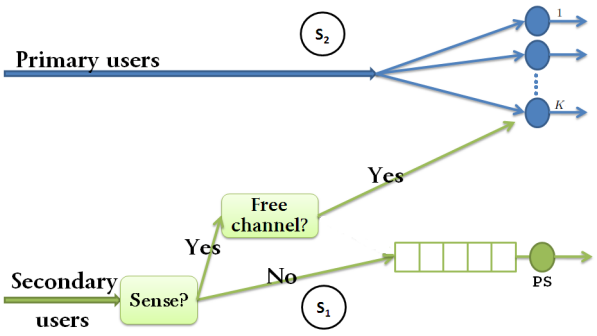


FIGURE 3.1: The OSA model for CR networks

The primary subsystem, namely $S_2$, is composed of the two following types of users:

- PUs,
- SUs that have sensed the licensed channels and have found, at least, one free channel.

The subsystem $S_2$ can be modeled using an M/M/K/K queue, known as the Erlang-B model, with arrival rate $\lambda_p + p\lambda_s$. Note that the Erlang-B model (M/M/K/K) was used in order to model CR networks in [71]. The blocking probability, which is the probability that any mobile finds all channels occupied, is given by the following Erlang-B formula:

$$\Pi(p, K) = \frac{\frac{\rho(p)^K}{K!}}{\sum_{n=0}^{K}(\frac{\rho(p)^n}{n!})}, \tag{3.3}$$

where $\rho(p) = \frac{(\lambda_p + p\lambda_s)}{\mu_p}$. This blocking probability depends not only on the number of licensed channels $K$, but also on the probability $p$ of sensing. In fact, if the sensing probability increases, the input rate in the subsystem $S_2$ increases, and the blocking probability $\Pi(p, k)$ increases. Note that, for simplicity reasons, we have considered in this section that PUs and SUs have the same priority to access licensed channels. The

paper [72] extended our model taking into account the priority of PUs in the expression of the blocking probability. However, They did not consider the possibility for a PU to reject a SU in service if it does not find a free channel. In the next section of this chapter, we consider a more general system taking into account the priority of PUs, where a PU that does not find a free channel may reject a SU in service.

In the next section, we focus on the optimal sensing probability or the optimal proportion of SUs that sense licensed channels, which minimizes two important metrics: the average sojourn time and the average total cost.

### 3.3.1 Optimization of global performances

The global analysis is well-suited for models where a CR base station transmits the traffic of SUs over multiple licensed channels in the wireless spectrum. We focus, in this section, on the average cost function of SUs. The arrival rate in the dedicated channel (subsystem $S_1$) is composed of SUs that have not sensed licensed channels. Then, the arrival rate of SUs for that dedicated channel is $\lambda_s(1-p)$. We assume that the maximum arrival rate, that is $\lambda_s$, which corresponds to the case where all SUs do not decide to sense, is lower than the service rate $\mu_s$. Thus, we have a sufficient stability condition for the M/M/1 queue with a PS policy, which models the subsystem $S_1$. As the dedicated channel is shared between all SUs, the more there are SUs transmitting over the dedicated channel, the higher is the sojourn time in the system (higher is the transmission delay). Note that QoS requirements for SUs may be achieved by using an admission control mechanism by the Service Provider (SP).

The average sojourn time $T_{S_1}$ for a SU, depending on the probability $p$ that SUs sense the licensed channels and the number of licensed channels, is expressed as follows:

$$T_{S_1}(p,K) = \frac{1}{\mu_s - \lambda_s(1-p)}. \tag{3.4}$$

If a SU decides to sense licensed channels, its average sojourn time depends on the arrival rate of the PUs $\lambda_p$, and the proportion of SUs $p\lambda_s$ that have decided to sense licensed channels. Then, we determine explicitly, in the following, the average sojourn time $T_{S_2}$ for a SU that decide to sense the licensed channels:

$$T_{S_2}(p,K) = \frac{1-\Pi(p,K)}{\mu_p}. \tag{3.5}$$

Note that a SU that decides to sense and does not find a free licensed channel is rejected from the system and try to retransmit at next time slot. Thus, the average sojourn time

of a SU in the system is expressed as follows:

$$T_S(p,K) = \frac{1-p}{\mu_s - \lambda_s(1-p)} + \frac{p(1-\Pi(p,k))}{\mu_p}. \tag{3.6}$$

For notation convenience, let us consider the following function: $X(p,K) = p(1-\Pi(p,K))$. By introducing the function $X(p,K)$ in the expression of the average sojourn time, we obtain the following simpler expression of the average sojourn time:

$$T_S(X(p,K)) = \frac{1-p}{\mu_s - \lambda_s + \lambda_s p} + \frac{X(p,K)}{\mu_p}. \tag{3.7}$$

In order to avoid the interference with PUs, SUs have to sense licensed channels before accessing them, and pay a cost for sensing. Note that SUs spend energy for sensing the spectrum. In fact, we model by the sensing cost, the energy spent for sensing licensed channels. The average cost function $U_S(p,K)$ for a SU that senses licensed channels with a probability $p$, is expressed as follows:

$$\begin{aligned} U_S(p,K) &= T_S(p,K) + \alpha p K. \\ &= \frac{1-p}{\mu_s - \lambda_s + \lambda_s p} + \frac{p(1-\Pi(p,K)}{\mu_p} + \alpha K p. \end{aligned} \tag{3.8}$$

We denote by $\Pi'(p,K)$ the derivative of the blocking probability with respect to the sensing probability $p$. The following proposition states the sensing probability that minimizes the average cost function.

**Proposition 3.1.** *For all values of $\alpha$ and $K$, the average cost function $U_S(p,K)$, defined in Equation (3.9), is minimized when the sensing probability is equal to:*

$$p = min(1, max(p_0, 0)) := p^*,$$

*where $p_0$ is the solution of the following equation:*

$$1 - \Pi(p,K) - p\Pi'(p,K) = -\alpha K \mu_p + \frac{\mu_p \mu_s}{(\mu_s - \lambda_s(1-p))^2}. \tag{3.9}$$

*Proof.* By replacing the function $X(p,K)$ in Equation (3.9), the average cost function can be rewritten as follows:

$$U_S(p,K) = \frac{1-p}{\mu_s - \lambda_s + \lambda_s p} + \frac{X(p,K)}{\mu_p} + \alpha p K.$$

After some algebra, the derivative of the average cost function, with respect to the sensing probability $p$, is expressed as follows:

$$\frac{\partial U_S}{\partial p}(p,K) = \frac{-\mu_s + \frac{\partial X}{\partial p}(p,K)(\mu_s - \lambda_s(1-p))^2 + \alpha K \mu_p (\mu_s - \lambda_s(1-p))^2}{\mu_p(\mu_s - \lambda_s(1-p))^2}.$$

Note that $\frac{\partial X(p,K)}{\partial p} = 1 - \Pi(p,K) - p\Pi'(p,K)$. Thus, the derivative of the cost function $U_S(p,K)$ equals 0 if and only if:

$$1 - \Pi(p,K) - p\Pi'(p,K) = -\alpha K \mu_p + \frac{\mu_p \mu_s}{(\mu_s - \lambda_s(1-p))^2}.$$

Therefore, the derivative of the average cost function with respect to the sensing probability equals 0 if and only if $p = min(1, max(p_0, 0)) := p^*$, where $p_0$ is the solution of Equation (3.9). □

The main drawback of the optimal sensing probability $p^*$, the solution of the global optimization, is that it needs a central controller, in order to develop an optimal OSA mechanism. Indeed, the SP has to design the network such that a proportion $p^*$ of SUs senses the licensed channels. In practice, it would be difficult to control and to design such centralized control. To overcome this hurdle, we look in the next section for a distributed mechanism, based on individual decisions of SUs about the OSA.

### 3.3.2 Individual opportunistic sensing policy

The main characteristic of the next generation networks is the transition from well-structured networks to infrastructure-less networks, and from centralized to decentralized networks. Recently, several researches focused on self-adaptive networks and autonomous devices. In this section, we consider that SUs decide individually whether to sense or not licensed channels. In fact, SUs try to minimize, solely, their average cost functions. Specifically, we model this system using a non-cooperative game with an infinite number of players (as we do not restrict neither the time horizon of the system nor the number of SUs). Note that game theory principle may be applied for resource allocation problems in a decentralized manner for wireless communications (see the survey paper [73] and [74] for some examples). Thus, we consider a game theoretical approach in order to design a decentralized OSA mechanism.

We consider that each SU decides on its probability $p$ to sense or not licensed channels. It looks for minimizing its average cost function $U(p,p',K)$, which depends on its probability $p$, and the probability $p'$ of all other SUs. The individual average cost function

$U(p, p', K)$ is expressed as follows:

$$U(p, p', K) \quad = \quad (1-p)T_{S_1}(p', K) + pT_{S_2}(p', K) + \alpha pK. \tag{3.10}$$

Note that the contribution to the cost by any individual SU is zero as we are not limited to a fixed number of SUs. Then, the equilibrium of this game is a Wardrop equilibrium, which was first studied in the context of road traffic since the 1950s in [75]. For notation convenience, we denote by $U_S(p, K) = U(p, p, K)$. Let us define, in the following theorem, the equilibrium for our non-cooperative game as a strategy that minimizes the cost function $U$, against others using the NE strategy.

**Theorem 3.2.** *The sensing probability $p^E$ is a NE policy for the OSA problem between SUs if and only if:*

$$p^E = \arg\min_p U(p, p^E, K), \ \forall p \in [0, 1].$$

The following proposition proves the existence of a NE strategy for our non-cooperative game between SUs.

**Proposition 3.3.** *For all values of $\alpha$ and $K$, the NE policy for the OSA problem between SUs exists. Moreover, the sensing probability at the NE is expressed as follows:*

- *if* $\frac{1}{\mu_s - \lambda_s} > \alpha K + \frac{1-\Pi(0,k)}{\mu_p}$;
  - *if* $\frac{1}{\mu_s} < \alpha K + \frac{1-\Pi(1,k)}{\mu_p}$ *then* $p^E = \{0, p', 1\}$.
  - *else* $p^E = 0$;
- *else*
  - *if* $\frac{1}{\mu_s} > \alpha K + \frac{1-\Pi(1,k)}{\mu_p}$ *then* $p^E = p'$;
  - *else* $p^E = 1$.

*where $p'$ is the solution of the following equation:*

$$\frac{1}{\mu_s - \lambda_s(1-p)} = \alpha K + \frac{1 - \Pi(p, K)}{\mu_p}. \tag{3.11}$$

*Proof.* From Equation (3.10), the first argument derivative of the average cost function is expressed as follows:

$$\frac{\partial U}{\partial p}(p, p') = T_{S_2}(p', K) - T_{S_1}(p', K) + \alpha K.$$

The probability $p'$ is a NE strategy for the OSA problem if and only if the first argument derivative of the average cost function equals 0:

$$\alpha K + T_{S_2}(p', K) = T_{S_1}(p', K).$$

This equation characterizes a NE strategy for SUs. After some algebra, this expression may be expressed as follows:

$$T_{S_1}(p', K) = \alpha K + \frac{1 - \Pi(p', K)}{\mu_p}.$$

Thus, the necessary and sufficient condition for the existence of a NE strategy for the OSA problem between SUs is:

$$\frac{1}{\mu_s - \lambda_s(1 - p^E)} = \alpha K + \frac{1 - \Pi(p^E, K)}{\mu_p}.$$

Let us prove that $\frac{1}{\mu_s - \lambda_s(1-p)}$ and $\alpha K + \frac{1-\Pi(p,K)}{\mu_p}$ intersect once in $[0, 1]$. Suppose that $\exists p_1 < p_2 \in [0, 1]$ such that $\frac{1}{\mu_s - \lambda_s(1-p_1)} = \alpha K + \frac{1-\Pi(p_1,K)}{\mu_p}$ and $\frac{1}{\mu_s - \lambda_s(1-p_2)} = \alpha K + \frac{1-\Pi(p_2,K)}{\mu_p}$. Therefore, we obtain:

$$\frac{\Pi(p_2, K) - \Pi(p_1, K)}{\mu_p} = \frac{\lambda_s(p_2 - p_1)}{(\mu_s - \lambda_s(1 - p_1))(\mu_s - \lambda_s(1 - p_2))}.$$

After some algebra, we obtain:

$$\frac{\mu_p \lambda_s(p_2 - p_1)}{\Pi(p_2, K) - \Pi(p_1, K)} \leq \mu_s^2 - 2\mu_s\lambda_s - \lambda_s^2 - \lambda_s p_1 p_2,$$

which leads to a contradiction as $\frac{\mu_p\lambda_s(p_2-p_1)}{\Pi(p_2,K)-\Pi(p_1,K)} > 0$ and $\mu_s^2 - 2\mu_s\lambda_s - \lambda_s^2 - \lambda_s p_1 p_2 < 0$. Note that we have assumed that $\mu_s \leq 2\lambda_s$ in order to give SUs incentive to sense and access licensed channels.

Consider that $\alpha K + \frac{1-\Pi(0,K)}{\mu_p} < \frac{1}{\mu_s - \lambda_s}$ and $\alpha K + \frac{1-\Pi(1,K)}{\mu_p} > \frac{1}{\mu_s}$. Thus, we have two equilibriums $p^E = 0$ and $p^E = 1$. These equilibriums represent a Follow The Crowd (FTC) phenomenon ( see [74]). In fact, there is an FTC behavior when the individual's tendency to choose an action increases with the probability of choosing this action by other individuals. For instance, when all SUs choose to sense licensed channels ($p' = 1$) the best response of a SU is to sense licensed channels, and therefore the equilibrium $p^E = 1$ exhibits an FTC characteristic. Moreover, there exists a unique equilibrium $p' = p' \in ]0, 1[$, where $p'$ is the unique solution of Equation 3.11. Therefore, $p^E = \{0, p', 1\}$.

Consider that $\alpha K + \frac{1-\Pi(0,K)}{\mu_p} > \frac{1}{\mu_s-\lambda_s}$ and $\alpha K + \frac{1-\Pi(1,K)}{\mu_p} > \frac{1}{\mu_s}$. It follows that $p^E = 1$ is the unique Nash equilibrium for the OSA game between SUs.

Consider that $\alpha K + \frac{1-\Pi(0,K)}{\mu_p} > \frac{1}{\mu_s-\lambda_s}$ and $\alpha K + \frac{1-\Pi(1,K)}{\mu_p} < \frac{1}{\mu_s}$. Therefore, $p^E = p'$ is the equilibrium strategy for our OSA game, where $p'$ is the solution of Equation 3.11.

Consider that $\alpha K + \frac{1-\Pi(0,K)}{\mu_p} < \frac{1}{\mu_s-\lambda_s}$ and $\alpha K + \frac{1-\Pi(1,K)}{\mu_p} < \frac{1}{\mu_s}$. It follows that $p^E = 0$ is the unique Nash equilibrium for the OSA game between SUs. □

Given the existence of a NE strategy for the OSA problem between SUs, the following proposition compares the sensing probability at the NE and the optimal sensing probability.

**Proposition 3.4.** *For all values of $\alpha$ and $K$, the optimal sensing probability is higher than the sensing probability at the NE, i.e. $p^E \leq p^*$.*

*Proof.* We prove this proposition by contradiction. Assume that there exists a sensing cost $\alpha_0 > 0$ and a number of licensed channels $K_0$ such that $p^E > p^*$. As $p^*$ minimizes the average cost function, we have:

$$T_S(p^*, K_0) + \alpha_0 p^* K_0 \leq T_S(p^E, K_0) + \alpha_0 p^E K_0.$$

However, $p^E$ is the sensing probability at the NE. Therefore, we have the following inequality:

$$T_S(p^*, p^E) + \alpha_0 p^* K_0 \geq T_S(p^E, K) + \alpha_0 p^E K_0.$$

After some algebra, combining the two previous inequalities, we obtain:

$$(1-p^*)T_{S_1}(p^*) + \frac{p^*(1-\Pi(p^*,K_0))}{\mu_p} \leq (1-p^*)T_{S_1}(p^E) + \frac{p^*(1-\Pi(p^E,K_0))}{\mu_p}.$$

It follows that:

$$(1-p^*)(T_{S_1}(p^*) - T_{S_1}(p^E)) \leq \frac{p^*}{\mu_p}(\Pi(p^*,K_0) - \Pi(p^E,K_0)).$$

Note that $T_{S_1}$ is decreasing with $p$ and $\Pi$ is increasing with $p$, then for $p^E > p^*$, the left hand side is positive and right hand one is negative which leads to a contradiction.

Finally, for all $\alpha$ and all $K$, the optimal sensing probability is higher than the sensing probability at the NE, i.e. $p^E \leq p^*$. □

This result is somehow intuitive. In fact, there is a lack of performance due to the selfishness of SUs in the decentralized system. In fact, SUs have less incentive to sense

licensed channels in a self-adaptive context than in a centralized network. Furthermore, the following proposition gives us a higher bound of the average cost function at the NE.

**Proposition 3.5.** *For all values of $\alpha$ and $K$, we have the following higher bound of the average cost function when using a NE policy:*

$$U_S(p^E, K) \leq \frac{1}{\mu_s - \lambda_s}.$$

*Proof.* Consider that $\frac{1}{\mu_s} < \alpha K + \frac{1-\Pi(1,k)}{\mu_p}$ and $\frac{1}{\mu_s - \lambda_s} < \alpha K + \frac{1-\Pi(0,k)}{\mu_p}$. Therefore, the average cost function is expressed as follows:

$$U_S(p^E, K) = \frac{1}{\mu_s - \lambda_s}.$$

Second, Consider that $\frac{1}{\mu_s} > \alpha K + \frac{1-\Pi(1,k)}{\mu_p}$ and $\frac{1}{\mu_s - \lambda_s} > \alpha K + \frac{1-\Pi(0,k)}{\mu_p}$. Thus, the average cost function verifies:

$$U_S(p^E, K) = \frac{1 - \Pi(1, K)}{\mu_p} + \alpha K \leq \frac{1}{\mu_s} \leq \frac{1}{\mu_s - \lambda_s}.$$

Otherwise, the average cost function can be bounded as follows:

$$U_S(p^E, K) = \alpha K + \frac{1 - \Pi(p^E, K)}{\mu_p} = \frac{1}{\mu_s - \lambda_s(1 - p^E)} \leq \frac{1}{\mu_s - \lambda_s}.$$

Finally, the higher bound of the average cost function is $U_S(p^E, K) \leq \frac{1}{\mu_s - \lambda_s}$. □

It is well known that the utility of the global optimization is higher than the utility when using NE strategies. Giving the existence of the NE strategy for SUs, we focus in the next section on the lack of performance (utility) induced by the competition between SUs. In order to measure this gap of performance, we introduce the metric of the PoA.

### 3.3.3 Price of anarchy

Koutsoupias and Papadimitriou [76] introduced the concept of *Price of Anarchy*, which captures the deterioration of the performance of a decentralized system, due to the selfishness of its agents. This metric is well studied in routing games [77], where the PoA describes the worst possible ratio between the total latency of a NE strategy and the latency of an optimal routing of the traffic. This metric describes the gap of performance in terms of individual utility between an optimal centralized system and a totally decentralized system.

The PoA is expressed as the ratio between the optimal utility (obtained with a centralized system) and the utility at the NE (obtained with a decentralized system when using a NE policy). In our context, we define the PoA as follows:

$$PoA = \frac{U_S(p^*, K)}{\max_{p \in p^E} U_S(p, K)} \leq 1. \tag{3.12}$$

Our aim is to determine an expression of the minimal value of the PoA or to bound it, in order to measure the worst performance of the decentralized system. The following proposition gives us the worst-case lack of performance when upgrading from centralized networks to self-adaptive networks

**Proposition 3.6.** *For all values of $\alpha$ and $K$, we have the following lower bound of the PoA:*

$$PoA(\alpha, K) \geq \frac{2(\lambda_s - \mu_s + \sqrt{\mu_s(\mu_s - \lambda_s)})}{\lambda_s} := \underline{PoA}. \tag{3.13}$$

*Proof.* The price of anarchy is expressed by the following ratio:

$$PoA(\alpha, K) = \frac{U_S(p^*, K)}{\max_{p \in p^E} U_S(p, K)}.$$

Suppose, first, that $\frac{1}{\mu_s} > \alpha K + \frac{1-\Pi(1,k)}{\mu_p}$ and $\frac{1}{\mu_s - \lambda_s} > \alpha K + \frac{1-\Pi(0,k)}{\mu_p}$. Therefore, we have $p^E = 1$. As we have proved in Proposition 3.4, $p^* \geq p^E$, then $p^* = 1$. Thus, we have $PoA(\alpha, K) = 1$. Let us focus on the gap between the utility function at the equilibrium and the optimal utility function. We have for all $p^*$, $\alpha$ and $K$

$$\begin{aligned} U_S(p^E, K) - U_S(p^*, K) &= \frac{1}{\mu_s - \lambda_s(1 - p^E)} - p^* \frac{1 - \Pi(p^*, K)}{\mu_p} - \alpha K p^* - \frac{1 - p^*}{\mu_s - \lambda_s(1 - p^*)} \\ &= -p^* \frac{1 - \Pi(p^*, K)}{\mu_p} - \alpha K p^* + \frac{p^* \mu_s - \lambda_s p^E (1 - p^*)}{(\mu_s - \lambda_s(1 - p^*))(\mu_s - \lambda_s(1 - p^E))} \end{aligned}$$

It's clear that the difference between the utility function at the equilibrium and the optimal utility function is maximal when $p^E = 0$. Note that the price of anarchy is minimal when $U_S(p^E) - U_S(p^*)$ is maximized. Then, the $PoA$ is minimized when $p^E = 0$. We focus on the analysis of the $PoA$ in this particular case. Suppose that $\frac{1-\Pi(p^*,K)}{\mu_p} + \alpha K < \frac{1}{\mu_s - \lambda_s}$. Then, we have for all $p^*$, $\alpha$ and $K$

$$U(p^*, p^*) < \frac{p^*}{\mu_s - \lambda_s} + \frac{1 - p^*}{\mu_s - \lambda_s(1 - p^*)}.$$

Thus, we obtain

$$U(p^*,0) < \frac{p^*}{\mu_s-\lambda_s} + \frac{1-p^*}{\mu_s-\lambda_s} = \frac{1}{\mu_s-\lambda_s},$$

which leads to a contradiction. In fact $U(0,0) = \frac{1}{\mu_s-\lambda_s} > U(p^*,0)$, and if $p^E = 0$ is an equilibrium, then $U(0,0) < U(p',0)$ for all $p'$. Finally, we have when $p^E = 0$, $\frac{1-\Pi(p^*,K)}{\mu_p} + \alpha K \geq \frac{1}{\mu_s-\lambda_s}$.

Moreover, when $p^E = 0$, we have the following expression of the price of anarchy:

$$PoA(\alpha,K) = \frac{\frac{p^*(1-\Pi(p^*,K))}{\mu_p} + \alpha p^* K + \frac{1-p^*}{\mu_s-\lambda_s(1-p^*)}}{\frac{1}{\mu_s-\lambda_s}}.$$

Thus, combining previous results, the price of anarchy is bounded by:

$$PoA(\alpha,K) \geq p^* + \frac{\frac{1-p^*}{\mu_s-\lambda_s(1-p^*)}}{\frac{1}{\mu_s-\lambda_s}}.$$

After some algebra, we obtain the following lower bound of the PoA:

$$PoA(\alpha,K) \geq p^* + \frac{(\mu_s-\lambda_s)(1-p^*)}{\mu_s-\lambda_s(1-p^*)} = \frac{\mu_s-\lambda_s(1-(p^*)^2)}{\mu_s-\lambda_s(1-p^*)}.$$

We denote the following function $F(X) = \frac{\mu_s-\lambda_s(1-X^2)}{\mu_s-\lambda_s(1-X)}$. The derivative of $F(X)$ with respect to $X$ is expressed as follows:

$$F'(X) = \frac{\lambda_s^2X^2 + (2\mu_s\lambda_s - 2\lambda_s^2)X + \lambda_s^2 - \lambda_s\mu_s}{(\mu_s-\lambda_s(1-X))^2}.$$

Note that $F'(X) = 0$ for $X = \frac{\lambda_s-\mu_s\pm\sqrt{\mu_s(\mu_s-\lambda_s)}}{\lambda_s}$. Moreover, we have $F(0) = 1$. Then, the function $F(X)$ is minimized when $X = \frac{\lambda_s-\mu_s+\sqrt{\mu_s(\mu_s-\lambda_s)}}{\lambda_s}$, and its minimum is $F(X) = \frac{\mu_s-\lambda_s(1-(\frac{\lambda_s-\mu_s+\sqrt{\mu_s(\mu_s-\lambda_s)}}{\lambda_s})^2)}{\mu_s-\lambda_s(1-\frac{\lambda_s-\mu_s+\sqrt{\mu_s(\mu_s-\lambda_s)}}{\lambda_s})}$.

Finally, for all $\alpha$ and $K$, we obtain the lower bound of the price of anarchy:

$$PoA(\alpha,K) \geq 2(\frac{\lambda_s-\mu_s+\sqrt{\mu_s(\mu_s-\lambda_s)}}{\lambda_s}).$$

□

This closed-form of the lower bound of the PoA is very interesting as it depends neither on the sensing cost $\alpha$ nor on the number of licensed channels $K$. Therefore, the SP may tune the service rate of the dedicated channel, $\mu_s$, and the arrival rate of SUs, $\lambda_s$, by

using some admission control for example, in order to minimize the gap between the NE and the global optimization's performance. In the following section, we present some numerical illustrations.

### 3.3.4 Numerical illustrations

This section presents the performance analysis of the proposed OSA mechanism. For this end, we have performed extensive numerical computations with different configurations of the system. Furthermore, two performance metrics are considered: the sensing cost and the capacity of the system (number of licensed channels). We fix the arrival rate for PUs (reps. SUs) at 0.6 (reps. 0.8), and we consider different service rates for the licensed channels ($\mu_p = 0.8$) and the dedicated channel ($\mu_s = 1.1$). Under these setting, the PoA is analytically evaluated to $PoA \geq 0.7524$ from Proposition 3.6.

We focus, first, on the case of one licensed channel, and we set the sensing cost to 0.1. Figure 3.2 illustrates the average total cost depending on the sensing probability of SUs. We observe that the average total cost is minimized when the SUs sense licensed channels with a probability $p = 1$, i.e. all SUs sense licensed channels. In fact, since the sensing cost is relatively low ($c_s = 0.1$), all SUs have incentive to sense licensed channels.

Secondly, we consider multiple licensed channels and we set $K$ to 10. As we have already assumed that the sensing cost is linear with the number of licensed channels, choosing to sense licensed channel become costly for SUs with the increase of the number of licensed channels ($c_s = 1$). We plot, in Figure 3.2, the average total cost, with $K = 10$ licensed channels, and we observe that SUs have less incentive to sense the licensed channels compared to the first scenario ($K = 1$). In fact, the average cost is minimal when SUs sense licensed channels with a probability of 0.427.

#### 3.3.4.1 Sensing cost

We evaluate, in the present section, the impact of the sensing cost parameter $\alpha$ on the performance of the proposed OSA mechanism, given a fixed number of licensed channels ($K = 10$). Mobile devices equipped with a CR have usually a limited battery, and have to be energy efficient. The main challenge of designing an energy-aware CR is to determine the appropriate OSA strategy, as SUs spend energy for sensing licensed channels. We plot, in Figure 3.4, the optimal probability of sensing for SUs $p^*$ and the sensing probability of SUs at the NE $p^E$. We remark that both probabilities are decreasing with the sensing cost $\alpha$. This result is intuitive, as increasing the sensing cost decreases the incentive of SUs to sense licensed channels. Furthermore, this observation

validates the analytical result obtained in Proposition 3.4. In fact, the optimal sensing probability $p^*$ (obtained from the global optimization of the centralized system) is always higher than $p^E$ (the sensing probability obtained at the NE).

It is straightforward that the non-cooperative behavior of SUs induces a worse performance compared to the centralized system. We focus on the gap of performance induced when migrating from centralized to decentralized networks. We illustrate the PoA, defined by Equation (3.12), in Figure 3.5. We observe that the minimum of the PoA equals 0.7559. Note that theoretically, the PoA is higher than 0.7524. Thus, the performances obtained by simulations are slightly better than the lower bound obtained analytically from Proposition 3.6. Given this result, we are able to design a decentralized OSA mechanism for energy-efficient SUs in self-adaptive CR networks, which is at worst 75% far from the optimal.

Note that the energy spent for sensing licensed channels depends not only on the cost of sensing $\alpha$, but also on the number of licensed channels $K$, as the sensing cost $c_s$ is assumed to be linear with $K$. We evaluate, in the following section, the impact of the capacity on the performance of the proposed OSA mechanism.

#### 3.3.4.2 Capacity

In this section, we are interested in the impact of the number of licensed channels on the proposed OSA mechanism. We fix the sensing cost $\alpha$ at 0.3, and we vary the number of licensed channel from 1 to 20. An interesting analysis of [78] shows that the average number of available licensed channels in TV white-bands is about 15. Note that under these settings, the blocking probability decreases with the number of licensed channels whereas the sensing cost increases.

Figure 3.3 depicts the impact of the number of licensed channels on both the optimal sensing probability and the sensing probability at the NE for SUs. We observe that both $p^*$ and $p^E$ are decreasing, and that $p^*$ is always higher than $p^E$. This result has already been proved analytically in Proposition 3.4. We plot, in Figure 3.6, the average total cost with the number of licensed channels. We remark that the average cost is minimal for $K = 2$. Note that increasing the capacity of the system increases the opportunities in the primary subsystem $S_1$, but also increases the sensing cost $c_s$.

Similarly to the sensing cost analysis, we measure the gap of performance between the global system and the decentralized system through the PoA. Figure 3.7 illustrates the PoA depending on the number of licensed channels $K$. The worst-case performance gap is 0.7619 obtained with 4 licensed channels. This result is slightly higher than the

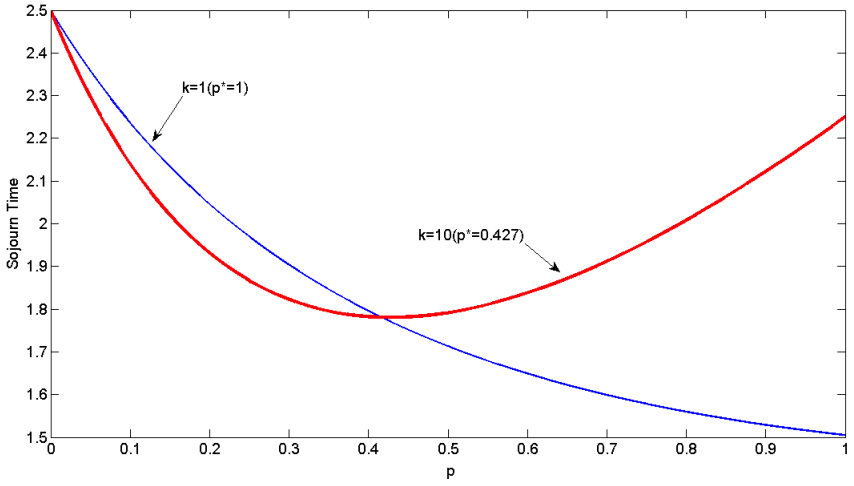


FIGURE 3.2: The average total cost function $U_S(p)$ for $\alpha = 0.1$, with one licensed channel, $K = 1$, and ten licensed channels, $K = 10$.

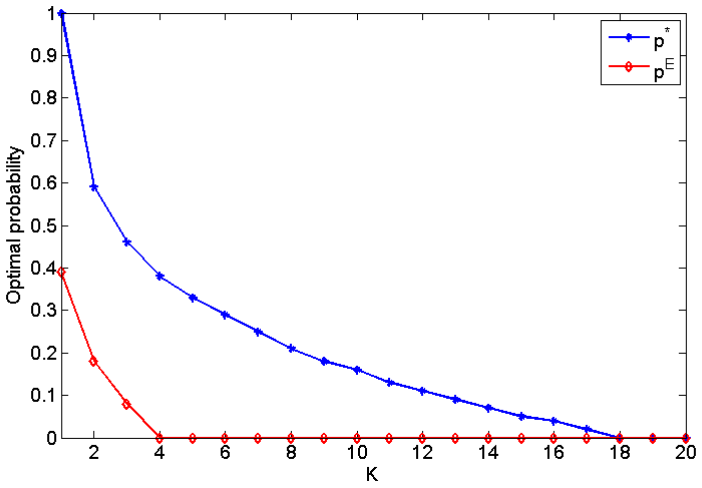


FIGURE 3.3: The probability of sensing depending on the number of licensed channels in both the centralized and the decentralized systems.

analytical result of the Proposition 3.6, which says that the lower bound of the PoA equals 0.7524.

### 3.3.5 Summary

In this section, we have defined an optimal OSA policy for SUs. Moreover, we have proposed a decentralized policy for self-interested SUs and we have evaluated the gap of performance between both approaches (global optimization and decentralized optimization) through the PoA metric. Nonetheless, we have taken the assumptions that PUs operate in a slotted model, and that they are perfectly synchronized with SUs. We

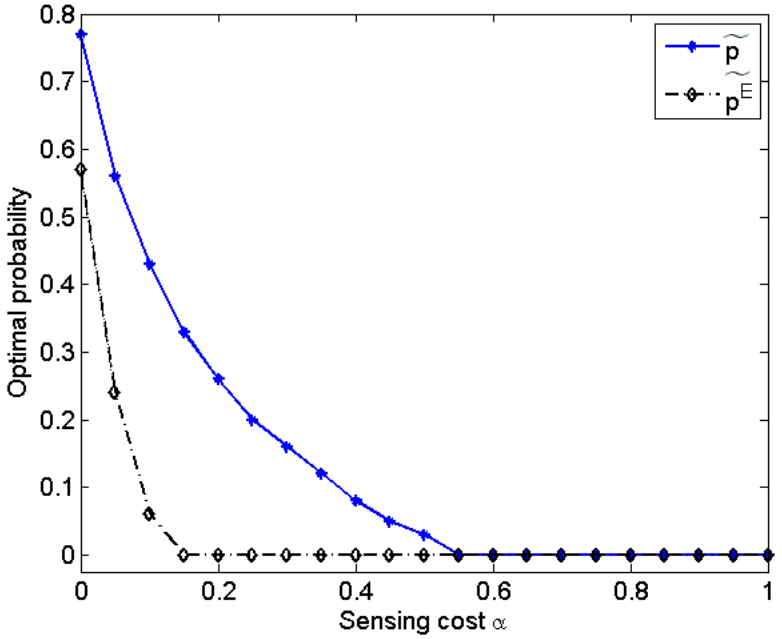


FIGURE 3.4: The optimal probability of sensing depending on the sensing cost $\alpha$.

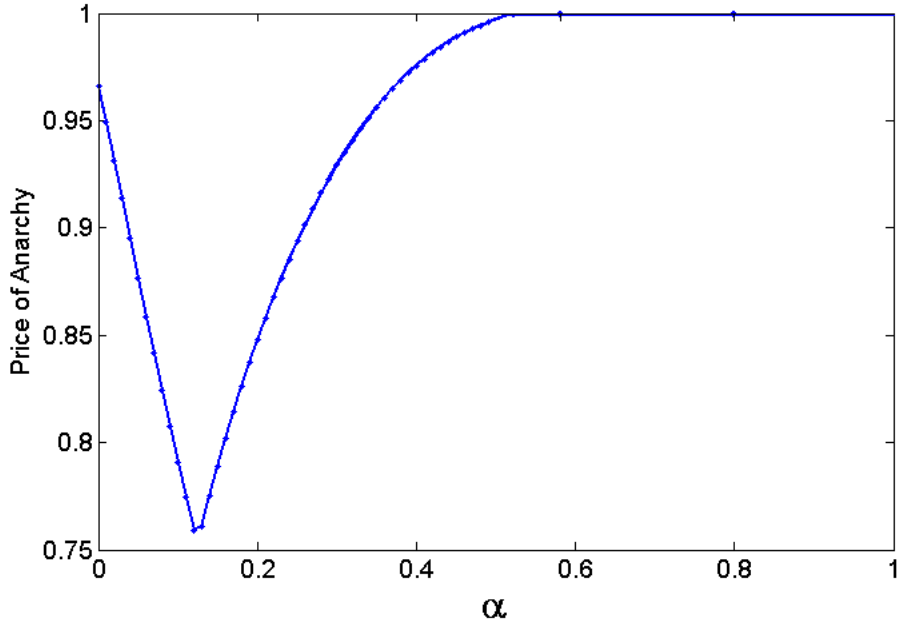


FIGURE 3.5: The price of anarchy depending on the sensing cost $\alpha$.

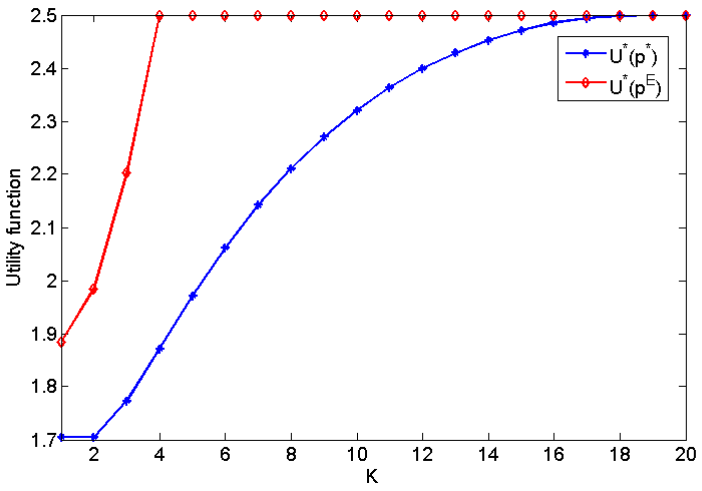


FIGURE 3.6: The average total cost depending on the number of licensed channels in both the centralized and the decentralized system for the slotted model.

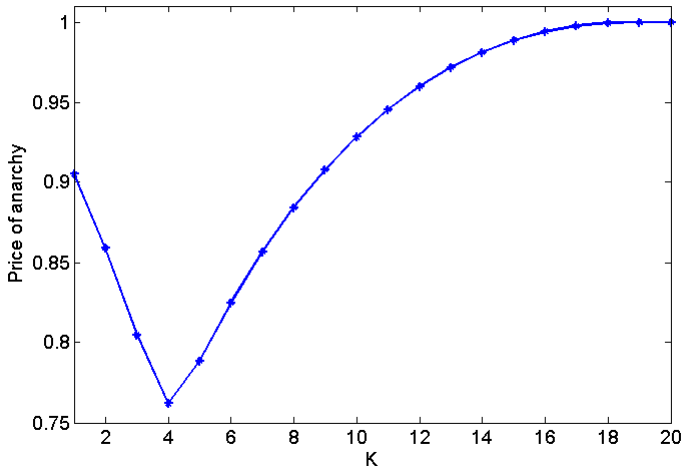


FIGURE 3.7: The price of anarchy depending on the number of licensed channels in the slotted model.

release theses assumptions in the next section by considering that the PUs evolve in a non-slotted regime, and that they may preempt a SU using licensed channels at their arrival. Releasing these assumptions significantly complicates the problem, as SUs have to face the reject form licensed channels by PUs, as well as the competition with each other.

## 3.4 The non-slotted model

In the present section, we relax some assumptions that were taken in order to simplify the study of the system. Indeed, we consider a more realistic model in which PUs evolve in a non-slotted mode, and have the highest priority to access licensed channels. Thus, if a PU does not find a free licensed channel, it rejects one SU (if there is one SU using licensed channels) and start transmission. We consider that SUs can detect that a PU is present and free immediately the channel. We further assume that if the SU is rejected, it gets no reward and is rejected from the system. When there are several SUs using licensed channels, a PU chooses randomly one SU to reject. Note that interruption from PUs is a key factor impacting the performance of SUs in CR networks. This assumption was also considered in [79]. We model, in the following section, the reject probability of SU in the primary subsystem.

### 3.4.1 Reject probability

We denote by $W_p(t)$ (resp. $W_s(t)$) the number of PUs (resp. SUs) using the licensed channels at the time slot $t$, where $W_p(t)+W_s(t) \leq K$. Specifically, the primary subsystem can be modeled using a bi-dimensional Markov process, $Z(t) = \{W_p(t), W_s(t)\}$. The probability that a SU will be rejected, when using a licensed channel, is denoted by $P_r(p, K)$. This probability depends on the proportion $p$ of SUs that senses licensed channels, and the number of licensed channels. Note that each SU that joins the system with a Poisson process observes the system in its stationary regime, according to the PASTA property (see Definition 2.6).

We denote by $P_0(n, m)$ the probability that a SU will be rejected, when it joins a licensed channel and the primary subsystem has already $n$ PUs and $m$ SUs. Note that we have necessary $n + m < K$, and the reject probability is expressed as follows:

$$P_r(p, K) = \sum_{n,m/n+m=0}^{n+m=K-1} P_0(n, m)\pi(n, m), \tag{3.14}$$

where $\pi(n, m)$ is the stationary probability of the Markov process $Z(t)$, described in Figure 3.8. The stationary probabilities $\pi(n, m)$ can be computed using standard tools of Markov theory. Let us focus on the reject probabilities $P_0(n, m)$, it is possible to express the relation between probabilities $P_0(n, m)$ as a linear system. Note that for all states $(W_p(t), W_s(t)) = (n, m)$, such that $n + m = K - 1$, $P_0(n, m)$ is expressed as

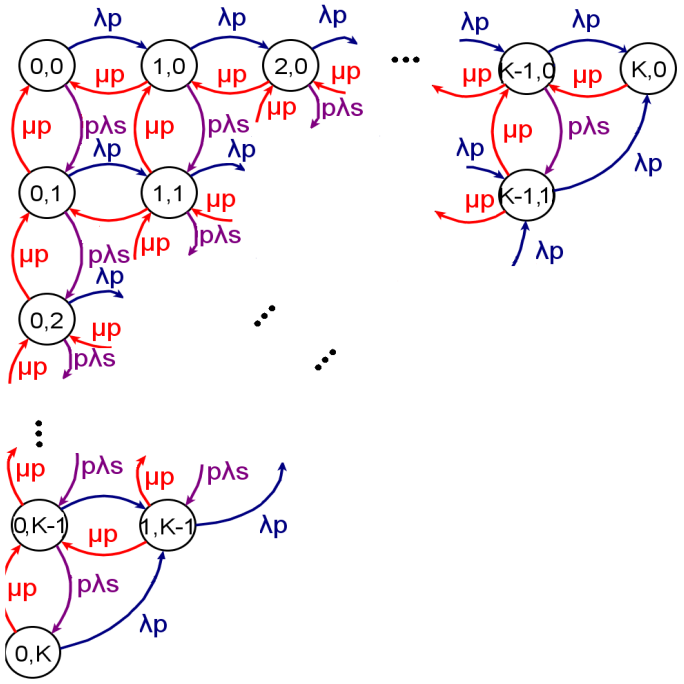


FIGURE 3.8: The bi-dimensional Markov chain of $Z(t)$.

follows:

$$P_0(n,m) = \begin{cases} \frac{1}{K}\frac{\lambda_p}{\lambda_p+\mu_p} + \frac{K-1}{K}\frac{\lambda_p}{\lambda_p+\mu_p}P_0(1,K-2) + \frac{\mu_p}{\lambda_p+\mu_p}P_0(0,K-2) & \text{if} \quad n=0, \\ \frac{\lambda_p}{\lambda_p+2\mu_p} + \frac{\mu_p}{\lambda_p+2\mu_p}P_0(K-2,0) & \text{if} \quad m=0, \\ \frac{1}{m+1}\frac{\lambda_p}{\lambda_p+2\mu_p} + \frac{m}{m+1}\frac{\lambda_p}{\lambda_p+2\mu_p}P_0(n+1,m-1) & \\ \quad + \frac{\mu_p}{\lambda_p+2\mu_p}(P_0(n-1,m)+P_0(n,m-1)) & \text{otherwise.} \end{cases}$$

Otherwise, for $n+m<K-1$, the probability $P_0(n,m)$ is expressed as follows:

$$P_0(n,m) = \begin{cases} \frac{p\lambda_s}{p\lambda_s+\lambda_p+\mu_p}P_0(n,m+1) + \frac{\lambda_p}{p\lambda_s+\lambda_p+\mu_p}P_0(n+1,m) & \\ \quad + \frac{\mu_p}{p\lambda_s+\lambda_p+\mu_p}P_0(n,m-1) & \text{if} \quad n=0, \\ \frac{p\lambda_s}{p\lambda_s+\lambda_p+2\mu_p}P_0(n,m+1) + \frac{\lambda_p}{p\lambda_s+\lambda_p+2\mu_p}P_0(n+1,m) & \\ \quad + \frac{\mu_p}{p\lambda_s+\lambda_p+2\mu_p}P_0(n-1,m) & \text{if} \quad m=0, \\ \frac{p\lambda_s}{p\lambda_s+\lambda_p+2\mu_p}P_0(n,m+1) + \frac{\lambda_p}{p\lambda_s+\lambda_p+2\mu_p}P_0(n+1,m) & \\ \quad + \frac{\mu_p}{p\lambda_s+\lambda_p+2\mu_p}(P_0(n-1,m)+P_0(n,m-1)) & \text{otherwise.} \end{cases}$$

We assume that the reject probability $P_r(p,k)$ is increasing with the sensing probability $p$. This assumption is somehow realistic. Indeed, the greater is the number of SUs that choose to sense, the higher is the probability to be rejected by PUs. In the following section, we study the impact of the reject probability on the average cost function, and we determine the optimal OSA policies for SUs.

### 3.4.2 Average total cost

The average sojourn time $T^r_{S_1}$ for a SU that chooses to join the dedicated channel without sensing licensed channels is given by:

$$T^r_{S_1}(p,K) = \frac{1}{\mu_s - \lambda_s(1-p)}. \tag{3.15}$$

Moreover, the average sojourn time of a SU that chooses to sense licensed channels is defined by:

$$T^r_{S_2}(p,K) = \frac{(1-\Pi(p,K))(1-P_r(p,K))}{\mu_p}. \tag{3.16}$$

Therefore, the average sojourn time of a SU in the non-slotted model is expressed as follows:

$$T^r_S(p,K) = \frac{1-p}{\mu_s - \lambda_s(1-p)} + \frac{p(1-\Pi(p,K))(1-P_r(p,K))}{\mu_p}. \tag{3.17}$$

The average cost function is expressed as follows:

$$U^r_S(p,K) = \frac{1-p}{\mu_s - \lambda_s(1-p)} + \frac{p(1-\Pi(p,K))(1-P_r(p,K))}{\mu_p} + \alpha p K.$$

For notation convenience, we define $Y(p,K) = p(1-\Pi(p,K))(1-P_r(p,K))$. By substituting $Y(p,K)$ in the expression of the average cost function, we obtain the following expression:

$$U^r_S(p,K) = \frac{1-p}{\mu_s - \lambda_s(1-p)} + \frac{Y(p,K)}{\mu_p} + \alpha p K.$$

The first intuition one can make is that releasing the assumption that PUs evolve in a slotted model induces a loss of performance. Let us denote by $p^*_r$ the optimal sensing probability of a SU in the non-slotted model.

**Proposition 3.7.** *For all values of $\alpha$ and $K$, the average cost function $U^r_S(p,K)$ is minimized when the sensing probability is equal to:*

$$p = min(1, max(p^r_0, 0)) := p^*_r,$$

*where $p^r_0$ is the solution of the following equation:*

$$\frac{\partial Y}{\partial p}(p,K) = -\alpha K \mu_p + \frac{\mu_p \mu_s}{(\mu_s - \lambda_s(1-p))^2}. \tag{3.18}$$

*Proof.* The proof of this proposition is analogous to the proof of Proposition 3.1 by replacing $X(p,K)$ by $Y(p,K)$. □

Furthermore, the following proposition gives us a relation between the average cost obtained with the slotted system and the average cost obtained with the non-slotted model.

**Proposition 3.8.** *For all values of $\alpha$ and $K$, the optimal value of the average cost function is higher in the non-slotted model than in the slotted one:*

$$U_S(p^*, K) \leq U_S^r(p_r^*, K).$$

*Proof.* Suppose, first, that $\mu_s - \alpha K(\mu_s - \lambda_s(1-p))^2 \leq 0$. Then, it follows from Proposition 3.7 that $p^* = 0$. Therefore, the average cost function is expressed as follows:

$$U_S(p^*, K) = \frac{1}{\mu_s - \lambda_s}.$$

Let us derive the average cost function with respect to the reject probability. After some algebra, we obtain the following expression of the derivative of the average cost function with respect to the reject probability:

$$\frac{\partial U_S^r}{\partial P_r}(P_r) = -\frac{p(1-\Pi(p,K))}{\mu_p} \leq 0.$$

We remark that $U_S^r(P_r)$ is decreasing with $P_r$. Thus, we have the following lower bound of the average cost function:

$$U_S^r(P_r) \geq U_S^r(1) = \frac{1}{\mu_s - \lambda_s} + \alpha p_r^* K \geq \frac{1}{\mu_s - \lambda_s},$$

which leads to:

$$U_S(p^*, K) \leq U_S^r(p_r^*, K).$$

Second, suppose that $\mu_s - \alpha K(\mu_s - \lambda_s(1-p))^2 > 0$. Therefore, we prove analogously that $U_S^r(P_r)$ is increasing with $P_r$, and we obtain that $U_S^r(P_r) \geq U_S^r(0)$.

Finally, the average cost function in the non-slotted model is higher than the average cost function in the slotted one, i.e. $U_S(p^*, K) \leq U_S^r(p_r^*, K)$. □

This result is somehow intuitive as the reject of a SU introduces a lack of performance to the system. We focus, in the next section, on the study of the non-slotted self-adaptive CR network model.

### 3.4.3 Individual optimization

We consider a distributed system in which each SU decides individually whether to sense or not licensed channels. In fact, each SU decides on its probability $p$ of sensing licensed channels. Note that a SU aims to minimize its average cost function $U_r(p, p', K)$, which depends on its probability $p$ and the probability $p'$ of other SUs. Thus, the average cost function is expressed as follows:

$$U_r(p, p', K) = (1-p)T^r_{S_1}(p', K) + pT^r_{S_2}(p', K) + \alpha pK. \tag{3.19}$$

We prove, in the following proposition, that the non-cooperative OSA for SUs has a NE.

**Proposition 3.9.** *For all values of $\alpha$ and $K$, the NE strategy for the OSA problem exists. Moreover the sensing probability at the NE is given by:*

- *if* $\frac{1}{\mu_s-\lambda_s} > \alpha K + \frac{(1-\Pi(0,k))(1-P_r(0,K))}{\mu_p}$;
  - *if* $\frac{1}{\mu_s} < \alpha K + \frac{(1-\Pi(1,k))(1-P_r(1,K))}{\mu_p}$ *then* $p_r^E = \{0, p'_r, 1\}$.
  - *else* $p_r^E = 0$;
- *else*
  - *if* $\frac{1}{\mu_s} > \alpha K + \frac{(1-\Pi(1,k))(1-P_r(1,K))}{\mu_p}$ *then* $p_r^E = p'_r$;
  - *else* $p_r^E = 1$.

*where $p'_r$ is the solution of the following equation:*

$$\frac{1}{\mu_s - \lambda_s(1-p)} = \alpha K + \frac{(1-\Pi(p,K))(1-P_r(p,K))}{\mu_p}. \tag{3.20}$$

*Proof.* The proof of this proposition is analogous to the proof of Proposition 3.3 by replacing $X(p, K)$ by $Y(p, K)$. □

For notation convenience, we denote for all $p$ and $K$, $U_r(p, p, K)$ by $U^r_S(p, K)$. Furthermore, the following proposition gives us a higher bound of the average total cost at the NE.

**Proposition 3.10.** *For all values of $\alpha$ and $K$, we have the following higher bound of the average cost function at the NE:*

$$U^r_S(p^E_r, K) \leq \frac{1}{\mu_s - \lambda_s}.$$

*Proof.* The proof of this proposition is analogous to the proof of Proposition 3.5 by replacing $X(p,K)$ by $Y(p,K)$. □

Given the existence of the NE for the proposed OSA mechanism in the non-slotted model, we study the gap of performance between the average cost at the NE and the average cost of the centralized system.

### 3.4.4 Price of anarchy

The PoA models the lack of performance between the utility at the NE and the optimal utility, and is defined by the following ratio:

$$PoA_r(\alpha,K) = \frac{U_s^r(p_r^*,K)}{\max_{p\in p_r^E} U_s^r(p,K)} \leq 1. \tag{3.21}$$

Let us focus on the expression of PoA. Similarly to the slotted model, our aim is to determine a lower value of the PoA or to bound it, in order to define the worst-possible lack of performance of the decentralized system. The following proposition gives us a lower bound of the price of anarchy, called $\underline{PoA_r}$.

**Proposition 3.11.** *For all values of $\alpha$ and $K$, we have the following lower bound of the PoA:*

$$PoA_r(\alpha,K) \geq \frac{2(\lambda_s - \mu_s + \sqrt{\mu_s(\mu_s-\lambda_s)})}{\lambda_s} := \underline{PoA_r}.$$

*Proof.* The proof of this proposition is analogous to the proof of Proposition 3.6 by replacing $X(p,K)$ by $Y(p,K)$. □

This closed-form lower bound of the PoA is interesting, as it depends neither on the sensing cost $\alpha$, nor on the number of licensed channel $k$. Thus, the SP has only to tune $\mu_s$ and $\lambda_s$ in order to maximize the performance of the decentralized system.

In the following section, we present some numerical illustrations that validate our theoretical findings.

### 3.4.5 Numerical illustrations

This section presents the performance analysis of the proposed OSA mechanism. For this end, we have performed extensive Matlab simulations with different configurations of the system. Furthermore, two performance metrics are considered: the sensing cost and the capacity of the system. We consider the same values of the system model parameters defined in Section 3.3.4. Moreover, we assume that PUs may preempt SUs

in service. Firstly, we focus on the sensing cost $\alpha$. Thereafter, we study the impact of the capacity (number of licensed channels) on the OSA mechanism.

#### 3.4.5.1 Sensing cost

We evaluate, in this section, the impact of the sensing cost $\alpha$ on the performance of the proposed OSA mechanism. Figure 3.9 illustrates the average cost function in both the slotted PUs transmissions and the non-slotted model. We observe that the average cost of SUs is always higher in the non-slotted model than in the slotted one, which validates the results of Proposition 3.8.

We observe, in Figure 3.10, that the optimal probability of sensing licensed channels is decreasing with $\alpha$ in both models. However, we remark that the optimal probability of sensing in the non-slotted model $p_r^*$ is more sensitive to the sensing cost $\alpha$ than the optimal probability of sensing in the slotted model $p^*$. In fact, in the non-slotted model, the reject probability decreases the benefit of sensing in term of utility.

Let us focus on the lack of performance induced by the non-cooperative behavior of SUs in the decentralized model. We obtain from Proposition 3.11 a lower bound of the price of anarchy $\underline{PoA_r} = 75.24\%$. This result is lower than the minimum value of the PoA obtained from Figure 3.11, which is 0.8289.

The number of licensed channels has a major leverage on the behavior of SUs and impacts not only the average sojourn time, but also the energy consumption, as the sensing cost grow linearly with the number of licensed channels. We depict, in the next section, the impact of the capacity on the performance of the proposed OSA policy.

#### 3.4.5.2 Capacity

In the present section, we are interested in the impact of the number of licensed channels on the performance of the proposed OSA mechanism for SUs. We set the sensing cost $\alpha$ to 0.3 and we vary the number of licensed channel from 1 to 20. Note that under these settings, the blocking probability decreases with the number of licensed channels, whereas the sensing cost increases.

Firstly, we observe, in Figure 3.12, that both the optimal sensing probability $p_r^*$ and the sensing probability at the NE $p_r^E$ are decreasing with number of licensed channel $K$. Moreover, we remark that the sensing probability at the NE is lower or equal than the optimal sensing probability. In fact, the non-slotted system is more sensitive to the number of licensed channels than the slotted one. Second, we obtain from Figure 3.13

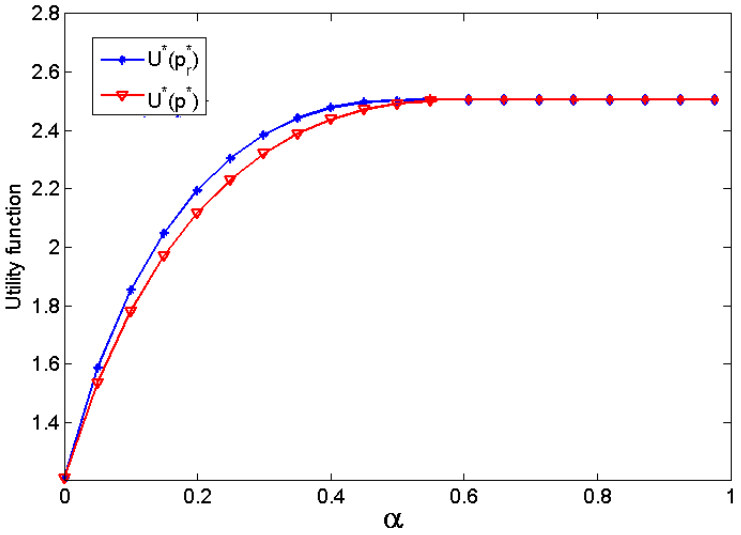


FIGURE 3.9: The global optimum depending on the sensing cost $\alpha$ in both the slotted and the non-slotted models.

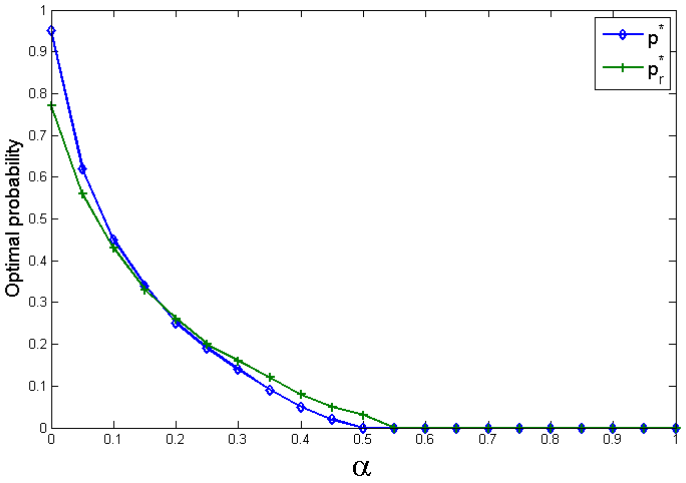


FIGURE 3.10: The optimal sensing probability depending on the sensing cost $\alpha$.

that the non-slotted model induces a higher average cost for SUs compared to the slotted model. Finally, we conclude with the analysis of the price of anarchy depending on the number of licensed channels $K$. In Figure 3.14, we observe that the minimal value of the price of anarchy is 0.8672, which is not so far from the lower bound given by Proposition 3.11, which is 75.24%.

Both the sensing cost and the capacity of the system are important factors in the performance of CR users. The SP may tune the system parameters in order to optimize the QoS for its SUs without the need for a centralized controller.

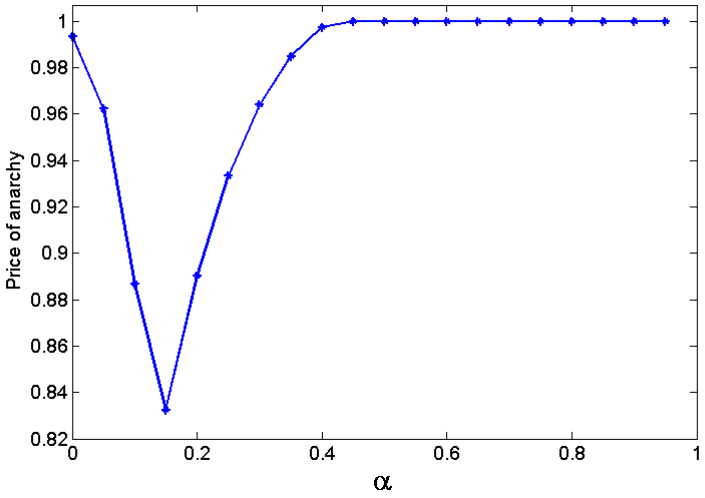


FIGURE 3.11: The price of anarchy depending on $\alpha$.

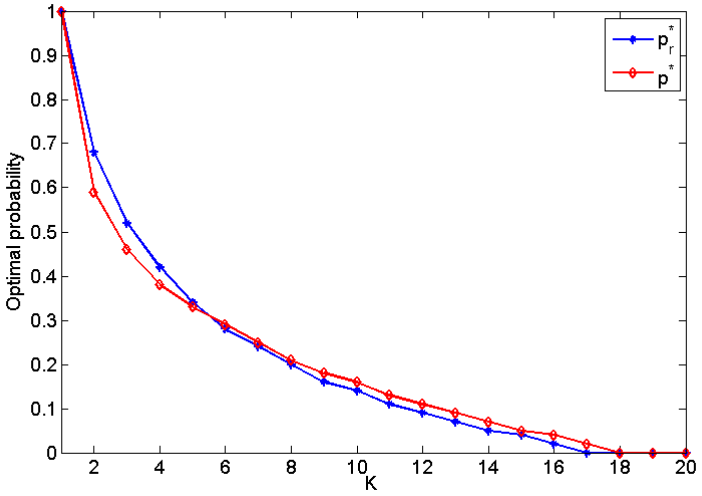


FIGURE 3.12: The probability of sensing depending on the number of licensed channels in non-slotted model.

### 3.4.6 Summary

As like as the slotted model, we have studied, in this section, the non-slotted OSA in both the centralized and the decentralized manners. We have proved the existence of a NE strategy, and we have evaluated the gap of performance in the decentralized system through the POA.

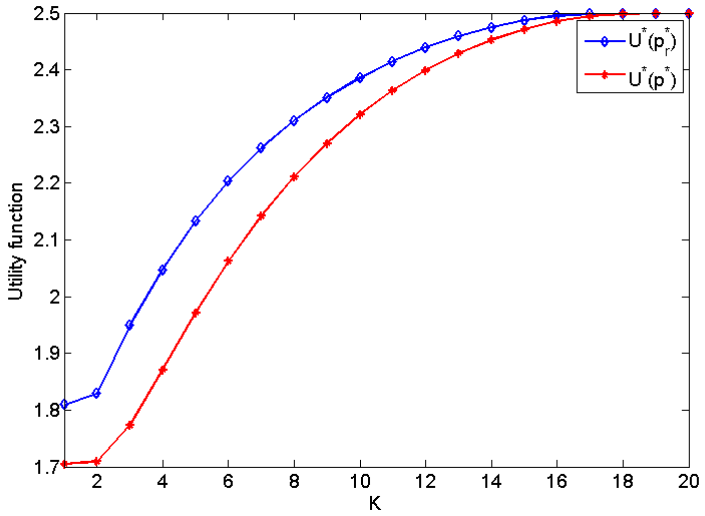


FIGURE 3.13: The average cost function with the number of licensed channels in both the slotted and the non-slotted models.

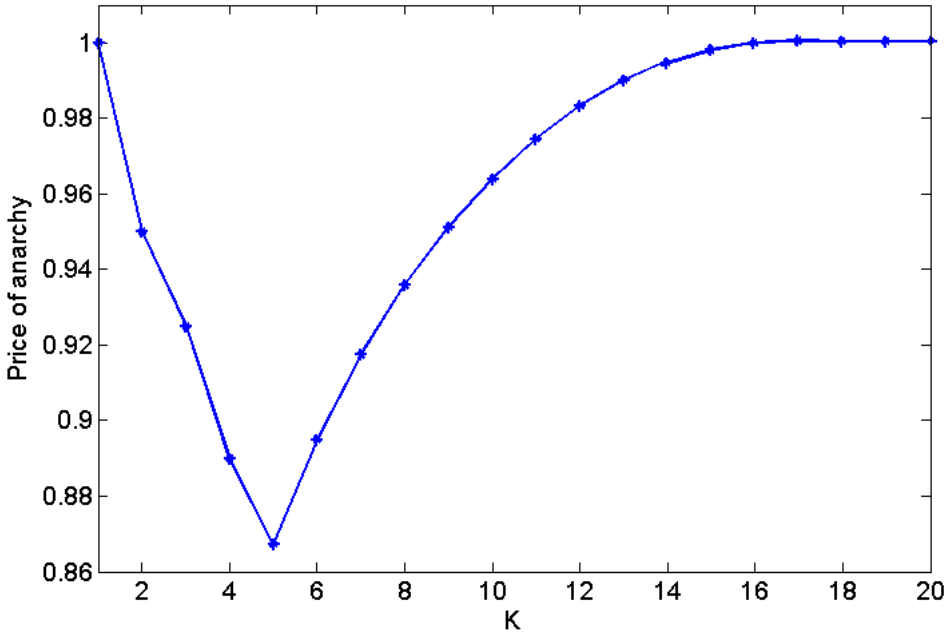


FIGURE 3.14: The price of anarchy with the number of licensed channels $K$ in the non-slotted model.

## 3.5 Conclusion

In this chapter, we have studied the performance of OSA in CR networks. We have considered both the slotted and the non-slotted models. We have considered the global optimization of the centralized system, and we have determined the optimal sensing probability. Furthermore, we have considered the the individual optimization in a decentralized manner, and we have proved the existence of a NE equilibrium between SUs. We have studied the performance of these approaches and we have evaluated the gap of performance between them using the well-studied metric: the PoA. Simulation results have validated our theoretical findings.

In the next chapter, we study the OSA for CR under energy and QoS constraints. Specifically, we formulate the model using a POMDP framework, and we present an optimal threshold-based OSA policy.

# Chapter 4

# Energy-efficient Opportunistic Spectrum Access in Cognitive Radio Networks

Contents



## 4.1 Introduction

The traditional spectrum management is based on agreements between the SP and PUs. CR is considered as the key technology that enables unlicensed users to access the licensed spectrum. Furthermore, the new spectrum-licensing paradigm, initiated by the FCC in 2008 [8], has promoted the idea of using the CR technology to face the spectrum scarcity problem. It allows unlicensed users to access the spectrum as long as they do not harm licensed users' transmissions.

Although the use of licensed bands by CR users is widely recognized, it is not well understood which applications are suitable for CR users, and what type of traffic a CR user may support. In fact, if CR users support multimedia applications, such as

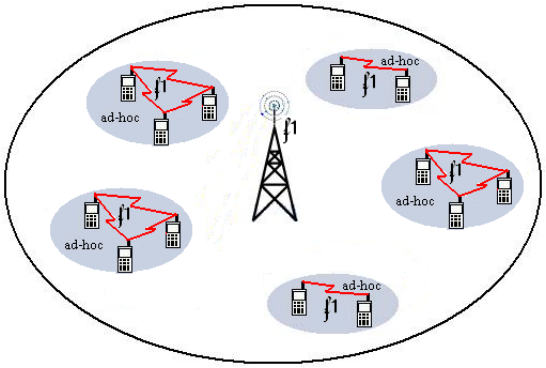


FIGURE 4.1: First use case: Using CR in ad-hoc communication. If the licensed frequency $f1$ is not used by PUs, SUs can communicate in ad-hoc mode using $f1$.

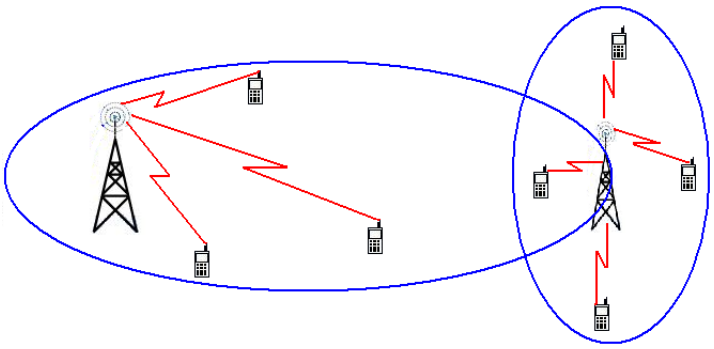

FIGURE 4.2: Second use case: using CR for BS's transmissions. If the licensed frequency $f1$ is not used by PUs, the BS serves its users using $f1$.

video streaming, VoIP or online gaming, they must be able to guarantee some QoS requirements. These motivations are behind the problem considered in this chapter.

The model that we are studying in this chapter is suited for several use-cases of the CR paradigm in wireless networks. Firstly, this model allows ad-hoc connections to use spectrum holes (frequencies that are not utilized by PUs), as illustrated in Figure 4.1. Second, we may consider that SUs are CR base stations, which are able to sense the activity of a primary base station, and take advantage of spectrum holes for transmitting on the downlink (see Figure 4.2).

Many works focused on the study of optimal OSA in CR networks (see [80], [81] and [82]). In [83], the authors focused on the OSA taking into account the energy consumption. They formulated their problem as a POMDP and derived some properties of the optimal OSA policy. Their control parameter is the duration of sensing used by a SU at each time slot in order to determine the PU's activity. They provided heuristic control policies by using grid-based approximations, which have low complexity but give suboptimal control policies. Authors of [53] incorporated the energy constraint in the design of the optimal OSA policy. They formulated the problem also using a POMDP with a

finite horizon criterion. They established a threshold structure of the optimal policy for the single channel model without providing analytical expression of the threshold. The main difference between these works and ours is that we consider not only the energy consumption, but also the transmission delay. Moreover, we consider a POMDP with an average reward criterion. Authors of [84] analyzed the latency of DSA in CR networks by considering a dedicated control with embedded control channel. In [85], authors considered an adaptive modulation scheme in order to guarantee a delay for SUs. The difference between their work and our's is that they considered a dynamic spectrum sharing and we are considering an OSA context.

It is noteworthy that the impact of the energy consumption or the capacity of CR users to support additional QoS requirements such as the expected delay, to the best of our knowledge, has been somehow ignored in the literature, partially due to the difficulties in analyzing it. In fact, it is very important for today's wireless networks to garantee a certain level of QoS. As SUs are not licensed to use the spectrum, the transmission delay of their packets depends not only on the PUs' activity but also on the competition with each other.

Our main contribution is to consider, in this CR setting, an optimal OSA mechanism that takes into account energy consumption and transmission delay. Note that, taking into account the delay as well as the energy consumption significantly complicates the optimization problem. For instance, without considering the delay constraint, the SU achieves the best tradeoff between trying to access licensed channel and sleeping to conserve energy. However, the design of energy-QoS tradeoff lies among several conflicting objectives: gaining immediate access, gaining spectrum occupancy information, conserving energy, and minimizing packet delays. The novelty of this work is to study an energy-QoS tradeoff OSA mechanism for SUs in a CR network. The major contributions of this chapter are:

- The problem is formulated as an infinite horizon POMDP with average criterion. Usually, OSA mechanisms for CR networks were modeled using POMDPs with expected total discounted reward (see [80], [83] and [27] for some examples). However, as decisions are taken frequently by SUs (every time slot) the discount rate is very close to 1. Thus, the average expected reward is more suited to model OSA mechanisms [86].

- In order to gain insights into the energy-delay constrained OSA, we derive structural properties of the value function. We are able to show that the value function is increasing with the belief and decreasing with packet delays. These structural results not only give us insights about the optimal OSA policy, but also reduce

the computational complexity when seeking for the optimal policies. In fact, the value function can be approximated by simple functions (see [87]).

- We show that the SUs maximize their average rewards by adopting a simple threshold policy, and we derive closed-form expressions of these thresholds.
- Since SUs may use a dedicated channel for their packets, the optimal threshold policy guarantees a bounded delay.
- We propose some learning algorithm to estimate the RF environment on-the-fly.

The organization of this chapter is as follows. In the next section, we describe the primary and the secondary user models. Section 4.3 presents our Markov decision process framework. In Section 4.4, we study the existence of an optimal threshold policy for our opportunistic spectrum access with an energy-QoS tradeoff. We propose two learning based protocols for the estimation of state transition rates in Section 4.5. Before concluding the chapter, we present, in Section 4.6, some numeric illustrations.

## 4.2 Model

We consider a wireless network where $N$ independent channels are licensed to PUs. The state of each channel $n \in \{1, \ldots, N\}$ is modeled by a time-homogeneous discrete Markov process $s_n(t)$. The state space is $\{0, 1\}$, where $s_n(t) = 0$ means that the channel $n$ is free for SUs' access, and $s_n(t) = 1$ means that the channel $n$ is occupied by PUs. The following matrix gives the transition probabilities of the channel $n$:

$$P_n = \begin{pmatrix} \alpha_n & 1-\alpha_n \\ \beta_n & 1-\beta_n \end{pmatrix}.$$

In fact, SUs observe a "good" channel (ON) if PUs are not using the licensed channel. On the other hand, the presence of PUs in the licensed channel results in a "bad" channel (OFF) for SUs. Therefore, the licensed channels can be modeled by the ON/OFF Gilbert-Elliot model [88], [89]. The transition rates evolve as illustrated in Figure 5.1. Note that this model was used in several works in the OSA area (see [80], [27], [83] and [53] for some examples).

The global state of the system, composed of the $N$ channels, is denoted by the vector $\mathbf{s}(t) = [s_1(t), ..., s_N(t)]$, and the global state space is $\mathcal{S} = \{0, 1\}^N$. The transition probabilities can be determined by statistics of the PUs' traffic, and are assumed to be known by SUs. We present, in Section 4.5, some methods allowing the SU to estimate these transition probabilities on-the-fly.

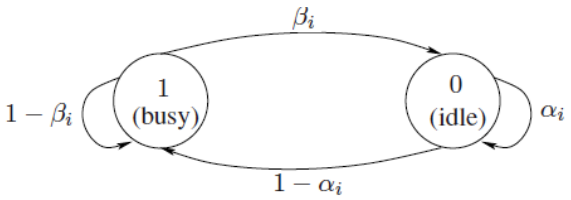


FIGURE 4.3: The channel transition probabilities for channel $i$.

We consider that all the $N$ licensed channels are open for SUs' transmissions when PUs are not using them. The aim of SUs is to find licensed channels that are not used by PUs during a given time slot. Note that looking for opportunities in licensed channels may induce not only a large packet delay, but also higher energy consumption, spent for sensing and transmissions over licensed channels. This may be caused by high traffic of PUs or collisions between SUs. For this end, we consider an OSA that takes into account packet delay, throughput and energy consumption. In order to introduce some QoS guarantee for SUs, we assume that, at any time slot, SUs have access to the network through another technology referred to as dedicated channel. This assumption ensures a higher bound of packet delays, while benefiting from licensed spectrum holes. Indeed, the aim of SUs is to find a tradeoff between the following conflicting objectives: transmitting with a guaranteed delay, but with higher cost using the dedicated channel, or transmitting with a lower cost using the licensed channels, but without delay guarantee.

The objective of SUs is to minimize the transmission cost accounting for energy consumption and transmission delay, i.e. a QoS guarantee with the lowest possible cost. In order to achieve such goal, a SU has to choose at each time slot one of the following actions:

- to be inactive during the time slot in order to save energy,
- to sense a licensed channel and to transmit if the channel is available during the time slot, else to wait for next time slot,
- to sense a licensed channel and to transmit if the channel is available during the time slot, else to use the dedicated channel.

Figure 4.4 illustrates the action diagram for SUs. Our important contribution is to consider the average packet delay in the optimal decision. Moreover, we consider that sensing licensed channels has a cost for SUs, which models the energy spent when sensing licensed channels. Given these constraints, we seek for an optimal OSA policy for SUs in CR networks. In the remainder of this chapter, we focus on the model of one SU accessing opportunistically licensed channels. The multi-user context will be studied in the following chapter.

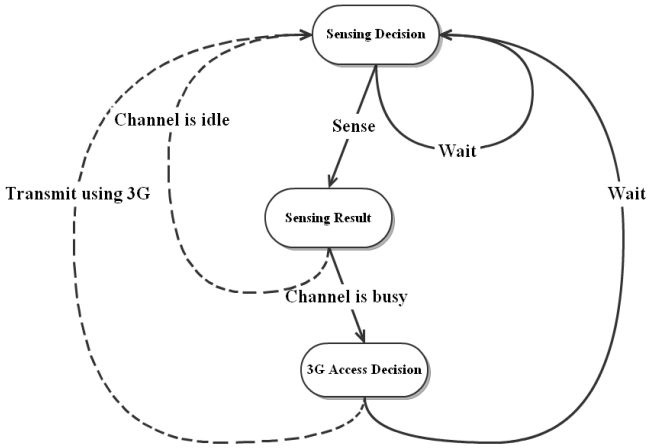


FIGURE 4.4: The action diagram for SUs. There are mainly two decision steps: the sensing decision and the access decision.

## 4.3 POMDP framework

The global system state $\mathbf{s}(t)$ cannot be directly observed by a SU. To overcome this difficulty, the SU infers the global state of the system based on observations that can be summarized in a belief vector $\omega(t) = \{\omega_1(t), ..., \omega_{2^N}(t)\}$, where $\omega_j(t)$ is the conditional probability (given observations and decisions history ) that the system state $\mathbf{s}(t) = j$, at the time slot $t$. Since the $N$ channels are independent, we may consider the following simpler belief vector:

$$\lambda(t) = [\lambda_1(t), .., \lambda_N(t)],$$

where $\lambda_i(t)$ is the conditional probability that the channel $i$ is available at the time slot $t$. This approximation was used in several analysis such as [90] and [91]. Hence, we study the OSA for SUs in CR networks as a POMDP problem. Our OSA mechanism can be formulated using a POMDP framework described as follows:

**State** We define the state of the system as a composition of belief and delay $(\lambda(t), l(t))$. The delay of a packet held by a SU is denoted by $l(t)$. When the SU receives a new packet, its delay equals one, and increases by one every time slot, except when the SU transmits the packet. We assume that the SU does not accept a new packet until it transmits the held one. We take this assumption in order to evaluate the impact of the OSA mechanism on the delay of the packets. As part of my future work, I will analyze the effect of traffic characteristics such as throughput and traffic model on the performance of the OSA mechanism. Note that the SU does not have accurate information about the first part of the system state, i.e. the belief vector $\lambda(t)$, but has a perfect knowledge about the packet delay.

**Action** The SU makes a two-level decision. It chooses, first, whether to sense the licensed channels or not. Then, it decides about the transmission over the licensed channels or using the dedicated one. It is straightforward that the SU transmits over the licensed channels if idle. Therefore, the second step decision is taken when there are no opportunities in licensed bands. In fact, when the licensed channel was sensed as occupied, the SU has two options:

- to wait for the next slot;
- to transmit using the dedicated channel.

Without loss of generality, we assume that both decisions are made at the beginning of the time slot. For each time slot $t$ and each state $(\lambda(t), l(t))$, we consider that the three possible actions for SUs are:

$$a(t) = \begin{cases} 0: & \text{to be inactive;} \\ 1: & \text{to sense and to transmit only if the channel is available during the slot;} \\ 2: & \text{to sense and to transmit if the channel is available during the time slot,} \\ & \text{else to transmit through the dedicated channel.} \end{cases}$$

**Observation and belief** When the SU decides to sense (i.e. to take action $a(t) \in \{1, 2\}$), one channel $n^*(t)$ is determined and the SU observes the channel occupancy state $s_{n^*(t)}(t) \in \{0, 1\}$. Let $\theta(t)$ be the observation outcome at the time slot $t$, where $\theta(t) = 0$ if the channel is sensed as idle, and $\theta(t) = 1$ otherwise. The SU takes into account the history of observations and actions by updating the belief vector based on observation outcomes. For each channel $n$, the conditional probability, $\lambda_n(t+1) := Pr(s_n(t+1) = 0 | a(t), \theta(t))$, is updated as follows:

$$\lambda_n(t+1) = \begin{cases} \beta_n + (\alpha_n - \beta_n)\lambda_n(t) & \text{if } a(t) = 0 \text{ or } n \neq n^*(t), \\ \alpha_n & \text{if } a(t) \neq 0, \text{or } \theta(t) = 0 \text{ and } n = n^*(t), \\ \beta_n & \text{if } a(t) \neq 0, \text{or } \theta(t) = 1 \text{ and } n = n^*(t). \end{cases} \quad (4.1)$$

The belief update function depends mainly on licensed channels' transition rates $\alpha$ and $\beta$. Indeed, these statistics may not be available for SUs. Specifically, we propose, in Section 4.5, some learning methods that allow the SUs to estimate the RF environment on-the-fly. Note that we can extend easily our model to sense not only one licensed channel, but also a subset of the licensed channels.

**Channel choice policy** At a given time slot $t$, the SU chooses a licensed channel $n^*(t) \in N$ to sense based on its belief vector $\lambda(t)$. There exists several channel choice policies in the literature like total sensing (see [66] and [59]), opportunistic sensing (see

[80] and [53]), randomized sensing (see [69] and [70]), and periodic sensing (see [67] and [68]). An example of opportunistic and greedy channel choice policy is to sense the channel that has the highest probability to be idle, i.e. $n^*(t) := \arg\max_n(\lambda_n(t))$.

**Policies** We define a sensing and access policy $\mu$ as a vector $[\mu_1, \mu_2, \ldots]$, where $\mu_t$ is a mapping from a state $(\lambda(t), l(t))$ to an action $a(t)$ at the time slot $t$. We denote by $\Gamma$ the set of all possible policies.

**Reward and costs** The SU tries to maximize its revenue by increasing the reward (obtained from successful transmissions) and decreasing the costs (spent for sensing and transmissions). Note that the cost of transmission over the dedicated channel is higher than the cost paid for transmission using the licensed channels. The different cost and rewards for SUs are denoted by:

- Reward: Let $\Phi$ be the reward representing the number of delivered bits when the SU transmits its packet.
- Costs: Let $c_s$ be the energy cost function for sensing a licensed channel, measured as monetary units. This function depends on the action $a(t)$ taken by the SU as follows:

$$c_s(a(t)) = \begin{cases} c_s, & \text{if } a(t) > 0, \\ 0, & \text{if } a(t) = 0. \end{cases}$$

  The PU and the SP (for the dedicated access), charge a price for each successfully transmitted packet. Those prices are respectively $P_p$ for a transmission over a licensed channel and $P_{3G}$ for a transmission over the dedicated channel. Indeed, $P_{3G}$ is higher than $P_p$. Therefore, when the SU transmits successfully a packet, it obtains the reward $z_t(a(t), \theta(t))$, which depends on the action $a(t)$ and the observation $\theta(t)$, and is expressed as follows:

$$z_t(a(t), \theta(t)) = \begin{cases} 0, & \text{if } a(t) = 0, \\ \Phi - P_p & \text{if } a(t) \geq 1 \text{ and } \theta(t) = 0, \\ \Phi - P_{3G}, & \text{if } a(t) = 2 \text{ and } \theta(t) = 1. \end{cases}$$

- Delay: In order to model the impact of the delay, we introduce an additional cost when a packet is not transmitted. This cost depends on the current delay $l(t)$ of the packet, and is defined by the function $f(l(t))$. This function is assumed to be increasing with $l(t)$, in order to growth the incentive of transmitting the packet when it becomes delayed.
- Instantaneous reward: At the time slot $t$, the instantaneous reward $r_t$ of a SU depends on the system state $(\lambda(t), l(t))$ and the action $a(t)$, and is expressed as

follows:

$$r_t((\lambda(t), l(t)), a(t)) = z_t(a(t), \theta(t)) - f(l(t)) - c_s(a(t)).$$

The problem faced by the SU consists of maximizing its average expected reward:

$$\bar{R}(\mu) = \lim_{T\to\infty} \frac{1}{T} I\!E_\mu \left( \sum_{t=1}^{T} r_t((\lambda(t), l(t)), a(t)) | \lambda(0), l(0) \right),$$

while $\lambda(0)$ is the initial belief vector. It is very important to consider the average reward rather than the total reward or the discounted cost as the SU takes frequently decisions. Then, our objective is to find an optimal sensing policy $\mu^*$ that maximizes the average expected reward $\bar{R}(\mu)$:

$$\mu^* = arg \max_{\mu\in\Gamma} \lim_{T\to\infty} \frac{1}{T} I\!E_\mu \left( \sum_{t=1}^{T} r_t((\lambda(t), l(t)), a(t)) | \lambda(0), l(0) \right). \tag{4.2}$$

For simplicity reasons, and to get deep theoretical analysis, we may restrict our study to the set of stationary policies. A stationary policy is a mapping that specifies for each state, independently of time slots, an action to be chosen. Note that looking for stationary policies reduce significantly the computational complexity of the OSA problem. In some particular MDP and POMDP problems, we are able to determine an optimal policy in a smaller set reduced to stationary policies. We prove in the following proposition that there exists an optimal stationary policy for our POMDP problem.

**Proposition 4.1.** *There exists an optimal stationary policy for our POMDP formulation of the OSA problem described in Equation (4.2).*

*Proof.* The proof results from Theorems 8.10.9 and 8.10.7 of [92]. Note that we have a POMDP with a discrete state space.

First, the immediate reward $r_t((s, l), a)$ is finite, i.e. $-\infty < r_t((s, l), a) < +\infty$ (as all costs and rewards are finite). Second, we prove that there exists a stationary policy $d^\infty$ for which the derived Markov chain is positive recurrent.

Let us focus on the following belief vector:

$$\Lambda_0 = (\lambda_1, \lambda_2, ..., \lambda_N) \quad \text{such that} \quad \lambda_j = \Omega^{j-1}(\beta_j|0), \quad \text{for} \quad j = 1, \ldots, N,$$

where $\lambda_j$ represents the belief of a channel that was not sensed for $j$ successive slots.

Denote by $d^\infty$ the stationary policy which senses licensed channels at every slot, with a greedy channel choice policy. Suppose that $\alpha \geq \beta$, the analysis of the other case

is analogous. Let us prove that the derived Markov chain is positive recurrent. Note that if a SU senses the licensed channel $i$ as busy, its belief equals $\beta$, the least belief probability over all the licensed channels, and therefore the SU will choose another licensed channel to sense in the next time slot as it is considering a greedy channel choice policy. The probability that the system returns to the initial belief from any state $\Lambda$ is $p(\lambda) = \prod_{k=0}^{N}(1 - \Omega^n(\lambda_j)) > 0, n \in \{O, ..., N\}$, and then the return time to the initial belief $\tau_j$ follow a geometric distribution so that $E\{\tau_j\} = \frac{1}{p(\Lambda_j)}$. Therefore, all state are positive recurrent under $d^\infty$.

Third, let us prove that $g^{d^\infty} > -\infty$ and the set $\{b \in \mathcal{S}_b : r_t((s,l),a) > g^{d^\infty}$ for some $a \in \mathcal{A}\}$ is finite and no empty. As the policy $d^\infty$ senses licensed channels every slot, $g^{d^\infty} = -f(l(t)) - c_s - (f(l(t)) + P_p - \Phi)\lambda_{n^*}$. If we have the following inequality

$$-f(l(t)) - c_s - (f(l(t)) + P_p - \Phi)\lambda_{n^*} > \max\{-f(l(t)), \Phi - c_s - P_{3G} - (P_p - P_{3G})\lambda_{n^*}\}$$

for all belief $b$, the policy always sense licensed channels is optimal and we have achieved our goal. Otherwise, the set $\{b \in \mathcal{S}_b : r_t((s,l),a) > g^{d^\infty}$ for some $a \in \mathcal{A}\}$ is finite and no empty.

Finally, we obtain from the theorems 8.10.9 and 8.10.7 of [92] that there exists an average optimal stationary policy. □

Given this result, we can restrict our problem to the set $\Gamma_S$ of stationary policies. Then, for the rest of this chapter, we omit the time index $t$, and we look for an optimal sensing policy that is a mapping from a system state $(\lambda, l)$ to an action $a$, independently of the time slot $t$. Before seeking for the optimal OSA policy, we make an analysis of the value function of the POMDP problem.

We denote by $\Omega^{ns}(\lambda|\theta)$, the function that updates the belief vector $\lambda$ when the user chooses to be inactive in the current slot, i.e. the SU takes action 0. The function $\Omega^{s}(\lambda|\theta)$ updates the belief vector $\lambda$ when the SU senses a licensed channel in the current slot and observes $\theta$, i.e. the SU takes the action 1 or 2.

We define, in the following, the value function $V(\lambda, l)$. Let us denote by $Q_a(\lambda, l)$ the action-value function of taking the action $a$ in the current slot when the information state is $(\lambda, l)$. Therefore, the value function is expressed as follows:

$$V(\lambda, l|\lambda_0, l_0) = \max_{a \in \mathcal{A}} \left(g_u(\lambda_0, l_0) + Q_a(\lambda, l|\lambda_0, l_0)\right), \tag{4.3}$$

where $(\lambda_0, l_0)$ is the initial state of the system and $g_u(\lambda_0, l_0)$ is a constant that depends on the initial state. Note that for any stationary policy, the state of the SUs is an irreducible Markov chain with one ergodic class. Thus, a unique steady state probability exists, and

we can omit the initial distribution. Thus, the value function for our POMDP problem can be expressed as follows:

$$g_u + V(\lambda, l) = \max_{a \in \mathcal{A}} Q_a(\lambda, l), \tag{4.4}$$

where $g_u$ is a constant. The optimal policy for our POMDP problem is the one that chooses the following action in the state $(\lambda, l)$ :

$$a^*(\lambda, l) = \arg\max_{a \in \mathcal{A}} Q_a(\lambda, l). \tag{4.5}$$

We determine the action-value function for each different action 0, 1 and 2. When the SU decides to wait, i.e. to take the action $a = 0$, we have:

$$Q_0(\lambda, l) = -f(l) + V(\Omega^{ns}(\lambda|\theta = 0), l + 1). \tag{4.6}$$

When the SU chooses to sense the channel $n^*$ and decides to wait for the next time slot if the channel $n^*$ is busy, i.e. to take action 1, we have:

$$\begin{aligned} Q_1(\lambda, l) &= -c_s + \lambda_{n^*}(\Phi - P_p + V(\Omega^s(\lambda|\theta = 0), 1)) \\ &\quad +(1 - \lambda_{n^*})(-f(l) + V(\Omega^s(\lambda|\theta = 1), l + 1)). \end{aligned} \tag{4.7}$$

When the SU chooses to sense the channel $n^*$ and to transmit using the dedicated channel if the channel $n^*$ is busy, i.e. to take action 2, we have:

$$\begin{aligned} Q_2(\vec{\lambda}, l) &= \Phi - c_s + \lambda_{n^*}(-P_p + V(\Omega^s(\lambda|\theta = 0), 1)) \\ &\quad +(1 - \lambda_{n^*})(-P_{3G} + V(\Omega^s(\lambda|\theta = 1), 1)). \end{aligned} \tag{4.8}$$

For the remainder of this chapter, we take some assumptions that simplify the analysis of the optimal policy. We focus on the case of one licensed channel. The multichannel case will be studied in Section 4.3.2. We take the assumption that there exists a packet delay $l^*$ such that the SU transmits its packet using the dedicated channel if the observation is $\theta = 1$. In fact, this assumption is somehow realistic, as the SU has no interest to keep the file in its buffer indefinitely. We denote by $\alpha$ and $\beta$ the transition rates of the licensed channel, and $\lambda$ the belief of the SU.

### 4.3.1 The single channel model

To solve the POMDP problem, the belief vector is a key element as it gives us insights about the system state. Firstly, we analyze the belief update function. The following

lemma gives us some properties of the belief update function $\Omega^{ns}$. We consider that $\alpha \geq \beta$. When $\alpha \leq \beta$, the analysis is similar and results are analogous.

**Lemma 4.2.** *We have the following properties of the belief update function* $\Omega^{ns}$.

1. *The update function* $\Omega^{ns}(\lambda|\theta)$ *is increasing with belief* $\lambda$.
2. *We have the following equivalence:*

$$\Omega^{ns}(\lambda|\theta) \geq \lambda \quad \Leftrightarrow \quad \lambda \leq \pi(0),$$

*and*

$$\Omega^{ns}(\lambda|\theta) \leq \lambda \quad \Leftrightarrow \quad \lambda \geq \pi(0),$$

*where* $\pi(0) = \frac{\beta}{1-\alpha+\beta}$ *is the stationary probability that the licensed channel is idle.*

*Proof.* First, the update function $\Omega^{ns}$ is linear with the belief because $\Omega^{ns}(\lambda) = \beta + (\alpha - \beta)\lambda$. As we have considered the case where $\alpha \geq \beta$, then the update function is increasing with the belief.

Second, let us prove that $\Omega^{ns}(\lambda) \geq \lambda$ for all beliefs $\lambda \leq \pi(0)$ by induction on the belief.

1. We have the initial condition: $\beta \leq \pi(0) = \frac{\beta}{1-\alpha+\beta}$ and $\Omega^{ns}(\beta) = \beta + (\alpha - \beta)\beta \geq \beta$.
2. Assume that we have: $\Omega^{ns}(\lambda) \geq \lambda$, for a given $\lambda \leq \pi(0)$.
3. The induction operator derives the following belief value: $\Omega^{ns}(\Omega^{ns}(\lambda)) = \beta + (\alpha - \beta)\Omega^{ns}(\lambda) \geq \beta + (\alpha - \beta)\lambda = \Omega^{ns}(\lambda)$.

Thus, $\Omega^{ns}(\lambda) \geq \lambda$ for all $\lambda \leq \pi(0)$. The analysis for $\lambda \geq \pi(0)$ is similar. □

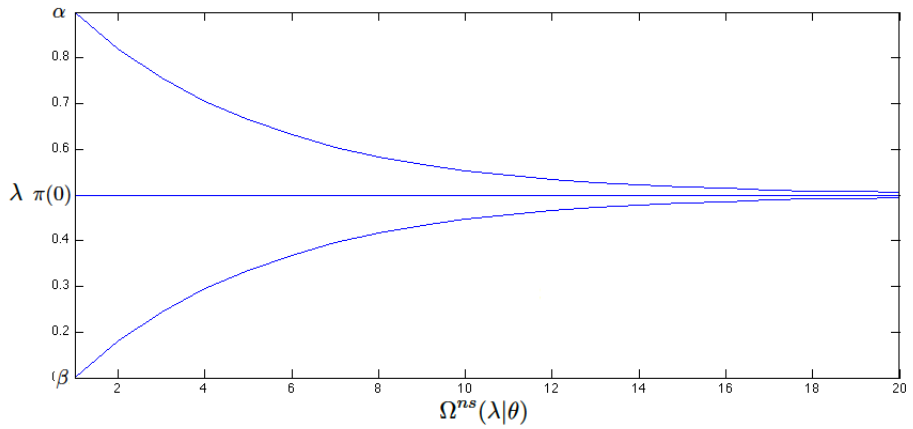


FIGURE 4.5: The belief update function $\Omega^{ns}$ with respect to number of time slots the channel was not sensed.

Figure 4.5 depicts the belief evolution. As shown in [34], the value function for a POMDP over a finite time horizon is PWLC with respect to the belief vector. In Proposition 4.3, we show that the value function for our POMDP problem over an infinite horizon with the average criterion has also this property.

**Proposition 4.3.** *The value function $V(\lambda, l)$, given by Equation (4.4), is PWLC with respect to the belief vector $\lambda$ over an infinite horizon with average criterion.*

*Proof.* The proof of the Proposition 4.3 is similar to [34] where the authors considered the finite time horizon problem. Hence, we briefly describe the procedure for this proof. For all belief vectors $\lambda$, the value function $V(\lambda, l^*)$ is linear with the belief:

$$\begin{aligned} V(\lambda, l^*) &= Q_2(\lambda, l^*) - g_u, \\ &= -g_u + \Phi - c_s - P_{3G} + V(\Omega^s(\lambda|\theta = 1), 1) + \\ &\quad \lambda_{n^*}(P_{3G} - P_p + V(\Omega^s(\lambda|\theta = 0), 1) - V(\Omega^s(\lambda|\theta = 1), 1)). \end{aligned}$$

Then the value function $V(\lambda, l^*)$ can be rewritten as an inner product of the belief vector and a $\Upsilon$-vector. As $Q_2(\lambda, l) = Q_2(\lambda, l^*)$, for all $l$, the action-value function $Q_2(\lambda, l)$ can be also rewritten as an inner product of the belief vector and a $\Upsilon$-vector. We suppose that Proposition 4.3 holds for all packet delays higher than $l+1$, and we prove that the proposition is true for packet delay $l$. After some algebra, we can rewrite the action-value functions given in Equations (4.6) and (4.8) in terms of $\Upsilon$-vector as follows:

$$Q_0(\lambda, l) = -f(l) + \max_{\Upsilon \in \Gamma_{l+1}} < \Omega^{ns}(\lambda|\theta), \Upsilon >= -f(l) + \sum_{s \in \mathcal{S}} \omega_s \left[ \sum_{s' \in \mathcal{S}} P(s'|s) \Upsilon_{l+1}^{\Omega^{ns}(\lambda|\theta)} \right], \quad (4.9)$$

and

$$\begin{aligned} Q_1(\lambda, l) &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1 - \lambda)(-f(l) + \max_{\Upsilon \in \Gamma_{l+1}} < \Omega^s(\lambda|\theta = 1), \Upsilon >) \\ &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1 - \lambda) \left( -f(l) + \sum_{s \in \mathcal{S}} \omega_s \left[ \sum_{s' \in \mathcal{S}} P(s'|s) \Upsilon_{l+1}^{\Omega^s(\lambda|\theta=1)} \right] \right), \quad (4.10) \end{aligned}$$

where $\Upsilon_{l+1}^{\Omega^{ns}(\lambda|\theta)}$ and $\Upsilon_{l+1}^{\Omega^s(\lambda|\theta=1)}$ are, respectively, the $\Upsilon$-vectors for the regions containing belief vectors $\Omega^{ns}(\lambda|\theta)$ and $\Omega^s(\lambda|\theta = 1)$, respectively. Each term in the square brackets of Equations (4.9) and (4.10) are elements $\Upsilon_{\lambda,l}$ of a $\Upsilon$-vector $\Upsilon_l$. Thus, the action-value functions can be rewritten as an inner product of the belief vector and a $\Upsilon$-vector $\Upsilon_l$. Moreover, there is only a finite number of such $\Upsilon$-vector $\Upsilon_l$, since we have a finite set of belief for all $l$. As the maximum of a finite set of piecewise linear and convex functions

is also piecewise linear and convex, the Proposition 4.3 holds for all beliefs and packet delays. □

Note that monotonicity results help us for establishing the structure of the optimal policies (see [93] for an example) and provide insights into the underlying problem. The following propositions states monotonicity results of the value function with respect to each of its parameters.

**Proposition 4.4.** *For a given belief vector $\lambda$, the value function is monotonically decreasing with packet delays, i.e. $V(\lambda, l) \leq V(\lambda, l')$ for $l \geq l'$.*

*Proof.* Let us prove that the value function $V(\lambda, l)$ is monotonically decreasing with packet delays, for a given belief vector $\lambda$. Note that SUs take the action 2 for all beliefs $\lambda$ when the packet delay is $l^*$. Therefore, we have:

$$V(\lambda, l^*) = \Phi - c_s + \lambda(-P_p + V(\alpha, 1)) + (1-\lambda)(-P_{3G} + V(\beta, 1)).$$

Note that the SU chooses the action that maximizes its average expected reward for the packet delay $l^* - 1$ and belief $\lambda$, as follows:

$$\begin{aligned} V(\lambda, l^* - 1) &= \max_a Q_a(\lambda, l^* - 1) - g_u \\ &\geq Q_2(\lambda, l^* - 1) - g_u, \\ &\geq \Phi - c_s + \lambda(-P_p + V(\alpha, 1)) + (1-\lambda)(-P_{3G} + V(\beta, 1)) - g_u, \\ &\geq V(\lambda, l^*). \end{aligned}$$

Let us prove that this propriety holds for all packet delays using a backward induction on packet delays:

1. Initial condition: For all belief vector $\lambda$, we have that: $V(\lambda, l^*) \leq V(\lambda, l^* - 1)$,
2. Suppose that $V(\lambda, l+2) \leq V(\lambda, l+1)$, $\forall \lambda$.
3. We have:

$$\begin{aligned} Q_0(\lambda, l) &= -f(l) + V(\Omega^{ns}(\lambda|\theta), l+1), \\ &\geq -f(l+1) + V(\Omega^{ns}(\lambda|\theta), l+2), \\ &= Q_0(\lambda, l+1). \end{aligned}$$

$$\begin{aligned} Q_1(\lambda, l) &= -c_s + \lambda(\Phi - P_p + V(\alpha, 1)) + (1-\lambda)(-f(l) + V(\beta, l+1)), \\ &\geq -c_s + \lambda(\Phi - P_p + V(\alpha, 1)) + (1-\lambda)(-f(l+1) + V(\beta, l+2)), \\ &= Q_1(\lambda, l+1). \\ Q_2(\lambda, l) &= -c_s + \Phi - P_{3G} + V(\beta, 1) + \lambda(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &= Q_2(\lambda, l+1). \end{aligned}$$

The inequalities come from the induction assumption and the monotonicity of the penalty function $f(l)$.

Finally, the value function is decreasing with packet delays. □

This result is intuitive as for the same belief $\lambda$ and for a given packet delay, the maximum expected remaining reward that can be accrued is lower than the one the SU can get with a smaller packet delay. We present, in the following lemma, a result that will be useful for the proof of the monotonicity of the value function with respect to the belief.

**Lemma 4.5.** *We have the following inequality:*

$$-P_p + V(\alpha, 1) \geq -P_{3G} + V(\beta, 1).$$

*Proof.* We prove this lemma by contradiction. Suppose that $-P_p + V(\alpha, 1) < -P_{3G} + V(\beta, 1)$. Let us prove, in the following, that the constant $g_u$ is higher than $\Phi - c_s - P_p$:

$$\begin{aligned} g_u + V(\alpha, 1) &\geq Q_2(\alpha, 1), \\ g_u + V(\alpha, 1) &\geq -c_s + \alpha(\phi - P_p + V(\alpha, 1)) + (1-\alpha)(\phi - P_{3G} + V(\beta, 1)), \\ g_u + V(\alpha, 1) &\geq -c_s + \phi - P_p + V(\alpha, 1), \\ g_u &> \Phi - c_s - P_p. \end{aligned}$$

We take the assumption that the immediate reward when the channel is idle is positive, i.e. $\Phi - c_s - P_p \geq 0$. We have already assumed that the SU takes the action 2 in the state $(\lambda, l^*)$ for all belief vector $\lambda$, i.e $a^*(\lambda, l^*) = 2, \forall \lambda$. Therefore, we have:

$$g_u + V(\lambda, l^*) = -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1-\lambda)(\phi - P_{3G} + V(\beta, 1)).$$

Let us focus on the packet delay $l^* - 1$. If $\lambda \leq \pi(0)$, the following inequality holds:

$$\begin{aligned} Q_0(\lambda, l^* - 1) &= -f(l^* - 1) + V(\Omega^{ns}(\lambda), l^*), \\ &= -g_u - f(l^* - 1) - c_s + \Omega^{ns}(\lambda)(\phi - P_p + V(\alpha, 1)) \\ &+ (1 - \Omega^{ns}(\lambda))(\phi - P_{3G} + V(\beta, 1)), \\ &= V(\lambda, l^*) - f(l^* - 1) + (\Omega^{ns}(\lambda) - \lambda)(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &< V(\lambda, l^*). \end{aligned}$$

The inequality is due to the assumption that $-P_p + V(\alpha, 1) < -P_{3G} + V(\beta, 1)$, the belief update function $\Omega^{ns}(\lambda) \geq \lambda$, and the delay penalty $f(l^* - 1)$ is positive. As the value function $V(\lambda, l)$ is decreasing with packet delays (see Proposition 4.4), then we have: $Q_0(\lambda, l^* - 1) < V(\lambda, l^*) < V(\lambda, l^* - 1)$. Note that we have already proved that $g_u$ is positive. Thus, the SU does not take the action 0 when the packet delay is $l^* - 1$. Let us focus on the action 1, we have the following inequality:

$$\begin{aligned} Q_1(\lambda, l^* - 1) &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1 - \lambda)(-f(l^* - 1) + V(\beta, l^*)), \\ &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1 - \lambda)\,(\,\phi - g_u - f(l^* - 1) - c_s \\ &\quad +\beta(-P_p + V(\alpha, 1)) + (1 - \beta)(-P_{3G} + V(\beta, 1)))\,, \\ &< -c_s + \lambda(\phi - P_p + V(\alpha, 1)) \\ &+ (1 - \lambda)(\phi - g_u - f(l^* - 1) - c_s - P_{3G} + V(\beta, 1)), \\ &< -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1 - \lambda)(\phi - P_{3G} + V(\beta, 1)), \\ &= Q_2(\lambda, l^* - 1). \end{aligned}$$

The first inequality is due to the assumption that $-P_p + V(\alpha, 1) < -P_{3G} + V(\beta, 1)$, and the second one is because $g_u$, $f(l^* - 1)$ and $c_s$ are positive. Thus, the optimal strategy is to take the action 2 when the packet delay is $l^* - 1$.

Let us prove by backward induction on $l$, that the optimal action is the action 2 for all belief vector $\lambda \leq \pi(0)$.

- If the SU takes the action 2 when the packet delay is $l^*$, then it takes also the action 2 when the packet delay is $l^* - 1$.
- We suppose that SU takes the action 2 when the packet delay is $l < l^* - 1$.

- We have the following inequality:

$$\begin{aligned} Q_0(\lambda, l-1) &= -f(l-1) + V(\Omega^{ns}(\lambda), l), \\ &= -g_u - f(l-1) - c_s + \Omega^{ns}(\lambda)(\phi - P_p + V(\alpha, 1)) \\ &+ (1 - \Omega^{ns}(\lambda))(\phi - P_{3G} + V(\beta, 1)), \\ &= V(\lambda, l) - f(l-1) + (\Omega^{ns}(\lambda) - \lambda)(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &< V(\lambda, l). \end{aligned}$$

The inequality is due to the assumption that $-P_p + V(\alpha, 1) < -P_{3G} + V(\beta, 1)$, $\Omega^{ns}(\lambda) \geq \lambda$, and $f(l-1)$ is positive. As the value function is decreasing with the packet delay (see Proposition 4.4), then $Q_0(\lambda, l-1) < V(\lambda, l-1) + g_u$, i.e. the SU does not take the action 0 with the packet delay $l-1$. Let us compare the action-value functions $Q_1(\lambda, l-1)$ and $Q_2(\lambda, l-1)$:

$$\begin{aligned} Q_1(\lambda, l-1) &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1-\lambda)(-f(l-1) + V(\beta, l)), \\ &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1-\lambda)\,(\,\phi - g_u - f(l-1) - c_s \\ &\quad +\beta(-P_p + V(\alpha, 1)) + (1-\beta)(-P_{3G} + V(\beta, 1)))\,, \\ &< -c_s + \lambda(\phi - P_p + V(\alpha, 1)) \\ &+ (1-\lambda)(\phi - g_u - f(l-1) - c_s - P_{3G} + V(\beta, 1)), \\ &< -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1-\lambda)(\phi - P_{3G} + V(\beta, 1)), \\ &= Q_2(\lambda, l-1). \end{aligned}$$

The first inequality is due to the assumption that $-P_p + V(\alpha, 1) < -P_{3G} + V(\beta, 1)$ and the second one is because $g_u$, $f(l-1)$ and $c_s$ are positive. Thus, The optimal strategy is to take action 2 when the delay of its packet equals $l-1$.

Finally, the SU takes action 2 for all packet delays and beliefs lower than $\pi(0)$.

Let us focus on the action-value function $Q_2(\alpha, 1)$, when the packet delay is $l = 1$, we have:

$$\begin{aligned} Q_2(\alpha, 1) &= -c_s + \alpha(\phi - P_p + V(\alpha, 1)) + (1-\alpha)(\phi - P_{3G} + V(\beta, 1)), \\ Q_2(\alpha, 1) &= \phi - c_s - P_{3G} + V(\beta, 1) + \alpha(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ -g_u + Q_2(\alpha, 1) &= -g_u + V(\alpha, 1) - P_p + \phi - c_s + (\alpha - 1)(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)). \end{aligned}$$

As the SU takes the action 2 also for the state $(\beta, 1)$, we have the following expression of the constant $g_u$:

$$\begin{aligned} g_u + V(\beta,1) &= -c_s + \beta(\phi - P_p + V(\alpha,1)) + (1-\beta)(\phi - P_{3G} + V(\beta,1)), \\ g_u + V(\beta,1) &= \phi - c_s - P_{3G} + V(\beta,1) + \beta(P_{3G} - P_p + V(\alpha,1) - V(\beta,1)), \\ g_u &= \phi - c_s - P_{3G} + \beta(P_{3G} - P_p + V(\alpha,1) - V(\beta,1)). \end{aligned}$$

Thus, we obtain:

$$-g_u + Q_2(\alpha,1) = V(\alpha,1) + P_{3G} - P_p + (\alpha - \beta - 1)(P_{3G} - P_p + V(\alpha,1) - V(\beta,1)).$$

As we have assumed that $P_{3G} - P_p + V(\alpha,1) - V(\beta,1) < 0$, and $P_{3G} > P_p$, we obtain: $V(\alpha,1) + g_u \leq Q_2(\alpha,1)$, and the SU takes also the action 2 in the state $(\alpha,1)$:

$$g_u + V(\alpha,1) = Q_2(\alpha,1) = -c_s + \alpha(\phi - P_p + V(\alpha,1)) + (1-\alpha)(\phi - P_{3G} + V(\beta,1)).$$

Finally, let us evaluate the difference between $V(\alpha,1)$ and $V(\beta,1)$. We have:

$$\begin{aligned} V(\alpha,1) - V(\beta,1) &= (\alpha - \beta)(P_{3G} - P_p + V(\alpha,1) - V(\beta,1)), \\ V(\alpha,1) - V(\beta,1) &< 0. \end{aligned}$$

and

$$\begin{aligned} V(\alpha,1) - V(\beta,1) &= (\alpha - \beta)(P_{3G} - P_p + V(\alpha,1) - V(\beta,1)), \\ (V(\alpha,1) - V(\beta,1))(1 - \alpha + \beta) &= (\alpha - \beta)(P_{3G} - P_p), \\ V(\alpha,1) - V(\beta,1) &= \frac{(\alpha - \beta)(P_{3G} - P_p)}{1 - \alpha + \beta}, \\ &> 0. \end{aligned}$$

which leads to a contradiction. Therefore, $-P_p + V(\alpha,1) \geq -P_{3G} + V(\beta,1)$. The analysis is similar when $\lambda > \pi(0)$. □

We study the monotonicity of the value function with respect to the belief. Intuitively, with a higher belief, for a given packet delay, the SU obtains better rewards. We prove, in the following proposition, that this intuition is true.

**Proposition 4.6.** *For a given packet delay $l$, the value function is monotonically increasing with beliefs $\lambda$, i.e. $V(\lambda, l) \geq V(\lambda', l)$ for $\lambda \geq \lambda'$.*

*Proof.* Let us prove that the value function $V(\lambda, l)$ is increasing with beliefs $\lambda$ for a given packet delay $l$. For all $\lambda_1 \leq \lambda_2$, we have:

$$\begin{aligned} V(\lambda_1, l^*) &= -g_u - c_s + \Phi - P_{3G} + V(\beta, 1) + \lambda_1(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &\leq -g_u - c_s + \Phi - P_{3G} + V(\beta, 1) + \lambda_2(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &= V(\lambda_2, l^*). \end{aligned}$$

This inequality results from the Lemma 4.5. Let us prove that this propriety holds for all packet delays using backward induction:

- Initial condition: There exists a packet delay $l^*$, such that $V(\lambda_1, l^*) \leq V(\lambda_2, l^*)$, $\forall \lambda_1 \leq \lambda_2$,
- Suppose that $V(\lambda_1, l+1) \leq V(\lambda_2, l+1)$, $\forall \lambda_1 \leq \lambda_2$,
- we have the following expressions of the action-value functions:

$$\begin{aligned} Q_0(\lambda_1, l) &= -f(l) + V(\Omega^{ns}(\lambda_1|\theta), l+1), \\ &\leq -f(l) + V(\Omega^{ns}(\lambda_2|\theta), l+1), \\ &= Q_0(\lambda_2, l). \end{aligned}$$

The inequality is a direct result from the induction assumption and the Lemma 4.2. Moreover, we have:

$$\begin{aligned} Q_2(\lambda_1, l) &= -c_s + \Phi - P_{3G} + V(\beta, 1) + \lambda_1(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &\leq -c_s + \Phi - P_{3G} + V(\beta, 1) + \lambda_2(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)), \\ &= Q_2(\lambda_2, l). \end{aligned}$$

The inequality comes also from the Lemma 4.5.

**First case** Assume that $\Phi + f(l) - P_p + V(\alpha, 1) - V(\beta, l+1) \geq 0$. Then, we have:

$$\begin{aligned} Q_1(\lambda_1, l) &= -c_s - f(l) + V(\beta, l+1) + \lambda_1(\Phi + f(l) - P_p + V(\alpha, 1) - V(\beta, l+1)), \\ &\leq -c_s - f(l) + V(\beta, l+1) + \lambda_2(\Phi + f(l) - P_p + V(\alpha, 1) - V(\beta, l+1)), \\ &= Q_1(\lambda_2, l). \end{aligned}$$

Finally, we have that $V(\lambda_1, l) \leq V(\lambda_2, l)$.

**Second case** Assume that $\Phi + f(l) - P_p + V(\alpha, 1) - V(\beta, l+1) < 0$. Then, for all beliefs $\lambda$, we have:

$$\begin{aligned} Q_1(\lambda, l) &= -c_s + \lambda(\phi - P_p + V(\alpha, 1)) + (1-\lambda)(-f(l) + V(\beta, l+1)), \\ &\leq -c_s - f(l) + V(\beta, l+1), \\ &\leq -c_s - f(l) + V(\Omega^{ns}(\lambda|\theta), l+1), \\ &\leq Q_0(\lambda, l). \end{aligned}$$

In fact, we have that $\beta \leq \Omega^{ns}(\lambda|\theta)$ for all beliefs, and the value function $V(\lambda, l)$ is increasing with beliefs for the packet delay $l+1$ (induction assumption). Thus, $g_u + V(\lambda, l) = \max\{Q_0(\lambda, l), Q_2(\lambda, l)\}$. Therefore, we have proved that $V(\lambda_1, l) \leq V(\lambda_2, l)$.

Finally, the value function is increasing with beliefs for all packet delays. □

In the following lemma, we prove that $g_u > -f(l)$.

**Lemma 4.7.** *The value function's constant $g_u$ is higher than $-f(l)$.*

*Proof.* we have:

$$\begin{aligned} g_u + V(\alpha, 1) &\geq Q_0(\alpha, 1), \\ g_u + V(\alpha, 1) &\geq -f(l) + V(\Omega^{ns}(\alpha), l+1), \\ g_u + V(\alpha, 1) - V(\Omega^{ns}(\alpha), l+1) &\geq -f(l), \\ g_u &> -f(l). \end{aligned}$$

The inequality comes from the monotonicity of the value function and $\Omega^{ns}(\alpha) < \alpha$. □

Once we have studied the monotonicity of the value function with respect to both of its parameters, we are able to show that the optimal OSA policy has a threshold structure.

### 4.3.2 The multichannel model

Note that Lemma 4.2 holds for the multichannel model. In fact, if $\vec{\lambda}_1 \leq \vec{\lambda}_2$, then $\lambda_{n_1^*} \leq \lambda_{n_2^*}$ and $\Omega^{ns}(\lambda_{n_1^*}) \leq \Omega^{ns}(\lambda_{n_2^*})$, and therefore, $\Omega^{ns}(\vec{\lambda}_1) \leq \Omega^{ns}(\vec{\lambda}_2)$. Second, consider that $\lambda_{n^*} \leq \pi(0)$. Then, we have $\Omega^{ns}(\lambda_{n^*}) \geq \lambda_{n^*}$, and thus $\Omega^{ns}(\vec{\lambda}) \geq \vec{\lambda}$. Otherwise, we have $\Omega^{ns}(\vec{\lambda}) \leq \vec{\lambda}$.

The Proposition 4.3 can be straightforwardly extended to the multichannel model. Furthermore, we have studied, in Proposition 4.4, the monotonicity of the value function

for a fixed belief value with respect to the packet delay. This proposition can be also straightforwardly extended to the multichannel model.

Let us focus on the Proposition 4.6. The monotonicity of the value function with respect to the belief vector depends on the order relationship over the belief set and also on the monotonicity of the belief update functions $\Omega^s(\lambda|\theta=0)$ and $\Omega^s(\lambda|\theta=1)$ depending on the belief vector. Note that the monotonicity of the value function with respect to the belief is the main difficulty for extending our study to the multichannel model, and will be considered as a part of our future works.

## 4.4 Optimal threshold policy

We determine, in this section, an optimal OSA policy for the SU, and we study the structure of such policy. An intuitive behavior of a SU that is accessing opportunistically the spectrum, and that is aware of both energy consumption and transmission delay, is:

- When the packet is recent, i.e. the delay of the packet is small, and the belief is small the SU chooses to wait for better opportunities at next time slots.
- For a delayed packet, the SU chooses to sense and access the dedicated channel if there are no free licensed channels.

We prove in this section, that the intuition is true, and there exists an optimal sensing policy, which has a threshold structure.

Note that SUs have a two-level decision. the first decision for a SU is whether to sense the licensed channels or to wait, depending on its belief, $\lambda$, and the current delay of the packet, $l$. Specifically, we have the following result, which gives us a threshold policy on the belief probability that answers this question.

**Proposition 4.8.** *For a given packet delay $l$, the optimal action for the SU is to wait for the next time slot, i.e. $a^*(\lambda,l)=0$ if and only if $\lambda \leq \lambda^*$, where $\lambda^*$ is the solution of the equation $\lambda^* = \max(0, \min\{Th1(\lambda^*,l), Th2(\lambda^*,l)\})$. The thresholds $Th1(\lambda^*,l)$ and $Th2(\lambda^*,l)$ are expressed as follows:*

$$Th1(\lambda^*,l) = \frac{V(\Omega^{ns}(\lambda^*|\theta),l+1) - V(\beta,l+1) + c_s}{f(l) + \Phi - P_p + V(\alpha,1) - V(\beta,l+1)}, \quad \textit{and}$$

$$Th2(\lambda^*,l) = \frac{V(\Omega^{ns}(\lambda^*|\theta),l+1) - V(\beta,1) + c_s - f(l) - \Phi + P_{3G}}{-P_p + V(\alpha,1) + P_{3G} - V(\beta,1)}.$$

*Proof.* In this proposition, we determine explicitly the best action $a^*(\lambda, l)$ for the SU depending on the belief $\lambda$ and the packet delay $l$. For a given information state $(\lambda, l)$, the SU decides to take the action 0 if and only if $Q_0(\lambda, l) \geq \max\{Q_1(\lambda, l), Q_2(\lambda, l)\}$.

- First, we assume that $Q_1(\lambda, l) > Q_2(\lambda, l)$. Let us compare $Q_0(\lambda, l)$ and $Q_1(\lambda, l)$. The inequality $Q_0(\lambda, l) \geq Q_1(\lambda, l)$ is equivalent to:

$$\begin{aligned} -f(l) + V(\Omega^{ns}(\lambda|\theta), l+1) &\geq -c_s + \lambda(\Phi - P_p + V(\alpha, 1)) \\ &\quad +(1-\lambda)(-f(l) + V(\beta, l+1)), \\ V(\Omega^{ns}(\lambda|\theta), l+1) &\geq V(\beta, l+1) - c_s + \lambda(f(l) \\ &\quad +\Phi - P_p + V(\alpha, 1) - V(\beta, l+1)). \end{aligned}$$

  As the value function $V(\lambda, l)$ is decreasing with packet delays and increasing with beliefs, we have that $V(\alpha, 1) \geq V(\beta, l+1)$. Moreover, we have already assumed that the immediate reward $\Phi$ is higher than the cost $P_p$. Thus, the expression $f(l) + \Phi - P_p + V(\alpha, 1) - V(\beta, l+1)$ is positive, and we obtain the following equivalence:

$$\begin{aligned} Q_0(\lambda, l) \geq Q_1(\lambda, l) \Leftrightarrow V(\Omega^{ns}(\lambda|\theta), l+1) &\geq V(\beta, l+1) - c_s + \lambda(f(l) + \Phi - P_p \\ &\quad +V(\alpha, 1) - V(\beta, l+1)). \end{aligned}$$

  Define the functions F and G as follows:

$$\begin{aligned} F(\lambda, l) &= V(\Omega^{ns}(\lambda|\theta), l+1), \\ G(\lambda, l) &= V(\beta, l+1) - c_s + \lambda(f(l) + \Phi - P_p + V(\alpha, 1) - V(\beta, l+1)). \end{aligned}$$

  Note that:

  - If $F(\lambda, l) \geq G(\lambda, l)$, then $Q_0(\lambda, l) \geq Q_1(\lambda, l)$, and the optimal action for the SU is $a(t) = 0$.
  - If $F(\lambda, l) < G(\lambda, l)$, then $Q_0(\lambda, l) < Q_1(\lambda, l)$, and the optimal action for the SU is $a(t) = 1$.

  We have proved, in Proposition 4.3, that the value function is PWLC with beliefs. Therefore, for all packet delays, the function $F(\lambda, l)$ is PWLC and increasing with $\lambda$, and the function $G(\lambda, l)$ is linear and increasing with $\lambda$. Let us study the sign of $F(\lambda, l) - G(\lambda, l)$. Under these setting, six cases rise up:

  1. $F(\lambda, l)$ is always higher than $G(\lambda, l)$, see Figure (4.6, case 1).
  2. $F(\lambda, l)$ is always lower than $G(\lambda, l)$, see Figure (4.6, case 2).

3. $F(\lambda, l)$ and $G(\lambda, l)$ intersect once and $F(\beta, l) < G(\beta, l)$, see Figure (4.6, case3).
4. $F(\lambda, l)$ and $G(\lambda, l)$ intersect once and $F(\beta, l) \geq (\beta, l)$, see Figure (4.6, case 4).
5. $F(\lambda, l)$ and $G(\lambda, l)$ intersect twice and $F(\beta, l) \geq (\beta, l)$, see Figure (4.6, case 5).
6. $G(\lambda, l)$ is tangent to $F(\lambda, l)$, see Figure (4.6, case 6).

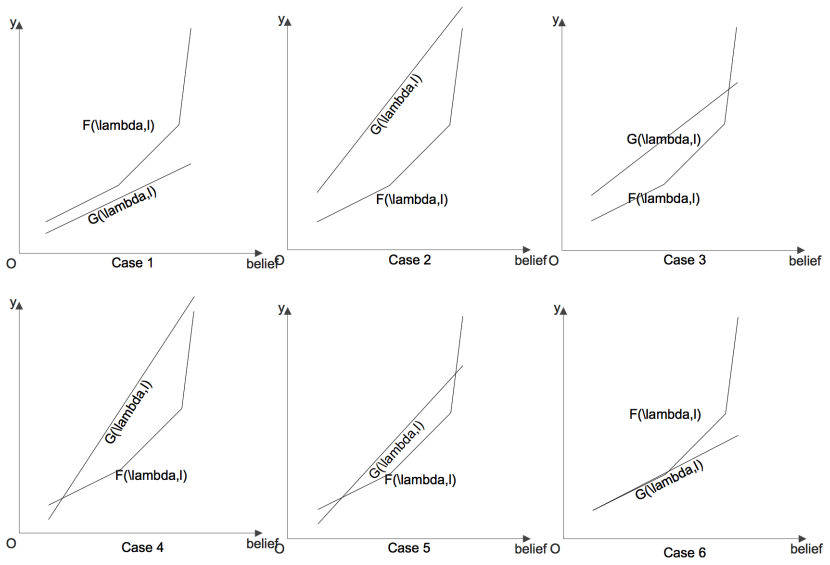


FIGURE 4.6: The analysis of the threshold: the functions $F(\lambda, l)$ and $G(\lambda, l)$.

Let us focus on the study of $F(\pi(0), l)$ and $G(\pi(0), l)$. Suppose that the SU chooses the action 0 for the state $(\pi(0), l)$. Then, we obtain the following inequality:

$$\begin{aligned} g_u + V(\pi(0), l) &= -f(l) + V(\Omega^{ns}(\pi(0)), l+1), \\ g_u + V(\pi(0), l) &\leq -f(l) + V(\Omega^{ns}(\pi(0)), l), \\ g_u + V(\pi(0), l) &\leq -f(l) + V(\pi(0), l), \\ g_u &\leq -f(l). \end{aligned}$$

This leads to a contradiction as $g_u > -f(l)$ (see Lemma 4.7). It follows that $Q_0(\lambda, l) < Q_1(\lambda, l)$, and $F(\pi(0), l) < G(\pi(0), l)$. Thus, F and G intersect once for belief probability in $[\beta, \alpha]$. Finally, the optimal OSA policy is depicted in the following:

– The SU takes the action 0 for all beliefs lower than the following threshold:

$$Th1(\lambda, l) = \frac{V(\Omega^{ns}(\lambda|\theta), l+1) - V(\beta, l+1) + c_s}{f(l) + \Phi - P_p + V(\alpha, 1) - V(\beta, l+1)},$$

and takes the action 1 otherwise.

- Second, we consider the case where $Q_2(\lambda, l) > Q_1(\lambda, l)$. Then, we have to compare the actions 0 and 2, which is equivalent to comparing the action-value functions $Q_0(\lambda, l)$ and $Q_2(\lambda, l)$. The SU takes the action 0 instead of the action 2 if and only if $Q_0(\lambda, l) \geq Q_2(\lambda, l)$, which is equivalent to:

$$\begin{aligned} -f(l) + V(\Omega^{ns}(\lambda|\theta), l+1) &\geq -c_s + \lambda(\Phi - P_p + V(\alpha, 1)) \\ &\quad +(1-\lambda)(\phi - P_{3G} + V(\beta, 1)), \\ V(\Omega^{ns}(\lambda|\theta), l+1) &\geq V(\beta, 1) + \Phi + f(l) - c_s - P_{3G} \\ &\quad +\lambda(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)). \end{aligned}$$

  Note that we have, from Lemma 4.5, that $P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1) \geq 0$. Then, we can provide the same analysis presented in the previous case with the function $F(\lambda, l) = V(\Omega^{ns}(\lambda|\theta), l+1)$ and the function $G(\lambda, l) = V(\beta, 1) + \Phi + f(l) - c_s - P_{3G} + \lambda(P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1))$. The former is PWLC and increasing with $\lambda$, and latter is linear increasing with $\lambda$. Thus, we obtain the following threshold policy:

  - The SU takes the action 0 for all beliefs lower than the following threshold:

$$Th2(\lambda, l) = \frac{V(\Omega^{ns}(\lambda|\theta), l+1) - V(\beta, 1) - \Phi - f(l) + c_s + P_{3G}}{P_{3G} - P_p + V(\alpha, 1) - V(\beta, 1)},$$

    and takes the action 2 otherwise.

□

This proposition gives us a necessary and sufficient condition on the sensing decision of SUs depending on the belief probability $\lambda$. Consequently, if $\lambda > \lambda^*$, then the optimal action for the SU is to sense licensed channels, i.e. $a^*(\lambda, l) \neq 0$.

Furthermore, we have proved, in Lemma 4.2, that the belief vector may be decreasing with the belief update function $\Omega^{ns}(.)$. It follows that there are less opportunities, at the next time slot, to transmit the packet. Thus, the SU should never decide to wait, i.e. action 0, if the belief decreases with $\Omega^{ns}(.)$. The following proposition proves that this intuition is true.

**Proposition 4.9.** *For all $\lambda > \pi(0)$ and $l$, the SU never takes the action 0 and thus, $Q_0(\lambda, l) < \max\{Q_1(\lambda, l), Q_2(\lambda, l)\}$.*

*Proof.* We have from the Lemma 4.2 that if $\lambda > \pi(0)$ then $\Omega^{ns}(\lambda) \leq \lambda$. Suppose that the SU takes the action 0 for a belief $\lambda$ and packet delay $l$. Thus we have:

$$\begin{aligned} g_u + V(\lambda, l) &= -f(l) + V(\Omega^{ns}(\lambda), l+1), \\ g_u + V(\lambda, l) &\leq -f(l) + V(\Omega^{ns}(\lambda), l), \\ g_u + V(\lambda, l) &\leq -f(l) + V(\lambda, l), \\ g_u &\leq -f(l). \end{aligned}$$

This leads to a contradiction as $g_u > -f(l)$. The first inequality is because the value function is decreasing with the packet delay, and the second one is because the value function is increasing with the belief and $\Omega^{ns}(\lambda) \leq \lambda$. Thus, if $\lambda > \pi(0)$, then the SU never takes the action 0 and then $Q_0(\lambda, l) < \max\{Q_1(\lambda, l), Q_2(\lambda, l)\}$. □

*Remark* 4.10. The SU never chooses the action 0 after it transmits a packet over the licensed channel because $\Omega^s(\lambda, \theta = 0) = \alpha > \pi(0)$.

Note that if licensed channels are often occupied, the SU would decide to transmit using the dedicated channel. We depict, in the following proposition, the threshold structure of the optimal decision about the use of the dedicated channel.

**Proposition 4.11.** *For all belief $\lambda$, the SU chooses to use the dedicated channel in spite of waiting for the next time slot if and only if the delay $l$ of the current packet verifies:*

$$-f(l) - \Phi + P_{3G} + V(\beta, l+1) - V(\beta, 1) > 0.$$

*Proof.* Let us compare the value-action functions $Q_1(\lambda, l)$ and $Q_2(\lambda, l)$ for all belief vector $\lambda$ and packet delay $l$. The SU waits for next time slot after sensing if $Q_1(\lambda, l) \geq Q_2(\lambda, l)$, which is equivalent to:

$$\begin{aligned} -c_s + \lambda(\Phi - P_p + V(\alpha, 1)) + (1-\lambda)(-f(l) + V(\beta, l+1)) &\geq -c_s + \lambda(\Phi - P_p + V(\alpha, 1)) \\ &\quad + (1-\lambda)(\phi - P_{3G} + V(\beta, 1)), \\ -f(l) + V(\beta, l+1)\phi - P_{3G} + V(\beta, 1) &\geq 0. \end{aligned}$$

Remark that this condition depends only on the packet delay $l$. □

Note that this expression depends neither on the cost of sensing $C_s$ nor on the belief vector $\lambda$. That is obvious, as this expression determines the best action to do after sensing a channel. We have the last property about the optimal threshold policy.

*Corollary* 1 (Never Wait After Sensing). For all $l$, if the penalty cost $-f(l)$ is lower than $\Phi - P_{3G}$, then the SU transmits on the dedicated channel when the licensed channel is sensed as busy.

*Proof.* If $-f(l)$ is lower than $\Phi - P_{3G}$, then $-f(l) - \Phi + P_{3G} + V(\beta, l+1) - V(\beta, 1)$ is always negative. In fact, $V(\beta, 2) - V(\beta, 1)$ is negative, and $-f(l) - \Phi + P_p + V(\beta, l+1) - V(\beta, 1)$ is decreasing with $l$. Therefore, the previous expression is negative for all delays $l \geq 1$. □

*Remark* 4.12. We obtain a two-level threshold structure of the optimal OSA policy, one threshold for each decision step (see Figure 4.4). In fact, the SU has to choose between the two following options: to sleep (action 0), or to sense the licensed channels (action 1). This decision is made using the threshold expressed in Proposition 4.8, based on the belief vector. Thereafter, the SU makes decision about using the dedicated channel or not (action 2), if it decides to sense licensed channels, depending on its packet delay, regardless the belief, based on Proposition 4.11.

Obviously, the optimal OSA policy depends on the transition rates $\alpha$ and $\beta$ of the PUs' activity. Most of researches in the OSA area assume that some information such that the statistics about the PUs' activity, or the licensed channel transition rates are priory known by the SUs, which may not be realistic in decentralized systems. In practice, an SDR that implement CR uses some learning methods to get insight about the RF environment. We present, in the following section, some learning methods that can be used in order to learn transition rates of the licensed channel on-the-fly.

## 4.5 Online learning of the RF environment

We have already proved that SUs have an optimal energy-delay constrained policy having a threshold structure, given perfect knowledge of channels transition rates. However, in practice, some information, such as transition rates $\alpha$ and $\beta$, are not available for the SU. In this section, we consider a model where the SU does not have external information about the state transition rates. In the following, we present two learning based protocols for SUs in order to estimate the licensed channels dynamics: rate estimator, and transition matrices estimator.

### 4.5.1 Rate estimator

In this approach, the SU starts with an initial arbitrary values of $\alpha$ and $\beta$. Then, it updates them every time slot depending on information about the system state. In fact, the SU computes its sensing policy based on the estimators $\hat{\alpha} = \{\hat{\alpha}_1, ..., \hat{\alpha}_N\}$ and

$\hat{\beta} = \{\hat{\beta}_1, ..., \hat{\beta}_N\}$, where $\hat{\alpha}_i$ (resp. $\hat{\beta}_i$) is the estimator of $\alpha_i$ (resp. $\beta_i$). In practice, the SU estimates the following parameters. First, the SU estimates $\hat{\alpha}_i$, which is the probability that the channel $i$ is sensed as idle, given that it was idle in the previous slot. Second, the SU estimates $\hat{\pi}_i(0)$, the stationary probability that the licensed channel is sensed as idle. Finally, the SU obtains the estimated value of $\beta_i$ based on the following relation:

$$\hat{\beta}_i = (1 - \hat{\alpha}_i)\frac{\hat{\pi}_i(0)}{1 - \hat{\pi}_i(0)}.$$

Formally, the licensed channels' transition rates are estimated based on the following counting processes:

- The vector $\hat{K} = \{\hat{K}_1, ..., \hat{K}_N\}$, where $\hat{K}_i$ represents the number of time slots a channel stays in the idle state, i.e. $\hat{K}_i$ is incremented if the channel $i$ is sensed and is idle at time slots $t$ and $t-1$.
- The vector $\hat{I} = \{\hat{I}_1, ..., \hat{I}_N\}$, where $\hat{I}_i$ represents the number of time slots that the channel is sensed and is idle.
- The vector $\hat{M} = \{\hat{M}_1, ..., \hat{M}_N\}$, where $\hat{M}_i$ represents the number of time slots that the channel is sensed.

Therefore, the SU estimates the state transition rates $\hat{\alpha}$ and $\hat{\pi}_i(0)$ based on the following expressions: $\hat{\alpha}_i = \frac{\hat{K}_i}{\hat{I}_i}$ and $\hat{\pi}_i(0) = \frac{\hat{I}_i}{\hat{M}_i}$.

The convergence of the previous estimators, $\hat{\alpha}$ and $\hat{\beta}$, depends on the occurrence of two successive sensing of the same channel. The SU may not sense frequently the same channel in two successive time slots. Therefore, this estimation method may be inaccurate, and may also harm the SU decision. We propose, in the next section, a more accurate, but also more complex, learning method named transition matrices.

### 4.5.2 Transition matrices estimator

We present, in this section, a learning protocol that estimates the transition matrices. We define the set of transition matrices $\{P_i(0), P_i(1), ...\}$, where $P_i(j)$ is the transition matrix of the channel $i$, when this channel was not sensed during $j$ consecutive slots. For example, if the channel $i$ was sensed, $j$ slots before as idle, then the current belief on the state of this channel is $(1, 0) * P_i(j)$.

Similarly to the rate estimator, the transition matrices are estimated using counting processes. Note that the previous learning protocol is somehow a particular case of this

TABLE 4.1: Simulation parameters

| Parameter | Value |
|---|---|
| $P_{3G}$ | 80 |
| $P_p$ | 10 |
| $c_s$ | 5 |
| $\Phi$ | 35 |

TABLE 4.2: Simulation scenarios

| Scenario | Description | $\alpha$ | $\beta$ |
|---|---|---|---|
| Scenario 1 | Licensed channels are often occupied | 0.15 | 0.1 |
| Scenario 2 | Licensed channels are often idle | 0.85 | 0.7 |
| Scenario 3 | Licensed channels have low transition rates | 0.95 | 0.05 |

approach. In fact, estimating $\alpha$ and $\beta$ is equivalent to estimating the set of transition matrices such that the channel was sensed in the previous slot $\{P_1(0), ..., P_N(0)\}$. Indeed, this learning based protocol gives a more accurate estimation of PUs' activity. Specifically, transition matrices estimator method updates the set of transition matrices every time slot in contrast to the rate estimators method which updates the transition rates only if SU senses as idle of the same licensed channel channel for two successive time slots. However, it needs more memory and computational complexity compared to the rates estimators method. Depending on the computational capacity of the SU, it may choose to implement either the rates estimator or the transition matrices estimator method.

## 4.6 Numeric illustrations

We make extensive numerical experimentations over important number of packets, in order to evaluate the performance of the proposed OSA mechanism, and validate the threshold structure of such policy. We consider 4 i.i.d licensed channels, i.e. $N = 4$ (with 4 licensed channels, we have approximately $10^6$ states). Furthermore, the system parameters are summarized in Table 4.1. We consider a model composed of four symmetric channels, and we simulate the system depicted in Figure 4.7. The three different scenarios studied in this chapter are illustrated in Table 4.2.

In this section, we describe the optimal threshold OSA policy, given perfect knowledge about the transition rates of licensed channels. We consider, first, the single channel case and then, we focus on the multichannel model. In the second part of this section, we present some results using estimated values of transition rates, and we compare the performance of the two proposed learning methods.

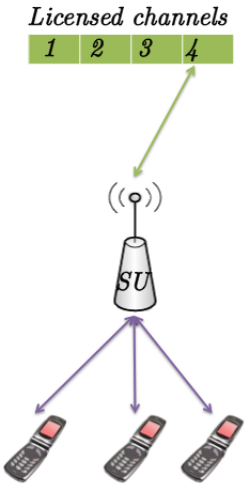


FIGURE 4.7: The simulation model.

### 4.6.1 Single channel model

In this section, we consider only one licensed channel having transition rates $\alpha = 0.15$ and $\beta = 0.1$. Figure 4.8 illustrates the optimal OSA policy of a SU depending on the belief and the packet delay. For each packet delay, the SU has a threshold policy depending on beliefs. We observe, in Figure 4.8, that the threshold belief probability $\lambda^*$ is decreasing with packet delays. Furthermore, the maximum packets delay is 13 time slots, regardless the belief vector.

Consider the same scenario with transition rates $\alpha = 0.7$ and $\beta = 0.85$. We observe, in Figure 4.9, that the optimal OSA policy of the SU has also a threshold structure. Furthermore, a packet has at most a delay of 3 time slots regardless its belief. Indeed, the SU always choose the dedicated channel for packets having a delay of 3 slots.

We have proved, in this section, that our numerical results validate our analytic finding. Indeed, the optimal OSA policy has a two-level threshold structure. We study in the next section the OSA mechanism in a multiple licensed channels context.

### 4.6.2 The multichannel model

In this section, we consider a model composed of 4 licensed channels. Note that when there are multiple licensed channels, SUs have to decide which one they have to sense and access. We implement, in our simulations, a natural greedy channel choice policy. In fact, we consider that the SU chooses the channel that has the highest belief.

We simulate the first scenario depicted in Table 4.2 and we illustrate, in Figure 4.10, the optimal OSA policy for SUs, depending on packet delays. For each packet delay $l$,

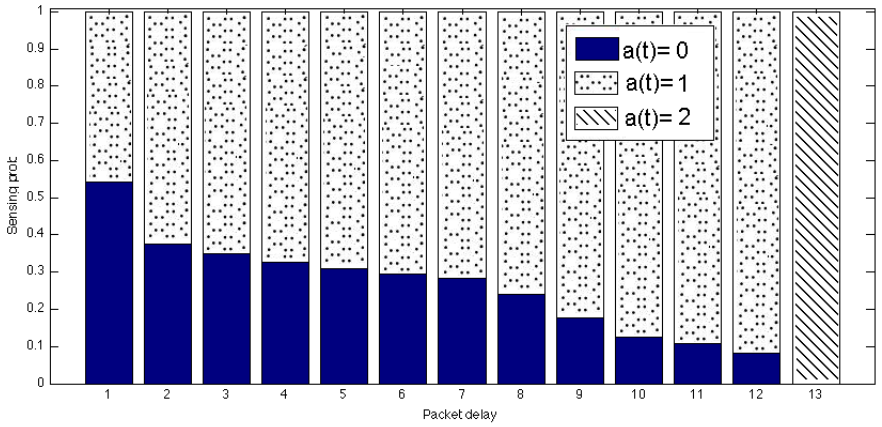


FIGURE 4.8: The optimal OSA policy with one licensed channel, with $\alpha = 0.15$ and $\beta = 0.1$.

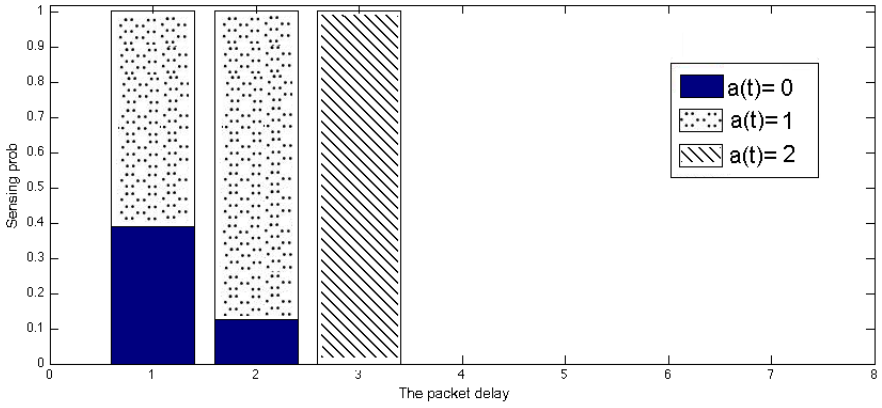


FIGURE 4.9: The optimal OSA policy with one licensed channel, with $\alpha = 0.7$ and $\beta = 0.85$.

the best action for the SU is to wait for the next slot if its belief probability is lower than $\lambda^*$. Otherwise, the SU decides to sense the licensed channels. We observe that the maximum packet delay $l^*$ equals 9. Then, when the packet delay is $l = 9$, the SU decides to sense and to transmit using the dedicated channel if the sensed channel is occupied (action 2). This observation validates the result of Proposition 4.11, as the choice of the action 2 depends only on the packet delay, regardless the belief vector.

We illustrate the optimal OSA policy for SUs obtained through simulating the second scenario of Table 4.2, in Figure 4.11. We observe that the SU chooses to transmit over the dedicated channel if there are no opportunities in the licensed spectrum, when the delay of the packet equals 5 slots, regardless its belief. Otherwise, it senses the licensed channels if its belief is higher than the threshold $\lambda^*$, and wait if its belief is lower. This result is intuitive as in this scenario, licensed channels are more often idle, inducing a

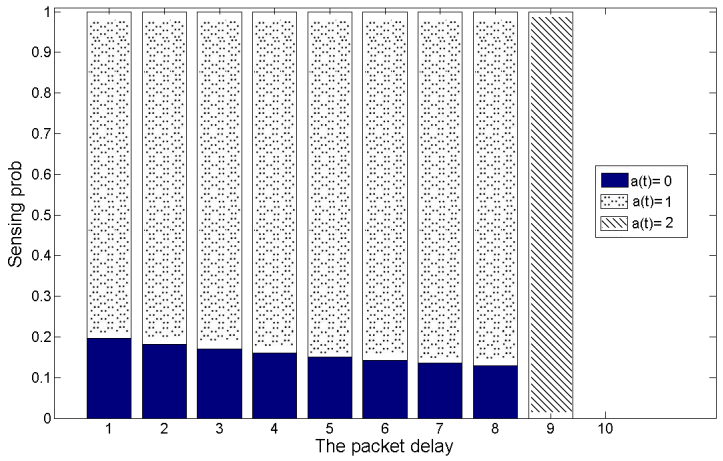


FIGURE 4.10: The optimal OSA policy in the scenario 1.

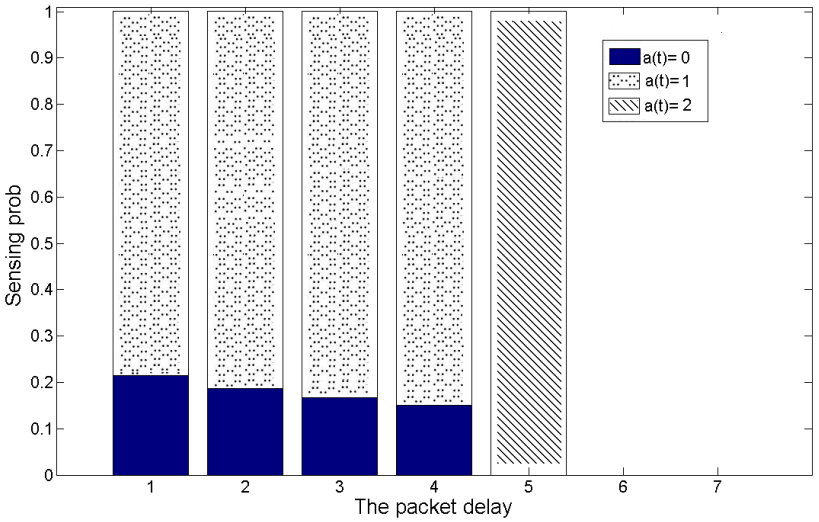


FIGURE 4.11: The optimal OSA policy in the scenario 2.

lower packet delay. Note that in both scenarios, the threshold belief $\lambda^*$ is decreasing with the packet delays.

Finally, we consider the last scenario depicted in Table 4.2. We observe, in Figure 4.12, that the maximum packet delay equals at most 5 time slots. We further observe that the OSA policy for SUs has also a threshold structure. However, the threshold belief probability $\lambda^*$ is not monotonous with packet delays. In fact, in this scenario, licensed channels are more static (the probability for each channel to stay occupied or idle is high enough). Thus, it appears one kind of periodic threshold strategy. One more observation, in this scenario, is that the SU changes the choice of the licensed channel to sense if it was sensed as occupied at the last time slot. Indeed, a channel sensed as occupied has a belief of $\beta$, the lowest possible belief, and will not be chosen at the next time slot, as we are using a greedy channel choice policy.

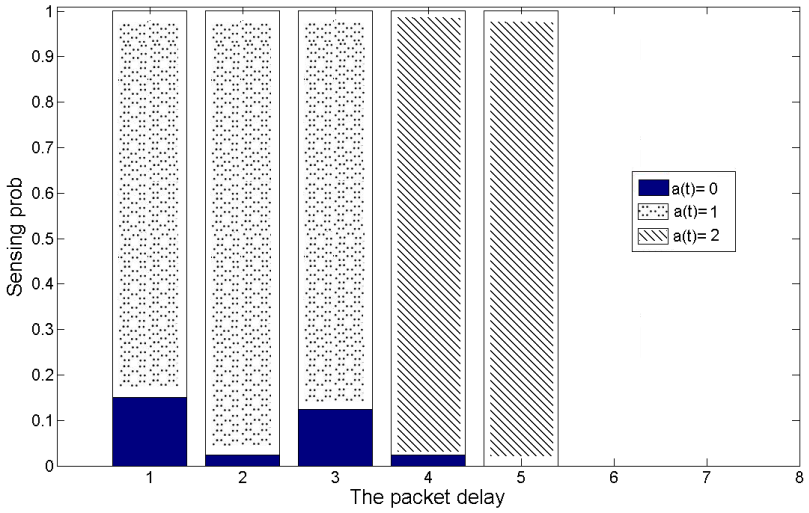


FIGURE 4.12: The optimal OSA policy in the scenario 3.

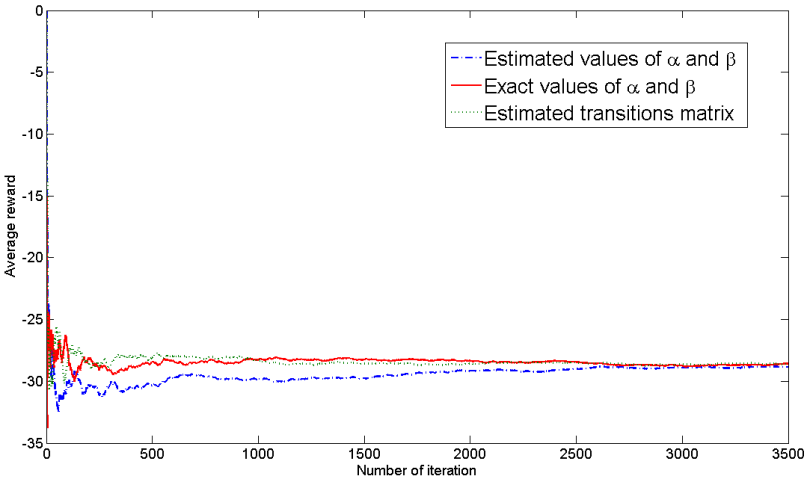


FIGURE 4.13: The average reward for scenario 1.

### 4.6.3 The multichannel model using estimated values of $\alpha$ and $\beta$

We simulate the three scenarios presented in Table 4.2, with estimated values of licensed channels' transition rates. Moreover, we consider both learning approaches presented in Section 4.5. In this section, we evaluate the performance of these learning methods using the two following metrics: The average reward and the average delay. We consider the model with known values of $\alpha$ and $\beta$ (studied in Section 4.6.2) as a reference model.

Figures 4.15 and 4.16 show that both learning protocols converge. In fact, we observe that both protocols converge before 400 iterations. However, in Figures 4.13 and 4.14, we can observe that the transition matrices estimation method converge 3 times faster (about 1000 iterations) than the rate estimators method (about 3000 iterations).

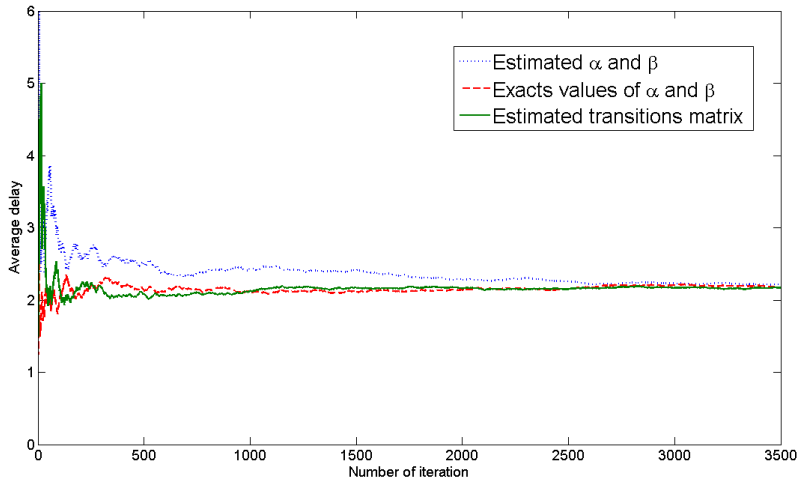


FIGURE 4.14: The average delay for scenario 1.

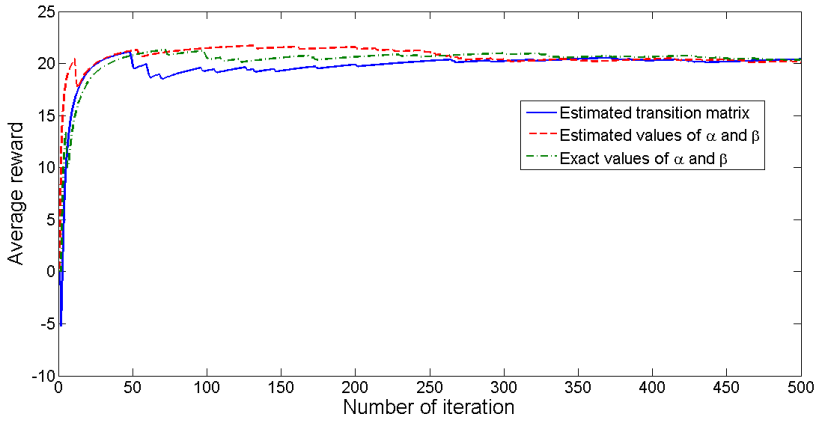


FIGURE 4.15: The average reward for scenario 2.

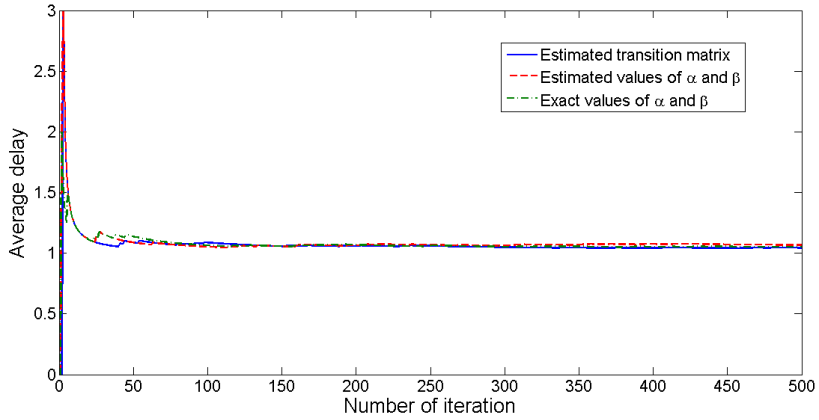


FIGURE 4.16: The average delay for scenario 2.

## 4.7 Conclusion

In this chapter, we have used a POMDP framework for designing an optimal OSA policy for CR networks taking into account an energy-delay tradeoff for SUs. Introducing a QoS metric in the OSA policy is very important, with the emergence of heterogeneous mobiles that are able to transmit their QoS-dependent traffic over different mediums of communication like 3G, WiFi and TV White Space. We have provided some structural properties of the value function and we have proved the existence of an optimal stationary OSA policy that has two-level threshold structure. We have been able to determine explicitly the threshold structure of the optimal policy.

Note that the interaction between several SUs has not been considered here, and in the literature very few, at the best of our knowledge. This perspective is also very important because if the channel choice policy is the same for all the SUs, there could have lots of collisions between several SUs that have sensed the same licensed channel. In the following chapter, we extend this study to the multichannel context. Indeed, we consider that SUs make decision individually and try to maximize their own benefits.

# Chapter 5

# Self-adaptive Spectrum Management in Partially Observable Environments

Contents



## 5.1 Introduction

Due to the recent and dramatic development of the wireless communication industry, the demand for wireless spectrum has been growing rapidly. Thus, the spectrum scarcity is becoming a challenge for several recent studies. Both academic and industry are recognizing that traditional fixed spectrum allocation is very inefficient, such that most of the time the bandwidth that was allocated is not optimally used and the corresponding channel is idle, which forms spectrum holes [8]. CR [1], which is a new paradigm for designing wireless communication systems, appeared in order to enhance the utilization of the radio frequency spectrum. It was considered as the key technology that enables SUs to access the licensed spectrum. Typically, SUs access opportunistically the spectrum when it is not used by PUs. The presence of several SUs in the same portion of spectrum

band enhanced the need to efficiently share the spectrum. Indeed, the utilization of the radio spectrum is reduced due to collisions among SUs under decentralized channel selection schemes. In order to optimize the utilization of the scarce spectrum resources, DSA become a promising approach to increase the efficiency of spectrum usage and to solve the scarcity problem.

Surprisingly, the impact of the energy constraint, due to the limited mobile users' battery, and the capacity of CR to support additional QoS were somehow ignored and not sufficiently studied in the literature. In many wireless systems, it is very important to provide reliable communications while sustaining a certain level of QoS. However, challenges in providing the QoS assurances increase due to the fact that SUs operate under constraints on the licensed channels' occupancy, and competition between each other.

We investigate an important problem for determining the OSA mechanism, and we propose a general model that allows us to study the impact of energy consumption and expected delay on the OSA policy. The main novelty of our approach is to consider a POSG framework. The theory of POMDP was widely and successfully used, like in [80], [53] and [90], to model and build OSA mechanisms in CR networks. However, those works do not consider the competition between SUs. Very few works proposed to model such competition(see [94] and [95] for example). Moreover, those works do not have significant results. In fact, using a DP approach to solve a POMDP is possible by transforming it into a completely observable MDP over belief states [95]. It is very difficult to generalize this technique for POSG as the SUs may have different beliefs. This problem was alleviated by introducing the notion of generalized belief state in [41], however the optimal algorithm becomes intractable beyond a small horizon. In our work, we focus on the existence of an SNE between SUs. The SNE is solved using a Linear Program (LP). Second, we identify paradoxical behaviors of SUs. One of the observed paradoxes here is a kind of Braess paradox, a well-studied paradox in routing context [96]. Our paradox indicates that decreasing the spectrum occupancy may lead degradation of the performance in term of the average throughput for SUs. This observation is due to the increase of the aggressiveness of SUs when the spectrum availability increases. We look further for a network control mechanism in order to optimize the average throughput of SUs at the SNE. For this end, we consider a Stackelberg game formulation [97].Note that Stackelberg game formulations was already proposed in the CR literature (see for example [39], [40] and [98]), as the natural hierarchy between PUs and SUs is very similar to the hierarchy between leaders and followers. Nevertheless, it was not used in order to enhance the network usage. In the second part of this chapter, we propose a control mechanism, for the network manager using a Stackelberg game formulation, such that the total average throughput of the SUs is maximized in this partially observable environment.

Many works focused on the study of optimal OSA policies in CR networks. In [80], the authors studied decentralized MAC protocols such that SUs search for spectrum opportunities without a central controller. They considered a POMDP and proposed an analytical framework based on this mathematical tool. However, the authors consider neither energy consumption nor any QoS constraint in their OSA policy. The problem of maximizing the throughput of traffic subject to some constraints on its delay received the extensive attention of pioneering work [99]. Authors of [100] described linear programming solvers for MDP, which are able to handle finite and infinite horizon problems. Moreover, authors of [101] considered a problem similar to ours but in a queueing context. They used the linear programming in order to solve an MDP and to study the equilibria for N players scenario in a stochastic game context. Few works focused on how SUs should operate in order to satisfy some QoS requirements and energy constraints. Authors of [53] incorporated the energy constraint in the design of the optimal OSA policy, in a single user context, and formulated their problem as a POMDP. The major difference between this work and ours is that the authors do not considered the competition between SUs. In [102], the authors presented a queueing analysis of a CR with multiple SUs. They proposed an adaptive algorithm to find the optimal contention probability that minimizes the expected delay. Authors of [103], proposed an energy-efficient non-cooperative strategy for resource allocation in CR networks based on a game theoretical approach. In summary, the main contributions of the chapter are as follows:

- We model a non-cooperative sensing and access game as a POSG. We prove the existence and uniqueness of an SNE for this OSA game.
- In the non-saturated regime, we exhibit an optimal sensing policy where SUs may sense licensed channels, even if they do not have any packets to transmit. Indeed, by sensing the licensed channels, a SU gets information on the RF environment.
- We highlight an interesting paradox, which says that increasing the spectrum occupancy may increase the SUs' average throughput. Indeed, SUs become less aggressive, which induces a better utilization of the spectrum holes (less collisions).
- Finally, we propose a control mechanism for the network manager in order to increase the average total throughput of the network at the SNE. For this purpose, we formulate the hierarchical framework as a Stackelberg game, where the network manager acts as the leader and SUs act as followers.

The remainder of the chapter is organized as follows. In Section 5.2, we introduce our system model. The utility function and the NE analysis are presented in Section 5.3. We

propose a Stackelberg-based mechanism for the network manager in order to optimize the licensed channels' utilization in Section 5.4. We present some simulation results in order to discuss the performance of the proposed model in Section 5.5, and we conclude the chapter in Section 5.6.

## 5.2 The model

We consider $M$ time-varying channels licensed for PUs and $N$ SUs accessing opportunistically the available channels. The occupancy of each channel $k \in \{1, \ldots, M\}$ is modeled by a time-homogeneous discrete Markov process denoted $s_k$, where the state $s_k = 0$ (resp. $s_k = 1$) means that the channel is idle (resp. busy). The licensed channels' transition rates are illustrated in Figure 5.1, where $\beta_k$ represents the probability that the licensed channel $k$ becomes idle, such that it was occupied in the previous time slot, and $\alpha_k$ represents the probability that the licensed channel $k$ becomes idle such that it was idle, in the previous time slot.

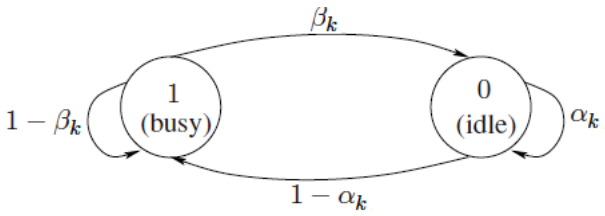


FIGURE 5.1: The discrete time Markov chain describing channel $k$ occupation state.

The global system state, at each time slot $t$, is composed of the states of the $M$ channels and is denoted by the vector $\mathbf{s}(t) = (s_1(t), ..., s_M(t))$. This global state is also called the Spectrum Occupancy State (SOS). The global state space is denoted by $\mathcal{S} = \{0, 1\}^M$.

We consider a slotted system, where SU access opportunistically the licensed channels when they are not used by PUs. Moreover, we consider a non-saturated regime such that the arrival of packets from upper layer to the transmission layer follows a Bernoulli process with parameter $q_a$. As long as the SU has a packet to transmit, a new packet is blocked and lost. The packet arrival processes for SUs are supposed to be independent and identically distributed. We further assume that a SU transmits, at most, one packet per time slot. Moreover, we consider an exclusive access to the licensed channels. In fact, when at least two SUs decide to transmit over the same channel, there is a collision and packets are lost (see Figure 5.2). This assumption is usual in CR networks problems related to the MAC layer (see [90] and [104]).

At each time slot $t$, we define the packet delay $l_i(t)$ for the SU $i$ as the number of elapsed time slot from the arrival of the packet into the transmission buffer until the time slot

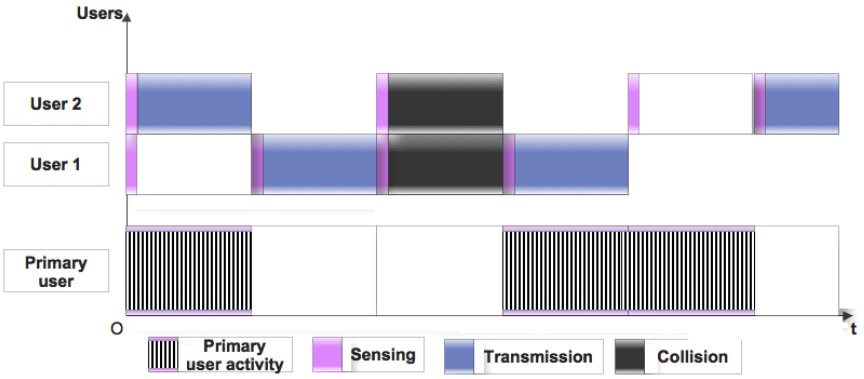


FIGURE 5.2: SUs transmissions

$t$. Therefore, $l_i(t) = 0$ means that the SU has no packet to transmit at the time slot $t$. At the beginning of each time slot, the SU $i$ has a perfect knowledge about the current packet delay $l_i(t)$, but ignores the SOS that can not be directly observed due to the partial spectrum sensing. Then, SUs have a partial observation of the global system state. Specifically, we study our problem using a POSG formulation.

A POSG is defined as a tuple $(\mathcal{N}, \mathcal{S}, b^0, \{\mathcal{A}_i\}, \{\mathcal{O}_i\}, \mathcal{P}, \{\mathcal{R}_i\})$, described as follows:

- $\mathcal{N}$ a finite set of SUs indexed $\{1, \ldots, N\}$,
- $\mathcal{S}$ a finite set of states, $|\mathcal{S}| = M$
- $b^0$ the initial state distribution,
- $\mathcal{A}_i$ the finite set of actions for SU $i$ (we define by $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_N$ the joint action set),
- $\mathcal{O}_i$ the finite set of observations for SU $i$ (we define by $\mathcal{O} = \mathcal{O}_1 \times \ldots \times \mathcal{O}_N$ the joint observation set),
- $\mathcal{P}$ a set of state transition and observation probabilities, i.e. $\mathcal{P}(s', o|s, a)$ is the probability that taking action $a$ in state $s$ results in observing $o$ and a transition to state $s'$,
- $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to I\!R$ the reward function for SU $i$.

**System state**: We denote the state of the users by $\mathbf{x}(t) = (x_1(t), \ldots, x_N(t))$, where $x_i(t) = (\lambda^i(t), l_i(t))$ represents the state of SU $i$, and $\mathbf{x_{-i}}(t)$ denotes the state of SUs other than $i$. Since the $M$ channels are independent, it was proved in [80] that we can consider the following simpler belief vector:

$$\lambda^i(t) = (\lambda^i_1(t), .., \lambda^i_M(t)),$$

where $\lambda_k^i(t)$ is the conditional probability for the SU $i$ that the channel $k$ is available at the time slot $t$. The state space of SU $i$ is referred to as $\mathcal{X}_i$, and $\mathcal{X} = \cup_i \mathcal{X}_i$ represents the set of all possible joint state of SUs.

**Belief**: Each SU senses at most one licensed channel in order to get information about the SOS. We denote by $\theta(t) = (\theta_1(t), \cdots, \theta_N(t))$ the set of observations of all the SUs, where $\theta_i(t) = 0$ means that the SU $i$ has sensed the licensed channel as idle. If $\theta_i(t) = 1$, then the licensed channel was sensed as occupied. The observation space is denoted by $\mathcal{O} = \{0, 1\}$. Each SU $i$ updates its belief vector $\lambda^i(t)$ based on its observation outcome $\theta_i(t)$. Define the observation probability $P_i(\theta_i(t) = \theta')$, the probability that the SU $i$ observes $\theta'$ at the time slot $t$. For each licensed channel $k$, the conditional probability $\lambda_k(t+1)$ depends not only on the observation of the SU, but also on its action. We denote by $\Omega(.|a_i(t), \theta_i(t))$ the update operator of the belief vector for each licensed channel.

**Actions and strategies**: Each SU has two actions to take sequentially, as illustrated in Figure 5.2. The first action, called *sensing-action*, is taken at the beginning of each time slot. This action determines whether the SU senses or not the licensed channels, based on the belief vector and the current packet delay. This sensing action induces an observation $\theta_i$. Then, the SU takes a second action, called *access-action*, which determines if it transmits its packet using the licensed channel or not. Certainly, this action has to be taken only if there are free licensed channels, and the SU has a packet to transmit. The joint action of all SUs is denoted by $\mathbf{a}(t) = (a_1(t), \cdots, a_N(t))$, where $a_i(t)$ denotes the action of SU $i$ and $\mathbf{a_{-i}}(t) = (a_1(t), \cdots, a_{i-1}, a_{i+1} \cdots, a_N(t))$ denotes the joint action set of SUs other than $i$. For notations convenience, we consider that the SU has only 3 possible actions:

- The action $a_i = 0$: the SU chooses to be inactive during the time slot. If the SU has a packet in its buffer, then the delay of the packet increases.
- The action $a_i = 1$: the SU chooses to sense licensed channels and not to transmit. Note that sensing licensed channels allows the SU to get more information that may improve the future rewards. If the SU has a packet in its buffer, then the delay of the packet increases.
- The action $a_i = 2$: the SU chooses to sense licensed channels and to transmit if idle. This action is possible only if the SU has a packet in its buffer.

Let us denote by $A_i(x_i)$ the action space of SU $i$, when it is in the state $x_i$, and by $A = \cup_i A_i$ the set of possible joint actions of SUs. Note that the action space for a SU depends on its state. For example, a SU that has no packet in its buffer ($l_i(t) = 0$)

cannot choose the action 2, i.e. $A_i = \{0, 1\}$. However, a SU having a packet to transmit chooses any action, i.e. $A_i = \{0, 1, 2\}$.

Based on the SU's action $a_i$ and its observation $\theta_i$, we have the following belief update, which comes from the Markov process. For all licensed channels $n \in \{1, \cdots, M\}$, the belief is updated as follows:

$$\lambda_n(t+1) := \Omega(\lambda_n(t)|a_i(t), \theta_i(t)) = \begin{cases} \beta_n + (\alpha_n - \beta_n)\lambda_n(t) & \text{if} \quad a_i(t) = 0; \\ \beta_n & \text{if} \quad a_i(t) \neq 0 \text{ and} \quad \theta_i(t) = 1; \\ \alpha_n & \text{if} \quad a_i(t) \neq 0 \text{ and} \quad \theta_i(t) = 0. \end{cases}$$

The strategy of SUs is defined by the probability of choosing a given action depending on its state $x_i(t) = (\lambda^i(t), l_i(t))$. We call a strategy for the SU $i$, a function $\mathbf{u_i}$ as a vector $[u_i(1), u_i(2), \ldots]$, where $u_i(t) : \mathcal{X}_i \times A_i \rightarrow [0, 1]$ is a mapping from a state $x_i(t)$ and an action $a_i(t)$ to a probability of taking the action $a_i(t)$ in the state $x_i(t)$. We denote by $\mathbf{u} := (u_1, \cdots, u_N)$ the multi-policy of all SUs (whose $i$th element is $u_i = [u_i(1), u_i(2), \ldots])$, and $\mathbf{u_{-i}}$ is the set of strategies of all SUs other than $i$. The set of all possible strategies is denoted $\mathcal{U}$.

**Instantaneous reward**: We denote by $c_s$ the energy spent for sensing and $c_t$ the energy spent for transmission. For each SU $i$, a natural definition of the instantaneous reward $r_i(t)$ is a composition of the throughput $\Phi$ and the energy costs. We introduce an additional cost, $f(l_i(t))$, in order to penalize the current packet delay. The instantaneous reward of a SU depends explicitly not only on its action $a_i(t)$, but also on the actions of all other SUs, denoted by $\mathbf{a_{-i}}(t)$. Furthermore, it depends on the state and the observation of SU $i$, $x_i$ and $\theta_i$. The instantaneous reward of the SU $i$ at the time slot $t$ is defined by:

$$r_i(x_i(t), \mathbf{a}(t), \theta_i(t)) = \begin{cases} \Phi - c_s - c_t, & \text{if } a_i(t) = 2,\ \theta_i(t) = 0 \text{ and } \forall j \neq i, a_j(t) \neq 2; \\ -c_s - c_t, & \text{if } a_i(t) = 2,\ \theta_i(t) = 0 \\ & \text{and } \exists j \neq i,\ a_j(t) = 2 \text{ (collision)}; \\ -f(l_i(t)) - c_s, & \text{if } a_i(t) = 1 \text{ or } a_i(t) = 2 \text{ and } \theta_i(t) = 1; \\ -f(l_i(t)), & \text{if } a_i(t) = 0. \end{cases} \tag{5.1}$$

where $\mathbf{a}(t) = [a_i(t)|\mathbf{a_{-i}}(t)]$, and $x_i(t) = (\lambda_i(t), l_i(t))$.

**Problem statement**: The objective of the SU $i$ is to maximize the average expected reward, given the initial condition $x_i(0) = x_0$. Usually, in OSA problems modeled using a POMDP formulation, the objective function is the expected total discounted reward like in [80], [105], [106] and [107]. In our context, we observe that decisions have to be taken frequently, at each time slot, which leads to a discount rate very close to 1

(see [86]). Thus, it is natural to consider policies on the basis of their average expected reward. Therefore, the SU $i$ seeks for the optimal strategy $u_i$ that maximize:

$$R_i(u_i, \mathbf{u_{-i}}) = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{\mathbf{u}} \left( \sum_{t=1}^{T} r_i(x_i(t), \mathbf{a}(t), \theta_i(t)) | x_0 \right). \tag{5.2}$$

We study the OSA problem in a non-cooperative setting, where each SU has its own state information and tries to maximize its average expected reward. Then, our problem will be studied in the following section through the concept of NE. Indeed, the SUs interact themselves through collisions when several SUs transmit over the same idle licensed channel. For simplicity reasons, and to get a deep theoretical analysis for the non-cooperative game between SUs, we consider only the set of stationary policies. A stationary policy is a mapping from a state $x_i$ and action $a_i$ to a probability $u_i(x_i, a_i)$, which does not depend on the time slot $t$. In the next section, we propose an analysis of the non-cooperative game. Our goal is to compute the set of all best responses strategies for a SU against a stationary multi-policy of all other SUs. Furthermore, we use a LP technique, which gives us a description of the NE for our non-cooperative game.

## 5.3 Nash equilibrium

In this section, we consider one licensed channel ($M = 1$), and $N$ SUs trying to access it. Note that SUs decide, solely, whether to access or not this licensed channel. Each SU looks for maximizing its average expected reward defined in Equation (5.2). Before analyzing the NE and its properties, we define, in the next section, the Best Response (BR) strategy, a standard concept in game theory (see [108]).

### 5.3.1 The best response function

In game theory, the best response is defined to be the strategy (or strategies) that produces the most favorable outcome for a player, given others' strategies. The concept of best response is central to John Nash's best-known contribution, the Nash equilibrium.

**Definition 5.1.** The best response strategy BR(.) is defined as follows:

$$\forall i \in \{1, \cdots, N\}, \quad BR_i(\mathbf{u_{-i}}) = arg \max_{u_i} R_i(u_i, \mathbf{u_{-i}}). \tag{5.3}$$

Note that the average expected reward function $R_i(u_i, \mathbf{u_{-i}})$ can be expressed as follows:

$$
\begin{aligned}
R_i(u_i, \mathbf{u_{-i}}) &= \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{a}\in A} \sum_{\theta'=0}^{1} \prod_{j\neq i} \pi_j^{u_j}(x_j) u_j(x_j, a_j) r_i(x_i, \mathbf{a}, \theta') \pi_i^{u_i}(x_i) u_i(x_i, a_i) P_i(\theta_i = \theta') \\
&= \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{a}\in A} \pi_i^{u_i}(x_i) u_i(x_i, a_i) \prod_{j\neq i} \sum_{\theta'=0}^{1} P_i(\theta_i = \theta') \pi_j^{u_j}(x_j) u_j(x_j, a_j) r_i(x_i, \mathbf{a}, \theta') \\
&= \sum_{x_i\in\mathcal{X}} \sum_{a_i\in A_i} \pi_i^{u_i}(x_i) u_i(x_i, a_i) \sum_{\mathbf{x_{-i}}} \sum_{\mathbf{a_{-i}}} \sum_{\theta'=0}^{1} \\
&\quad \prod_{j\neq i} P_i(\theta_i = \theta') \pi_j^{u_j}(x_j) u_j(x_j, a_j) r_i(x_i, \mathbf{a}, \theta'),
\end{aligned} \tag{5.4}
$$

where $\pi_i^{u_i}(x_i)$ is the stationary probability that the state of the SU $i$ is $x_i$, which depends on the strategy $u_i$ of the SU. The following lemma gives us a simpler expression of the average expected reward.

**Lemma 5.2.** *The average expected reward $R_i(u_i, \mathbf{u_{-i}})$ of the SU $i$ is expressed as follows:*

$$
\begin{aligned}
R_i(u_i, \mathbf{u_{-i}}) &= \sum_{x_i\in\mathcal{X}_i} \sum_{a_i=0}^{1} \pi_i^{u_i}(x_i) u_i(x_i, a_i) r_i(x_i, \mathbf{a}, \theta_i) + [\Phi(1 - \bar{P}_{tr}(\mathbf{u_{-i}}))\Pi(0) \\
&\quad -(1 - \Pi(0)) f(l_i) - c_s - \Pi(0) c_t] u_i(x_i, 2),
\end{aligned} \tag{5.5}
$$

*where $\Pi(0)$ is the stationary probability that the licensed channel is idle, and $\bar{P}_{tr}(\mathbf{u_{-i}})$ represents the probability that at least one SU $j \neq i$ transmits over the licensed channel during the current time slot.*

*Proof.* The average reward function, that a SU is trying to maximize, is expressed by:

$$
R_i(u_i, \mathbf{u_{-i}}) = \sum_{\mathbf{x}} \sum_{\mathbf{a}} \sum_{\theta'=0}^{1} P_i(\theta_i = \theta') \prod_{j\neq i} \pi_j^{u_j}(x_j) u_j(x_j, a_j) r_i(x_i, \mathbf{a}, \theta_i) \pi_i^{u_i}(x_i) u_i(x_i, a_i).
$$

Let us define the set $A^*_{-i} = \{a_{-i} | \exists j \neq i \; s.t. \; a_j = 2\}$. The expected reward can be expressed by:

$$
\begin{aligned}
R_i(u_i, u_{-i}) &= \sum_{x_i} \sum_{x_{-i}} \sum_{a_i=0}^{1} \sum_{a_{-i}} \sum_{\theta'=0}^{1} \prod_{j \neq i} P_i(\theta_i = \theta') \pi_j^{u_j}(x_j) u_j(x_j, a_j) r_i(x_i, \mathbf{a}, \theta_i) \pi_i^{u_i}(x_i) u_i(x_i, a_i) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i} \in A^*_{-i}} \prod_{j \neq i} P_i(\theta_i = 0) \pi_j^{u_j}(x_j) u_j(x_j, a_j) [-c_s - c_t] \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i} \in A^*_{-i}} \prod_{j \neq i} P_i(\theta_i = 1) \pi_j^{u_j}(x_j) u_j(x_j, a_j) [-c_s - f(l_i)] \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i} \in A/A^*_{-i}} \prod_{j \neq i} P_i(\theta_i = 0) \pi_j^{u_j}(x_j) u_j(x_j, a_j) [\Phi - c_s - c_t] \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i} \in A/A^*_{-i}} \prod_{j \neq i} P_i(\theta_i = 1) \pi_j^{u_j}(x_j) u_j(x_j, a_j) [-c_s - f(l_i)] \pi_i^{u_i}(x_i) u_i(x_i, 2)
\end{aligned}
$$

$$
\begin{aligned}
R_i(u_i, u_{-i}) &= \sum_{x_i} \sum_{a_i=0}^{1} \pi_i^{u_i}(x_i) u_i(x_i, a_i) r_i(x_i, \mathbf{a}, \theta_i) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i}} \prod_{j \neq i} P_i(\theta_i = 0) \pi_j^{u_j}(x_j) u_j(x_j, a_j) [-c_s - c_t] \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i}} \prod_{j \neq i} P_i(\theta_i = 1) \pi_j^{u_j}(x_j) u_j(x_j, a_j) [-c_s - f(l_i)] \pi_i^{u_i}(x_i) u_i(x_i, 2)
\end{aligned}
$$

$$
\begin{aligned}
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i} \in A/A^*_{-i}} \prod_{j \neq i} P_i(\theta_i = 0) \pi_j^{u_j}(x_j) u_j(x_j, a_j) \Phi \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
R_i(u_i, u_{-i}) &= \sum_{x_i} \sum_{a_i=0}^{1} \pi_i^{u_i}(x_i) u_i(x_i, a_i) r_i(x_i, \mathbf{a}, \theta_i) \\
&- \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i}} \prod_{j \neq i} \pi_j^{u_j}(x_j) u_j(x_j, a_j) c_s \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&- \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i}} \prod_{j \neq i} P_i(\theta_i = 0) \pi_j^{u_j}(x_j) u_j(x_j, a_j) c_t \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&- \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i}} \prod_{j \neq i} P_i(\theta_i = 1) \pi_j^{u_j}(x_j) u_j(x_j, a_j) f(l_i) \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&+ \sum_{x_i} \sum_{x_{-i}} \sum_{a_{-i} \in A/A^*_{-i}} \prod_{j \neq i} P_i(\theta_i = 0) \pi_j^{u_j}(x_j) u_j(x_j, a_j) \Phi \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
R_i(u_i, u_{-i}) &= \sum_{x_i} \sum_{a_i=0}^{1} \pi_i^{u_i}(x_i) u_i(x_i, a_i) r_i(x_i, \mathbf{a}, \theta_i(t)) - c_s \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&- \sum_{x_i} f(l)(1 - \Pi(0)) \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&- \sum_{x_i} \Pi(0) c_t \pi_i^{u_i}(x_i) u_i(x_i, 2) \\
&+ \sum_{x_i} \Phi(1 - \bar{P}^*) \Pi(0) \pi_i^{u_i}(x_i) u_i(x_i, 2).
\end{aligned}
$$

□

Note that $\bar{P}_{tr}(\mathbf{u_{-i}})$ can be expressed as follows:

$$\bar{P}_{tr}(\mathbf{u_{-i}}) = 1 - \prod_{j \neq i} \sum_{x_j \in \mathcal{X}_j} \sum_{a_j=0}^{1} \pi_j^{u_j}(x_j) u(x_j, a_j). \tag{5.6}$$

Note that the interaction between the SU $i$ and other SUs is summarized in the probability $\bar{P}_{tr}(\mathbf{u_{-i}})$. We are able now to define the expected instantaneous reward $\bar{r}_i$ for SU $i$ as follows:

$$\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) = \sum_{\theta'=0}^{1} I\!E_{\mathbf{u}}[r_i(x_i, \mathbf{a}, \theta_i)] P_i(\theta_i = \theta') \tag{5.7}$$

$$\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) = \begin{cases} (\Phi(1 - \bar{P}_{tr}(\mathbf{u_{-i}})) + f(l) - c_t)\Pi(0) - f(l) - c_s, & \text{if } a_i = 2, \\ -f(l_i) - c_s, & \text{if } a_i = 1, \\ -f(l_i), & \text{if } a_i = 0. \end{cases}$$

Note that $\bar{r}_i(x_i, a_i, \mathbf{u_{-i}})$ represents the instantaneous reward that the SU $i$ expect when taking the action $a_i$ in the state $x_i$, and the multi-policy of all other SUs is $\mathbf{u_{-i}}$. Thus, the average expected reward $R_i(u_i, \mathbf{u_{-i}})$, given by Lemma 5.2, can be rewritten as follows:

$$R_i(u_i, \mathbf{u_{-i}}) = \sum_{x_i} \sum_{a_i} \pi_i^{u_i}(x_i) u_i(x_i, a_i) \bar{r}_i(x_i, a_i, \mathbf{u_{-i}}). \tag{5.8}$$

The set of best response strategies for a SU, given fixed strategies for all other SUs, can be computed using a LP, as proposed in [100]. In the following, we present such a LP, which determines the set of all best response strategies for player $i$ against a stationary policy $\mathbf{u_{-i}}$ of all its opponents. We denote by $z_{i,u_i}(x_i, a_i) = \pi_i^{u_i}(x_i) u_i(x_i, a_i)$, the steady state probability that the system state of SU $i$ is $x_i \in \mathcal{X}$, and that the action $a_i \in \mathcal{A}_i$ is chosen. The following LP gives us the best response policies, for all SUs $i \in \{1, \cdots, N\}$, and for all multi-policy $\mathbf{u} \in \mathcal{U}$.

**LP(i,u):** Find $z^*_{i,u_i}(x_i, a_i)$, where $(x_i, a_i) \in \mathcal{X}_i \times \mathcal{A}_i$, that maximizes:

$$\sum_{x_i} \sum_{a_i} z_{i,u_i}(x_i, a_i) \bar{r}_i(x_i, a_i, \mathbf{u_{-i}}),$$

subject to:

$$\begin{aligned} \sum_{a_j} z_{i,u_i}(r, a_j) - \sum_{x_i} \sum_{a_i} z_{i,u_i}(x_i, a_i) p_{x_i a_i r} &= 0, \forall r \in \mathcal{X}, \\ \sum_{x_i} \sum_{a_i} z_{i,u_i}(x_i, a_i) &= 1, \\ z_{i,u_i}(x_i, a_i) &\geq 0, \end{aligned}$$

where $p_{xay}$ is the probability that the system switches from state $x$ to state $y$ by taking the action $a$.

Let $M_1(A)$ denote the set of probabilities measures over a set $A$, and let us define $\Gamma_i(\mathbf{u})$ as the set of optimal solutions of **LP(i,u)**. A point to set mapping $\gamma_i(z_i)$, given a non-negative real numbers $z_i = \{z_i(\mathbf{x}, \mathbf{a}), (x_i, a_i) \in \mathcal{X}_i \times \mathcal{A}_i\}$, is defined as follows:

- if $\sum_{a_i} z_i(x_i, a_i) \neq 0$ then $\gamma_i(x_i, a_i, z_i) := \{\frac{z_i(x_i,a_i)}{\sum_{a_i'} z_i(x_i,a_i')}\}$ is a singleton. Note that $\gamma_i(x_i, z_i) = \{\gamma_i(x_i, a_i, z_i) : a_i \in A_i(x_i)\}$ is a point in $M_1(A_i(x_i))$.
- Otherwise, $\gamma_i(x_i, z_i) := M_1(A_i(x_i))$, the set of all probabilities measures over $A_i(x_i)$.

Define $g_i(z_i)$ as the set of stationary policies that choose the action $a_i$ in the state $x_i$ with a probability in $\gamma_i(a_i, x_i, z_i)$. Moreover, we define the occupancy measures $f(x_0, \mathbf{u})$ for a multi-policy $\mathbf{u}$ as $\{f_i(x_0, \mathbf{u}), (a_i, x_i) \in \mathcal{X}_i \times \mathcal{A}_i, \forall i | f_i(x_0, \mathbf{u}) = \pi_i^{u_i}(x_i) u_i(x_i, a_i)\}$. Note that for each player $i$ and stationary policy $u_i$, the state of that player is an irreducible Markov chain with one ergodic class. Thus, a unique steady-state probability exists. Therefore, we can omit the initial state distribution $x_0$.

**Proposition 5.3.** *For any stationary multi-policy OSA for SUs, we have the following properties:*

1. *If $z^*_{i,\mathbf{u}}$ is an optimal solution of* **LP(i,u)**, *then any element $v \in g_i(z^*_{i,\mathbf{u}})$ is an optimal stationary response for SU $i$ against the stationary policy $\mathbf{u_{-i}}$ of other SUs. Moreover, the multi-policy $\mathbf{w} = [v|\mathbf{u_{-i}}]$ satisfies $f_i(\mathbf{w}) = z^*_{i,\mathbf{u}}$.*
2. *The optimal sets $\Gamma_i(\mathbf{u}), \forall i$ are convex, compact, and upper semi-continuous in $\mathbf{u_{-i}}$, where $\mathbf{u}$ is identified with points in $\prod_{i=1}^{N} \prod_{x_i} M_1(A_i(x_i))$.*
3. *For all $i$, $g_i(z_i)$ is upper semi-continuous in $z$ over the set of solutions for* **LP(i,u)**.

*Proof.* The proof of (1) follows from Theorem 2.6 of [109]. The first part of (2) is a direct result of the LP. However, the second part follows by applying the theory of sensitivity analysis of LP by Dantzig et al. [110] in the Theorem 3.6 of [111] to **LP(i,u)**. The last property follows from the definition of $g_i(z_i)$. □

### 5.3.2 The Nash equilibrium

We model the interaction between SUs as a non-cooperative game. Let us define the concept of NE between SUs in our model.

**Definition 5.4.** The NE is defined as a set of strategies (one for each player) $\mathbf{u}^* = (u_1^*, u_2^*, ..., u_N^*)$, such that:

$$\forall i \in \{1, \ldots, N\}, \quad u_i^* = \arg\max_{u_i} R_i(u_i, \mathbf{u}_{-\mathbf{i}}^*). \tag{5.9}$$

A successful transmission for a SU over the licensed channel depends not only on the PUs' activity but also on the competition with other SUs. When a SU senses the channel as idle, it transmits successfully its packet if and only if the action of all other SUs is not to transmit on the licensed channel during the current slot. Indeed, a SU that chooses an action $a \in \{0, 1\}$ does not impact the instantaneous reward of other SUs. Given this remark, we have the following theorem, which states the existence of a NE multi-policy for our OSA problem between SUs.

**Theorem 5.5.** *There exists a stationary multi-policy* $\mathbf{u}^*$ *that is a Nash equilibrium.*

*Proof.* Consider a fixed value of the stationary probability that the channel is idle, $\Pi(0)$. Note that for each SU $i$ and any stationary policy $u_i$, the state process of that SU is an irreducible Markov chain with one ergodic class. Moreover, the strategies chosen by any SU does not depend on the cost realization. Otherwise, a SU could use the costs to estimate the state and actions of other SUs. Thus, from the Theorem of fixed point of Kakutani, a fixed point $u_i \in BR(\mathbf{u}_{-\mathbf{i}})$ exists. Proposition 5.3 implies that the stationary multi-policy $g = \{g_i(z_i) \forall i\}$ is a NE. □

After proving the existence of a NE of our game, the second problem we address now is to determine a particular type of equilibrium: the Symmetric Nash Equilibrium. A symmetric multi-policy $\mathbf{u}^* = (u^*, u^*, \cdots, u^*)$ is an SNE if and only if:

$$R_i(u^*, \mathbf{u}_{-\mathbf{i}}^*) \geq R_i(u_i, \mathbf{u}_{-\mathbf{i}}^*), \ \forall i \text{ and } \forall u_i \neq u^*. \tag{5.10}$$

In order to find an SNE, we assume that $N-1$ SUs use a strategy $u_0$, and a tagged SU (without loss of generality, the user $N$) uses the strategy $u_N$. Therefore, a multi-policy $\mathbf{u} = (u_0, \cdots, u_0, u_N) := (\mathbf{u}_{-\mathbf{N}}, u_N)$ is an SNE if and only if:

$$u_N = u_0 \in BR(\mathbf{u}_{-\mathbf{N}}). \tag{5.11}$$

### 5.3.3 Properties of the Nash equilibrium

Let us define by $P_{tr}(u_i)$ the attempt rate for a SU $i$. $P_{tr}(u_i)$ is expressed as follows:

$$P_{tr}(u_i) = \sum_{x_i' \in \mathcal{X}} \pi_i^{u_i}(x_i') u_i(x_i', 2), \tag{5.12}$$

where $\pi_i^{u_i}(x_i)$ is the stationary probability that the state of the SU $i$ is $x_i$, and $u_i$ is the mixed strategy of the SU $i$. The following proposition states that for each SU $i$, its attempt rate is always the same at different SNE of the game.

**Proposition 5.6.** *Consider two SNE* $\mathbf{u_1^*} \neq \mathbf{u_2^*}$*, such that* $\mathbf{u_1^*} = (u_1^*, \cdots, u_1^*)$ *and* $\mathbf{u_2^*} = (u_2^*, \cdots, u_2^*)$*. Therefore, the attempt rates for any SU* $i$ *at the SNE are unique and equal:*

$$\forall i \in \{1, \cdots N\}, \quad P_{tr}(u_1^*) = P_{tr}(u_2^*) := P^*.$$

*Proof.* Consider $z_0^*$ the solution of the LP that maximizes $\bar{r}_i(x_i, a_i, \mathbf{u_{-i}})$, and $z_\epsilon^*$ the solution of the LP that maximizes $\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) + \epsilon \mathbb{1}_{a_i=2}$. Note that, in the second problem, the reward for the action 2 is increased, compared to the first one. Assume that $\sum_{x_i} z_0^*(2, x_i) > \sum_{x_i} z_\epsilon^*(2, x_i)$, then we obtain:

$$\begin{aligned}
&\sum_{x_i}\sum_{a_i} z_\epsilon^*(a_i, x_i)\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) + \epsilon \sum_{x_i} z_\epsilon^*(2, x_i), \\
\leq &\sum_{x_i}\sum_{a_i} z_0^*(a_i, x_i)\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) + \epsilon \sum_{x_i} z_\epsilon^*(2, x_i), \\
< &\sum_{x_i}\sum_{a_i} z_0^*(a_i, x_i)\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) + \epsilon \sum_{x_i} z_0^*(2, x_i).
\end{aligned} \tag{5.13}$$

Therefore, $z_0^*$ is the optimal solution that maximizes $\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) + \epsilon \mathbb{1}_{a_i=2}$, which leads to a contradiction as $z_\epsilon^*$ is assumed to be the optimal solution of $\bar{r}_i(x_i, a_i, \mathbf{u_{-i}}) + \epsilon \mathbb{1}_{a_i=2}$. The first inequality is because $z_0^*$ maximizes $\bar{r}_i(x_i, a_i, \mathbf{u_{-i}})$, and the second one is due to the assumption. Then, we obtain that $\sum_{x_i} z_0^*(x_i, 2) \leq \sum_{x_i} z_\epsilon^*(x_i, 2)$.

Note that the attempt rate of the SU $i$ is expressed by $P_{tr}(u_i)$, and the attempt rate of other SUs is expressed by $\bar{P}_{tr}(\mathbf{u_{-i}})$. Therefore, if the attempt rate of other SUs decreases, the reward $\bar{r}_i(x_i, 2)$ increases and then the attempt rate $P_{tr}(u_i)$ increases. In fact, a SU decreases its attempt rate if all the other SUs increase their attempt rates. Finally, the BR function of SU $i$ decreases with the attempt rate of other users $\bar{P}_{tr}(\mathbf{u_{-i}})$.

Since we are considering SNE strategies, we have $P_{tr}(u_i) = \bar{P}_{tr}(\mathbf{u_{-i}})$. Suppose that there are two Nash equilibrium strategies, $\mathbf{u^1}$ and $\mathbf{u^2}$ having different attempt rates,

$P_{tr}(u^1) < P_{tr}(u^2)$. As both $\mathbf{u^1}$ and $\mathbf{u^2}$ are SNE, we have the following inequality:

$$P_{tr}(BR_i(\mathbf{u^1_{-i}})) = P_{tr}(u^1) < P_{tr}(u^2) = P_{tr}(BR_i(\mathbf{u^2_{-i}})), \tag{5.14}$$

which lead to a contradiction, as $BR_i(.)$ is a decreasing function with respect to the attempt rate. □

We denote by $P^*$ the attempt rate of a SU when all SUs use a SNE strategy. As usual in non-cooperative games, the utilization of the resource is suboptimal at the NE. In the following section, we look for a network manager's control mechanism in order to optimize an important global metric of the system, the average total throughput.

## 5.4 Network management

The SNE between SUs has been deeply investigated using a LP technique in the previous section. Note that interactions between SUs induce collisions. Henceforward, we focus on the impact of the PUs' activity on the performance of the global system. Since the resource utilization at the SNE is generally suboptimal, we propose to introduce some control in order to enhance the spectrum utilization. We propose a simple mechanism by introducing some kind of hierarchy in the OSA game. We obtain this hierarchy by introducing a controller, named the network manager. This controller plays as a leader in the Stackelberg game, and the SUs play as followers.

We formulate the problem of maximizing the average total throughput of the system as a Stackelberg game. The objective of the network manager is to maximize the average total throughput of the system at the SNE. Note that the average total throughput of the system is defined as follows:

$$U^* = \frac{1}{N}\sum_{i=1}^{N} P_{tr}(u_i^*)\prod_{j\neq i}(1 - P_{tr}(u_j^*)).$$

From Proposition 5.6 , the attempt rates at the SNE of all SUs are equals. Thus, we obtain:

$$U^* = P^*(1-P^*)^{N-1}.$$

The following proposition gives us the attempt rate at the SNE that maximizes the average total throughput of the system.

**Proposition 5.7.** *When the attempt rate at the SNE, $P^*$, is equal to $\frac{1}{N}$, the average total throughput $U^*$ is maximized.*

*Proof.* As we have N users transmitting over the same licensed channel, with an average probability of $P$, we have a successful transmission, if the channel is idle, with probability $P(1-P)^{N-1}$. The probability $P^*$ maximizes $P(1-P)^{N-1}$ if and only if $(1-P^*)^{N-1} - P^*(N-1)(1-P^*)^{N-2} = 0$, then $(1-NP^*)(1-P^*)^{N-2} = 0$. Therefore, when $P^* = \frac{1}{N}$, the utility for SUs is optimal. □

Note that the attempt rate $P^*$ obtained from a multi-policy SNE, given by Theorem 5.5, does not necessarily equal the optimal attempt rate obtained from Proposition 5.7. Then, the network manager makes a decision (an intervention) in order to influence the SNE multi-policy.

The question that we have to answer is how the network manager can impact SUs' policies in order to maximize the average total throughput of the system at the SNE. Before, we state, in the following proposition, some properties of the attempt rate and the channel occupancy. The following proposition shows that increasing the channel occupancy decreases the attempt rate of SUs at the SNE.

**Proposition 5.8.** *$P^*$ is decreasing when $\Pi(0)$ decreases.*

*Proof.* Consider two stationary probabilities that the channel is idle $\Pi_1(0)$ and $\Pi_2(0)$, such that $\Pi_1(0) < \Pi_2(0)$. Consider two SNE strategies, $\mathbf{u^1}$ obtained with the stationary probability $\Pi_1(0)$, and $\mathbf{u^2}$ obtained with the stationary probability $\Pi_2(0)$. Note that, for a given value of attempt rate $P^*$, the immediate reward for the action $a_i = 2$ is higher for the channel having a stationary probability of $\Pi_2(0)$ than for the channel having a stationary probability of $\Pi_1(0)$ (see Equation (5.8)). Let us denote by $P_{tr}(u^1)$ the attempt rate obtained with strategy $\mathbf{u^1}$, and by $P_{tr}(u^2)$ the attempt rate obtained with strategy $\mathbf{u^2}$. We obtain from Proposition 5.6 that $P_{tr}(u^1) < P_{tr}(u^2)$ (decreasing $\Pi(0)$ decreases the instantaneous reward for the action $a_i = 2$).

Finally, we obtain that the attempt rate $P^*$ decreases when the stationary probability that the licensed channel is idle decreases. □

We have the following relationship between $\Pi(0)$ and $\beta_0$.

**Lemma 5.9.** *$\Pi(0)$ is increasing with $\beta_0$.*

*Proof.* The stationary probability $\Pi(0)$ is defined as follows:

$$\Pi(0) = \frac{\beta_0}{1-\alpha+\beta_0}.$$

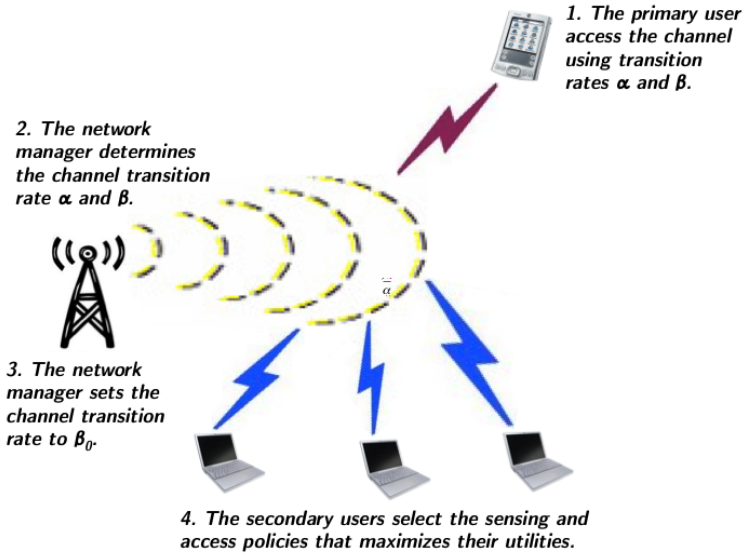


FIGURE 5.3: The Stackelberg game model of the SU throughput maximization.

The derivative of $\Pi(0)$ with respect to $\beta_0$ is:

$$\frac{\partial \Pi(0)}{\partial \beta_0} = \frac{1-\alpha}{(1-\alpha+\beta_0)^2}.$$

As $\alpha \in [0,1]$, then the derivative of $\Pi(0)$ with respect to $\beta_0$ is always positive. Therefore $\Pi(0)$ is increasing with $\beta_0$. □

Given this result, the network manager varies the channel occupancy state in order to maximize the average total throughput of SUs at the SNE. Figure 5.3 depicts the relationships between PUs, the network manager and SUs.

Moreover, the stationary probability that the licensed channel is idle is given by $\Pi(0) = \frac{\beta}{1-\alpha+\beta}$. It is obvious that the stationary probability $\Pi(0)$ is increasing with $\beta$. Thus, by reducing $\beta$, the network manager can reach a target value of stationary probability $\Pi(0)$ that maximizes the average total throughput of SUs at the SNE. We denote by $\beta_0$ the transition rate that maximizes the average total throughput of SUs at the SNE.

*Remark* 5.10. Note that if $P^* > \frac{1}{N}$, then $\beta_0 < \beta$, and the network manager increases the channel occupancy in order to maximizes the average total throughput of SUs at the SNE. However, if $P^* < \frac{1}{N}$, then the target value $\beta_0$ that maximizes the average total throughput at the SNE is above the PUs' transmission rate, i.e. $\beta_0 > \beta$. Therefore, the network manager cannot improve the performance of the system. Indeed, the network manager can only decrease the transition rate from state occupied to idle, by occupying the licensed channel after it was already occupied. Figure 5.4 illustrates the impact of the transition rate $\beta_0$ on the attempt rate when using an SNE policy.

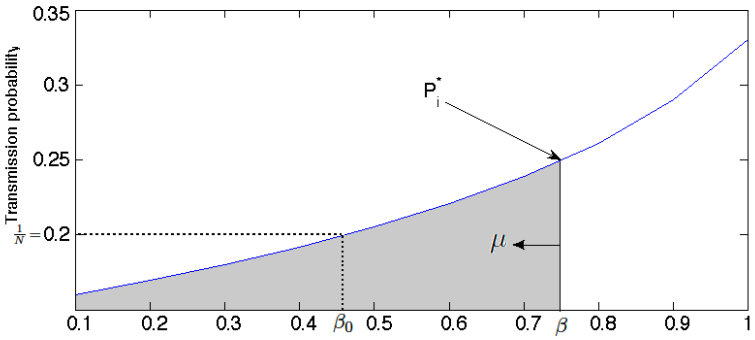


FIGURE 5.4: The attempt rate when using a SNE policy with respect to the transition rate $\beta_0$.

Let us define the network manager's (leader) actions by:

- $a_1^p$: the network manager occupies the licensed channel if this channel was already occupied in the previous slot and becomes idle in the current slot;
- $a_2^p$: the network manager does not occupy the channel if this channel was occupied in the previous slot and becomes idle in the current slot.

In fact, when the leader chooses the action $a_1^p$, the licensed channel is not used by PUs but appears occupied for the followers (SUs). Then, the leader's action impacts the SNE of the followers. The set of the leader's actions is denoted $\mathcal{A}_l = \{a_1^p, a_2^p\}$. We define a mixed strategy of the leader by a mapping $\mu : \mathcal{A}_l \to [0, 1]$, where $\mu(a)$ is the probability that the leader takes the action $a$. Note that we have $\mu(a_2^p) = 1 - \mu(a_1^p)$. Given a strategy $\mu$ of the network manager, the induced transition rate $\beta'$ is:

$$\beta'(\mu) \quad = \quad (1 - \mu(a_1^p)) \times \beta, \tag{5.15}$$

where $\beta$ is the transition rate of PUs. Denote by $\mathbf{u}^*(\mu)$ the SNE of the followers when the leader's strategy is $\mu$. In fact, the action of the leader $\mu$ changes the transition rate from $\beta$ to $\beta'(\mu)$, which impacts the SNE of the followers. The objective of the leader (network manager) is therefore to find a strategy $\mu$ that maximizes the average throughput of the system:

$$\bar{U}(\mu, \mathbf{u}^*(\mu)) = \frac{1}{N} \sum_{i=1}^{N} Thr_i(\mathbf{u}^*(\mu)) = P^*(\mathbf{u}^*(\mu))(1 - P^*(\mathbf{u}^*(\mu)))^{N-1}. \tag{5.16}$$

The network manager problem can be expressed as follows:

$$\mu^* = \arg\max_{\mu} U(\mu, \mathbf{u}^*(\mu)), \tag{5.17}$$

where $\mathbf{u}^*(\mu)$ is an SNE among the $N$ SUs taking into account the strategy of the leader. The vector of actions $(\mu^*, \mathbf{u}^*(\mu^*))$ is by definition a Stackelberg equilibrium [108], and we have the following theorem, which proves the existence of such equilibrium.

**Theorem 5.11.** *There exists a Stackelberg equilibrium for our hierarchical game with a network manager and $N$ SUs.*

*Proof.* We have proved, in Proposition 5.7, that the attempt rate at the SNE $P^*$, which maximizes the leader's utility should be equal to $P^* = \frac{1}{N}$, where $N$ is the number of SUs. Moreover, we have proved, in Proposition 5.8, that $P^*$ decreases when $\Pi(0)$ decreases, and that $\Pi(0)$ is increasing with $\beta$. Thus, the leader computes the value of $\beta' = \min\{\beta_0, \beta\}$, and uses the following strategy:

$$\mu(a_1^p) = 1 - \frac{\beta'}{\beta}, \text{ and } \mu(a_2^p) = \frac{\beta'}{\beta}.$$

Note that SUs converge to an SNE where every SU maximizes its own utility taking into account the new channel transition rates $(\alpha, \beta')$. Therefore, there exists a Stackelberg equilibrium between the network manager and SUs. □

## 5.5 Numerical illustrations

We illustrate, in this section, some Matlab-based simulation results in both saturated ($q_a = 1$) and non-saturated regimes ($q_a < 1$). We consider five SUs ($N = 5$) transmitting opportunistically, and we assume that the deadline delay is 3 slots. The deadline delay is the time by which the packet must be transmitted. We set the transmission cost $c_t = 100$; the sensing cost $c_s = 5$ and the throughput $\Phi = 200$kbit/s. Moreover, we consider a delay penalty function $f(l) = \min\{l, l_{max}\}$, where $l_{max}$ is the deadline delay.

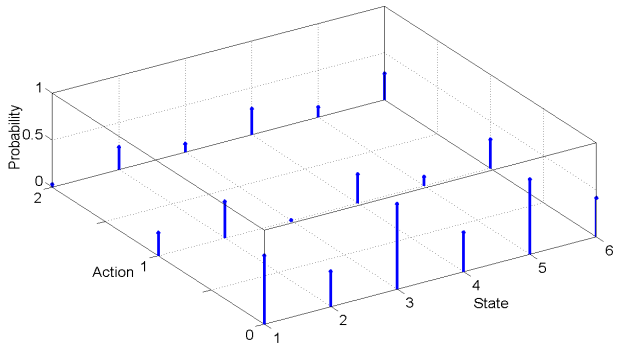


FIGURE 5.5: The equilibrium policy in the saturated case with $\alpha = 0.1$, $\beta = 0.9$ and $c_t = 100$.

### 5.5.1 Symmetric Nash equilibrium

Consider, first, the saturated regime, where SUs have always packets to transmit. Therefore, we obtain the following set of states:

| State index | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $l$ | 1 | 1 | 2 | 2 | 2 | 2 |
| $\lambda$ | $\alpha$ | $\beta$ | $\alpha$ | $\beta$ | $\Omega(\alpha)$ | $\Omega(\beta)$ |

We can observe, in Figure 5.5 obtained with $\alpha = 0.1$ and $\beta = 0.9$, that a SU chooses a mixed strategy composed of the three possible actions: sleeping; sensing; sensing and transmitting. Moreover, when the transmission cost increases $c_t = 500$, we observe, in Figure 5.6, that SUs have less incentive to sense and transmit.

Secondly, we focus on the non-saturated regime with $q_a = 0.85$. When a SU transmits a packet, its local state $l$ becomes 1 if it receives a new packet at the time slot $t$ (with probability $q_a$), otherwise $l = 0$. Therefore, we obtain the following set of states:

| State index | 1 | 2 | 3 | 4 | 5 | 6 | ... | 18 |
|---|---|---|---|---|---|---|---|---|
| $l$ | 0 | 0 | 0 | 0 | 0 | 0 | ... | 2 |
| $\lambda$ | $\alpha$ | $\beta$ | $\Omega(\alpha)$ | $\Omega(\beta)$ | $\Omega^2(\alpha)$ | $\Omega^2(\beta)$ | ... | $\Omega^2(\beta)$ |

Consider $\alpha = 0.9$ and $\beta = 0.1$, a scenario where the licensed channel stays in the same state during long periods, as it is the case with TV white bands [78]. We plot, in Figure 5.7, the multi-policy SNE obtained after solving the LP. We observe that the probability of sensing when the SU has no packet to transmit, i.e. $a_i = 1$, is increasing with the number of consecutive time slots the SU have not sensed the licensed channel. It means that the SU tries to get information about licensed channels by sensing even if it has no packet to transmit.

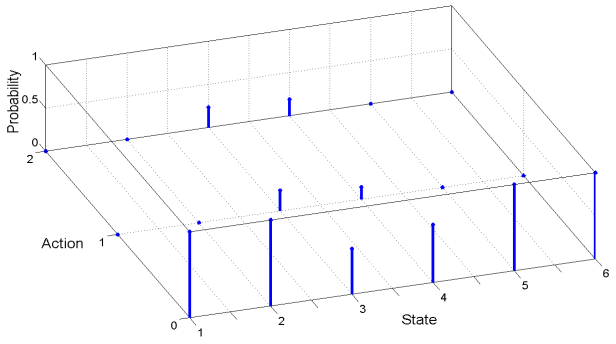


FIGURE 5.6: The equilibrium policy in the saturated case with $\alpha = 0.1$, $\beta = 0.9$ and $c_t = 500$.

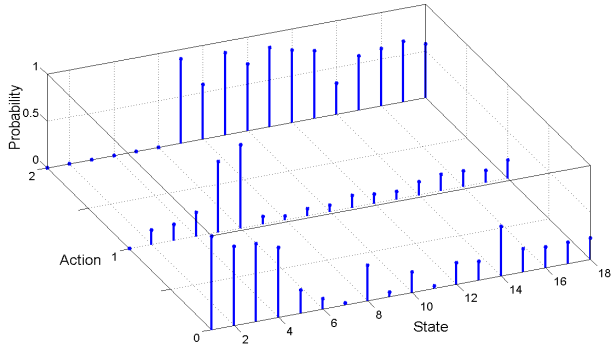


FIGURE 5.7: The equilibrium policy in the non saturated case with $\alpha = 0.9$, $\beta = 0.1$ and $q_a = 0.85$.

### 5.5.2 Braess paradox

Figure 5.8 illustrates the attempt rate $P^*$ depending on the number of SUs. We observe that the attempt rate at the SNE is decreasing with the number of SUs, which is somehow intuitive, as the collision probability $1 - P^*(1 - P^*)^{N-1}$ increases due to the competition between SUs. In Figure 5.9, a Braess kind of paradox is illustrated. Indeed, there is a degradation of the performance of the system when additional resource is added. Specifically, we have an opposite formulation, saying that reducing system resources induce better performances. When the average spectrum occupancy (stationary probability that the licensed channel is occupied, i.e. $\frac{1-\alpha}{1-\alpha+\beta}$) is less than 0.5, the average throughput of the system increases with the average occupation of the channel.

In order to understand this phenomenon, we study the impact of the average channel occupancy on the average total throughput of the system. The SUs' attempt rate is decreasing when the channel is less available. Surprisingly, the average throughput is not always increasing with the offered channel opportunities. In fact, we observe, in Figure 5.9, that when the channel is available more than 50% of time, the average SUs' throughput is decreasing when the licensed channel is idler. The attempt rate is $P = \frac{1}{5}$

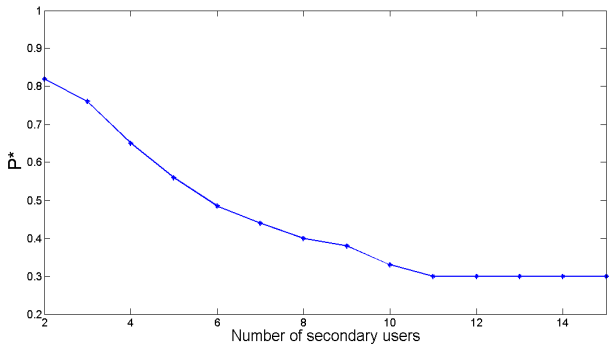


FIGURE 5.8: The attempt rate at the SNE depending on the number of SUs for $\alpha = 0.95$ and $\beta = 0.9$.

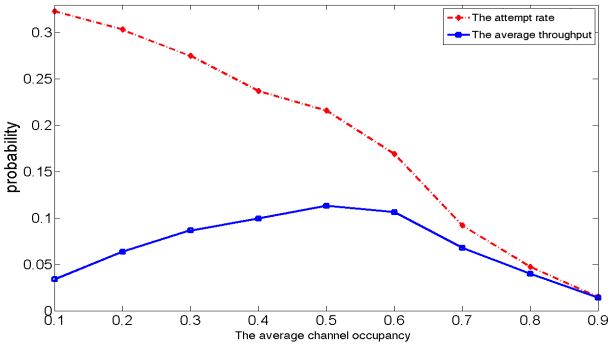


FIGURE 5.9: The attempt rate and the average throughput with the channel occupancy for $c_t = 100$.

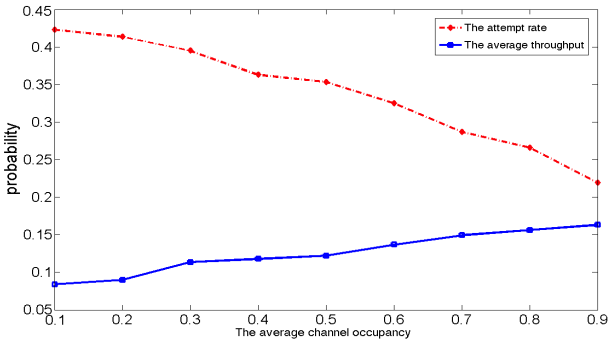


FIGURE 5.10: The attempt rate and the average throughput with the channel occupancy for $c_t = 900$.

when the channel availability is 0.5, and the average throughput is maximal for this channel availability. Note that it has been already proved that the SUs' attempt rate, that maximizes the average total throughout is $\frac{1}{N}$, where $N$ is the number of SUs. In Figure 5.10, there is another example in which the average throughput is always increasing with the average channel occupancy.

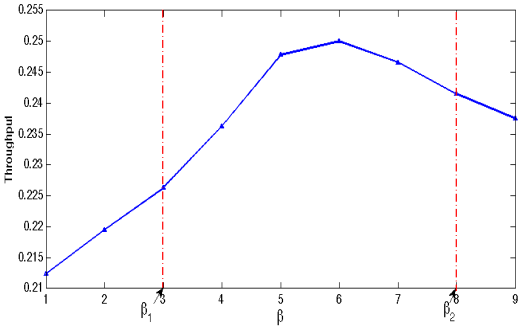


FIGURE 5.11: The average throughput depending on $\beta$.

### 5.5.3 Stackelberg equilibrium

Let us consider a scenario where two SUs are competing in order to access a licensed channel. The PUs' transition rate $\alpha$ is set to 0.1. We consider, first, that $\beta = 0.8$, and we illustrate, in Figure 5.11, the average throughput of the SUs depending on the transition rate $\beta$. We observe that the optimal value of $\beta_0$, which is also the transition rate at the Stackelberg equilibrium, is equal to 0.6. Therefore, the network manager has to decrease the transition rate from the occupied state to the idle state (i.e. $\beta$) from 0.8 to 0.6, which increases the average throughput of SUs from 0.2415 to 0.25.

Secondly, we consider that the PUs' requirement is $\beta = 0.3$. Thus, the network manager has to increase $\beta_0$ in order to increase the average throughput of SUs, which require that PUs use less the licensed channel. However, as we have already assumed that the SUs' access is opportunistic, PUs are unaware of the presence of SUs, and the network manager cannot increase $\beta_0$. Thus, the optimal action of the network manager is to be inactive ($\beta_0 = \beta$), as it cannot improve the actual SUs' performance.

Finally, Figure 5.12 illustrates the average channel availability ($\Pi(0)$) that maximizes the throughput for SUs at the SNE. We considered that PUs occupy the licensed channel with a probability $\Pi(1) = 0.5$. Then, when the cost is higher than 100, there is no paradox, as we cannot increase the channel availability (the network manager has to increase $\Pi(0)$).

## 5.6 Conclusion

In this chapter, we have set up a non-cooperative OSA mechanism for CR networks, and we have considered that SUs are in competition in order to access a licensed channel. Both the saturated and the non-saturated regimes have been studied, and we have

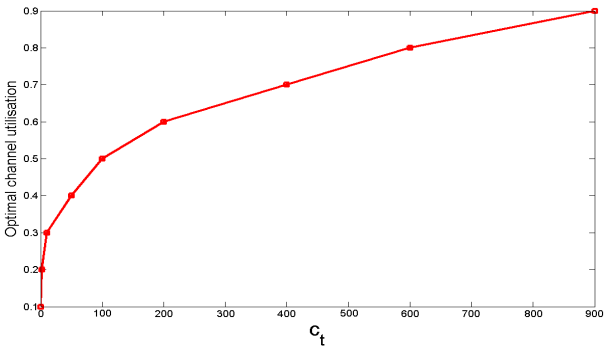


FIGURE 5.12: The optimal channel utilization with the transmission cost.

proved the existence of an SNE multi-policy for the OSA problem, modeled as a non-cooperative game between SUs. Moreover, we have proved that the attempt rate at the SNE is unique. The impact of both the arrival rate and the transmission cost on the system performances has been deeply studied. Simulation results have shown that more opportunities of transmission may decreases the average throughput of the system due to the aggressiveness and the competition between SUs. In fact, we have found Braess paradox where reducing system resources induce better performance. In order to optimize the average throughput of the system, we have proposed a Stackelberg game model for the network manager. We have proved the existence of an optimal strategy for the network manager. This strategy is defined by increasing the average time that the licensed channel is occupied.

In the following part of this thesis, we study self-adaptive congestion control at the transport layer, especially for multimedia applications. We focus, in the next chapter, on the resources management in wireless networks at upper layer of the protocol stack, the transport layer. Specifically, we propose some content-aware congestion control mechanisms for partially observable environments.

# Part III

# Self-adaptive and Learning Mechanisms for Congestion Control at the Transport Layer

# Chapter 6

# Learning-TCP: A Media-aware Congestion Control Algorithm for Multimedia Transmission

**Contents**



TCP dominates today's communication protocols at the transport layer in both wireless and wired networks, due to its simple and efficient solutions for end-to-end flow control, congestion control and error control of data transmission over IP networks (see [9] and [11]). However, despite the success of TCP, the existing TCP congestion control is considered unsuitable for delay-sensitive, bandwidth-intense, and loss-tolerant multimedia applications, such as real-time audio streaming and video-conferences (see [9] and [11]). There are two main reasons for this. First, TCP is error-free and trades transmission delay for reliability. Packets may be lost during transport due to network congestion and errors, but TCP keeps retransmitting lost packets until they are successfully transmitted, even if this requires a large delay. The error-free restriction ignores delay deadlines of multimedia packets, i.e. the time by which they must be decoded. Note that even if multimedia packets are successfully received, they are not decodable if they are received after their respective delay deadlines. TCP congestion control adopts an AIMD algorithm. This results in a fluctuating TCP throughput over time, which

significantly increases the end-to-end packet transmission delay, and leads to poor performance for multimedia applications [11]. To mitigate these limitations, a plethora of research focused on smoothing the throughput of AIMD-based congestion control for multimedia transmission (see [112] and [113]). These approaches adopt various congestion window updating policies to determine how to adapt the congestion window size to the network congestion. However, these approaches seldom explicitly consider the characteristics of the multimedia applications, such as delay deadlines and distortion impacts of multimedia packets.

In this chapter, we propose a media-aware POMDP-based congestion control, referred to as Learning-TCP, which exhibits an improved performance when transmitting multimedia packets. Unlike the current TCP congestion control protocol that only adapts the congestion window to the network congestion (e.g. the packet loss rate in TCP Reno and the RTT in TCP Vegas), the proposed congestion control algorithm also takes into account multimedia packets' distortion impacts and delay deadlines when adapting its congestion window size. Importantly, the proposed media-aware solution only changes the congestion window updating policy of the TCP protocol at the sender side, without requiring modifications to feedback mechanisms at the receiver.

Note that the multimedia quality obtained by receivers is impacted by the network congestion incurred at bottleneck links, which is only partially observable by senders based on feedback of network congestion signals. In order to capture dynamics of the network congestion and optimize the expected long term quality of multimedia transmissions, we formulate the media-aware congestion control problem using a POMDP framework. The proposed framework allows users to evaluate the network congestion variations over time, and provides the optimal threshold-based congestion window updating policy that maximizes the long-term discounted reward. In this chapter, the considered reward is the multimedia quality, measured using the well-known Peak Signal to Noise Ratio (PSNR).

In practice, the sender needs to learn the network environment during transmission in order to adapt its congestion control policy. Hence, we also propose an online learning approach for solving the POMDP-based congestion control problem. A comparative study of several existing congestion control mechanisms for multimedia applications and the proposed solution is presented in Table 6.1.

TABLE 6.1: Learning-TCP vs current congestion control solutions for multimedia streaming

| **Algorithm** | **Name of the congestion control** | **TCP-Friendliness** | **Multimedia support** | **Content dependency** | **Decision Type** |
|---|---|---|---|---|---|
| Rejaie 1999 [114] | RAP | AIMD-based | Source rate adaptation | No | Myopic |
| Cai 2005 [112] | GAIMD | AIMD-based | Playback buffering | No | Myopic |
| Bansal 2001 [113] | Binomial Algorithm | Binomial scheme | Source rate adaptation | No | Myopic |
| Our approach | Learning-TCP | AIMD-like media aware | Quality-centric congestion control | Yes | Foresighted |

This Chapter presents a TCP-like window-based congestion control schemes that use history information, in addition to the current window size and congestion feedback. In summary, this chapter makes the following contributions:

*Media-aware congestion control*: The proposed Learning-TCP provides a media-aware approach to adapt the AIMD-like congestion control policy to both varying network congestion and multimedia characteristics taking into account source rates, distortion impacts and delay deadlines of multimedia packets. Hence, the media-aware approach leads to a significantly improved multimedia streaming performance.

*POMDP-based adaptation*: We propose a POMDP framework to formulate the media-aware congestion control problem. It allows the TCP senders to optimize the congestion window updating policy that maximizes the expected long-term quality of multimedia applications. Furthermore, the network user has a partial knowledge about the bottleneck link status. In fact, the number of packets in transit over the bottleneck link queue depends not only on the congestion window of the user, which is known, but also on the congestion windows of all the other users, which cannot be observed. Therefore, the long term prediction and adaptation of the POMDP framework under partial observation of the system state is essential for multimedia streaming, since it can consider, predict, and exploit the dynamic nature of the multimedia traffic and the transmission environment, in order to optimize the application performance.

The POMDP solution is based on a set of updating policies composed of generic congestion control algorithms, with general increase and decrease functions like: AIMD, Inverse Increase/Additive Decrease (IIAD), Square Root inversely proportional Increase/proportional Decrease (SQRT), and Exponential Increase/Multiplicative Decrease (EIMD).

*Online learning for delay-sensitive multimedia applications*: We present some structural properties of the optimal solution. Thereafter, we propose a practical low-complexity

online learning method to solve the POMDP-based congestion control problem on-the-fly. The proposed learning method is designed for multimedia transmission that takes advantage of structural results of the value function.

The chapter is organized as follows. In Section 6.1, we present the media-aware congestion control problem that maximizes the performance of multimedia applications. Thereafter, in Section 6.2, we formulate the problem using a POMDP-based framework. Structural results and the proposed online learning method are presented in Section 6.3. Section 6.4 provides some simulation results that validate the congestion control algorithm, and Section 6.5 concludes the chapter.

## 6.1 Media-aware congestion control formulation

### 6.1.1 Network settings

We assume that the network has a set of $N$ end users indexed $\{1, \cdots, N\}$. Each user is composed of a sender node and a receiver node that establish an end-to-end transport layer connection. Let $w_n$ represents the congestion window size of the user $n$. The network system has some bottleneck links, which results in packet losses when buffers are overloaded. Note that a user cannot observe the traffic generated by other users. In fact, an end user $n$ can only infer the congestion status by observing feedback information from transmitted acknowledgments per RTT. For each acknowledgment, the end user $n$ observes congestion event $o_n \in \{success, fail\}$ (the packet being received successfully or not by the receiver). We consider a time-slotted system with a slot duration of one RTT. Moreover, we assume that the user $n$ has a delay vector $delay_n$ of all packets in its output queue, with $delay_n^i(t+1) = delay_n^i(t) + RTT$ if the $i$-th packet in the queue is not transmitted during the $t$th $RTT$. Before transmitting a packet, the user verifies if $delay_n^i(t) < D_n$, where $D_n$ is the deadline delay of the packet. If not, it drops the packet. The observed information $o_n$ is available to the sender through transmission acknowledgments (ACK) built into the protocol.

### 6.1.2 Two-level congestion control adaptation

A TCP-like window-based congestion control scheme increases the congestion window after successful transmission of a window of packet, and decreases the congestion window upon the detection of a packet loss event. A general description regarding the congestion

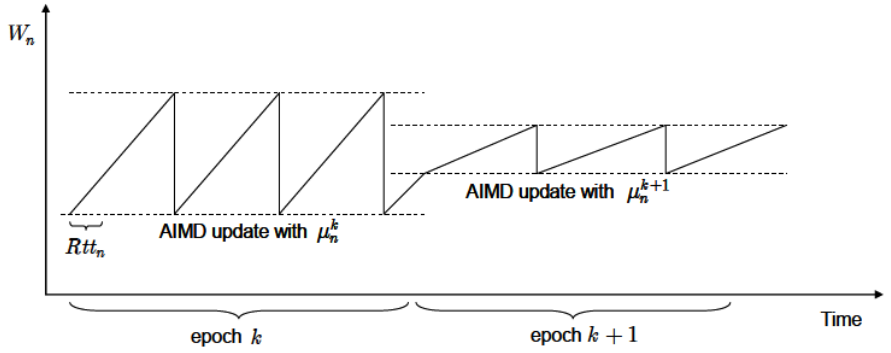


FIGURE 6.1: Congestion window size over time with different update policies per epoch

control window size variation is:

$$w_n \leftarrow \begin{cases} w_n + f(w_n), & \text{if } o_n = success; \\ w_n - g(w_n)w_n, & \text{if } o_n = fail. \end{cases} \tag{6.1}$$

Let us define $\mu_n(w_n) = [f(w_n), g(w_n)] \in \mathcal{A}$, as the updating policy that specifies the two congestion window size variation functions (we refer to $f(w_n)$ as the increasing function and $g(w_n)$ as the decreasing function), where $\mathcal{A}$ represents the set of all updating policies. Some existing examples of updating policies can be found in [112] and [113].

Unlike the existing TCP congestion control that fixes the congestion window updating policy without considering applications' characteristics, the proposed Learning-TCP uses a two-level adaptation to update the congestion window. We define the congestion control epoch $Epoch_n$ as $T \times RTTs$ for user $n$ to periodically change its congestion window updating policy. In fact, we allow the sender to update its policy at the beginning of each epoch, which it cannot change until the next epoch (see Figure 6.1). Indeed, this chapter focuses on how to optimally determine the updating policy, at each epoch, in order to improve the quality of multimedia applications.

### 6.1.3 Expected multimedia quality per epoch

In this section, we discuss the objective of the proposed media-aware congestion control. Denote application parameters as $\phi_n^k = (R_n^k, D_n^k, A_n^k)$ for user $n$ in the $k$th epoch, where $R_n^k$ represents the source rate of the multimedia application. The source rate is the average number of packets that arrives at the transmission buffer per second. For example, in a VoIP call, the source rate can be controlled and adapted to the network environment, since there are usually some rate control modules implemented in VoIP software. We further assume an additive distortion reduction function for multimedia applications as in [115], and $A_n^k$ is the additive distortion reduction per packet in epoch

$k$. $A_n^k$ can be thought of as the media quality improvement of each packet. The following equation depicts the expected distortion reduction per packet for the end user $n$:

$$E[Q_n^{t,k}(w_n^t, \phi_n^k)] = A_n^k(1 - p_n^k(w_n^t)) \sum_{i=1}^{\min\{w_n^t, buf_n\}} I(delay_n^i(t) \leq D_n^k), \tag{6.2}$$

where $buf_n$ represents the number of packet in the buffer of the user $n$. The average distortion reduction in the $k$th epoch is expressed as follows:

$$E[Q_n^k(\mu_n^k, \phi_n^k)] = \frac{1}{T}\sum_{t=1}^{T} E[Q_n^{t,k}(w_n^t, \phi_n^k)]. \tag{6.3}$$

Specifically, a POMDP framework allows users to evaluate the network congestion without perfect knowledge of the overall system state. For each epoch, the proposed Learning-TCP allows the user $n$ to select an optimal updating policy $\mu_n^{opt,k}$ that maximizes the expected distortion reduction in the epoch $k$, given application parameters $\phi_n^k$. Thus, the proposed algorithm performs the following optimization:

$$\mu_n^{opt,k} = \arg\max_{\mu_n^k} \sum_{k=1}^{\infty} \gamma^k E[Q_n^k(\mu_n^k, \phi_n^k)], \tag{6.4}$$

where $\gamma$ is a discount factor. Note that when the application has no delay deadline, i.e. $D_n^k = \infty$, the objective function in Equation (6.4) is equivalent to maximizing the exponential moving average throughput in the epoch.

During periods of severe congestion, our algorithm may not be TCP-friendly, and therefore penalize other TCP flows. We describe, in the next section, how we adapt our algorithm to be quality-centric and TCP-friendly.

### 6.1.4 TCP-Friendliness

TCP is not well-suited for emerging multimedia applications because it ignores QoS requirements of the multimedia traffic. To address this issue, some approaches were proposed using end-to-end congestion control schemes [116]. Since TCP is widely used for traffic transport over the Internet, new congestion control schemes should be TCP-Friendly. Therefore, TCP-Friendly congestion control for multimedia has recently become an active research topic (see [117] and [112]). TCP-Friendliness requires that the average throughput of applications using new congestion control schemes does not exceed that of traditional TCP-transported applications under the same circumstances (see [118]). Therefore, we examine the competitive behaviors between TCP and Learning-TCP.

It is well known that the TCP congestion control strategy increases by one or decreases by half the congestion window. Let us consider a scenario with a link having a capacity of $r$ packets per RTT, shared between two flows, one TCP-transported and the other using our media-aware congestion control algorithm.

It is straightforward that updating policies in $\mathcal{A}$ are not necessarily TCP-Friendly (for example, $f(w) = w$ and $g(w) = 1$). However, there exists a non-empty subset of $\mathcal{A}$, whose policies do not violate the TCP-friendliness rule. Proposition 6.1 states that the Learning-TCP algorithm can be TCP-Friendly.

**Proposition 6.1.** *For all updating policies $\mu$ chosen from the set $\mathcal{A}_{fr} = \{\mu(w) = [f(w), g(w)] | f(w) = \frac{3g(w)}{2-g(w)}\}$, the proposed Learning-TCP algorithm is TCP-Friendly.*

*Proof.* The proof of this proposition is a generalization of the proof of [112] and [119] made for AIMD$(\alpha, \beta)$. We extend this result for a general updating policies $f(w), g(w) : \mathcal{R} \to \mathcal{R}$. Denote by $w_{L-TCP}$ and $w_{TCP}$ the congestion windows of the Learning-TCP transported flow and the TCP transported flow respectively. Assume that both flows have the same RTT and MSS. The effect due to different RTT and MSS is beyond the scope of this dissertation and is an issue in our future work. On one hand, when $w_{L-TCP} + w_{TCP} < r$, the link is in the underload region and thus, the congestion windows $w_{L-TCP}$ and $w_{TCP}$ evolves as follows:

$$w_{L-TCP}(t + \Delta t) = w_{L-TCP}(t) + f(w_{L-TCP}(t))\Delta t \quad (6.5)$$

$$w_{TCP}(t + \Delta t) = w_{TCP}(t) + \Delta t. \quad (6.6)$$

On the other hand, when $w_{L-TCP} + w_{TCP} \geq r$, the link is overloaded and congestion occurs. We assume that both flows receive the congestion signal once congestion occurs and we denote $t_i$ the $i$th time that the link is congested. Both flows decrease simultaneously their window based on the following expression:

$$w_{L-TCP}(t_i) + w_{TCP}(t_i) = r \quad (6.7)$$

$$w_{L-TCP}(t_i^+) = w_{L-TCP}(t_i) - g(w_{L-TCP}(t_i))w_{L-TCP}(t_i) \quad (6.8)$$

$$w_{TCP}(t_i^+) = \frac{1}{2}w_{TCP}(t_i). \quad (6.9)$$

The duration between $t_i$ and $t_{i+1}$ is referred to as the $i$th cycle during which both flows increase their window. Therefore, we have:

$$w_{L-TCP}(t_{i+1}) - w_{L-TCP}(t_i) = -\frac{2g(w_{L-TCP}(t_i)) + f(w_{L-TCP})}{2(f(w_{L-TCP}) + 1))}w_{mL-TCP}(t_i) + \frac{rf(w_{L-TCP})}{2(f(w_{L-TCP}) + 1))} \quad (6.10)$$

Thus, independent of the initial values of $w_{L-TCP}$ and $w_{TCP}$, after a sufficient number of cycles, the congestion windows of these two flows in the overloaded region converge to:

$$w_{L-TCP}(th) = \frac{f(w_{L-TCP})r}{2g(w_{L-TCP}) + f(w_{L-TCP})}, \tag{6.11}$$

$$w_{TCP}(th) = \frac{2g(w_{L-TCP})r}{2g(w_{L-TCP}) + f(w_{L-TCP})}. \tag{6.12}$$

Therefore, in the steady state, $w_{L-TCP}$ and $w_{TCP}$ increase and decrease periodically. Their average throughput in steady state are expressed by the following:

$$\bar{w}_{L-TCP} = \frac{(2 - g(w_{L-TCP}))f(w_{L-TCP})r}{4g(w_{L-TCP}) + 2f(w_{L-TCP})}, \tag{6.13}$$

$$\bar{w}_{TCP} = \frac{3g(w_{L-TCP})r}{4g(w_{L-TCP}) + 2f(w_{L-TCP})} \tag{6.14}$$

Finally, to guarantee the fairness between the flows, the necessary and sufficient condition is:

$$f(w) = \frac{3g(w)}{2 - g(w)}. \tag{6.15}$$

□

## 6.2 POMDP framework for media-aware congestion control

In the proposed framework, users have a partial knowledge about the congestion status of bottleneck links. We define the congestion factor $C_g$, which represents the impact of all users on the congestion status at the bottleneck link. The congestion factor can be seen as a congestion level or occupation level of the bottleneck link. $\mathcal{C}_n$ represents the set of all possible congestion factors. Since the user cannot observe the traffic generated by other users and transmitted over the bottleneck links, it estimates solely the average congestion factor based on history of its observations and actions. Therefore, we formulate the problem with a POMDP framework. Moreover, the objective function to optimize can be rewritten as follows:

$$U_n = \sum_k \gamma^k \sum_{t=1}^{T} A_n^k (1 - p_n^k(w_n^t)) \sum_{i=1}^{\min\{w_n^t, buf_n\}} I(delay_n^i(t) \leq D_n^k). \tag{6.16}$$

Note that the end user tries to maximize the number of packets successfully transmitted before their delay deadlines.

### 6.2.1 POMDP-based congestion control

Based on Equation (6.16), we define a POMDP-based congestion control of user $n$ as a follows:

**Action**: The user selects the congestion window updating policy $\mu_n^k \in \mathcal{A}$, where $\mu_n^k$ is the updating policy of user $n$ in the $k$th epoch.

**State**: The state is defined as $X_n^k = \{C_g, \phi_n^k\} \in \mathcal{X}_n$. The application parameters $\phi_n^k$ are known by the user $n$. However, the congestion factor $C_g \in \mathcal{C}_n$, which is impacted by the overall traffic transiting in the bottleneck link, cannot be directly observed by the users. The user $n$ has to infer the congestion factor based on the observed information and feedback.

At each time slot, the system has a congestion factor $C_g$. The user takes an action $\mu_n$, which causes the environment to transit to $C_g'$ with probability $T(C_g', \mu_n, C_g)$. Having the congestion factor $C_g'$, the user observes $o_n$ with probability $O(o_n, C_g', \mu_n)$. The belief about the congestion factor is defined as $b : \mathcal{C}_n \to [0, 1]$. The function $b(.)$ represents the probability distribution of the congestion factor at the $k$th epoch. Denote the chosen congestion factor (i.e., inferred by the end user as the most likely of all possible congestion factors) at the $k$th epoch by $C_g^k$. The belief distribution of the congestion factor $b(C_g)$ is updated as follows:

$$\begin{aligned} b_n^k(C_g') &= \frac{Pr(o_n|C_g', \mu_n^k, b) Pr(C_g'|\mu_n^k, b)}{Pr(o_n|\mu_n^k, b)}; \\ &= \frac{O(o_n, C_g', \mu_n^k) \sum\limits_{C_g \in \mathcal{C}_n} T(C_g', \mu_n^k, C_g) b_n^{k-1}(C_g)}{Pr(o_n|\mu_n^k, b)}. \end{aligned} \tag{6.17}$$

The denominator, $Pr(o_n|\mu_n, b)$, can be treated as a normalizing factor, independent of $C_g'$ that causes $b$ to sum to 1.

The probability $p_n^k(w_n)$ represents the average packet loss rate in the $k$th epoch when the congestion window size is $w_n$, which can be calculated as follows:

$$p_n^k(w_n) = \sum_{C_g \in \mathcal{C}_n} Prob(C_g \geq \widetilde{C}_g|w_n) b_n(C_g), \tag{6.18}$$

where $\widetilde{C}_g$ is the congestion level at the bottleneck link, which is not observable by users. However, the average packet loss rate itself is observable by users, given a certain congestion window $w_n$.

**Utility**: Based on Equation (6.16), the utility of user $n$ is defined as the discounted long-term expected reward:

$$U_n = \sum_{k=1}^{\infty} \gamma^k \sum_{C_g \in \mathcal{C}_n} u_n(X_n^k, \mu_n^k) b(C_g), \tag{6.19}$$

where $u_n(X_n^k, \mu_n^k) = \sum_{t=1}^{T} A_n^k (1 - p_n^k(w_n^t)) \sum_{i=1}^{\min\{w_n^t, buf_n\}} I(delay_n^i(t) \leq D_n^k)$ represents the immediate reward in the $k$th epoch.

A policy $\mu_n^{opt} = \{\mu_n^{opt,1}, \mu_n^{opt,2}, ...\}$ that maximizes $U_n$ is called an optimal policy that specifies for each epoch $k$ the optimal updating policy $\mu_n^{opt,k}$ to use. The optimal value function $U_n^k$ satisfies the following Bellman equation:

$$U_n^k(C_g^k) = \max_{\mu_n^k \in \mathcal{A}} \{u_n(X_n^k, \mu_n^k) + \gamma \sum_{C_g' \in \mathcal{C}_n} T(C_g' | \mu_n^k, C_g) U_n^{k+1}(C_g')\}. \tag{6.20}$$

The optimal policy at the $k$th epoch is expressed as follows:

$$\mu_n^{opt,k} = \arg \max_{\mu_n^k \in \mathcal{A}} \{u_n(X_n^k, \mu_n^k) + \gamma \sum_{C_g' \in \mathcal{C}_n} T(C_g' | \mu_n^k, C_g) U_n^{k+1}(C_g')\}. \tag{6.21}$$

We prove in the next section the existence of optimal stationary policy and we show how to determine such policy for our POMDP problem.

### 6.2.2 Existence of optimal stationary policy

Because of the difficulty of computation and implementation of the optimal solution for POMDP-based problems, we would like to restrict attention to stationary policies when seeking optimal solution. Note that we formulate our problem as an infinite horizon POMDP with expected discounted reward.

The belief set is continuous, which may lead to an explosion of the solution size and the computation complexity. Therefore, we transform the belief set to a discrete set. We use an aggregation function that maps the belief states into a discrete set of beliefs. An example of aggregation function is presented in Section 6.3. Moreover, for each belief, we assume that there is a finite set of actions $\mathcal{A}$. Under these assumptions, Theorem 6.2.10 of [92] can be applied and we can prove the existence of an optimal stationary policy for our POMDP problem. Therefore, we restrict our problem to the set of stationary policies. We are able to determine an algorithm that computes one such policy. We can now omit the epoch index $k$, as the optimal stationary policies depend only on $\phi$ and $C_g$. The goal of this POMDP problem is therefore to find a sequence of updating

policies $\mu_n$ that maximizes the expected reward. For each belief, the value function can be formulated as follows:

$$U_n(C_g) = \max_{\mu_n \in \mathcal{A}} \{u_n(X_n, \mu_n) + \gamma \sum_{C'_g \in \mathcal{C}_n} T(C'_g | \mu_n^k, C_g) U_n(C'_g)\} \tag{6.22}$$

Specifically, a powerful result of [34] and [35] says that the optimal value function for our POMDP problem is PWLC in the belief. Then, every value function can be represented by a set of hyper-planes denoted $\Upsilon$-vectors, $\Gamma_k$, where $U_n(C_g) = \max_{\Upsilon \in \Gamma_k} b(C_g)\Upsilon$. $\Gamma_k$ is updated using the value iteration algorithm through the following sequence of operations:

$$\Gamma_{k+1}^{\mu,o_n} \leftarrow \Upsilon_{\mu}^{o_n}(X_n) = \frac{u_n(X_n, \mu)}{|o_n|} + \gamma \sum_{X' \in \mathcal{X}} T(X_n, \mu, X') O(o_n, C'_g, \mu) \Upsilon(X'), \forall \Upsilon \in \Gamma_k, \tag{6.23}$$

$$\Gamma_{k+1}^{\mu} = \oplus_{o_n} \Gamma_{k+1}^{\mu,o_n}; \tag{6.24}$$

$$\Gamma_{k+1} = \cup_{\mu \in \mathcal{A}} \Gamma_{k+1}^{\mu}. \tag{6.25}$$

Note that each $\Upsilon$-vector is associated with an action that defines the best updating policies for the previous $(k-1)$ epochs. The $k$th horizon value function can be expressed as follows:

$$U(C_g) = \max_{\mu_n \in \mathcal{A}} \left[ u_n(X_n^k, \mu_n) + \gamma \sum_{o_n} \max_{\Upsilon \in \Gamma_k^{\mu_n, o}} \sum_{C'_g \in \mathcal{C}_n} P_n(C'_g | C_g) O(o_n, C'_g, \mu) \Upsilon \right]. \tag{6.26}$$

Many algorithms were proposed to implement solutions for POMDP problems by manipulating the $\Upsilon$-vector using a combination of set projection and pruning operations (see [34],[95] and [120]).

The main difficulty of POMDP-based optimization is the prohibitively high computational complexity and the assumption that statistics, such as the state transition probability are priory known, which may be not true in practice. To overcome this obstacle, we propose an online learning method that allows the sender to determine the optimal congestion control policy on-the-fly, with a low computational complexity.

## 6.3 Online Learning

Solving a POMDP is an extremely difficult computational problem. In this section, we show how the value function can be updated on-the-fly, and with a low computation complexity, in order to solve the POMDP problem described in the previous section. In the proposed learning model, the user maintains the state-value function $Q(\mu_n, \phi, C_g)$ as a lookup table, which determines the optimal policy in the current slot. In fact, the state-value function $Q(\mu_n, \phi, C_g)$ is updated as follows:

$$Q(\mu_n^{k-1}, \phi^{k-1}, C_g^{k-1}) \leftarrow \beta_k Q(\mu_n^{k-1}, \phi^{k-1}, C_g^{k-1}) + (1-\beta_k)(U_n + \gamma Q(\mu_n^k, \phi^k, C_g^k)), \quad (6.27)$$

where $\beta_k$ is a learning rate factor satisfying $\sum_{k=1}^{\infty} \beta_k = \infty, \sum_{k=1}^{\infty} (\beta_k)^2 < \infty, e.g. \beta_k = \frac{1}{k}$. At the epoch $k-1$, the user gets the application parameters $\phi^{k-1}$, estimates the congestion factor $C_g^{k-1}$, and chooses the policy $\mu_n^{k-1}$ that maximizes $Q(\mu_n^{k-1}, \phi^{k-1}, C_g^{k-1})$. At the epoch $k$, the user obtains the new application parameters, estimates the congestion factor, chooses a congestion window updating policy, and updates the state-value function $Q(\mu_n^{k-1}, \phi^{k-1}, C_g^{k-1})$.

The large state space $\mathcal{X}_n$, due to the continuous space of congestion factors, may prohibit an efficient learning solution, due to the complexity and the long convergence time. We propose to adopt an effective state aggregation mechanism to reduce the complexity and the convergence time of the learning algorithm. As an example of the aggregation function, we may quantize the congestion factor to the nearest integer.

### 6.3.1 Adaptive state aggregation

We propose to use an aggregation function that maps the congestion factor space $\mathcal{C}_n$ into a discrete space, as we have assumed in Section 6.2.2. This function aggregates the adjacent average congestion factors $C_g' \in \tau_n \subset \mathcal{C}_n$ into a representative average congestion factor value $C_g$. In this chapter, we propose an adaptive state aggregation method that iteratively adapts the aggregation function. Let $\Delta(C_g, U_n^k, \delta)$ represent the adaptive aggregation function, defined as follows:

$$\Delta(C_g, U_n^k, \delta) = C_g^n = \frac{C^L + C^H}{2}, \quad (6.28)$$

where $C^L = invU_n^k(U^{min} + (l-1)\delta)$, $C^H = invU_n^k(U^{min} + l\delta)$, and $(l-1)\delta \leq U_n^k(w|C_g) - U^{min} < l\delta$. Note that $invU_n^k$ represents the inverse function of $U_n^k(w|C_g)$, $U^{min}$ denotes the minimum value of the expected utility of the user starting from the previous epoch,

and $\delta$ is referred to as the utility spacing that determines the aggregation function from the expected utility-to-go domain.

### 6.3.2 Structural Properties

In this section, we develop some structural properties of the optimal policy and corresponding value function, based on which we will then discuss approximation results of the value function. This approximation allows us to represent compactly the value function. It was proved, in [35], that the optimal value function $U_n^*$ is PWLC with respect to the belief vector. As we are considering a discrete set of average congestion factors, the value function can be approximated using a PWLC function. Importantly, we are able to control the computational complexity and achievable performance by using different predetermined approximation error thresholds $\delta$.

**Algorithm 2** Online learning algorithm for POMDP-based congestion control

Initialize $Q(\mu_n^k, \phi_n^k, C_g) = 0$ for all possible application parameters, congestion factor and updating policy;
Initialize $\phi$, $\mu_n$ and $C_g$;
$U_n = 0$;
**while** true **do**
  $\phi^{prev} = \phi$;
  $\mu_n^{prev} = \mu_n$;
  $C_g^{prev} = C_g$;
  Get the new application parameters $\phi$;
  Select the policy and congestion factor such as: $(\mu_n^k, C_g) = \arg\max_{\mu_n, C_g} Q(\mu_n, \phi, C_g) b_n(C_g)$ with probability $(1-\epsilon)$, else choose a random policy and congestion factor;
  $Q(\mu_n^{prev}, \phi^{prev}, C_g^{prev}) \leftarrow \beta_k Q(\mu_n^{prev}, \phi^{prev}, C_g^{prev}) + (1-\beta_k)(U_n + \gamma Q(\mu_n, \phi, C_g))$;
  $U_n = 0$;
  **for** $t = 1 \rightarrow T$ **do**
    Transmit packets using the updating policy $\mu_n$ and the congestion factor $C_g$;
    Update the congestion window based on Equation (6.1);
    $U_n = U_n + A_n^k \times recPkt$, where $recPkt$ is the number of packets received before their delay deadlines.
  **end for**
  Update the beliefs based on Equation (6.17);
**end while**

We propose, in this section, a low-complexity online learning algorithm based on an extension of the on-policy TD-$\lambda$ Algorithm [121], described in Algorithm 3. The proposed learning method is greatly impacted by the utility spacing $\delta$, and the number of states in an epoch depends on the aggregation function $\Delta(C_g, U_n^k, \delta)$. The size of the average congestion set in the $k$th epoch is $\lceil \frac{U^{k,max} - U^{k,min}}{\delta} \rceil + 1$.

At the beginning of epoch $k$, the user receives the application parameters $\phi_n^k$ from the upper layer, and selects the updating policy and the congestion factor that maximize its state-value function. Then, the user transmits its packets during the epoch using the chosen policy. At the end of the epoch, the user updates the state-value function based on observation during the epoch. The following lemma proves the convergence of the proposed algorithm.

**Lemma 6.2.** *The proposed learning algorithm converges to the optimal value function w.p.1.*

*Proof.* The proof of this lemma follows from the Theorem 1 of [122]. In fact, Sarsa algorithm converges to the optimal values function whenever the following assumptions hold:

1. The state space and the action space are finite,
2. $\beta_k$ satisfies $\sum_{k=1}^{\infty} \beta_k = \infty, \sum_{k=1}^{\infty} (\beta_k)^2 < \infty, e.g. \beta_k = \frac{1}{k}$,
3. The reward function is bounded.

It is straightforward that the previous assumptions hold for our problem, and therefore, the Algorithm 3 converges to the optimal values function. □

### 6.3.3 Implementation and complexity

Although value iteration algorithms give an exact solution of POMDP optimization problems, those algorithms require a time and space complexity that may be prohibitively expensive. In fact, to better understand the complexity of exactly solving the POMDP problem, let $|\Gamma_k|$ be the number of $\Upsilon$-vectors in the $k$th epoch. In the worst case, the $\Upsilon$-vectors size in the $(k+1)$-th epoch is $|\mathcal{A}| \times |\Gamma_k|$ (see [123]), and the running time will be $|\mathcal{X}_n|^2 \times |\mathcal{A}| \times |\Gamma_k|$. It also requires solving a number of LPs for pruning vectors.

Interestingly, the proposed algorithm has a state space of $|\mathcal{A}| \times |\mathcal{C}_n| \times |\Phi|$, and has a polynomial time complexity. Therefore, this algorithm can be implemented on mobile devices as it takes only a polynomial time when seeking for the optimal policy. Moreover, the proposed algorithm is implemented only at the transmitter side and is transparent for the receiver. We do not even require any change at routers. Moreover, as we have proved that Learning-TCP is TCP-Friendly, any other congestion control algorithm can be implemented in parallel. For first epochs, the Learning-TCP algorithm may give suboptimal performance. However, a near-optimal result can be obtained after a sufficient number of epochs. Interestingly, we can significantly speed up the learning and

avoid this problem if the state-value functions are initialized with the values obtained the last time Learning-TCP was used.

## 6.4 Simulations

In this section, we present some simulation results using MATLAB-based simulations of the proposed Learning-TCP algorithm. Note that we do not study the performance of congestion control schemes (AIMD, Binomial,...) as they were already deeply investigated. Instead, we analyze the performance of LearningTCP that chooses one congestion control schema every epoch. We consider that multimedia users are transmitting video sequences at a variable bit rate of $\mathcal{R} = \{1, 1.25, 1.5, ..., 5.75, 6\}$ Mbps. We assume that packets can tolerate a delay of $\mathcal{D} = \{133, 266, 400, ..., 800\}$ ms, and we set the packet length to 1024 Bytes. Moreover, we assume that each frame has an additive distortion per packet in the set $\mathcal{A}_{distor} = \{0.05, 0.06, ..., 0.16\}$. We consider also a set of policies $\mathcal{A}$ composed of IIAD and SQRT policies defined as follows:

$$\text{IIAD: } f(w) = \frac{3\beta}{2w - \beta} \text{ and } g(w) = \frac{\beta}{w}; \tag{6.29}$$

$$\text{SQRT: } f(w) = \frac{3\beta}{2\sqrt{w+1} - \beta} \text{ and } g(w) = \frac{\beta}{\sqrt{w+1}}; \tag{6.30}$$

where $\beta \in \{0.1, 0.2, ..., 0.9\}$. We consider the set of average congestion factors $\mathcal{C}_n = \{1, 2, .., 50\}$, and we set $\gamma$ to 0.1.

### 6.4.1 TCP-fairness

We focus, first, on the fairness of our proposed Learning-TCP. Figure 6.2 shows how the proposed algorithm interacts with TCP transported flows depending on QoS parameters chosen from the set $\Phi = \mathcal{R} \times \mathcal{D} \times \mathcal{A}_{distor}$. In order to study this effect, we simulate 10 connections: 5 with TCP and 5 connections using the Learning-TCP algorithm, within different QoS requirements and application parameters. We illustrate, in Figure 6.2, the fairness ratio depending on the delay deadline and source rate. The fairness ratio (see [113] and [114]) is defined by the ratio between the total throughput of Learning-TCP connections and total throughput of TCP connections. The closer the fairness ratio is to 1, the friendlier will the congestion control be to other TCP flows. We observe that Learning-TCP has a fairness ratio close to 1 except with hard deadline delay and high source rate. In fact, as we can see in Figure 6.2, when the delay deadline is lower than 300 ms and the source rate is higher than 4 Mbps, the fairness ratio is between 1.2 and

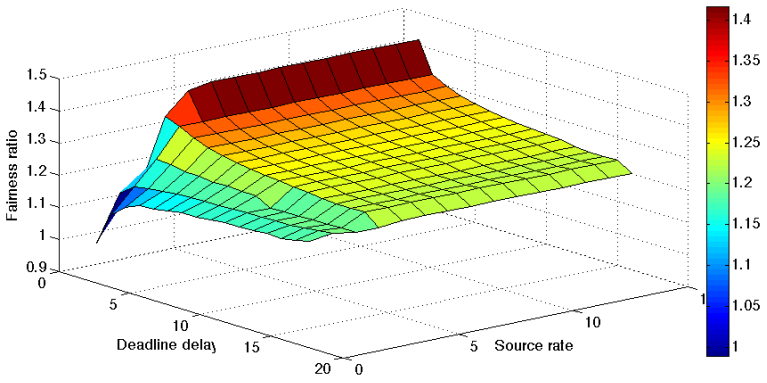


FIGURE 6.2: Fairness ratio of Learning-TCP for different source rates and delay deadlines .

1.45. Indeed, with hard deadline delays and high source rates, the user needs higher throughput in order to satisfy its QoS requirement.

### 6.4.2 Learning-TCP algorithms and fixed-policy algorithms

In this section, we investigate the interactions between Learning-TCP and other multimedia congestion control algorithms. We consider a bottleneck link of capacity 100 Mbps shared between 10 users (one Learning-TCP; one TCP and 8 users using Binomial congestion control) as described in Table 6.2. We simulate a video transmission application during 350 time slots, and we assume that users receive a new set of application parameters every epoch, T=50 RTT, where the RTT duration is 100 ms. In order to illustrate the impact of the delay on the congestion control algorithms, we assume that the deadline delay is 133 ms in the first epoch, and that it increases by 133 ms every epoch. A real use-case of these simulation settings is a streaming application, where the user may change the required quality at each epoch. Let us consider a congested network, the user decreases at the end of each epoch the required quality of the streaming, and increases the deadline delay, thereby decreases the packet loss probability. We observe, in Figure 6.3, that the Learning-TCP uses different policies for each delay deadline. For hard delay deadlines, we observe that the throughput of the Learning-TCP user is higher than the throughput of other users. Figure 6.5 illustrates the throughput of TCP user and Figure 6.4 illustrates the throughput of Binomial congestion control users. The Binomial-CC users obtain the highest average throughput (9.2 Mbps Versus 7.65 Mbps for TCP and 8.36 Mbps for Learning TCP). In fact, as we can see in Figure 6.3, the Learning-TCP gives the highest throughput for hard delay deadlines. However, it is still TCP-friendlier in the average. Interestingly, we show in the next section, how the proposed algorithm gives better video quality when obeying the friendliness rule.

TABLE 6.2: Users in the network

| | IIAD1 | IIAD2 | IIAD3 | IIAD4 | TCP |
|---|---|---|---|---|---|
| I | $\frac{0.3}{w-0.1}$ | $\frac{0.6}{w-0.2}$ | $\frac{0.9}{w-0.3}$ | $\frac{1.2}{w-0.4}$ | 1 |
| D | $\frac{0.2}{w}$ | $\frac{0.4}{w}$ | $\frac{0.4}{w}$ | $\frac{0.8}{w}$ | 0.5 |
| | SQRT1 | SQRT2 | SQRT3 | SQRT4 | LEARNING-TCP |
| I | $\frac{3}{8\sqrt{w+1}-1}$ | $\frac{3}{4\sqrt{w+1}-1}$ | $\frac{9}{8\sqrt{w+1}-3}$ | $\frac{3}{2\sqrt{w+1}-1}$ | f(w) |
| D | $\frac{0.25}{\sqrt{w+1}}$ | $\frac{0.5}{\sqrt{w+1}}$ | $\frac{0.75}{\sqrt{w+1}}$ | $\frac{1}{\sqrt{w+1}}$ | g(w) |

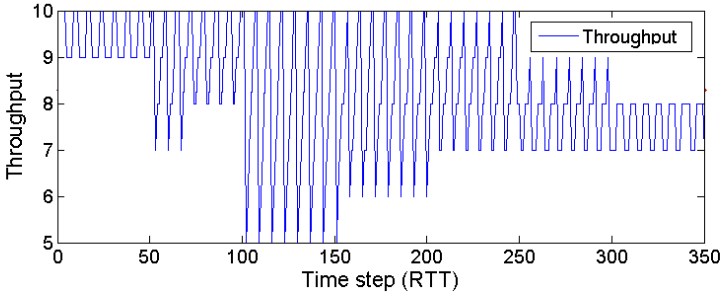


FIGURE 6.3: Throughput of Learning-TCP.

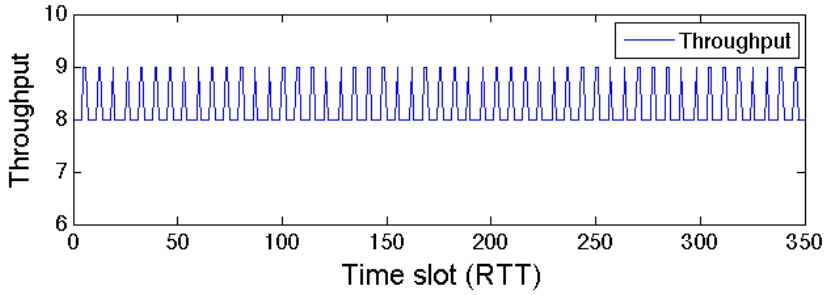


FIGURE 6.4: Throughput of Binomial-CC.

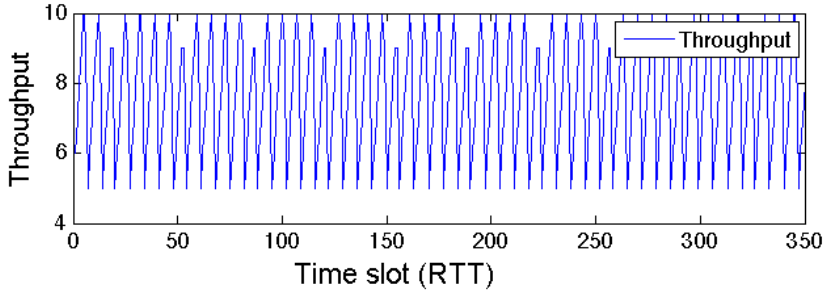


FIGURE 6.5: Throughput of TCP.

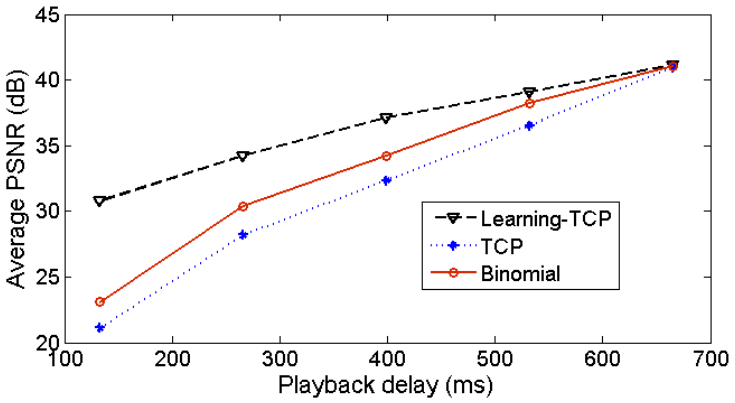


FIGURE 6.6: Average received video quality using different congestion control for multimedia transmission.

### 6.4.3 Performances of Learning-TCP against others multimedia congestion control algorithms

In order to evaluate the video quality (measured through the average PSNR, in decibels) using different congestion control algorithms, we consider the previous scenario where 4 users use Binomial congestion control algorithm; 4 users use TCP and two users using a Learning-TCP algorithm.

We simulate the transmission of a video sequence with length of 50 s (CIF resolution, 50 Hz frame-rate) and compressed by an H.264/AVC codec (any codec can be used, we used this one just for illustrative purposes). We assume that users receive different values of source rate and additive distortion per packet at every epoch. The delay deadline varies between 133 ms and 800 ms. Figure 6.6 illustrates the video quality obtained with different congestion control algorithms. We observe that the Learning-TCP leads to better video quality. Therefore, our proposed approach outperforms others, especially for real-time applications with hard deadline delay such as video-conferencing applications for example. In fact, as illustrated in Figure 6.7, Learning-TCP users obtain the highest percentage of packets delivered before their delay deadline. Indeed, our algorithm is able to optimize the congestion window by considering the distortion impact, delay deadline and the source rate.

## 6.5 Conclusion

In this chapter, we have formulated the media-aware congestion control problem as a POMDP that considers the distortion impact, delay deadline and the multimedia source rate. We have considered a set of generic TCP-friendly updating functions for the

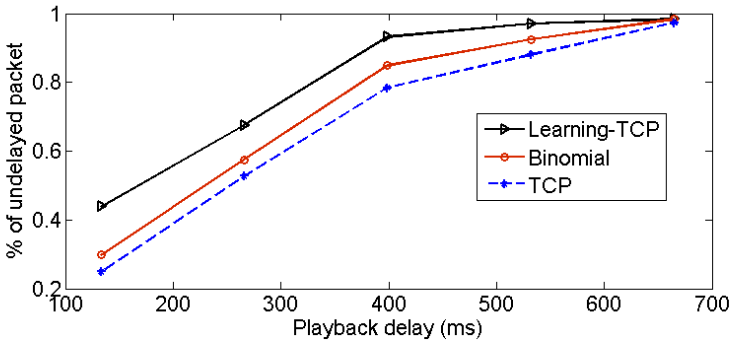


FIGURE 6.7: The percentage of packets delivered before their delay deadlines.

congestion window adaptation. The optimal policy allows the sender to optimize the congestion window size that maximizes the long-term expected quality of the multimedia application. We have also proposed an online learning method to solve the Learning-TCP on-the-fly. Simulation results have shown that the proposed congestion control algorithm outperforms conventional TCP-friendly congestion control schemes in terms of quality, especially for real-time applications with hard delay deadlines. Moreover, the proposed Learning-TCP algorithm is implemented only at the sender side, and is transparent to the routers and the receiver.

Note that we have considered only the impact of QoS parameters (delay, source rate, etc.) on the congestion control. In the next chapter, we focus on the quality perceived by end users through a QoE-based congestion control algorithm. In fact, we consider that users maximize the QoE, based on MOS feedbacks from receivers.

# Chapter 7

# QoE-aware Congestion Control Algorithm for Conversational Services in Wireless Environments

Contents



## 7.1 Introduction

When the bottleneck link is overloaded or channel conditions are bad, the TCP throughput decreases and cannot satisfy the source rate of the multimedia application. This increases, generally, the jitter and the packet loss rate that could impact the user-perceived quality, which is also known as the QoE. Although the QoE is affected by some factors, such as the audio quality, devices, echo, etc., we focus, in this chapter, on improving the QoE through a novel congestion control algorithm. The impact of non-networking factors could be cataloged into a protocol stack to form a conceptual

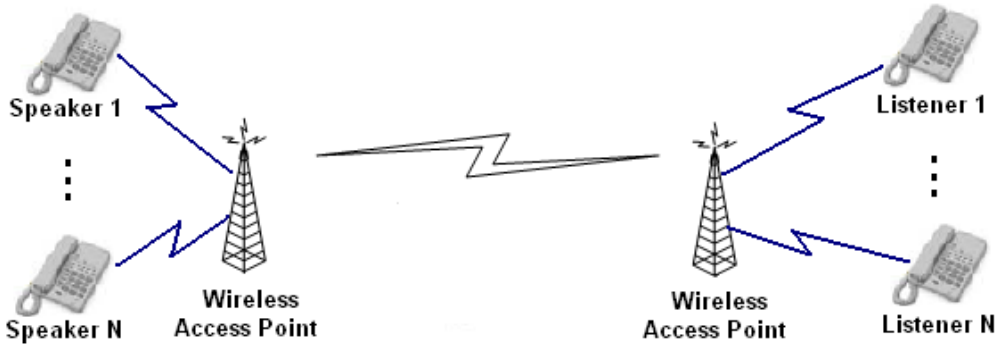


FIGURE 7.1: The experimental model

relationship between QoS and QoE (see [124] and [125]). The QoE is measured by MOS values. In a subjective test, the QoE is rated on a scale of 1 (bad) to 5 (excellent) by a significant number of people, and the average of the scores is called a MOS. Note that, the ITU-T Recommendation P.911 [126] provides the reference for carrying out subjective measurement of audiovisual materials.

In this chapter, we propose a QoE-aware POMDP-based congestion control algorithm, referred to as MOS-TCP, which exhibits an improved performance when transporting multimedia applications, specifically over a wireless path. Our algorithm is suited for networks containing wireless branches, like the model depicted in Figure 7.1. The goal of the MOS-TCP algorithm is to control the end-to-end congestion in order to maximize the QoE, where packets can be lost due to congestion or randomly due to errors encountered across the wireless path. Unlike the current TCP congestion control protocol that only adapts the congestion window to the network congestion (e.g. based on network congestion signals, such as the packet loss rate in TCP Reno, or the round-trip time in TCP Vegas), the proposed congestion control algorithm adopts a two-level congestion control mechanism. Indeed, it adapts over time the congestion window size according to the source rate and the QoE feedbacks. Moreover, we consider a set of updating policies composed of generic congestion control algorithms with general increase and decrease functions, such as AIMD, IIAD, SQRT, and EIMD. In order to capture dynamics of the network congestion and optimize the QoE, we formulate the congestion control using a POMDP framework. The proposed POMDP framework allows users to evaluate the network congestion variations over time, and determines an optimal threshold-based congestion window updating policy in order to maximize the long-term discounted reward. In this chapter, the QoE measured through the multimedia quality (MOS) defines the reward.

In summary, we address the following contributions:

*QoE-aware congestion control*: The proposed MOS-TCP adapts the AIMD-like congestion control policy to both varying network congestion and multimedia characteristics.

*POMDP-based adaptation*: We formulate the QoE-aware congestion control problem using a POMDP framework. The framework allows senders to optimize the congestion window updating policy that maximizes the long-term expected QoE. Furthermore, users have a partial knowledge about the bottleneck link status. In fact, the number of packets in the bottleneck link queue depends on the congestion windows of all users, which cannot be observed. Therefore, the long term prediction and adaptation of the POMDP framework is essential for optimizing the performances of multimedia applications.

*Online learning for QoE-sensitive multimedia applications*: Since the computation of an optimal policy is usually time/process consuming, and as wireless devices are capacity-limited, we propose practical learning method to solve the POMDP-based congestion control problem on-the-fly. The proposed model-free learning algorithm is based on TD-$\lambda$ reinforcement learning, and is designed for QoE-sensitive multimedia applications.

This chapter is organized as follows. We introduce the QoE and explain the MOS calculation in Section 7.2. In Section 7.3, we model the QoE-aware congestion control problem that maximizes the performance of multimedia applications. Thereafter, in Section 7.4, we formulate the problem using a POMDP-based framework. We present a low-complexity algorithm to solve the POMDP in Section 7.5. Section 7.6 provides experimental results that validate the proposed congestion control method, and Section 7.7 concludes the chapter.

## 7.2 QoE-aware networking and MOS measurement

To overcome the limitation of QoS-based optimization, QoE-based approaches are introduced as a more effective way to optimize transmission algorithms and protocols with respect to user satisfaction. QoE metrics are defined as a set of quantitative measures to assess the perceived QoS of end users [127]. Moreover, a new approach, namely *QoE-aware networking*, is proposed to re-formalize the service optimization problem and to improve the user experience. Because the QoE metrics reflect the end user's experience, QoE-based approaches may improve the subjective service quality, optimize the use of network resources, and provide services to more users without noticeable degradation of users' experience. Recently, QoE metrics are used to optimize various types of network services. In cellular systems, authors of [128] used a QoE-based approach to allocate downlink wireless resources among different applications. They defined several QoE models for different types of applications such as file downloading, voice call and video

streaming, and adopt QoE-based utility maximization to improve the user perceived quality. In [129], authors applied QoE metrics to optimize IEEE 802.11 wireless LAN. They used a machine learning approach to generate real-time QoE measurements and used the QoE feedbacks to manage wireless network resources. In [130], authors used QoE metrics for packet scheduling in multi-hop wireless networks. The packet scheduler determines the packet drop pattern that minimizes the degradation of MOS values. In P2P networks [131], scalable video coding and QoE metrics are used to optimize the performance of P2P video streaming systems. In this chapter, we seek to enable QoE-awareness in a more general setting. We integrate the QoE metrics within the TCP protocol. Since TCP is a widely adopted building block in many network services, our approach is applicable to a much wider spectrum of applications.

Since QoE metrics are subjective, the standard QoE measuring process should involve human observers, e.g., when measuring VoIP quality, the MOS are often used as a subjective rating ranging from 1 (poor) to 5 (excellent). However, to enable QoE-awareness in multimedia services, it is infeasible to use subject human tests for real-time applications. Instead, some QoE online prediction methods should be used to estimate QoE from the service output. The QoE prediction methods are dependent on the types of content. A number of models are proposed for predicting QoE with different kinds of contents including web service[132], voice services [133], audio/video content [134], etc. Instead of proposing another new approach of QoE prediction, we base our experiments on the QoE prediction results produced by an existing real system, i.e. Microsoft Lync system [135] (previous known as Office Communication Server and Office Communicator [136]). In the Lync system, the VoIP software measures a set of variables, which may affect the QoE throughout the communication sessions. Based on the collected measurements, it can predict the subjective QoE metrics in real-time. Furthermore, the QoE metrics are normalized and represented in the standard MOS. Our considered Lync software provides several types of MOS values (NetworkMOS, ListeningMOS, conversationalMOS) in order to represent the degradation in different phases of the whole communication process (see Figure 7.2). The MOS prediction mechanism provides a quantitative approach to evaluate the communication quality that end users have experienced.

- NetworkMOS is calculated purely based on obtained network statistics (information), which include the packet loss, bit errors, packet delay, and jitter.
- ListeningMOS is not only decided by network parameters, but also by the choice of audio codec and audio devices, as well as the recording conditions such as echo, background noise level, talk-over, etc. It captures the perceived quality of an audio stream at the receiver side. Note that both NetworkMOS and ListeningMOS are only measured for unidirectional traffic.

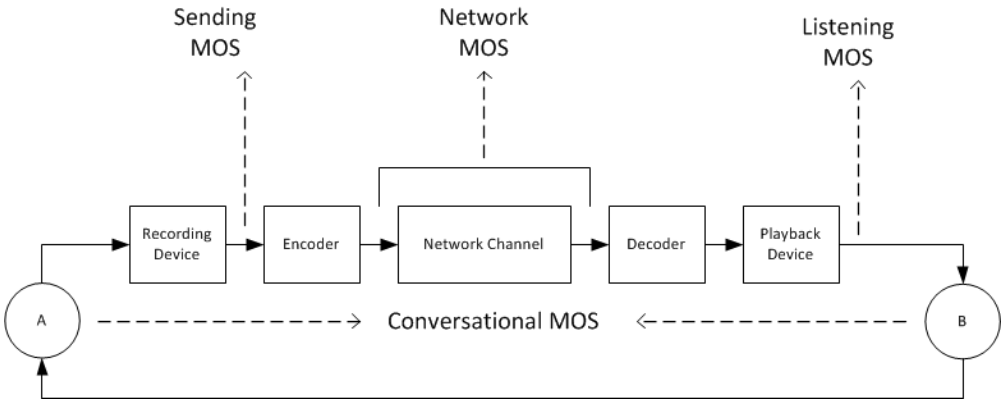


FIGURE 7.2: Different MOS measurements in Microsoft Lync system

- ConversationalMOS is measured for both sending and receiving streams. It takes into account the round-trip delay in addition to all the above-mentioned factors.

Observing these different MOS values gives us a clear perspective on the performance of the entire communication process. A congested network, for example, will cause degradation in NetworkMOS, while a bad recording device can be identified from low ListeningMOS values.

## 7.3 QoE-aware congestion control problem

We consider the same network model and congestion window adaptation defined in Chapter 6. In this section, we discuss the objective of the proposed QoE-aware congestion control. Denote by $R_n^k$ the source rate of a multimedia application for user $n$ in the $k$-th epoch. The source rate is the average number of packets that arrives at the transmission buffer per second at the transport layer. In fact, in a VoIP call, the source rate can be controlled and adapted to the network environment, since there are usually some rate control modules implemented in VoIP software.

We propose, in this chapter, a congestion control algorithm that dynamically changes the congestion window updating policy in order to maximize the QoE. Therefore, it is straightforward and somehow intuitive that each user has as objective to maximize its own QoE. As we can see, in Figure 7.3, the MOS is correlated with the listener satisfaction. The higher MOS, the greater the listener's satisfaction. Therefore, the objective of users is to maximize the expected future MOS starting from the current slot. A similar utility function was used in [137]. Each user tries to optimize, selfishly,

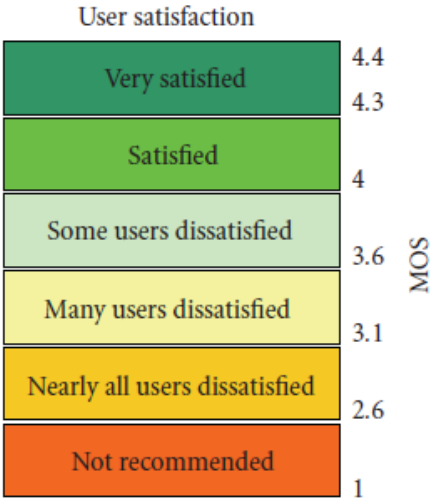


FIGURE 7.3: Relation between MOS and user satisfaction [134].

the following expected total discounted MOS:

$$U_n = \sum_{k=1}^{\infty} \gamma^k u_n^k(\mu_n^k, R_n^k), \tag{7.1}$$

where $\gamma$ is a discount factor, and $u_n^k = MOS_n^k(\mu_n^k, R_n^k)$ is the received MOS by the user $n$ at the $k$-th epoch, when the source rate at the $k$-th epoch is $R_n^k$, and the user $n$ uses the congestion window updating policy $\mu_n^k$. Since the network user do not take frequently decision (it chooses a congestion control every $T$RTT), we have used the expected discounted reward criterion instead of the average reward criterion. MOS-TCP mechanism allows the user to maximize its expected total discounted MOS. In fact, the QoE varies with the source rate, the congestion window updating policy, and the congestion status at the bottleneck links. The latter depends not only on the user $n$ but also on other users. We show, in the next section, how the proposed POMDP-based congestion control algorithm determines the optimal updating policy given partial knowledge of bottleneck links status.

## 7.4 POMDP-based congestion control

The network user maximizes its expected discounted QoE expressed by Equation (7.1). We formulate our problem using a POMDP-based framework as the global system state is not well known for users. In fact, the user has a partial knowledge about the congestion status at the bottleneck links. The latter depends on the congestion windows of all users, which is unknown by the user $n$. Thus, the user $n$ has to estimate solely the impact of all the other users based on the history of observations and actions. In

fact, the user $n$ estimates the packet loss rate when it transmits data using the congestion window $w_n$. We define a POMDP-based congestion control of user $n$ in a tuple $\{\mathcal{A}, \mathcal{X}_n, O_n, \Omega_n, P_n, U_n\}$ as follows:

**Action**: The user selects the congestion window updating policy $\mu_n = \{\mu_n^1, \mu_n^2, ...\} \in \mathcal{A}$, with $\mu_n^k$ is the updating policy of user $n$ in the $k$-th epoch.

**State**: The state is defined as $X_n^k = \{p_n^k, R_n^k\} \in \mathcal{X}_n$. The source rate $R_n^k$ is known by the user $n$. However, the packet loss rate $p_n^k$, which is impacted by other users' windows, cannot be directly observed. The user $n$ has to infer the congestion status of the bottleneck links using congestion observations and QoE feedbacks. The belief of the packet loss rate is defined as $b : [0, 1] \rightarrow [0, 1]$. The function $b(.)$ represents the probability distribution of the packet loss rate.

**Observed information and observation probability**: The observed information is defined by congestion events $o_n \in O_n$. The observation probability is defined as a function $\Omega_n : \mathcal{T}_n \times O_n \rightarrow [0, 1]$. Let $\Omega_n^{k-1}(o_n = fail|w_n)$ represent the probability of packet loss when the congestion window size is $w_n$ at the $(k-1)$-th epoch.

The conventional POMDP updates the belief function per time slot (RTT), but in the proposed POMDP framework, $b_n(p_n^k)$ is updated per epoch. In fact, the belief distribution is kept the same within the epoch, which reduces the computational complexity and also the memory requirement for calculating the optimal policy. Note that by updating the belief per epoch, we also tolerate delayed MOS feedbacks.

**State transition**: The average packet loss rate $p_n^k$ when using the congestion window updating policy $\mu_n^k$ at the $k$-th epoch cannot be known by $n$ until the end of the epoch. Instead, the user estimates it based on the following expression:

$$b(p_n^{k+1}|\mu_n^{k+1}) = \frac{Prob(p_n^{k+1}|p_n^k, \mu_n^{k+1})}{\sum_p Prob(p|p_n^k, \mu_n^{k+1})}, \tag{7.2}$$

where $Prob(p_n^{k+1}|p_n^k, \mu_n^{k+1})$ is the probability that the packet loss rate will be $p_n^{k+1}$ at the $(k+1)$-th epoch when choosing the policy $\mu_n^{k+1}$, given that the packet loss rate is $p_n^k$.

Based on the MOS feedbacks obtained at the end of every epoch, the user chooses the updating policy that maximizes the QoE, as illustrated in Figures 7.4 and 7.5.

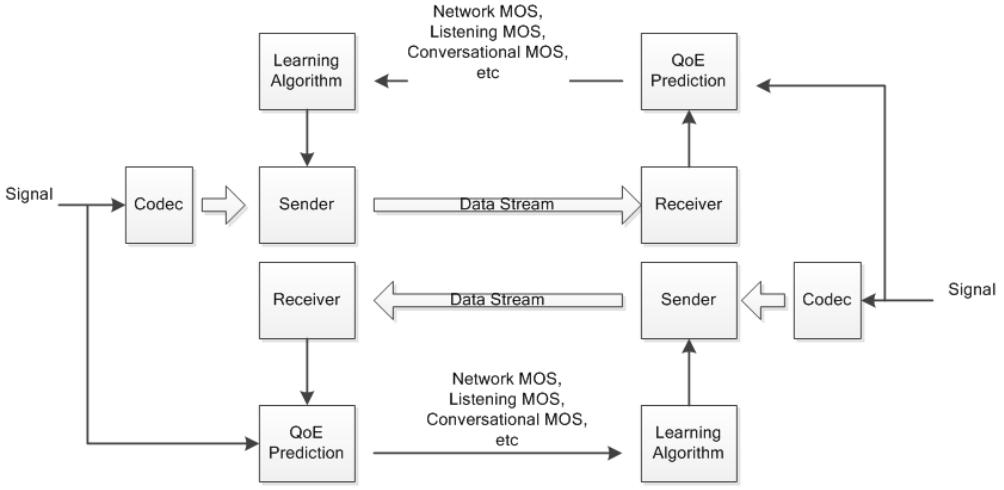


FIGURE 7.4: MOS exchange in bidirectional conversation.

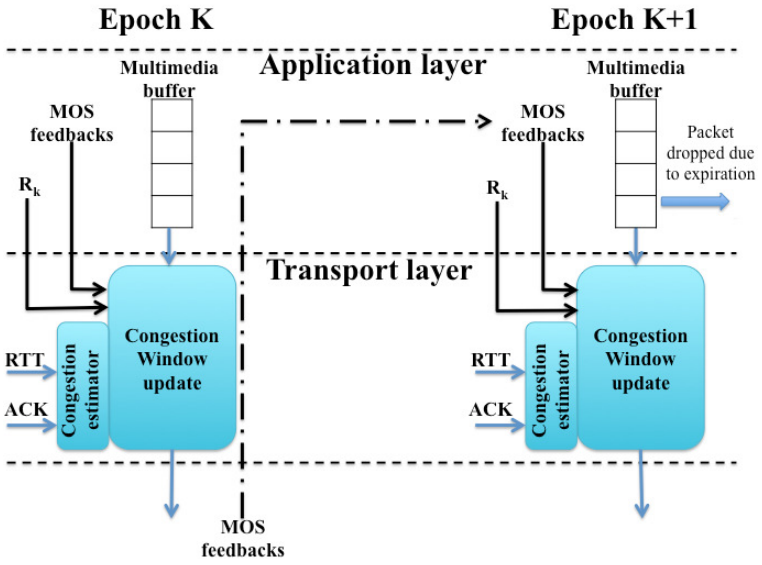


FIGURE 7.5: System diagram of MOS-TCP in time epoch $k$ and $k+1$.

## 7.5 MOS-based POMDP algorithm

We propose, in this section, a POMDP-based algorithm in order to maximize the QoE for multimedia applications. Every epoch, MOS-TCP users receive three feedbacks: NetworkMOS, ListeningMOS and ConversationalMOS. These feedbacks reflect the listener's satisfaction, and the user has to choose the action that improves the total expected QoE. Therefore, based on these feedbacks, we propose a POMDP-based algorithm that maximizes the expected QoE. Furthermore, as solving POMDPs is an extremely difficult computational problem, we present a low computation complexity online learning algorithm in order to solve the proposed congestion control. Note that learning algorithms are very useful in wireless systems as they require low complexity.

### 7.5.1 Packet-loss differentiation

The main obstacle, that wireless networks have to face, is physical impairments. The fast recovery algorithm solves the single packet loss within one window. However, due to the nature of wireless networks, a fading channel may cause contiguous packet losses. Therefore, the key idea of designing a wireless TCP is to distinguish the cause of packet loss. Many schemes were proposed in the literature. For example, TCP Veno [32] estimates the backlogged packets in the buffer of the bottleneck link. It determines the optimal throughput the network can accommodate based on the minimal RTT. The difference between the optimal throughput and the actual throughput can be used to derive the amount of backlogged packets. TCP Veno suggests that the loss is said to be random if the number of backlogged data is below a threshold, otherwise the loss is considered as congestive. In our congestion control algorithm, we implement the same methodology, depicted in Algorithm 1, in order to distinguish random packet loss from congestive loss.

### 7.5.2 The objective function

Since our objective is to avoid the congestion at bottleneck links and improve the QoE, the MOS represents a consistent feedback that gives us information about the impact of the congestion status on the multimedia quality. MOS feedbacks vary with the packet loss rate and the jitter interval, which are related to the congestion status at the bottleneck links. The higher the MOS, the better the QoE and the lower the packet loss rate and the jitter interval. Note that our objective is to maximize the total expected received MOS. Depending on the multimedia application, the user maximizes the expected QoE using NetworkMOS, ListeningMOS or ConversationalMOS feedback. All these MOS feedbacks depend on the packet loss rate and on the jitter interval, both of which depend on the source rate and the congestion window updating policy.

### 7.5.3 The optimal policy

A policy that maximizes $U_n$ is called an optimal policy $\mu_n^{opt} = \{\mu_n^{opt,1}, \mu_n^{opt,2}, ...\}$, it specifies for each epoch $k$ the optimal updating policy $\mu_n^{opt,k}$. The optimal value function $U_n^{opt}$ satisfies the following Bellman equation:

$$U_n^{opt,k}(p_n^k) = \max_{\mu_n^k \in \mathcal{A}} \{u_n^k(\mu_n^k, R_n^k) + \gamma \sum_{p'} b(p_n^k) T(p'|p_n^k) J_n^{k+1}(p')\}. \tag{7.3}$$

The optimal policy at the $k$-th epoch is expressed as follows:

$$\mu_n^{opt,k} = \arg\max_{\mu_n^k \in \mathcal{A}} \{u_n^k(\mu_n^k, R_n^k) + \gamma \sum_{p'} b(p_n^k) T(p'|p_n^k) J_n^{k+1}(p')\}. \quad (7.4)$$

### 7.5.4 Online learning

Solving the presented POMDP is expensive in terms of time (calculation) and space (memory) complexity. Then, it is not suitable for wireless systems with small capacity multimedia devices. In this section, we present a low-complexity online learning algorithm. Our online learning is an extension of the on-policy TD-$\lambda$ algorithm Sarsa [121] for POMDPs.

Each MOS-TCP user estimates the state-values $Q(\mu_n^k, R_n^k, p_n^k)$, defined as the expected future reward starting from state $(R_n^k, p_n^k)$ and taking the action $\mu_n^k$. The MOS-TCP user chooses, at every epoch, the optimal policy based on Algorithm 3. Specifically, this algorithm supports delayed MOS feedbacks, as it changes the congestion window updating policy per epoch. As illustrated in Figure 7.5, the user gets some feedbacks at the end of each epoch, which reflects the impact of the network on the listening quality. Therefore, the user applies the online learning algorithm in order to choose the congestion window policy that maximizes the expected future MOS starting from the current slot. At the beginning of epoch $k$, the user receives the source rate $R_n^k$ from the upper layer and selects the congestion window updating policy that maximizes its state-value function. Then, the user transmits its packets during the epoch using the chosen policy. At the end of the epoch, the user computes the packet loss rate and updates the state-value function $Q(\mu_n^k, R_n^k, p_n^k)$ based on the observed MOS feedback. Moreover, the user updates the belief probability of the packet loss rate. Depending on the MOS feedback considered in the objective function, we denote Network-CC the MOS-TCP algorithm that maximizes the NetworkMOS, Listening-CC the MOS-TCP that maximizes the ListeningMOS, and Conversational-CC the algorithm that maximizes the expected ConversationalMOS.

### 7.5.5 Implementation and complexity

Although the value iteration algorithms give exact solutions for the POMDP optimization problems (see [34]), they need expensive time and space complexities. In fact, the sender needs a large storage space, and spends an exponential time when seeking for the optimal policy. As we can see in Table 7.1, the complexity of the exact POMDP solutions grows exponentially with the number of epoch. Importantly, our online learning algorithm can be implemented on mobile devices that do not have a sophisticated

**Algorithm 3** MOS-TCP online learning algorithm for POMDP-based congestion control

Initialize $Q(\mu_n^k, R_n^k, p_n^k) = 0$;
$k \leftarrow 1$;
Get application parameters $R_n^1$;
Choose arbitrarily the updating policy $(\mu_n^1)$;
**while** true **do**
  **for** $t = 1 \rightarrow T$ **do**
    Update the congestion window using policy $\mu_n^k$ ;
    Update the observation probability $\Omega_n^k$ based on congestion event observation;
  **end for**
  Evaluate the packet loss rate $p_n^k$;
  The user gets the QoE feedbacks: MOS;
  Update the beliefs based on Equation (7.2);
  Get application parameters $R_n^{k+1}$;
  Choose updating policy

$$(\mu_n^{k+1}) = \arg\max_{\mu_n^{k+1}} Q(\mu_n^{k+1}, R_n^{k+1}, p_n^{k+1}) b_n(p_n^{k+1}),$$

  with probability $(1 - \epsilon)$;
  Else choose a random policy in $\mathcal{A}$;

$$Q(\mu_n^k, R_n^k, p_n^k) \leftarrow Q(\mu_n^k, R_n^k, p_n^k) + \alpha[MOS + \gamma \times \sum_{p_n^{k+1}} Q(\mu_n^{k+1}, R_n^{k+1}, p_n^{k+1}) b_n(p_n^{k+1}) - Q(\mu_n^k, R_n^k, p_n^k)];$$

  $k \leftarrow k + 1$;
**end while**

TABLE 7.1: Comparisons of exact POMDP solution and the proposed online learning algorithms

| | **Exact solution** | **MOS-TCP** |
|---|---|---|
| Consumed Memory | $O(\lvert\mathcal{A}\rvert\lvert\mathcal{V}_{k-1}\rvert^{\lvert\mathcal{O}\rvert})$, with $\mathcal{V}_k$ is the solution of the $(k-1)$-th epoch | $O(\lvert\mathcal{A}\rvert\lvert\mathcal{X}\rvert)$ |
| Time complexity | $O(\lvert\mathcal{X}\rvert^2\lvert\mathcal{A}\rvert\lvert\mathcal{V}_{k-1}\rvert^{\lvert\mathcal{O}\rvert})$ | $O(\lvert\mathcal{A}\rvert\lvert\mathcal{X}\rvert)log(\lvert\mathcal{A}\rvert\lvert\mathcal{X}\rvert)$ |

calculation or a large memory space. Moreover, the proposed algorithm is implemented only on the transmitter side and is transparent to the receiver side. We do not even require any change at the routers. Interestingly, this algorithm supports the delay of MOS feedbacks as it updates the congestion window updating policy per epoch.

## 7.6 Numerical illustrations

### 7.6.1 Testbed experiments

Microsoft Lync is an integrated software-based communication and collaboration platform, which is mainly designed for enterprise users. It provides various real-time communication features such as instant messaging, software-based voip, and video/audio conferencing through the same user interface. The system includes a set of server components that can be deployed in the enterprise network. After installing the client-side component, enterprise users can initiate audio/video calls with others or set up a group conference through the IP network. Furthermore, it supports communications with traditional phone through some PSTN gateway.

The system supports the standard Session Initiate Protocol (SIP) for signaling and RTP/RTCP protocols for transmitting media packets. For the two-way communications, clients can directly connect with each other and transmit data in a peer-to-peer way. For multi-users conferencing sessions, a Multimedia Controller Unit (MCU) server can help to coordinate the session and to replicate data packets to all receivers. When users are behind some Network Address Translator (NAT) or firewalls, a mediation server allows clients to relay data packets. The MOS prediction module in Lync is implemented at the application layer and is independent on the transport protocol. The underlying transport protocol in Lync can be TCP, UDP, or even server-relayed tunnels (e.g. Traversal Using Relay NAT (TURN) protocol), depending on the connectivity of the Lync clients.

The proposed algorithm is implemented only at the sender side, and is transparent to routers and receiver. However, an end-to-end signaling mechanism needs to be implemented at the application layer on both the transmitter and the received side. Note that a library-based MOS feedback mechanism can be adopted to help developers of multimedia applications to design QoE-based multimedia applications without the need of run-time training and signaling.

MOS feedbacks need to be sent from the receiver side to the sender at every epoch. The MOS prediction and feedback are located at the application layer. Thus, there is no need to modify the receiver part of the TCP code. Meanwhile, the TCP sender part can be designed to be backward compatible, i.e. the sender works in normal mode when there is no MOS feedback and switches to the MOS-based congestion control mode only when the application layer has indicated it to do so. In this way, MOS-TCP clients can still interact with the old non-MOS version ones.

In our experiments, the QoE trace is captured and anonymized from a deployed Microsoft Lync 2010 Service in Microsoft Labs. The duration of the collected trace is about three

months. The average length of each session is 11 minutes. From the original trace, we extract only the PC-to-PC audio streams since it reflects the voice quality over pure IP networks. The extracted part contains 1,935,110 end-to-end audio streams in total. The audio codec used by clients is Microsoft RTAudio Speech codec with the clock rate 16KHz. We use the Gilbert model to model the wireless channel conditions. This approach was introduced in [138]. By generating synthetic traces that simulate the wireless network being tested, multiple users can access the network simultaneously and perform experiments.

We consider a set of policies $\mathcal{A}$ composed of AIMD, IIAD and SQRT, defined as follow:

$$\text{AIMD: } f(w) = \frac{3\beta}{2-\beta} \text{ and } g(w) = \beta;$$

$$\text{IIAD: } f(w) = \frac{3\beta}{2w-\beta} \text{ and } g(w) = \frac{\beta}{w};$$

$$\text{SQRT: } f(w) = \frac{3\beta}{2\sqrt{w+1}-\beta} \text{ and } g(w) = \frac{\beta}{\sqrt{w+1}};$$

where $\beta \in \{0.1, 0.2, ..., 0.9\}$. Note that the conventional TCP is AIMD(0.5). We compare our proposed algorithms with other congestion control algorithms for multimedia applications. We focus, especially, on AIMD and Binomial congestion control algorithms. In fact, authors of [113] proved that the AIMD-based Binomial congestion control algorithms IIAD and SQRT are well-suited for multimedia applications. We consider that the data is transmitted over an IEEE 802.11a wireless link and the playback delay is 200 ms. We use IEEE 802.11a in our numerical study only for illustrative purposes, and any kind of wireless device can be used instead.

### 7.6.2 Unidirectional communications

In this section, we focus on the unidirectional communications with a speaker and a listener in each session. We present a comparative study between Listening-CC, Network-CC and other congestion control algorithms. We compare, in different scenarios, the QoE (ListeningMOS) and we consider the following congestion control algorithms: Listening-CC, Network-CC, Binomial congestion control and AIMD algorithms. We do not compare with Conversational-CC as we are considering unidirectional communications. We consider that each pair is composed of a transmitter (speaker) and a receiver (listener). Let us focus on the first scenario of Table 7.2. We run audio transmissions with different source rates and we plot, in Figure 7.6, the obtained QoE for different type of users.

TABLE 7.2: Experimental scenarios in unidirectional communications

| | AIMD | IIAD | SQRT | NetworkCC | ListeningCC |
|---|---|---|---|---|---|
| Scenario 1 | 2 | 2 | 2 | 2 | 2 |
| Scenario 2 | 4 | 4 | 4 | 4 | 4 |
| Scenario 3 | 8 | 8 | 8 | 8 | 8 |

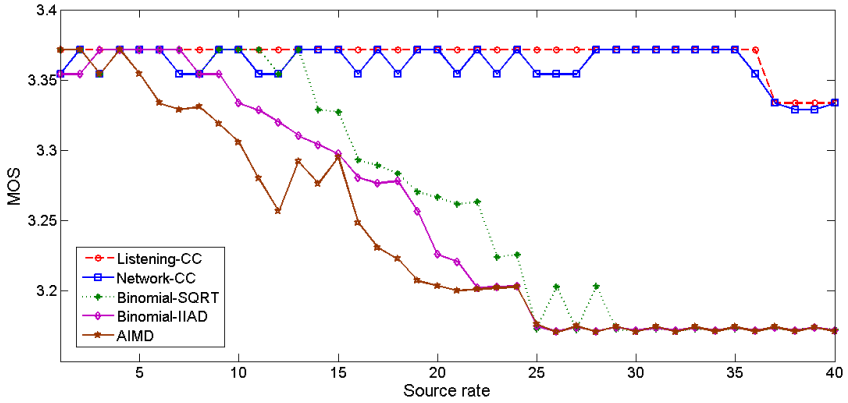


FIGURE 7.6: ListeningMOS with different source rates in the first scenario.

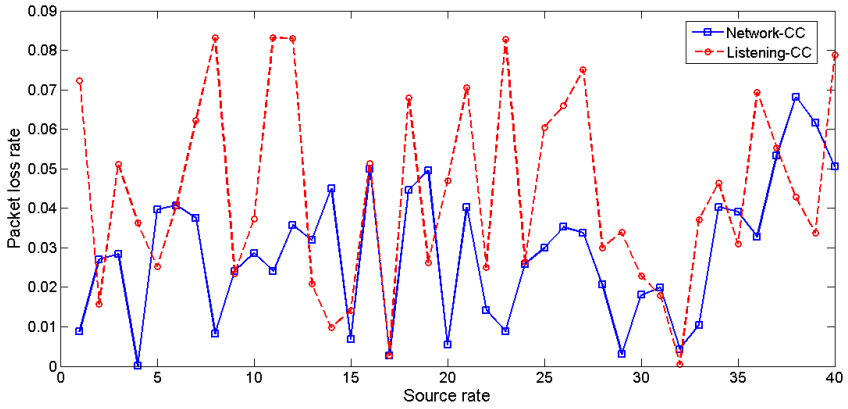


FIGURE 7.7: Packet loss rate depending on the source rate in the first scenario.

We observe that the Listening-CC and Network-CC algorithms improve significantly the QoE compared to AIMD and Binomial congestion control algorithms. Moreover, the MOS obtained with Listening-CC is slightly better than the MOS obtained by the Network-CC algorithm. Furthermore, as we can see in Figure 7.7, the packet loss rate for Listening-CC users is higher than Network-CC. In fact, as the NetworkMOS depends only on network factors, maximizing this MOS minimizes the packet loss rate and the jitter interval. However, Listening-CC bases on ListeningMOS, which depends on other factors than the network ones. Then, users can choose a policy that maximizes the ListeningMOS even with higher values of packet loss rate and jittering. Consider the second scenario of Table 7.2. We observe, in Figure 7.8, that Listening-CC and

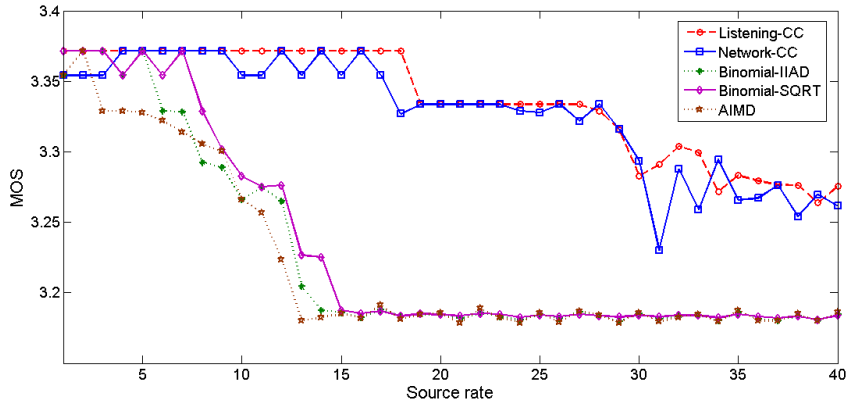


FIGURE 7.8: ListeningMOS with different source rates in the second scenario.

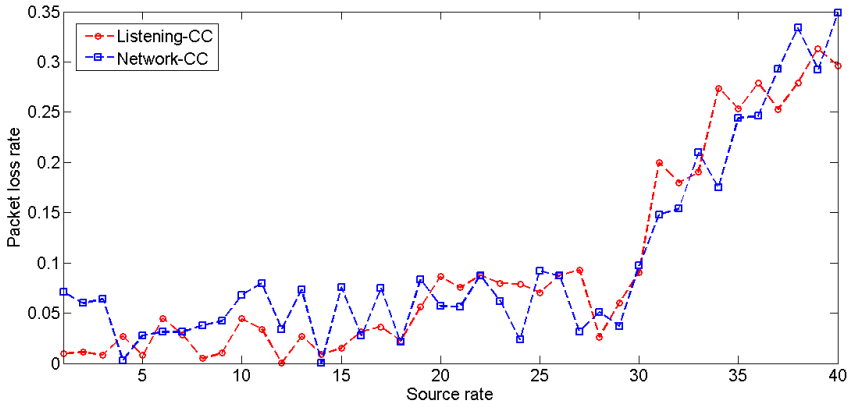


FIGURE 7.9: Packet loss rate depending on the source rate in the second scenario.

Network-CC algorithms lead to better QoE than Binomial and AIMD users. Moreover, Listening-CC leads to slightly better QoE than Network-CC. Figure 7.9 illustrates that the packet loss rate for both Listening-CC and Network-CC algorithms is increasing with the source rate as the bottleneck link become overloaded. The fluctuation of packet loss rate is due to the imperfect characteristics of the wireless link. In the third scenario of Table 7.2, we consider more load on the bottleneck link. Figures 7.10 and 7.11 illustrates the ListeningMOS and the packet loss rate for different congestion control algorithms. It is clear that the MOS-TCP frameworks lead to better QoE. However, the improvement decreases with the source rate, and all congestion control algorithms give the same QoE for high values of source rate. In fact, with such number of audio sessions and source rates, the wireless link is always overloaded and the source rates requested by users cannot be satisfied. Therefore, the packet loss rate increases for all the users, and the QoE decreases. In summarize, both Listening-CC and Network-CC algorithms improve the QoE compared to other AIMD-based congestion control algorithms for multimedia transmission. Moreover, Listening-CC is slightly better than Network-CC algorithm,

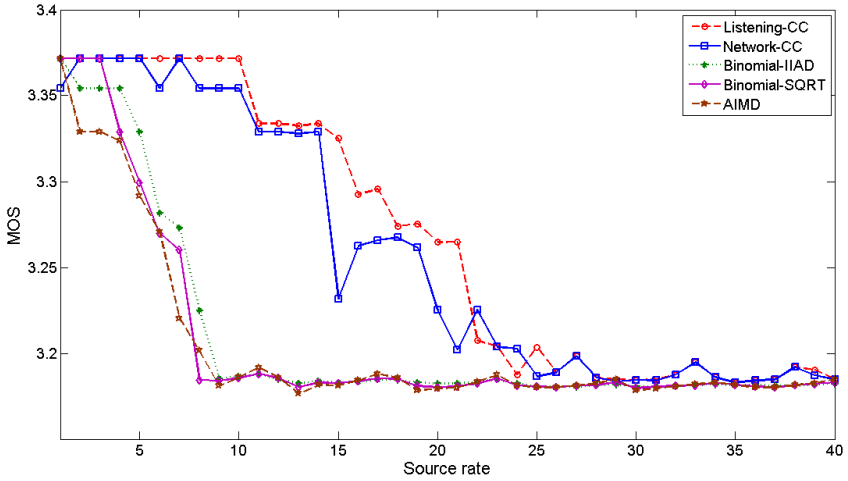


FIGURE 7.10: ListeningMOS with different source rates in the third scenario.

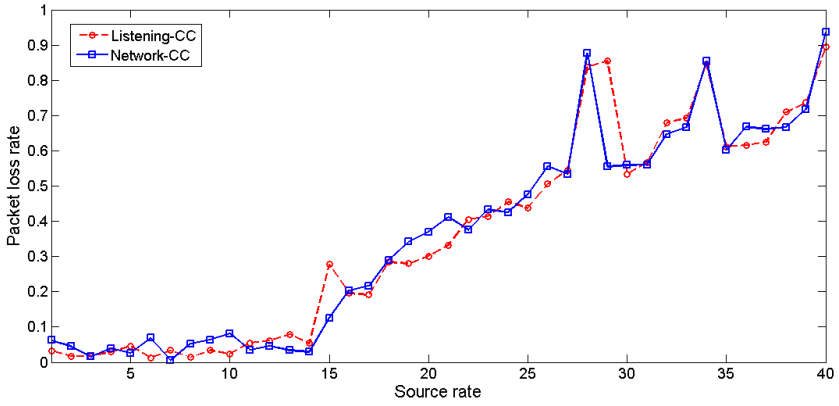


FIGURE 7.11: Packet loss rate depending on the source rate in the third scenario.

TABLE 7.3: Experimental scenarios in bidirectional communications

| | AIMD | IIAD | SQRT | NetworkCC | ListeningCC | ConversationalCC |
|---|---|---|---|---|---|---|
| Scenario 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| Scenario 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| Scenario 3 | 8 | 8 | 8 | 8 | 8 | 8 |

as it considers not only packet loss rate and jitter but also the impact of non-network factors.

### 7.6.3 Bidirectional communications

We consider, in this section, bidirectional audio conversations. We run the three scenarios presented in Section 7.6.2 with a bidirectional communication.

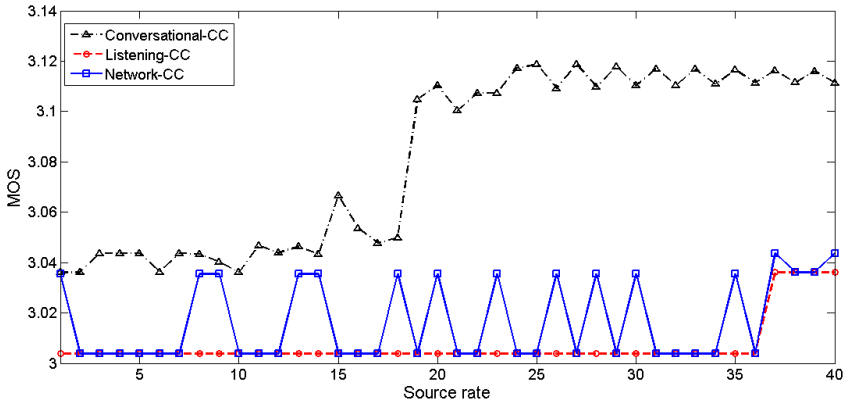


FIGURE 7.12: ConversationalMOS with different source rates in the first scenario.

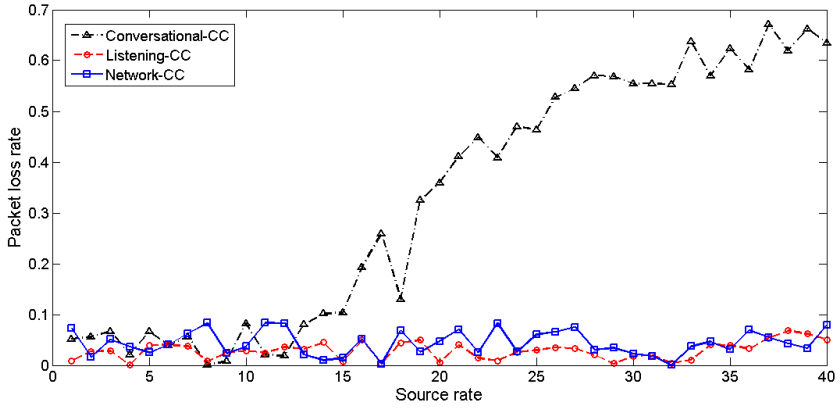


FIGURE 7.13: Packet loss rate depending on the source rate in the first scenario.

In the first scenario of Table 7.3, we run the conversations over the wireless link. Figure 7.12 illustrates the QoE (conversationalMOS) for Conversational-CC and other congestion control algorithms with different values of the source rate. We observe that the Conversational-CC leads to better QoE than Listening-CC and Network-CC algorithms. Surprisingly, the improvement of Conversational-CC compared to Listening-CC and Network-CC algorithms is more important for higher source rate. In fact, for high values of source rate, we observe, in Figure 7.13, that Conversational-CC algorithm is more aggressive than other congestion control algorithms as it leads to significantly higher packet loss rate.

In the second scenario of Table 7.3, we plot, in Figures 7.14 and 7.15, the Conversational-MOS and the packet loss rate for different congestion control algorithms depending on the source rate. We observe that the Conversational-CC algorithm outperforms other congestion control algorithms. Moreover, we remark that for some values of the source

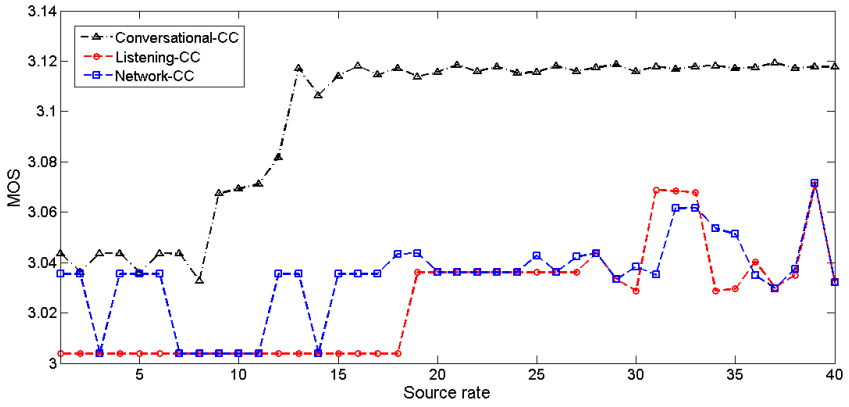


FIGURE 7.14: ConversationalMOS with the source rates in the second scenario.

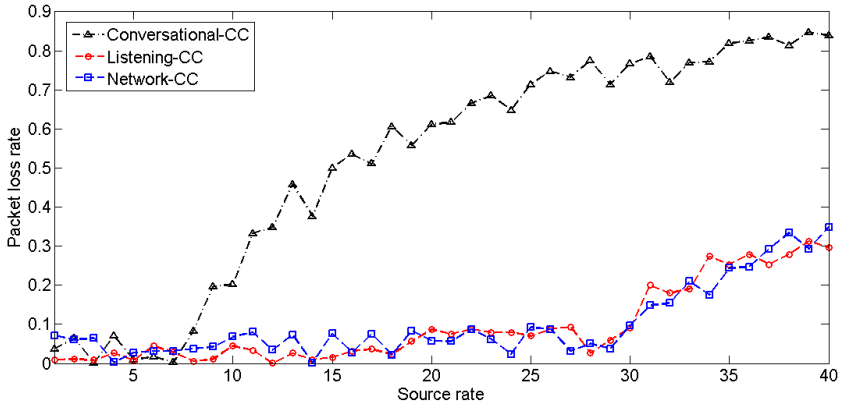


FIGURE 7.15: Packet loss rate depending on the source rate in the second scenario.

rate, the Listening-CC in better than Network-CC and for other values Network-CC is better.

Let us focus on the third scenario of Table 7.3. Figure 7.16 shows that the Conversational-CC leads to better QoE than other congestion control algorithms. In fact, it bases on the conversationalMOS feedback which takes into consideration both sent and received audio streams, and is less sensitive to the network factors, such as packet loss rate and jittering, than ListeningMOS and NetworkMOS. However, as we can see, in Figure 7.17, when the Conversational-CC algorithm is higher than Listening-CC and Network-CC, it leads to higher packet loss rate. Indeed, when the wireless link is overloaded, all congestion control algorithms lead to worst performances. Finally, the Conversational-CC is more suitable for bidirectional communications.

Although the improvements in MOS do not seem to be very large (0.1-0.3) in absolute values, the relative improvements are actually significant. In the practical system (e.g., Microsoft Lync), only few users have MOS values below 3 or above 4. The dynamic

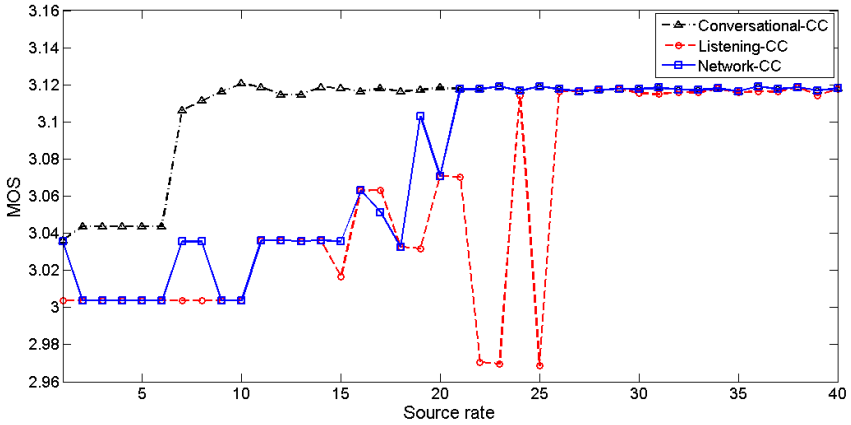


FIGURE 7.16: ConversationalMOS with different source rates in the third scenario.

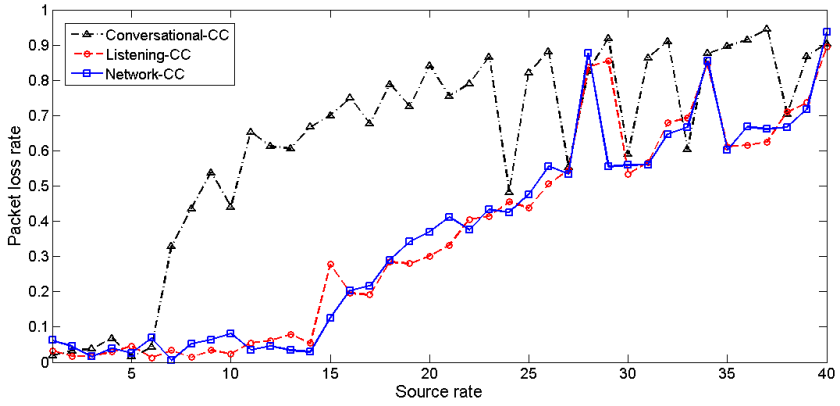


FIGURE 7.17: Packet loss rate depending on the source rate in the third scenario.

range of the MOS values is about 1.0. The region between 3.0 and 4.0 is a quite sensitive interval of MOS for users. Our improvement is about 10% to 30% in the range. Because the sessions in our traces are using the same audio codec and software version, this means that the actual degradation of ListeningMOS is relatively small. However, if we focus on the NetworkMOS, the improvements are significant. As we can see in Figure 7.18, the improvement of MOS-TCP user is about 1 in NetworkMOS.

## 7.7 Conclusion

We have formulated, in this chapter, the QoE-aware congestion control problem as a POMDP that maximizes the QoE for multimedia users. We have considered a set of generic AIMD-like updating functions for the congestion window. The optimal policy allows the sender to optimize the congestion window updating policy that maximizes the expected discounted QoE. We have also proposed an online learning method to solve

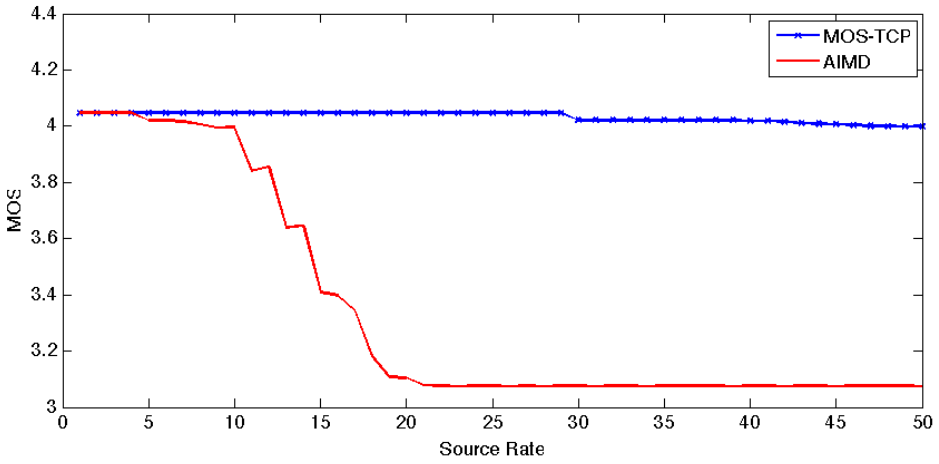


FIGURE 7.18: NetworkMOS for MOS-TCP user and a AIMD user.

the MOS-TCP on-the-fly. Experimental results have shown that the proposed algorithm outperforms other congestion control schemes in terms of QoE. Note that the friendliness of MOS-TCP can be studied similarly to the study of Learning-TCP in Chapter 6, and will be a part of our future work.

# Chapter 8

# Conclusion and perspectives

## 8.1 Summary of contributions

In this dissertation, we have proposed some applications of game theory, self-adaptivity and learning in wireless networks at different layers of the protocol stack. We have focused, first, on the MAC layer, and we have studied the OSA in CR networks. We have studied both the slotted and the non-slotted models using a queueing analysis. In fact, we have proposed centralized and decentralized OSA mechanisms for SUs in CR networks, and we have proved the existence of a NE between SUs. Moreover, we have evaluated the gap of performance between the optimal solution, obtained from the centralized system, and the performance at the NE, obtained from the decentralized system, using the well-studied metric: the price of anarchy.

Secondly, we have used a POMDP framework in order to design an optimal OSA policy for SUs in CR networks, taking into account energy and delay constraints. Introducing a QoS metric in the spectrum sensing policy is very important with the emergence of heterogeneous mobile devices that are able to transmit their traffic over different mediums of communication like 3G, WiFi and TV White Space. We have provided some structural properties of the value function, and we have proved the existence of an optimal stationary OSA policy that has threshold structure. We have been able to determine explicitly the threshold structure of the optimal policy.

Furthermore, we have focused on the non-cooperative OSA mechanism for CR networks, and we have considered that SUs are in competition in order to access a licensed channel. Specifically, we have proved the existence of an SNE multi-policy for the OSA problem, modeled as a non-cooperative game between SUs, and that the attempt rate at the SNE is unique. Simulation results have shown that more opportunities in the spectrum

may decreases the average throughput of the system due to the aggressiveness and the competition between SUs. In fact, we have found a Braess kind of paradox, where reducing system resources induce better performance. In order to optimize the average throughput of the system, we have proposed a Stackelberg game model for the network manager. Furthermore, we have shown that a Stackelberg equilibrium strategy for our problem exists. This strategy is defined by increasing the average time that the licensed channel is occupied.

We have also proposed two learning and knowledge extraction mechanisms. Specifically, we have presented two learning-based protocols for SUs in order to estimate licensed channels' dynamics: rate estimator, and transition matrices estimator.

Thereafter, we have focused on the transport layer, and we have formulated the media-aware congestion control problem as a POMDP that considers the distortion impact, delay deadline and the multimedia source rate. We have considered a set of generic TCP-friendly updating functions, in order to optimize the congestion window adaptation by maximizing the long term expected quality of multimedia applications. Moreover, we have proposed an online learning method to solve the POMDP on the fly. Simulation results have shown that the proposed congestion control algorithm outperforms conventional TCP-friendly congestion control schemes in terms of quality, especially for real-time applications with hard delay deadlines. Moreover, the proposed algorithm is implemented only at the sender side, and is transparent to routers and receivers.

Finally, we have formulated the QoE-aware congestion control problem as a POMDP that maximizes the QoE for multimedia users. We have considered a set of generic AIMD-like updating functions for the congestion window. The optimal policy allows the sender to optimize the congestion window size in order to maximize the long term expected QoE. We have also proposed an online learning method to solve the MOS-TCP on-the-fly. Experimental results have shown that the proposed algorithm outperforms other congestion control schemes in terms of QoE.

## 8.2 Perspectives

### 8.2.1 Cooperative OSA in CR networks

It is well known that sensing the licensed spectrum is time and energy consuming. Therefore, a cooperative OSA for SUs is welcome. Specifically, we may consider that SUs decides individually whether to cooperate or not with each other, in order to maximize their own benefits. Note that a SU obtains more information about the spectrum

occupancy if it cooperates with other SUs. However, it may have lower collision probability over a free licensed channel if it does not cooperate. In fact, if SUs does not cooperate, other SUs ignore, generally, the status of the licensed channel sensed by that user. Intuitively, there is a tradeoff for SUs between cooperating or not with each other. This model can be studied using a game-theoretical approach. Moreover, depending on the information exchanged between SUs, we may use Decentralized POMDP (Dec-POMDP) or Interactive POMDP (I-POMDP) in order to design the optimal OSA policy. In fact, if SUs exchange only the spectrum information, the optimal solution may be obtained using a I-POMDP. However, if SUs exchange their internal states and beliefs, Dec-POMDP determines the optimal OSA policy. Several challenges may be considered in this model, such as message exchange protocol, the cost of sending messages, the cooperative OSA, the impact of the exchanged information on the performances of SUs, etc.

### 8.2.2 CR in TV white spaces

The TVWS are located in the VHF and UHF bands, and have some characteristics that make them highly motivating for wireless communications. In fact, the TVWS have the ability to cover a significant area with a relatively lower cost. Moreover, the Non-line-of-sight performance of TVWS offers SUs the ability to penetrate obstacles such as trees and buildings. Note that the new spectrum licensing allows unlicensed users to access the spectrum as long as they do not harm the licensed users. Therefore, SUs need to communicate with a database to obtain a list of currently available TVWS.

In our future works, we consider CR base stations that sense a subset of the spectrum in order to locate some free frequencies. Thereafter, a SU that need to communicate through the TVWS send a request to a CR base station to obtain information about free licensed channels. Note that CR base stations may be either cooperating or competing in order to take advantage from opportunities in the TVWS. We may also consider that CR base stations sense the TVWS and sublease available channels for SUs that seek for opportunities in the licensed spectrum. Intuitively, there is a tradeoff between the number of sensed channels, as CR base stations care about the sensing cost and the price that a CR base station charges for SUs. We study the cooperation between CR base station and we analyze the performance of SUs as well as the benefit of CR base station. Furthermore, we focus on the non-cooperative model, and we study the impact of the competition between CR base stations on the benefit of each other. We also study the impact of the competitive behavior of CR base stations on the performance of SUs.

### 8.2.3 Media-aware TCP congestion control

Evolutionary games have been developed in biological sciences in the aim of studying the evolution and equilibrium behavior (called Evolutionary Stable Strategies ESS) of large populations. As the TCP is widely adopted building block in many network services, there is a large population using TCP-based congestion control algorithms. Note that the number of users that uses Learning-TCP has an impact on the expected performances of network users. In our future works, we focus on the impact of the fraction of users that uses Learning-TCP on the overall performances of bottleneck links, and on the performance of Learning-TCP users. Specifically, this system can be modeled using evolutionary game theoretic approach. Furthermore, an interesting perspective is to evaluate the impact of multiple flows (multiple learning-TCP connections) on the performance of Learning-TCP users, as well as the fairness between TCP flows and Learning-TCP flows.

The proposed QoE-based adaptation can be straightforwardly extended to other kind of content such as video applications. The only difference is that video or graphics based QoE feedback is needed to train the QoE-decision based engine, which adapts the TCP transmission. Moreover, we will develop a library-based MOS feedback mechanism, which may be adopted in order to help developers of multimedia applications to design QoE-based multimedia applications without the need of run-time training and signaling.

The proposed QoE-based congestion control algorithm can be extended to support a wider set of applications. In fact, we will propose a QoE-based congestion control mechanism for multicast multimedia streaming applications. Moreover, we may use in our MOS-TCP an aggregation of different kind of MOS feedback: NetworkMOS, ListeningMOS and conversationalMOS. This approach will be tested in our future works.

# Appendix A

# Publications of the thesis

## A.1 Journal papers:

1. Oussama Habachi, Yezekael Hayel, and Rachid El Azouzi, "Optimal Energy-Delay Tradeoff for Opportunistic Spectrum Access in Cognitive Radio Networks", submitted to Computer Networks, 2012

2. Oussama Habachi, Rachid El Azouzi, and Yezekael Hayel, "A Stackelberg Model for Opportunistic Sensing in Cognitive Radio Networks", submitted to Transactions on Wireless Communications, 2012

3. Ousssama Habachi, Hsien-Po Shiang, Mihaela van der Schaar and Yezekael Hayel, "Online Learning based Congestion Control for Adaptive Multimedia Transmission", submitted to Transactions on Signal Processing, 2012

4. Oussama Habachi, Yusuo Hu, Mihaela van der Schaar, Y. Hayel and Feng Wu, "MOS-based Congestion Control for Conversational Services in Wireless Environments", accepted in IEEE Journal on Selected Areas in Communications (JSAC), 2012

5. Oussama Habachi and Yezekael Hayel, "Optimal Opportunistic Sensing in Cognitive Radio Networks", accepted in IET Communications, special issue on Cognitive Communications, 2012

## A.2 Conference papers:

1. Oussama Habachi, Yezekael Hayel, and Rachid El Azouzi, "Optimal Energy-Delay Tradeoff Policies in Cognitive Radio Networks", IEEE Globecom 2012

2. Oussama Habachi, Yusuo Hu, Mihaela van der Schaar, Y. Hayel and Feng Wu, "QoE-aware Congestion Control Algorithm for Conversational Services", IEEE International Conference on Communication (ICC), 2012

3. Oussama Habachi, Rachid El Azouzi, and Yezekael Hayel, "Distributed energy-delay framework for opportunistic spectrum access", IEEE Wireless Days, 2011

4. Oussama Habachi and Y. Hayel, "Optimal sensing strategy for opportunistic secondary users in a cognitive radio network", in the 13th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2010

# Bibliography


[1] Joseph Mitola III. Cognitive radio: An integrated agent architecture for software defined radio dissertation. *Scientific American*, 294(3):66–73, 2000. URL http://www.ncbi.nlm.nih.gov/pubmed/20007053.

[2] Simon Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Area in Communications*, 23, Feb. 2005.

[3] Petri Mähönen, Janne Riihijärvi, Marina Petrova, and Zach Shelby. Hop-by-hop toward future mobile broadband IP. *IEEE Communications Magazine*, 42(3): 138–146, 2004.

[4] Chris Ramming. Cognitive Networks. In *DARPATech*, 2004. URL http://www.darpa.mil/DARPAtech2004/pdf/scripts/RammingScript.pdf.

[5] Robert Berezdivin and Robert Breinig. Next-generation wireless communications concepts and technologies. *IEEE Communications Magazine*, 40:108–116, 2002.

[6] Angeliki Alexiou and Martin Haardt. Smart antenna technologies for future wireless systems: trends and challenges. *IEEE Communications Magazine*, 42 (9):90–97, 2004. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1336725.

[7] 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*, 50:2127–2159, 2006.

[8] FCC Spectrum Policy Task Force. Report of the spectrum efficiency working group. Nov., 2002. URL http://www.fcc.gov/sptf/reports.html.

[9] Bing Wang, Jim Kurose, Prashant Shenoy, and Don Towsley. Multimedia streaming via tcp: An analytic performance study. *ACM Transaction Multimedia Computer Communication Application*, 4(2):16:1–16:22, May 2008. ISSN 1551-6857. doi: 10.1145/1352012.1352020. URL http://doi.acm.org/10.1145/1352012.1352020.

[10] Jitendra Padhye, bbVictor Firoiu, Donald F. Towsley, and James F. Kurose. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, 2000.

[11] Alex Balk. Adaptive video streaming: pre-encoded mpeg-4 with bandwidth scaling. *Computer Networks*, 44(4):415–439, 2004. URL http://linkinghub.elsevier.com/retrieve/pii/S138912860300433X.

[12] Jianwei Huang, Randall A. Berry, and Michael L. Honig. Spectrum sharing with distributed interference compensation. In *Proc. of IEEE DYSPAN*, pages 88–93, 2005.

[13] Dusit Niyato and Ekram Hossain. Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of nash equilibrium, and collusion. *IEEE Journal on Selected Areas in Communications*, 26(1):192 –202, jan. 2008. ISSN 0733-8716. doi: 10.1109/JSAC.2008.080117.

[14] Chunyi Peng, Haitao Zheng, and Ben Y. Zhao. Utilization and fairness in spectrum assignment for opportunistic spectrum access. *Mobile Network Application*, 11(4): 555–576, August 2006. ISSN 1383-469X. doi: 10.1007/s11036-006-7322-y. URL http://dx.doi.org/10.1007/s11036-006-7322-y.

[15] Sudharman K. Jayaweera, Gonzalo Vazquez-Vilar, and Carlos Mosquera. Dynamic spectrum leasing: A new paradigm for spectrum sharing in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59(5):2328 –2339, jun 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2042741.

[16] Haitao Zheng and Chunyi Peng. Collaboration and fairness in opportunistic spectrum access. In *Proc. of IEEE International Conference on Communications*, volume 5, pages 3132 – 3136 Vol. 5, may 2005. doi: 10.1109/ICC.2005.1494982.

[17] Omer Ileri. Demand responsive pricing and competitive spectrum allocation via a spectrum server. In *Proc. of IEEE DYSPAN*, pages 194–202, 2005.

[18] Nie Nie and Cristina Comaniciu. Adaptive channel allocation spectrum etiquette for cognitive radio networks. *Mob. Netw. Appl.*, 11(6):779–797, December 2006. ISSN 1383-469X. doi: 10.1007/s11036-006-0049-y. URL http://dx.doi.org/10.1007/s11036-006-0049-y.

[19] Beibei Wang, Yongle Wu, and K.J. Ray Liu. Game theory for cognitive radio networks: An overview. *Comput. Netw.*, 54(14):2537–2561, October 2010. ISSN 1389-1286. doi: 10.1016/j.comnet.2010.04.004. URL http://dx.doi.org/10.1016/j.comnet.2010.04.004.

[20] ET Docket No. 02-380 FCC. ET Docket No. 04-186. Second report and order and memorandum opinion and order. Nov. 2008.

[21] BBN Technologies. DARPA XG WG. The xg vision rfc version 2.0. 2003.

[22] Working Group on Wireless Regional Area Networks (WRANs). Ieee 802.22. URL http://ieee802.org/22/.

[23] IEEE Standards Coordinating Committee 41. Dynamic spectrum access networks. URL http://www.scc41.org/.

[24] European Telecommunications Standards Institute. Etsi reconfigurable radio. URL http://www.etsi.org/WebSite/technologies/RRS.aspx.

[25] Yonghong Zeng, Ying-Chang Liang, Zhongding Lei, Ser Wah Oh, Francois Chin, and Sumei Sun. Worldwide regulatory and standardization activities on cognitive radio. In *Proc. of IEEE DYSPAN*, pages 194–202, 2010.

[26] Ian F Akyildiz, Won-Yeol Lee, Mehmet Can Vuran, and Shantidev Mohanty. A survey on spectrum management in cognitive radio networks. *IEEE Communications Magazine*, 46(4):40–48, 2008. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4481339.

[27] Qing Zhao and Brian M. Sadler. A survey of dynamic spectrum access. *IEEE Signal Processing Magazine*, 24(3):79–89, 2007. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4205091.

[28] Ye Tian, Kai Xu, and N. Ansari. TCP in wireless environments: problems and solutions. *IEEE Communications Magazine*, 43(3):S27–S32, 2005. doi: 10.1109/MCOM.2005.1404595. URL http://dx.doi.org/10.1109/MCOM.2005.1404595.

[29] Yang Su, P. Steenkiste, and T. Gross. Performance of tcp in multi-hop access networks. In *Proc. of 16th International Workshop on Quality of Service, 2008. IWQoS 2008*, pages 181 –190, june 2008. doi: 10.1109/IWQOS.2008.26.

[30] Lawrence S. Brakmo, Sean W. O'Malley, and Larry L. Peterson. Tcp vegas: new techniques for congestion detection and avoidance. *SIGCOMM Comput. Commun. Rev.*, 24(4):24–35, October 1994. ISSN 0146-4833. doi: 10.1145/190809.190317. URL http://doi.acm.org/10.1145/190809.190317.

[31] David X. Wei, Cheng Jin, Steven H. Low, and Sanjay Hegde. Fast tcp: motivation, architecture, algorithms, performance. In *Proc. of IEEE INFOCOM*, 2004.

[32] Cheng Peng Fu and S C Liew. Tcp veno: Tcp enhancement for transmission over wireless access networks. *IEEE Journal on Selected Areas in Communications*, 21 (2), 2003.

[33] Edward J. Sondik. The optimal control of partially observable markov decision processes. *Doctoral dissertation, Stanford University*, 1971.

[34] Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5): 1071–1088, 1973. URL http://www.jstor.org/stable/168926.

[35] Edward J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26:282–304, 1978.

[36] Richard Bellman. *The theory of dynamic programming*, volume 60. Bull. Amer. Math. Soce, 1954.

[37] Leonard Kleinrock. *Queueing systems*. Wiley Interscience, 1975.

[38] John F. Nash, Jr. Equilibrium points in $n$-person games. In *Proc. of the National Academy of Sciences*, volume 36, pages 48–49. National Academy of Sciences of the United States of America, January 1950.

[39] Ashraf Al Daoud, Tansu Alpcan, Sachin Kumar Agarwal, and Murat Alanyali. A stackelberg game for pricing uplink power in wide-band cognitive radio networks. In *Proc. of the 47th IEEE Conference on Decision and Control, 2008*, pages 1422 –1427, dec. 2008. doi: 10.1109/CDC.2008.4738975.

[40] Jin Zhang and Qian Zhang. Stackelberg game for utility-based cooperative cognitive radio networks. In *Proc. of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 23–32, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-624-3. doi: http://doi.acm.org/10.1145/1530748.1530753. URL http://doi.acm.org/10.1145/1530748.1530753.

[41] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proc. of the National Conference on Artificial Intelligence*, 2004.

[42] Lloyd Shapley. Stochastic games. In *Proc. of the National Academy of Sciences*, pages 1095–1100, 1953.

[43] Harold W. Kuhn. Extensive games and the problem of information. *Annals of Mathematics Studies*, 28, 1953.

[44] Andrew G. Barto, Steven J. Bradtke, Satinder P. Singh, The Thank Rich Yee, Vijay Gullapalli, and Brian Pinette. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:81–138, 1995.

[45] Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: reinforcement learning with less data and less time. *Machine Learning*, pages 103–130, 1993.

[46] Christopher J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, Kings College, 1989. URL http://linkinghub.elsevier.com/retrieve/pii/092188909500026C.

[47] Gavin Adrian Rummery and Mahesan Niranjan. On-line q-learning using connectionist systems. Technical report, 1994.

[48] Christopher G. Atkeson and Juan Carlos Santamaria. A comparison of direct and model-based reinforcement learning. In *Proc. of International Conference on Robotics and Automation*, pages 3557–3564. IEEE Press, 1997.

[49] Amir Massoud Farahmand, Azad Shademan, Martin Jägersand, and Csaba Szepesvári. Model-based and model-free reinforcement learning for visual servoing. In *Proc. of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, pages 4135–4142, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2788-8. URL http://dl.acm.org/citation.cfm?id=1703775.1704113.

[50] Changqing Luo, F.R. Yu, Hong Ji, and V.C.M. Leung. Cross-layer design for tcp performance improvement in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59(5):2485 –2495, jun 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2041802.

[51] Keqin Liu and Qing Zhao. Learning from collisions in cognitive radio networks: Time division fair sharing without pre-agreement. In *Proc. of The 2010 Military Communications Conference*, 2010.

[52] Isameldin Suliman and J. Lehtomaki. Queueing Analysis of Opportunistic Access in Cognitive Radios. In *Proc. of CogArt'09*, pages 1–9, 2009. doi: 10.1109/INFCOM.2010.5461942. URL http://dx.doi.org/10.1109/INFCOM.2010.5461942.

[53] Yunxia Chen, Qing Zhao, and Ananthram Swami. Distributed spectrum sensing and access in cognitive radio networks with energy constraint. *IEEE Transaction on Signal Processing*, Feb. 2009.

[54] Song Gao, Li jun Qian, and Dhadesugoor R. Vaman. Distributed energy efficient spectrum access in cognitive radio wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 8, 2009.

[55] Xin Liu and Sai Shankar. Sensing-based opportunistic channel access. In *Proc. of Mobile Network Application*, pages 577–591, 2006.

[56] Mark Felegyhazi, Mario Cagalj, Shirin Saeedi Bidokhti, and Jean pierre Hubaux. Noncooperative multi-radio channel allocation in wireless networks. In *Proc. of IEEE INFOCOM*, 2007.

[57] Nguyen Duy Duong and A. S. Madhukumar. Non-cooperative power control and spectrum allocation in cognitive radio networks: a game theoretic perspective. *Wireless Communications and Mobile Computing*, pages n/a–n/a, 2012. ISSN 1530-8677. doi: 10.1002/wcm.2202. URL http://dx.doi.org/10.1002/wcm.2202.

[58] Siva Subramani andTamer Basar, Simon Armour, Dritan Kaleshi, and Zhong Fan. Noncooperative equilibrium solutions for spectrum access in distributed cognitive radio networks. *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 1–5, 2009. URL http://hdl.handle.net/1983/1480.

[59] Krishna Jagannathan, Ishai Menache, Eytan Modiano, and Gil Zussman. Non-cooperative spectrum access: The dedicated vs . free spectrum choice. *Electrical Engineering*, (July):1–12, 2011. URL http://web.mit.edu/people/gilz/pub_files/spectrum_access_MOBIHOC11.pdf.

[60] Prasanna Chaporkar and Alexandre Proutiére. Optimal joint probing and transmission strategy for maximizing throughput in wireless systems. *IEEE Journal on Selected Areas in Communications*, 26(8):1546–1555, 2008.

[61] Mohammad M. Rashid, Md. J. Hossain, Ekram Hossain, and Vijay Bhargava. Opportunistic spectrum scheduling for multiuser cognitive radio: a queueing analysis. *IEEE Transactions on Wireless Communications*, 8(10):5259–5269, October 2009. ISSN 1536-1276. doi: 10.1109/TWC.2009.081536. URL http://dx.doi.org/10.1109/TWC.2009.081536.

[62] Vamsi Krishna Tumuluru, Ping Wang, and Dusit Niyato. A novel spectrum-scheduling scheme for multichannel cognitive radio network and performance analysis, 2011. URL http://www3.ntu.edu.sg/home/WangPing/papers/TVT_vamsi.pdf.

[63] Amine Laourine, Shiyao Chen, and Lang Tong. Queuing Analysis in Multichannel Cognitive Spectrum Access: A Large Deviation Approach. In *Proc. of IEEE INFOCOM*, pages 1–9, 2010. doi: 10.1109/INFCOM.2010.5461942. URL http://dx.doi.org/10.1109/INFCOM.2010.5461942.

[64] Ioannis Krikidis, J. Nicholas Lanemanand John Thompson, and Steve McLaughlin. Stability Analysis for Cognitive Radio with Cooperative Enhancements. In *Proc. IEEE Information Theory Workshop (ITW), Volos, Greece*, 2009.

[65] Caoxie Zhang, Xinbing Wang, and Jun Li. Cooperative cognitive radio with priority queueing analysis. In *Proceedings of the 2009 IEEE international conference on Communications*, ICC'09, pages 4672–4676, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3434-3. URL http://dl.acm.org/citation.cfm?id=1817770.1818142.

[66] Hyoil Kim and Kang G Shin. Adaptive mac-layer sensing of spectrum availability in cognitive radio networks. *Electrical Engineering*, 7(5):–518–06, 2006. URL http://www.eecs.umich.edu/techreports/cse/2006/CSE-TR-518-06.pdf.

[67] Xiaohong Guan Lang Tong Xin Li, Qianchuan Zhao. On the performance of cognitive access with periodic spectrum sensing. In *Proc. of ACM workshop on Cognitive radio networks*, 2009.

[68] Ekram Hossain, Dusit Niyato, and Zhu Han. *Dynamic spectrum access and management in cognitive radio networks*. Cambridge University Press, 2009.

[69] Senhua Huang, Xin Liu, and Zhi Ding. Opportunistic spectrum access in cognitive radio networks. In *Proc. of IEEE INFOCOM*, 2008.

[70] Chunsheng Xin, Min Song, Liangping Ma, George Hsieh, and Chien-Chung Shen. On random dynamic spectrum access for cognitive radio networks. In *GLOBECOM*, pages 1–5, 2010.

[71] Peter J. Smith, Abdulla Firag, Pawel A. Dmochowski, and Mansoor Shafi. Analysis of the m/m/n/n queue with two types of arrival process: Applications to future mobile radio systems. *J. Applied Mathematics*, 2012, 2012.

[72] Yassin Belkasmi et al. Channel allocation strategies in opportunistic-based cognitive networks. In Proceedings of IEEE IWCMC, 2012.

[73] Eitan Altman, Thomas Boulogne, Rachid El-Azouzi, Tania Jiménez, and Laura Wynter. A survey on networking games in telecommunications. *Comput. Oper. Res.*, 33:286–311, February 2006. ISSN 0305-0548. doi: 10.1016/j.cor.2004.06.005. URL http://dl.acm.org/citation.cfm?id=1114732.1114734.

[74] Refael Hassin and Moshe Haviv. To queue or not to queue: Equilibrium behavior in queueing systems. Elsevier, 2003.

[75] John Glen Wardrop. Some theoretical aspects of road traffic research. In *Proc. Inst. Civil En*, pages 325–378, 1952.

[76] Christos Papadimitriou Elias Koutsoupias. Worst-case equilibria. In *Proc. of STACS'99*, 1999.

[77] Tim Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67, 2003.

[78] Stephen J. Shellhammer and Ahmed K. Sadek. Technical challenges for cognitive radio in the tv white space spectrum. In *Proc. of Information Theory and Applications Workshop*, pages 323–333. Ieee, 2009. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5044964.

[79] Husheng Li. Impact of primary user interruptions on data traffic in cognitive radio networks: Phantom jam on highway. In *GLOBECOM*, pages 1–5, 2011.

[80] Qing Zhao, Lang Tong, Ananthram Swami, and Yunxia Chen. Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A pomdp framework. *IEEE Journal on Selected Areas in Communications*, 25(3):589–600, 2007. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4155374.

[81] Hua Liu, Bhaskar Krishnamachari, and Qing Zhao. Cooperation and learning in multiuser opportunistic spectrum access. In *IEEE International Conference on Communications*, 2008.

[82] Haitao Zheng. Collaboration and fairness in opportunistic spectrum access. In *Proc. of IEEE International Conference on Communications*, pages 3132–3136, 2005.

[83] A. T. Hoang, Y. C. Liang, D. T. C. Wong, Y. Zeng, and R. Zhang. Opportunistic spectrum access for energy-constrained cognitive radios. *IEEE Transactions on Wireless Communications*, 2008.

[84] Li-Chun Wang, Yin-Chih Lu, Chung-Wei Wang, and David S. L. Wei. Latency analysis for dynamic spectrum access in cognitive radio: Dedicated or embedded control channel? In *Proc. of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC07)*, 2007.

[85] Mohsen Karimzadeh Kiskani, Babak Hossein Khalaj, and Shahin Vakilinia. Delay qos provisioning in cognitive radio systems using adaptive modulation. In *Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks*, Q2SWinet '10, pages 49–54, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0275-3. doi: 10.1145/1868630.1868639. URL http://doi.acm.org/10.1145/1868630.1868639.

[86] K. J. Ray Liu and Beibei Wang. *Cognitive Radio Networking and Security: A Game-Theoretic View*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521762316, 9780521762311.

[87] Qing-Shan Jia. Engine maintenance policy optimization with succinct value function representation. In Proceedings of the 7th Asian Control Conference, 2009.

[88] Edgar N. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39:12531265, 1960.

[89] E. O. Elliott. Estimates of error rates for codes on burst-noise channels. *Bell System Technical Journal*, 42:19771997, 1963.

[90] Lang Tong Qiang Alex Zhao and Ananthram Swami. Decentralized cognitive mac for dynamic spectrum access. In *Proc. 1st IEEE Symp. New Frontiers Dynamic Spectrum Access Networks*, 2005.

[91] Keqin Liu, Qing Zhao, and Yunxia Chen. Distributed sensing and access in cognitive radio networks, 2008. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4621370.

[92] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., 1994. URL http://portal.acm.org/citation.cfm?id=528623.

[93] William S. Lovejoy. Some monotonicity results for partially observed markov decision processes. *Operations Research*, 35:736–743, September 1987. ISSN 0030-364X. doi: 10.1287/opre.35.5.736. URL http://dl.acm.org/citation.cfm?id=50950.50957.

[94] Bharaneedharan Rathnasabapathy, , and Piotr Gmytrasiewicz. Formalizing multi-agent pomdp's in the context of network routing. In *Proc. of the 36th Hawaii International Conference on System Sciences (HICSS03*, 2003.

[95] Anthony R. Cassandra. *Exact and approximate algorithms for partially observable markov decision processes*. PhD thesis, Brown University, Department of Computer Science, 1998.

[96] Andreas Lazar Yannis A. Korilis and Ariel Orda. Avoiding the braess paradox in noncooperative networks. In *Proc. of IEEE Conf. Decision Control*, 1997.

[97] G. Gordon J. Pineau and S. Thrun. The theory of the market economy, 1952.

[98] Samson Lasaulce, Yezekael Hayel, Rachid El Azouzi, and Mérouane Debbah. Introducing hierarchy in energy games. *IEEE Transactions on Wireless Communications*, 8(7):3833–3843, 2009.

[99] Andreas Lazar. Optimal flow control of a class of queuing networks in equilibrium. *IEEE Transaction on Automatic Control*, 1983.

[100] David Bello and German Riano. Linear programming solvers for markov decision processes. *IEEE Systems and Information Engineering Design Symposium*, pages 90–95, 2006.

[101] Eitan Altman, Konstantin Avrachenkov, Nicolas Bonneau, Merouane Debbah, Rachid El-Azouzi, and Daniel Menasché. Constrained Stochastic Games in Wireless Networks. In *Proc. of IEEE Global Telecommunications Conference*, November 2007.

[102] L. Tong S. Wang, J. Zhang. Delay analysis for cognitive radio networks with random access: A fluid queue view. In *Proc. of IEEE INFOCOM*, 2010.

[103] Enrico Del Re, Renato Pucci, and Luca Simone Ronga. Energy efficient resource allocation game for cognitive radio. In *Proc. of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management*, CogART '11, pages 57:1–57:6, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0912-7. doi: 10.1145/2093256.2093313. URL http://doi.acm.org/10.1145/2093256.2093313.

[104] Oussama Habachi and Yezekael Hayel. Optimal sensing strategy for opportunistic secondary users in a cognitive radio network. In *Proc. of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, MSWIM '10, pages 343–350, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0274-6. doi: http://doi.acm.org/10.1145/1868521.1868577. URL http://doi.acm.org/10.1145/1868521.1868577.

[105] Gao Yang and Wang Yiming. Multi-channel access algorithm with channel state information unknown. In *Proc. of the Fifth International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pages 427 –430, jan. 2012. doi: 10.1109/ICICTA.2012.113.

[106] Sarah Filippi, Olivier Cappe, Fabrice Clerot, and Eric Moulines. A near optimal policy for channel allocation in cognitive radio. In Sertan Girgin, Manuel Loth, Rmi Munos, Philippe Preux, and Daniil Ryabko, editors, *Recent Advances in Reinforcement Learning*, volume 5323 of *Lecture Notes in Computer Science*, pages 69–81. Springer Berlin / Heidelberg, 2008.

[107] Haji Ali Ahmad, Mingyan Liu, Tara Javidi, Qing Zhao, Senior Member, and Bhaskar Krishnamachari. Optimality of myopic sensing in multi-channel opportunistic access. *IEEE Transactions on Information Theory*, pages 4040–4050, 2009.

[108] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, August 1991. ISBN 0262061414. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262061414.

[109] Eitan Altman and Adam Shwartz. Markov decision problems and state-action frequencies. *SIAM Journal Control Optimisation*, 29(4):786–809, July 1991. ISSN 0363-0129. doi: 10.1137/0329043. URL http://dx.doi.org/10.1137/0329043.

[110] George Dantzig, Jon H. Folkman, and Norman Shapiro. On continuity of the minimum set of a continuous function. *Journal of Mathematical Analysis and Applications*, pages 519–548, 1967.

[111] Eitan Altman and Adam Shwartz. Sensitivity of constrained markov decision problems. *Operations Research*, pages 1–22, 1991.

[112] Lin Cai, Student Member, Xuemin Shen, Senior Member, Jianping Pan, Jon W. Mark, and Life Fellow. Performance analysis of tcp-friendly aimd algorithms for multimedia applications. *IEEE Transactions on Multimedia*, 7:339–355, 2005.

[113] Deepak Bansal and Hari Balakrishnan. Binomial Congestion Control Algorithms. In *Proc. of IEEE INFOCOM*, Anchorage, AK, April 2001.

[114] Reza Rejaie, Mark Handley, and Deborah Estrin. Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In *Proc. of IEEE INFOCOM*, pages 1337–1345, 1999.

[115] Hsien-Po Shiang and Mihaela van der Schaar. Multi-user video streaming over multi-hop wireless networks: a distributed, cross-layer approach based on priority queuing. *IEEE Journal on Selected Areas in Communications*, 25(4):770–785, 2007.

[116] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, 1999.

[117] Jitendra Padhye, Jim Kurose, Don Towsley, and Rajeev Koodli. A model based tcp-friendly rate control protocol, 1999.

[118] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-based congestion control for unicast applications. In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Proc. of SIGCOMM, pages 43–56, New York, NY, USA, 2000. ACM. ISBN 1-58113-223-9. doi: 10.1145/347059.347397. URL http://doi.acm.org/10.1145/347059.347397.

[119] Kang-Won Lee, Rohit Puri, Tae-Eun Kim, Kannan Ramchandran, and Vaduvur Bharghavan. An integrated source coding and congestion control framework for video streaming in the internet. *IEEE Transactions on Multimedia*, 2000.

[120] Nevin L. Zhang and Weihong Zhang. Speeding up the convergence of value iteration in partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 14:2001, 2001.

[121] Richard S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, pages 1038–1044. MIT Press, 1996.

[122] Satinder Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesv Ari. Convergence results for single-step on-policy reinforcement-learning algorithms. In *Machine Learning*, pages 287–308, 1998.

[123] Geoff Gordon Joelle Pineau and Sebastian Thrun. Point-based value iteration: An anytime algorithm for pomdps, 2003.

[124] Shigeru Tasaka and Yutaka Ishibashi. Mutually compensatory property of multimedia qos. In *Proc. of IEEE International Conference on Communications*, 2002.

[125] Wanmin Wu, Ahsan Arefin, Raoul Rivas, Klara Nahrstedt, Renata Sheppard, and Zhenyu Yang. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In *Proc. of the 17th ACM international conference on Multimedia*, Proc. of ACM MM '09, pages 481–490, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-608-3. doi: 10.1145/1631272.1631338. URL http://doi.acm.org/10.1145/1631272.1631338.

[126] Telephone Installations and Local Line. Subjective video quality assessment methods for multimedia applications. *Networks*, 910 (P.910 (09/99)):37, 1999. URL http://www.mendeley.com/research/itut-recommendation-p910-subjective-video-quality-assessment-methods-multimedia-

[127] Vocabulary for performance and quality of service. new appendix i definition of quality of experience (qoe). *Networks*, 10(P.10), 2008.

[128] Srisakul Thakolsri, Wolfgang Kellerer, and Eckehard Steinbach. Qoe-based cross-layer optimization of wireless video with unperceivable temporal video quality fluctuation. In *Proc. of IEEE International Conference on Communications (ICC 2011)*, Kyoto, Japan, Jun 2011.

[129] Kandaraj Piamrat, Adlen Ksentini, Csar Viho, and Jean-Marie Bonnin. Qoe-aware admission control for multimedia applications in ieee 802.11 wireless networks. In

*Proc. of VTC Fall*, pages 1–5. IEEE, 2008. URL http://dblp.uni-trier.de/db/conf/vtc/vtc2008f.html#PiamratKVB08.

[130] Andre B. Reis, Jacob Chakareski, Andreas Kassler, and Susana Sargento. Quality of experience optimized scheduling in multi-service wireless mesh networks. In *Proc. of IEEE ICIP*, pages 3233–3236, 2010. ISBN 978-1-4244-7994-8. URL http://dblp.uni-trier.de/db/conf/icip/icip2010.html#ReisCKS10.

[131] Maximilian Michel, Sachin Agarwal, Wolfgang Kellerer, and Anja Feldmann. Toward qoe-aware optimum peer cache sizes for p2p video-on-demand systems. In *Proc. of IEEE International Conference on Communications*, pages 1–5, 2010. ISBN 978-1-4244-6402-9. URL http://dblp.uni-trier.de/db/conf/icc/icc2010.html#MichelAKF10.

[132] Eva Ibarrola, Fidel Liberal, Ianire Taboada, and Rodrigo Ortega. Web qoe evaluation in multi-agent networks: Validation of itu-t g.1030. In *Proc. of the Fifth International Conference on Autonomic and Autonomous Systems*, pages 289–294. Ieee, 2009. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4976618.

[133] Telephone transmission quality, telephone installations, local line networks. *ITU-T P Series*, 562(P.562).

[134] Jan A. Bergstra and Kees Middelburg. Itu-t recommendation g.107 : The e-model, a computational model for use in transmission planning. Technical report, 2003.

[135] Microsoft lync. URL http://lync.microsoft.com/.

[136] Office communications server 2007 quality of experience monitoring server guide. 2007. URL http://technet.microsoft.com/en-us/library/dd627288(office.12).aspx.

[137] Shoaib Khan, Svetoslav Duhovnikov, Eckehard Steinbach, and Wolfgang Kellerer. Mos-based multiuser multiapplication cross-layer optimization for mobile multimedia communication. *Adv. MultiMedia*, 2007(1):6–6, January 2007. ISSN 1687-5680. doi: 10.1155/2007/94918. URL http://dx.doi.org/10.1155/2007/94918.

[138] Sheldon M. Ross. *Stochastic Processes*. John Wiley and Sons, 1996.