UTILITY-BASED RESOURCE ALLOCATION STRATEGIES
AND PROTOCOL DESIGN FOR SPECTRUM-ADAPTIVE
WIRELESS NETWORKS

by

Fan Wang

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA

2009
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Fan Wang entitled “Utility-based Resource Allocation Strategies and Protocol Design for Spectrum-adaptive Wireless Networks” and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Marwan Krunz, Ph.D. Date: August 7, 2009

William E. Ryan, Ph.D. Date: August 7, 2009

Loukas Lazos, Ph.D. Date: August 7, 2009

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director: Marwan Krunz, Ph.D. Date: August 7, 2009
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: ___________________________ Fan Wang
I would like to express my sincerest gratitude to my supervisor Professor Marwan Krunz not just for all the precious advices, but also for the encouragements and for placing trust in me. He guided me through the years and helped me to grow passion in academic research Because of his support and understanding, I had a pleasant and valuable experience during my Ph.D study in the ECE department, University of Arizona. This dissertation would have been impossible without his guidance.

I would like to thank Professor William Ryan and Professor Loukas Lazos for agreeing to be members of my dissertation committee and all the valuable suggestions that make this dissertation better.

I wish to give special thanks to Professor Shuguang Cui of University of Texas A&M for his valuable discussions and feedbacks on many of my research papers. I am grateful to my labmates in the wireless networking group for their help and friendship, including Ossama Younis, Tao Shu, Satyajeet S. Ahuja, Haythem Bany-Salameh, Mohammad Siam, Raed Al-Zubi, Alaa Muqattash, and Sisi Liu. Their suggestions and discussions enlightened me a lot and refined my understanding of the field.

I would also thank Tami Whelan for handling all the paperwork necessary for the completion of my degree.

The support and encouragement I have received from my wife, Yang, my parents and my parents in law throughout my life cannot be measured. Their enlightenment and positive attitude gave me the strength and confidence to follow my dreams.
DEDICATION

To my parents and my wife Yang.
# TABLE OF CONTENTS

## LIST OF FIGURES
   ................................................................. 9

## LIST OF TABLES
   ........................................................................ 11

## ABSTRACT
   ........................................................................ 12

## CHAPTER 1 INTRODUCTION ........................................... 14
   1.1 Background ......................................................... 14
      1.1.1 Mobile Ad hoc Network ................................. 15
      1.1.2 Resource Allocation Schemes ......................... 15
      1.1.3 Medium Access Protocol Design ..................... 17
   1.2 Dissertation Overview ......................................... 18
      1.2.1 Game Theory ................................................. 19
      1.2.2 Challenges and Motivations ........................... 20
   1.3 Organizations ...................................................... 24

## CHAPTER 2 RESOURCE ALLOCATION AND CHANNEL-ACCESS DESIGN FOR SINGLE-CHANNEL MOBILE AD-HOC NETWORKS ......................................................... 26
   2.1 Introduction ......................................................... 26
   2.2 Related Work ....................................................... 31
   2.3 Formulation of the Power-Control Game .................. 33
      2.3.1 Defining the Utility Function ......................... 33
      2.3.2 Computing the NE ....................................... 35
      2.3.3 Selecting the Pricing Factor ............................ 39
   2.4 Proposed GMAC Protocol ....................................... 42
      2.4.1 Protocol Overview ....................................... 42
      2.4.2 Operational Details ...................................... 43
      2.4.3 Computing the AW Size ................................. 49
      2.4.4 Terminals Within Two Master Clusters .............. 50
      2.4.5 Fairness Issues ............................................ 50
      2.4.6 Protocol Overhead ....................................... 52
   2.5 Performance Evaluation ......................................... 53
      2.5.1 Simulation Setup .......................................... 53
      2.5.2 Single-Neighborhood Network Configuration ....... 53
TABLE OF CONTENTS – Continued

2.5.3 Multi-Neighborhood Network Configuration with a Fixed Field Size ........................................... 54
2.5.4 Multi-Neighborhood Network Configuration with a Variable Field Size ......................................... 55
2.5.5 Effect of the Pricing Factor $\alpha_i$ ................................................................................................. 55
2.5.6 Effect of Packet Size ...................................................................................................................... 57
2.5.7 Comparison with Centralized Scheduling ......................................................................................... 57
2.5.8 Effect of Traffic Model .................................................................................................................... 59
2.6 Conclusions ........................................................................................................................................ 60

CHAPTER 3 RESOURCE ALLOCATION AND CHANNEL-ACCESS DESIGN FOR MULTI-CHANNEL WIRELESS NETWORKS - EXCLUSIVE CHANNEL OCCUPANCY CASE ................................................................. 62
3.1 Introduction ........................................................................................................................................ 62
3.2 System Model .................................................................................................................................... 65
3.3 MAC Protocol Design ...................................................................................................................... 67
  3.3.1 Protocol Overview ...................................................................................................................... 67
  3.3.2 Operational details ...................................................................................................................... 68
  3.3.3 Channel, Rate and Power Assignment ......................................................................................... 71
  3.3.4 Channel Capacity Constraints .................................................................................................. 74
3.4 Adaptive Load Control ..................................................................................................................... 75
  3.4.1 Control Channel Bottleneck ...................................................................................................... 76
  3.4.2 Data Contention Delay .............................................................................................................. 78
  3.4.3 Load Control Algorithm ............................................................................................................ 79
3.5 Performance Evaluation .................................................................................................................... 81
  3.5.1 Simulation Setup ...................................................................................................................... 81
  3.5.2 Performance under different control rates .................................................................................. 82
  3.5.3 Adaptive Load Control Performance ......................................................................................... 84
3.6 Conclusions ........................................................................................................................................ 85

CHAPTER 4 THROUGHPUT-ORIENTED CHANNEL ACCESS DESIGN FOR INTERFERENCE-BASED MULTI-CHANNEL WIRELESS NETWORKS .................................................................................................................. 88
4.1 Introduction ........................................................................................................................................ 88
4.2 System Model .................................................................................................................................... 92
4.3 Problem Formulation ....................................................................................................................... 94
  4.3.1 Utility Function .......................................................................................................................... 95
  4.3.2 Game Formulation ..................................................................................................................... 97
  4.3.3 Optimal Pricing Function ........................................................................................................... 98
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Iterative Algorithms</td>
<td>102</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Sequential Price-based Iterative Water-filling</td>
<td>103</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Parallel Price-based Iterative Water-filling</td>
<td>107</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Relaxation Algorithms</td>
<td>109</td>
</tr>
<tr>
<td>4.5</td>
<td>MAC Protocol Design</td>
<td>110</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Assumptions</td>
<td>111</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Protocol Overview</td>
<td>111</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Operation Details</td>
<td>113</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Simplified Packet-based MAC Design</td>
<td>115</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Implementation of Relaxed Algorithms</td>
<td>116</td>
</tr>
<tr>
<td>4.6</td>
<td>Performance Evaluation</td>
<td>116</td>
</tr>
<tr>
<td>4.7</td>
<td>Conclusions</td>
<td>121</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>123</td>
</tr>
<tr>
<td>5.2</td>
<td>Problem Formulation</td>
<td>126</td>
</tr>
<tr>
<td>5.2.1</td>
<td>System Model</td>
<td>126</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Resource Constraints</td>
<td>127</td>
</tr>
<tr>
<td>5.3</td>
<td>Resource Allocation Strategies</td>
<td>127</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Selfish Update Strategy</td>
<td>129</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Optimal Update Strategy</td>
<td>130</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Simplified Incentive-based Strategy</td>
<td>131</td>
</tr>
<tr>
<td>5.4</td>
<td>Performance Evaluation</td>
<td>133</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusions</td>
<td>137</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions and Future Research</td>
<td>139</td>
</tr>
<tr>
<td>A</td>
<td>Appendix A</td>
<td>141</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>142</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 An opportunistic CRN in the presence of two legacy PRNs.................. 22
2.1 Example of two transmissions that can proceed simultaneously if
transmission powers are computed appropriately...................... 29
2.2 Best response functions of two links A-B and C-D......................... 36
2.3 Best response of link $i$ as a function of $\alpha_i$.............................. 41
2.4 An example of three concurrent transmissions............................... 43
2.5 Exchange of control and data packets in GMAC............................... 46
2.6 Access slot in the GMAC protocol.............................................. 49
2.7 Example of a slave terminal that is within two clusters.................... 51
2.8 Example of two unsynchronized schedules................................... 51
2.9 Network goodput vs. traffic load for the single-neighborhood configu-
ration............................................ 55
2.10 Relative frequency of having $m$ concurrent transmissions................. 56
2.11 Performance for a multi-neighborhood configuration under a fixed
field size........................................ 57
2.12 Performance under a fixed number of nodes.................................. 58
2.13 Effect of the pricing factor $\alpha_i$ on the performance of GMAC............ 59
2.14 Performance under different packet sizes.................................... 60
2.15 Comparison with optimal centralized scheduling............................ 61
2.16 Performance under IPP traffic generation model............................ 61
3.1 Rate/SNR_{th} relationship.................................................... 66
3.2 RTS/CTS packet format...................................................... 69
3.3 Two scenarios that motivate delaying the ACK packet....................... 71
3.4 RTS-CTS-DATA-ACK exchange.............................................. 71
3.5 Control contention delay...................................................... 76
3.6 Control channel virtual queue............................................... 77
3.7 An example of load control mechanism in the case of CC bottleneck....... 81
3.8 Performance under difference control rates.................................. 86
3.9 Performance of the adaptive load control mechanism....................... 87
4.1 Example of channel allocation for 4 CR links................................. 93
4.2 Nash equilibrium and Pareto-optimal Frontier................................ 98
4.3 IWF versus PIWF............................................................ 103
4.4 Normalized system sum-rate versus iterations................................ 105
LIST OF FIGURES – Continued

4.5 Example network with three CR links. .......................... 107
4.6 Convergence speed of the sequential/parallel PIWF. ............ 109
4.7 Overview of the MAC operation with two CR transmissions ($A \rightarrow B$
and $C \rightarrow D$). .................................................. 112
4.8 Performance when $\alpha = 0.1$. .................................. 119
4.9 Performance under traffic rate $\Lambda/\mu = 0.5$. ..................... 120
4.10 Performance under traffic rate $\Lambda/\mu = 0.7$ and $\alpha = 0.1$. .... 122

5.1 Power efficiency vs. traffic rate (power mask = 10 mW). ........... 135
5.2 Connection blocking probability vs. traffic rate (power mask = 10 mW). ................................. 135
5.3 System throughput vs. traffic rate (power mask = 10 mW). ....... 136
5.4 Connection blocking probability vs. power mask ($\lambda/\mu = 0.5$). .... 137
5.5 Average power vs. power mask ($\lambda/\mu = 0.5$). .......................... 138
LIST OF TABLES

2.1 Simulation parameters ........................................ 54
3.1 Simulation parameters ........................................ 82
3.2 Control Rates .................................................. 83
ABSTRACT

Resource allocation strategies, which include power/rate controls and dynamic spectrum access, are the keys to improving the performance of dynamic (mobile) wireless networks. In this dissertation, we propose several resource optimization schemes for various wireless network architectures, with the goal of maximizing the system throughput and/or minimizing the total energy consumption. We also integrate these schemes into the design of distributed medium-access control (MAC) protocols. First, we propose a game theoretic power control scheme for single-channel ad-hoc networks, and design an efficient MAC protocol, called GMAC, that implements such a scheme in a distributed fashion. GMAC allows for multiple potential nodes to contend for the channel through an admission phase that allows these nodes to determine their appropriate transmission powers. Successful contenders proceed concurrently following the admission phase. We then study the operation of spectrum-agile (cognitive) radios in a multi-channel, multi-hop wireless network setting. Two principal cases are considered: exclusive-occupancy and interference-based channel models. In the case of exclusive-occupancy channel models, we design a MAC protocol that exploits the "dual receive" capability of the radios to maximize the network throughput. We then propose a cross-layer framework for joint adaptive load/medium access controls. Under this framework, the traffic loads of individual node are adapted based on "local" parameters. In the case of interference-based channel model, when system throughput is the primary performance metric, we apply "price-based" iterative water-filling (PIWF) algorithms for resource allocation. When energy consumption is the primary metric, we propose a selfish update algorithm and an incentive-based update algorithm for minimizing the power consumption while satisfying rate demands and power-mask requirements. An efficient multi-channel MAC protocol is proposed to facilitate the radio negotiation and co-
vergence phase. Simulation results indicate that our proposed protocols achieve significant throughput/energy improvements over existing protocols.
CHAPTER 1

INTRODUCTION

1.1 Background

In the past decade, we have witnessed significant progress in the area of wireless communication and networks. This can be attributed to high demand for wireless services such as data, voice, video, and the development of new wireless standards [40][2][39]. The market potential is also a major factor pushing the research and development in wireless. In a recent study [45], the value of the mobile entertainment market is forecasted to increase from 17.3 Billion in 2007 to nearly 77 Billion by 2011. The growth of wireless networks is expected to gradually outpace landline communications because advancements in these technologies have continued to enable higher broadband speeds.

In general, the wireless networks can be categorized into two types: infrastructure-based networks and infrastructure-less networks. In infrastructure-based wireless networks, the mobile users access the network through a direct (or single-hop) link-layer connection with a network attachment point, which can be named a base station (BS) or access point (AP) based on the specific technology. Typical examples of infrastructure-based network include WiMAX (IEEE 802.16 [39]) and most cellular networks. In infrastructure-less wireless networks, instead of relying on a wired BS to coordinate the flow of messages to each user, the individual users can form their own network and forward packets to and from each other. This adaptive behavior allows a network to be quickly formed even under the most adverse conditions. Infrastructure-less wireless network is also known as ad hoc network, and is the focus of this dissertation.
1.1.1 Mobile Ad hoc Network

An ad hoc network is an autonomous collection of users that communicate over bandwidth-limited wireless links. The network is decentralized, in which all network activities, including discovering the topology and delivering the messages, must be executed by the nodes themselves. With current technology and the increasing popularity of notebook computers, interest in ad hoc networks has grown significantly. Future advances in technology will allow us to form small ad hoc networks on campuses, during conferences, and even in our own homes. The advantages of an ad hoc network are the easiness and speed of deployment, which are important requirements for certain civilian and military applications. Examples of applications using ad hoc networks include:

1. Rescue team operations - Members of rescue teams need to be in constant communication during a rescue mission in order to exchange real-time information.

2. Underdeveloped territories - Third world countries with rough terrain would be able to set up ad hoc networks without first spending the time, money, and energy involved in setting up a wired network.

Many of these applications require that nodes be mobile. This refers to a specific type of network, namely mobile ad hoc networks (MANETs). In MANETs, the network topology may change rapidly over time because of the node mobilities.

1.1.2 Resource Allocation Schemes

In most MANETs, the two principal wireless network resources, bandwidth and energy, are scarce. The main challenge in designing MANETs is to allocate network resources as efficiently as possible while providing the QoS required by the users.

Resource allocation involves several aspects, including transmission power control (TPC), rate adaptation and spectrum allocation. TPC can serve several purposes, including reducing co-channel interferences (CCI), improving spatial reuse
and reducing energy consumption. There is a dilemma for the power allocation strategy. Increasing the transmit power (TP) will increase the link’s signal-to-interference-plus-noise ratio (SINR), leading to better performance for this link (e.g., the link can use a high-order modulation scheme to achieve high throughput). At the same time, increasing TP for one link can cause more interference to its neighboring links, leading to the degradation of the system performance. Rate adaptation is a link layer mechanism that exploits the multi-rate capability at the physical layer. With rate adaptation, a sender can select the best transmission rate and dynamically adapt its decision to the time-varying and location-dependent channel quality.

Spectrum allocation has become crucial for multi-channel wireless systems, as the need for the spectrum resources has been dramatically increasing. Traditional static spectrum allocation policies have been to grant each wireless service exclusive usage of certain frequency bands, leaving several spectrum bands unlicensed for industrial, scientific and medical (ISM) purposes. The tremendous growth in ubiquitous low-cost wireless devices that utilize the ISM bands has laid increasing stress on the limited and scarce radio spectrum resources. On the other hand, most of the licensed spectrum has been largely under-utilized. According to a recent FCC report [25], the channel occupancy of licensed band in rural areas is less than 10% on average.

Consequently, dynamic spectrum allocation (DSA) has been proposed so that unlicensed users are allowed to use the under-utilized licensed spectrum (or white space) conditional on the interference to the licensed user being below an acceptable level [7]. The DSA process consists of four major steps:

1. Spectrum sensing: When a user aims to transmit a packet, it first needs to be aware of the spectrum usage around its vicinity.

2. Spectrum allocation: Based on the spectrum availability, the users can allocate one or multiple channels. This allocation not only depends on spectrum availability, but is also determined based on internal (or external) policies.

3. Spectrum Access: Since there may be multiple users trying to access the spectrum, this access should be coordinated in order to prevent multiple users
colliding in overlapping portions of the spectrum.

4. Transmitter-receiver handshake: Once a portion of the spectrum is determined for communication, the receiver of this communication should be indicated about the selected spectrum.

In this dissertation, we focus on the last three procedures of the DSA process. The first procedure, spectrum sensing and its hardware implementations, is out of the scope of this document. Cognitive radio (CR) technology is the key technology that enables a wireless network to use spectrum in a dynamic manner. The term “cognitive radio” was first coined by Mitola [55] as “the point in which wireless personal digital assistants (PDAs) and the related networks are sufficiently computationally intelligent about radio resources and related computer-to-computer communications to: (a) detect user communications needs as a function of use context, and (b) to provide radio resources and wireless services most appropriate to those needs.” Mitola’s definition, however, does not specify the radio architecture for the physical and link layers. More recently, the FCC [25] suggested that any radio with adaptive spectrum awareness is to be referred to as a CR. Specifically, a CR should be able to adapt its transmission parameters to the neighborhood environment. A cognitive radio network (CRN) can be regarded as an ad hoc network consisted of CRs.

1.1.3 Medium Access Protocol Design

Medium Access Control (MAC) protocols are responsible for coordinating the access from active nodes. The first wireless MAC protocols for multi-hop networks were designed to control only mutual exclusion. A typical example is the IEEE 802.11 standard [40]. It always uses maximum power for transmitting a packet, and aims to establish communication on a predefined link rate. Medium access contention is resolved by CSMA/CA (carrier sense multiple access with collision avoidance) and RTS/CTS (request to send/clear to send) packets. The combination of RTS/CTS and CSMA can be seen as a mean to enforce exclusion regions around sources and
destinations: nodes that hear and decode RTS/CTS packets (thus in the exclusion regions) will be excluded from transmitting in parallel, while nodes that do not hear RTS/CTS (thus outside of the exclusion regions) may proceed with transmitting.

More recent MAC protocols target on the medium access in non-exclusive channel usage environment. Power and/or rate control have been used at the MAC layer as a way to improve spatial reuse (e.g., [56; 90; 58]) or reduce energy consumption (e.g., [6; 47]). In many of these MAC schemes, terminals broadcast some collision avoidance information (CAI) to neighboring terminals. This information is used to bound the transmission powers of potential future transmitters in the neighborhood.

In recent years, with the proliferation of the multi-channel wireless networks, a number of multi-channel MAC protocols have been proposed (e.g., [89] [80] [10] [22]). Designing an efficient MAC scheme that exploits multiple channels is important but not an easy task. For example, when a terminal is listening on a particular channel, it cannot hear communication taking place on a different channel. Due to this, a new type of hidden terminal problem occurs in the multi-channel environment, which we refer to as multi-channel hidden terminal problem. Existing work resolve this problem by using radios with multiple transceivers (e.g., [89]), or restricting to single-hop operations (e.g., [80]). For example, MMAC [80] is one of the most prominent multi-channel MAC protocols applicable to single-hop networks. It defines a default control channel which all terminals must periodically switch to and synchronize to for a specific time duration.

1.2 Dissertation Overview

Due to the fading nature of wireless channels, power and energy constraints of mobile users, user mobilities, QoS requirements and many other factors, efficient resource allocation and cross-layer protocol design have been the keys to ensure the system level performance. Specifically, MANETs need efficient distributed algorithms to determine network organization and link scheduling. Many tools can be used to analyze such a system. In this dissertation, we use game theory as the mathematical
tool to analyze the resource allocation problem.

1.2.1 Game Theory

Game-theoretic approaches to radio resource allocation have recently attracted much attention (e.g., [94] [65] [37]). Game theory has been traditionally used in a variety of fields such as economics, political science, and biology. It is a mathematical tool for analyzing the interaction of two or more decision makers. A (strategic) game consists of three components: a set of players, the strategy set for each player and a utility (payoff) function for each player measuring the degree of “happiness” of the player [63]. Recently, game theory has also been applied to telecommunications and particularly wireless communications (e.g., [70] [18] [9]). In a wireless system, the users’ interaction can be modeled as a game in which the wireless users are the players competing for network resources (i.e., bandwidth and energy). Any action taken by a user affects the performance of other users in the network.

Game theory offers certain benefits as a tool to analyze distributed algorithms and protocols for MANETs [82]. We highlight two of those benefits:

1. Analysis of distributed systems: Game theory allows us to investigate the existence, uniqueness and convergence to a steady state operating point when network nodes perform independent adaptations. Hence it serves as a strong tool for a rigorous analysis of distributed protocols.

2. Design of incentive schemes: Mechanism design, being an important part of game theory analysis, concerns itself with how to engineer incentive schemes that will lead independent, self-interested participants towards outcomes that are desirable from a system point of view. This is especially helpful in the design of incentive schemes for MANETs.

It should be noted that, tools from optimization theory have also been employed to study resource allocation in wireless networks using the network utility maximization framework (see for example [93]). While there is considerable overlap between the game-theoretic and optimization-theoretic approaches, game theory
tends to focus on the multiuser competitive nature of the problem and on the users’ interaction.

1.2.2 Challenges and Motivations

A key design objective in MANETs is to achieve high network throughput while maintaining energy-efficient wireless communications for mobile terminals [1; 66]. To achieve this objective, efficient design of the resource allocation algorithm and the channel access scheme are necessary. The difficulty of resource allocation in MANETs lies in the distributed and un-coordinated nature of the network. In cellular systems, a base station instructs mobile users to adjust their transmit powers/rates. An ad-hoc network on the other hand does not have a centralized arbiter which can tell each node the TP to use to communicate with a particular receiver. Thus resource allocation in an ad hoc network is not trivial and needs to be administered in a distributed manner.

In this dissertation, we study the resource allocation problem in various types of ad hoc networks, each of which has its own challenges and requires its own solutions. The first network we consider is a single-channel multi-hop MANET, motivated by the “ad hoc” mode of the IEEE 802.11 standard [40]. Several studies have documented the inadequate performance of this standard, which is attributed to its conservative treatment of potential interferers (the RTS/CTS packets are used to silence all overhearing terminals), its use of fixed transmission powers (TPs), and its inefficient handling of ACK packets (the exposed terminal problem). These problems are particularly acute in dense networks, where the transmitter-receiver distances are relatively small. To overcome these problems, researchers considered the use of power control at the MAC layer as a way to improve spatial reuse (e.g., [56; 90; 58]) and/or reduce energy consumption (e.g., [6; 47]). Despite their demonstrated performance gains, previously proposed TPC schemes suffer from several problems, including backward incompatibility with the 802.11 architecture, extra hardware, and lack of ACK protection. In our work, we overcome the above deficiencies and further improve the spatial reuse of the network. We formulate the
channel contention problem as a non-cooperative power-control game, and allow each node to distributedly select its appropriate TPs with the goal of maximizing the special reuse of the entire network. We also present a novel MAC protocol (GMAC) that implements this game in a distributed fashion.

We also consider the resource allocation problem for operating spectrum-agile radios in multi-channel multi-hop CRNs. Several scenarios can be envisioned for operating CRs within a network. In this dissertation, we focus on one popular scenario in which CRs are used to enable opportunistic communications. In such a scenario, CR nodes act as secondary users with no pre-allocated spectrum. They form an opportunistic ad hoc cognitive radio network that coexists geographically with several primary radio networks (PRNs) – see Figure 1.1. Each PRN consists of several legacy primary radios (PRs) that are licensed to operate over a given portion of the spectrum. CR users continuously scan the spectrum, identify potential spectrum holes and exploit them for their transmissions. Besides selecting what frequency bands to use, CRs must also set their transmission powers and rates appropriately, ensuring that their opportunistic transmissions do not interfere with any of the PRs. The problem of identifying spectrum holes and selecting appropriate powers/rates is overcomplicated by the presumingly non-cooperative nature of the PRNs, which usually do not provide feedback (e.g., interference margins) to CR users.

The operations in CRN can be further categorized into two models. The first model enforces an “exclusive channel reservation” policy, whereby a channel cannot be allocated to more than one CR link in the same contention region (a.k.a., “neighborhood”). Such a restriction is in line with the classic CSMA/CA scheme. It simplifies the analysis by allowing us to ignore the CR-to-CR interference. Many of the existing research efforts fall in this category (e.g., [102; 101; 48; 44; 17; 37]). Many of these protocols assumed that each CR is equipped with multiple radios and is capable of multi-channel transmissions and receptions. This assumption comes with the cost of extra hardware, but greatly simplifies the task of multi-channel MAC design. Issues such that hidden terminals, exposed terminals and connectiv-
Figure 1.1: An opportunistic CRN in the presence of two legacy PRNs.
ity can be overcome easily. In our design, we assume that only a single radio is equipped at each CR, but each radio is capable of receiving over two channels simultaneously. This is referred to as the “dual receive” capability of the radio. The dual-receive capability has been readily supported by some recent devices (e.g., [4] and [3]). Our work [83] [51] provide a distributed multi-channel MAC protocol for multi-hop ad hoc networks based on this half-duplex and dual-receive radios. To the best of our knowledge, this is the first MAC protocol that exploits the radio’s dual-receive capability in multi-channel ad hoc networks. In addition, due to the contention nature of the MAC protocol, a wireless link is prone to be a bottleneck because of the limited spectrum, channel contention delays, and possible collisions. To solve the above issues, our work [83] [51] also include a cross-layer adaptive load control framework for joint adaptive load control and medium access control. In this framework, the traffic load of each CR is adjusted according to the local MAC parameters.

The second operational model is called “channel sharing” model. It relaxes the “exclusive channel reservation” requirement and allows neighboring CR transmissions to overlap in the frequency spectrum. By relaxing this requirement, the channel sharing model has a potential to further improve the network performance. To achieve this goal, an efficient design should accommodate more CRs in the network and maximize the CRN’s performance without disturbing PR transmissions. A typical measure of the CRN’s performance is the achievable sum-rate across all CR pairs. Unfortunately, the problem of maximizing the sum-rate over a multi-user, interference channel subject to individual power constraints is a non-convex optimization problem [81]. Such a problem becomes even more intractable when we allow CRs to share the same channel, as we now have to consider CR-to-CR interferences in addition to PR-to-CR and CR-to-PR interferences. Several attempts have been made to solve the aforementioned interference channel problem (e.g., [94; 35; 16; 96; 38]). However, these solutions are either based on centralized controllers, or resulting in inefficient channel usage (Details in Chapter 4). We are motivated to design a spectrum/power/rate allocation scheme that overcomes the inefficiency of previous
approaches and yet can be implemented in a distributed fashion [85; 86; 84; 79].

Since most of the terminals in a wireless network are battery-powered, energy efficiency is crucial to prolonging the life of the terminals. We are motivated to study the power minimization problem in the multi-channel CRNs. It is known in the literature [12] that the power minimization problem is a “dual form” of the Rate Maximization problem. We can apply our incentive idea in the sum-rate maximization problem to the power minimization problem, and study various incentive-based techniques to improve the energy utilization while satisfying the rate requirement of each user. Simulations are used to study and compare the performance of our formulations and demonstrate their effectiveness.

1.3 Organizations

The remainder of the dissertation is organized as follows. In Chapter 2, we investigated the power/rate adaptation algorithms and MAC protocol design for single channel MANETs. We propose a distributed, single-channel MAC protocol (GMAC) that is inspired by game theory, and compare its effectiveness with the 802.11 scheme and other existing single-channel power-controlled MAC protocols.

In Chapter 3, we consider the resource allocation problem for “exclusive usage” multi-channel wireless networks. We propose a distributed multi-channel MAC protocol for operating spectrum-agile half-duplex radios in a multi-hop ad hoc network. We show that the system performance greatly outperforms the traditional MAC, and is maximized at some saturation points. In order for the system to operate at these points autonomously, we propose a cross-layer framework for joint adaptive load and medium access controls. Simulation results are followed to show its effectiveness.

In Chapter 4, we consider the resource allocation problem in the channel sharing model of the multi-channel CRN. We propose a novel joint power/channel allocation scheme that improves the system level end-to-end throughput through a distributed pricing approach. Our algorithms are then integrated into the design of a distributed MAC protocol. This protocol allows CRs to dynamically select channels and adapt
to different transmission powers and rates. We discuss how the various versions of algorithms impact the MAC design. Simulations are conducted to compare the performance of the proposed protocol against other adaptive protocols.

In Chapter 5, we study the energy minimization problem in the channel sharing model of the multi-channel CRN. As in Chapter 4, we allow neighboring CRs to share the spectrum. The resource allocation problem is modeled as a non-cooperative game, where CRs repeatedly negotiate their best powers and spectrum selection to reach a NE. Due to the selfishness of the CR users, this NE is generally socially inefficient. Accordingly, we propose two incentive mechanisms to improve the social efficiency of the NE. Simulations are used to demonstrate the effectiveness of our models.

Finally, Chapter 6 provides some conclusions of this dissertation and suggestions for future work.
CHAPTER 2

RESOURCE ALLOCATION AND CHANNEL-ACCESS DESIGN FOR
SINGLE-CHANNEL MOBILE AD-HOC NETWORKS

In this chapter, we propose a game-theoretic based power control scheme for single-channel ad hoc networks, and design an efficient MAC protocol, called GMAC. GMAC allows for multiple potential transmitters to contend through an admission phase that enables them to determine the transmission powers. Successful contenders proceed concurrently with their transmissions.

2.1 Introduction

In MANETs, a key design objective is to achieve high network throughput while maintaining energy-efficient wireless communications for mobile terminals [1; 66]. To achieve this objective, efficient design of the MAC layer is necessary in order to resolve channel-contention and reduce packet collisions.

The “ad hoc” mode of the IEEE 802.11 standard [40] has so far been used as the de facto MAC protocol for MANETs. This standard is based on CSMA/CA with an optional RTS/CTS (request-to-send/clear-to-send) handshake to coordinate channel access and resolve contention. Several studies have documented the inadequate performance of this protocol, which is attributed to its conservative treatment of potential interferers (the RTS/CTS packets are used to silence all overhearing terminals), its use of fixed transmission powers (TPs), and its inefficient handling of ACK packets (the exposed terminal problem). These problems are particularly acute in dense networks, where the transmitter-receiver distances are relatively small.

To overcome these problems, researchers considered the use of power control at the MAC layer as a way to improve spatial reuse (e.g., [56; 90; 58]) and/or reduce energy consumption (e.g., [6; 47]). Our emphasis in this chapter is on the first
objective, i.e., to improve the network throughput. In throughput-oriented power-controlled MAC schemes, terminals broadcast some collision avoidance information (CAI) to neighboring terminals. This information is used to bound the transmission powers of potential future transmitters in the neighborhood. For example, CAI may include information about the maximum tolerable interference (MTI), defined as the amount of interference power that a receiver of a data or ACK packet is able to tolerate from one future interferer [58]. Future transmitters use the overheard MTI values along with other information (e.g., channel gains, load tolerance, etc.) to determine their TPs. This way, multiple interference-limited transmissions can take place concurrently in the vicinity of the same receiver.

**Motivation and contributions.** Despite their demonstrated performance gains, previously proposed TPC schemes suffer from several problems, including backward incompatibility with the 802.11 architecture, extra hardware, and lack of ACK protection. For instance, many of them offer dual-channel solutions, which often require two transceivers per terminal. In [58], the authors presented a single-channel, single-transceiver power-controlled MAC protocol called POWMAC, which allows for multiple concurrent interference-limited transmissions and provides protection for ACK packets. Simulations indicate that POWMAC achieves good performance improvement (up to 40%) over the classic (fixed-power) CSMA/CA. However, because it relies on heuristics for determining the MTI value (and hence the TPs), the protocol does not fully exploit the potential of power control and may sometimes unnecessarily silence some possible transmissions.

The following simple example illustrates the impact of setting the MTI heuristically in the POWMAC protocol. Consider a MANET of four terminals: A, B, C, and D, as depicted in Fig. 2.1, where $P_{\text{max}}$ denotes the maximum transmission power. Let $G_{ij}$ and $d_{ij}$ denote, respectively, the channel gain and distance between any two terminals $i$ and $j$. Suppose that terminals A and C wish to transmit to terminals B and D, respectively, and suppose that A succeeds in transmitting its RTS before C sends its RTS. Assume that $G_{AB} = G_{CD}$, and that A and C are within the maximum transmission range of each other. Let $\text{SNR}_{\text{th}} = 6$
dB be the required signal-to-noise ratio at the receiver. Assume a two-ray channel propagation model with a path loss factor of 4. For simplicity, we ignore the thermal noise. Clearly, the spatial reuse is maximized when both transmissions $A \rightarrow B$ and $C \rightarrow D$ are allowed to proceed concurrently. The necessary conditions for that are 

$$ \frac{G_{AB} p_A}{G_{CB} p_C} \geq \text{SNR}_th \quad \text{and} \quad \frac{G_{CD} p_C}{G_{AD} p_A} \geq \text{SNR}_th, $$

where $p_A$ and $p_C$ are the transmission powers of terminals $A$ and $C$, respectively. Combining these two inequalities, we get

$$ \frac{G_{AD} G_{CD}}{G_{AB} G_{CB}} \leq \frac{1}{\text{SNR}_th}. $$

Since we assume $G_{AD} = G_{CB}$ and $G_{AB} = G_{CD}$, the above condition becomes:

$$ \left( \frac{G_{AD}}{G_{AB}} \text{SNR}_th \right)^2 \leq 1 $$

which leads to $\frac{G_{AD}}{G_{AB}} \leq \frac{1}{\text{SNR}_th}$. At $\text{SNR}_th = 6$ dB and a path loss factor of 4, the necessary condition for two concurrent transmissions reduces to $d_{AC} \geq 0.41 d_{AB}$.

Now suppose that the POWMAC protocol is to be applied to the same example network, with terminal $A$ capturing the channel before terminal $C$. According to POWMAC, terminal $A$ calculates the minimum power needed for correct reception, heuristically inflates this power by adding to it some “interference margin,” and evenly distributes this margin among multiple future interfering terminals (whose number is not known in advance). Specifically, $A$’s transmission power ($p_A$) will be inflated to allow for future interference from up to $N_{AW}$ of $B$’s neighbors. So, $p_A$ will be chosen such that $\frac{G_{AB} p_A}{N_{AW} p_{MTI}} \geq \text{SNR}_th$, where $p_{MTI}$ is the maximum tolerable interference a future interferer is allowed to add to receiver $B$. After $p_{MTI}$ is set for link $A \rightarrow B$, when $C$’s transmission power is computed, it must satisfy the following MTI requirement: $G_{CB} p_C \leq p_{MTI}$. In addition, $C$’s power must satisfy the SNR requirement $\frac{G_{CD} p_C}{G_{AD} p_A} \geq \text{SNR}_th$. Combining these two inequalities, we get:

$$ \frac{G_{AB} p_A \text{SNR}_th}{G_{CD} G_{AD} p_A} \leq p_C \leq \frac{G_{AB}}{G_{CB} N_{AW} \text{SNR}_th} p_A. $$

Since we assume $G_{AD} = G_{CB}$ and $G_{AB} = G_{CD}$, the above condition reduces to $\frac{G_{AD}}{G_{AB}} \leq \frac{1}{\sqrt{N_{AW} \text{SNR}_th}}$. Given a typical $N_{AW}$ value in POWMAC of 5, the necessary condition for two concurrent transmissions in POWMAC becomes $d_{AC} \geq 0.73 d_{AB}$. That is, if $d_{AC} < 0.73 d_{AB}$, then POWMAC allows only one of these two transmissions to proceed. This inefficiency
is related to how the MTI is determined. Setting the value for this parameter is nontrivial. A value that is too large adds unnecessary interference in the network and wastes energy, while a value that is too small prevents some feasible concurrent transmissions.

![Diagram](image)

Figure 2.1: Example of two transmissions that can proceed simultaneously if transmission powers are computed appropriately.

To select the appropriate TPs that maximize the spatial reuse, we formulate the channel contention problem as a non-cooperative power-control game and present a novel MAC protocol (GMAC) that implements this game in a distributed fashion. Game theory provides a powerful mathematical tool for decision-making among contending transmissions. It has been applied in the context of infrastructure-based (cellular) wireless networks (e.g., [70; 8; 43; 91]). In these approaches, each user attempts to adjust his individual power in order to maximize a given utility function that may incorporate conflicting goals (e.g., signal quality and energy consumption). Though driven by similar game-theoretic analysis, our work is proposed for MANETs, and hence is quite different from its cellular counterparts in two aspects. First, in MANETs, multi-hop communication is common, whereas in cellular networks, terminals use a single hop to communicate with the base station (BS). Second, for uplink transmissions in cellular networks, interference occurs at the BS, which can be easily estimated. On the other hand, in MANETs, each reception will endure interference from all active transmissions in the network, particularly those within the receiver’s proximity. For this reason, it is difficult for a terminal to
estimate all possible sources of interference.

Unlike previous work on game-theoretic MAC design, our proposed GMAC protocol does not rely on any correlations in the offered traffic. It makes power control decisions on a per packet basis. Instead of making power decisions in parallel, as in previous work, each receiver uses its knowledge of previously scheduled transmissions in its vicinity to decide whether to join this schedule or not (this will be explained in Section 2.4). In contrast to previous work which mainly focused on computing the optimal powers using theoretical analysis but did not address how to exploit such analysis in a practical MAC protocol, our solution also incorporates the admission of active links and computing their corresponding transmission powers in a distributed fashion. It also addresses practical design issues, such as which information to include in the control packets and how to coordinate control and data packet transmissions.

We provide a MAC protocol that exploits game theory in computing the TPs for a number of contending transmitters. GMAC supports a CSMA/CA-like access mechanism for distributing these powers to contending transmitters, hence enabling more concurrent transmissions than in previously proposed techniques. The protocol is asynchronous, completely distributed, and uses only a single channel for both data and control packets. This ensures hardware compatibility with the 802.11 scheme. We also introduce a linear pricing function to obtain Pareto improvements in the achieved NE solution (this will be explained in Section 2.3).

**System model.** We consider a MANET with the following node capabilities:

- All the nodes have the same maximum transmission range and they all have the same $P_{\text{max}}$.

- The channel gain is stationary for the transmission duration of a few control packets and one data packet. This assumption holds for typical mobility patterns and transmission rates.

- Channel gains between any two terminals are symmetric. This is the underlying assumption in any RTS/CTS-based protocol, including the IEEE 802.11
standard.

- Each terminal is capable of estimating the channel gains from the received signal strength of control packets, which are transmitted at power $P_{\text{max}}$.

**Organization.** The rest of the chapter is organized as follows. In Section 2.2, we briefly discuss TPC schemes for MANETs and outline previous work on the application of game-theoretic power control analysis. Section 2.3 introduces our game-theoretic formulation for resolving channel contention in MANETs. Based on the analysis in Section 2.3, we present our GMAC protocol design in Section 2.4. In Section 2.5, we provide simulation results and insights on the performance of GMAC compared to previous approaches. Finally, we draw our conclusions in Section 2.6 and propose some extensions for future work.

### 2.2 Related Work

TPC protocols that were proposed in the literature focused on either reducing energy consumption (e.g., [6; 47]), or increasing network throughput (e.g., [56; 90; 58]). For example, Agarwal et al. [6] assumed that each terminal possesses a fixed number of power levels, and proposed a mechanism that uses the minimum required power level for transmission. A similar approach was proposed in [47] to reduce energy consumption in networks with non-uniformly distributed terminals. These approaches reduce the overall energy consumption, but achieve comparable throughput performance to that of the 802.11 standard.

Throughput-oriented TPC MAC protocols include PCMA [56], PCDC [57], and POWMAC [58] (see [50] for a complete overview of various TPC MAC protocols in MANETs). The work in this chapter belongs to this class of protocols. In PCMA [56], the receiver advertises its interference margin by sending busy-tone pulses over a dedicated control channel. The use of a control channel along with a busy tone scheme was proposed in [90], where the sender transmits the data packets and busy tones at a reduced power, while the receiver transmits its busy tones at the maximum possible power. The PCDC protocol [57] uses two frequency-separated
channels for data and control packets, allowing for interference-limited concurrent transmissions in the vicinity. In contrast to the protocols above, the POWMAC protocol [58] uses a single channel for both data and control packets. Data packets are transmitted after several RTS/CTS exchanges take place. This enables the scheduling of multiple concurrent transmissions in the same vicinity, provided that a certain interference margin is not exceeded at each transmitter-receiver pair.

Historically, TPC was used in cellular networks to improve various performance metrics, including throughput and energy efficiency. Some protocols (e.g., [70; 43; 91; 8]) used game theory to compute the transmission powers of terminals in a single cell. In [70], Saraydar et al. proposed a game-theoretic power control algorithm for data transmissions in cellular networks. The number of efficient bits transmitted per unit of energy was used as the utility function. By using a linear pricing function for the transmission power, Pareto improvement of the game is achieved. Ji and Huang [43] formulated a game for uplink power control in cellular networks. Their game’s utility is a decreasing concave function of the transmission power and an increasing concave function of the signal-to-interference ratio. An iterative algorithm that searches for the equilibrium solution was introduced and analyzed under different scenarios. The same framework was adapted in [91] by using a different sigmoid utility function. Alpcan et al. [8] adopted a cost function defined as the difference between a pricing function based on power consumption and a utility function based on Shannon capacity. They also proposed two iterative algorithms to obtain the NE and proved that these algorithms converge under certain conditions. All these game-theoretic protocols consider infrastructure-based cellular networks. Extending game theory to MANETs is challenging since such networks typically lack a centralized infrastructure. Very recently, there has been an increasing interest in using game theory to compute the transmission powers in wireless ad hoc networks [34; 18]. In these works, individual terminals use the feedback information (e.g., received SNR) to update their power levels. Then, they perform a few iterations in which the transmission power is selected and refined at each terminal until an optimal power allocation vector is determined. This approach implicitly assumes that the set of
active contenders and their relative channel gains remain static over several data packet durations. Such assumptions may not be valid in MANETs, as terminals may join and leave the network frequently and channel conditions may be rapidly changing.

2.3 Formulation of the Power-Control Game

In this section, we formulate the power-control game and use it to select the links that can be activated concurrently within a neighborhood. We then provide our rationale for selecting an appropriate value for the pricing factor of the game’s utility function. We consider a general multi-hop ad hoc network, where the mobile terminals contend for the channel whenever they have data to send. Our GMAC protocol, presented in Section 2.4, intelligently uses this analysis to enable concurrent transmissions across the entire network.

2.3.1 Defining the Utility Function

We first define an appropriate utility function for the power-controlled channel access game. Our goal is to maximize the overall network throughput while preventing terminals from unnecessarily using high power. We use the capacity (maximum achievable rate) as the figure of merit, and let the physical layer decide on the appropriate coding/modulation scheme that achieves this capacity\(^1\). We approximate the achievable rate by Shannon’s capacity [64]. Accordingly, the utility function for an active link \(i\) is defined as follows (similar to [8]):

\[
u_i(p_i, \mathbf{p}_{-i}) = \log(1 + \gamma_i) - \alpha_i p_i, \quad i = 1, 2, \ldots, n
\]

where \(p_i\) is the transmission power of \(i\)'s transmitter (to be computed), \(\mathbf{p}_{-i} \triangleq [p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n]\) is a vector that represents the transmission powers of all links other than \(i\), \(\gamma_i\) is the received SINR at the receiver, and \(\alpha_i\) is the pricing factor.

\(^1\)Some approaches incorporate the modulation scheme in the utility function [70].
factor. The SINR is given by:

$$\gamma_i = \frac{h_{ii}p_i}{\sum_{j \neq i} h_{ji}p_j + \sigma^2}$$ (2.2)

where $h_{ii}$ denotes the channel gain between the two terminals of link $i$, $h_{ji}$ denotes the channel gain between the transmitter of link $j$ and the receiver of link $i$, and $\sigma^2$ is the thermal noise power$^2$. Note that, in general, $h_{ji} \neq h_{ij}$.

The second term in (2.1) is a linear pricing function that represents the “price” for consuming a specific amount of power. Each terminal selects its transmission power such that its own utility function is maximized. This results in a standard non-cooperative game theory problem [27] of the following form:

$$\max_{p_i} u_i(p_i, p_{-i}), \text{for all } i = 1, 2, \ldots, n$$ (2.3)

subject to the constraint $C_1$:

$$C_1 : p_i \in S_i \triangleq [0, P_{\max}].$$ (2.4)

We assume that all terminals use the same coding/modulation scheme, and that SNR$_{th}$ (the SINR threshold required to achieve a target bit error rate at the receiver) and $P_{\max}$ are the same for all terminals. The solution to the above game is the one that achieves the NE. Previous work [70] stated that the NE exists if the following two conditions are satisfied:

1. $S_i$ is a nonempty and convex subset of some Euclidean space; and

2. $u_i$ is a continuous and quasi-concave function in $p_i$.

The first condition is readily satisfied. To show that the second condition is also satisfied, we take the second-order derivative of $u_i$:

$$\frac{\partial^2 u_i}{\partial p_i^2} = -\frac{h_{ii}^2}{(h_{ii}p_i + \sum_{j \neq i} h_{ji}p_j + \sigma^2)^2}.$$ (2.5)

$^2$We use the term link to refer to a tentative transmission between a transmitter and a receiver.
Since $\frac{\partial^2 u_i}{\partial p_i^2} < 0$, the second condition is satisfied. Therefore, the NE exists.

However, the existence of the NE does not guarantee that the SNR threshold is satisfied for all links. Thus, we require another constraint:

$$C_2 : \gamma_i \geq \text{SNR}_{th}.$$  \hfill (2.6)

In GMAC, if $C_2$ is not satisfied for some links, we do not allow these links to proceed. As we will explain later in our protocol design, we handle this problem during the admission control phase. A terminal competing for the channel will compute the NE powers. If these powers are feasible (i.e., $C_1$ and $C_2$ are both satisfied), the terminal will decide whether or not to proceed concurrently with previously scheduled transmissions. This decision-making process is made serially by terminals during a contention period known as the access window (AW) (described in Section 2.4). Effectively, we can always obtain a feasible power solution after all control packets have been exchanged.

### 2.3.2 Computing the NE

To find the NE of the game, we construct and analyze the players’ best response functions [63]. The best response of link $i$ is the transmission power that maximizes its utility function and satisfies the constraint $C_1$. The power that maximizes $i$’s utility function can be obtained by equating the first-order derivative of $u_i$ to zero:

$$\frac{\partial u_i}{\partial p_i} = \frac{h_{ii}}{h_{ii}p_i + \sum_{j \neq i} h_{ji}p_j + \sigma^2} - \alpha_i = 0.$$  \hfill (2.7)

Therefore,

$$p_i = \frac{1}{\alpha_i} - \frac{\sum_{j \neq i} h_{ji}p_j + \sigma^2}{h_{ii}}.$$  \hfill (2.8)

The computed $p_i$ may or may not satisfy the constraint $C_1$. Thus, the best response function of link $i$, $p^*_i$, is given by:
\[
p_i^* = \begin{cases} 
0, & \text{if } p_i \leq 0 \\
p_i, & \text{if } 0 < p_i < P_{\text{max}} \\
P_{\text{max}}, & \text{if } p_i \geq P_{\text{max}} 
\end{cases}
\]  

We now use the best response function to obtain the NE. Consider the simple example in Fig. 2.1 and assume that \( \alpha_i \) is fixed to \( 1/P_{\text{max}} \) (the reason for such a selection will be explained later). Fig. 2.2 represents the best response functions for the links A-B and C-D, and the cross-point of these two functions is the NE. As stated in Section 2.1, the achieved NE satisfies the SNR constraint as long as \( d_{AC} \geq 0.41d_{AB} \). If \( d_{AC} \) gets smaller, the slope of the best response function becomes flatter, and so the NE power will eventually not satisfy the constraint \( C_2 \).

![Best response functions of two links A-B and C-D.](image)

In general, if we have \( n \) links, the NE is the cross-point of the best response hyperplanes of all the links. By rearranging the terms in (2.8) and writing \( n \) simultaneous
equations, (2.8) can be expressed in a matrix form as:

$$HP^* = G$$  \hspace{1cm} (2.10)$$

where $H = [h_{ij}]_{i,j}$ is an $n \times n$ matrix representing the channel gains between the transmitter/receiver pairs. Note that the matrix $H$ is, in general, asymmetric. $G = [g_1, g_2, ..., g_n]^T$ is an $n \times 1$ vector with $g_i \triangleq h_{ii} - \sigma^2$. If the constraint $C_2$ is satisfied for all links, then $P^* = [p_1^*, p_2^*, ..., p_n^*]$ is the NE solution, and can be calculated using:

$$P^* = H^{-1}G.$$  \hspace{1cm} (2.11)$$

Equation 2.11 is valid only when $H$ is invertible, which is often the case because the elements of $H$ (the channel gains) are, in general, independent random variables. As discussed in Section 2.4.2, in the unlikely event that $H$ becomes non-invertible upon the inclusion of a link $i$, the GMAC protocol handles this situation conservatively by simply dropping link $i$ from the set of links to be concurrently activated.

We use (2.8) to reformulate the constraint $C_2$ as a lower bound on $p_i$ as follows:

$$p_i \geq \frac{1}{\alpha_i} \frac{\text{SNR}_{th}}{1 + \text{SNR}_{th}}.$$  \hspace{1cm} (2.12)$$

Therefore, the constraints on $p_i$ become:

$$P_{\min} \equiv \frac{\text{SNR}_{th}}{1 + \text{SNR}_{th}}P_{\max} \leq p_i \leq P_{\max}.$$  \hspace{1cm} (2.13)$$

With $\alpha_i$ set to $1/P_{\max}$, the computed $p_i$, $i = 1, 2, ..., n$, are guaranteed not to exceed $P_{\max}$ (based on (2.8)). It is possible, however, that one or more of the computed powers may fall below $P_{\min}$. If this happens, one of several possible approaches can be taken to remedy the situation, depending on whether the computation is being done when a link is first being considered for admission or if this is the final power computation that is performed by the master receiver. In the first case, a link is trying to admit itself, but the resulting NE leads to an infeasible set of powers.
Accordingly, this link will not be allowed to transmit, i.e., the corresponding receiver will send a negative CTS (NCTS) packet\(^3\). In the second case, the master receiver drops the link with the least TP, i.e., the one that is farthest from satisfying the SNR constraint. This is done by setting the final TP for that link to zero. The TPs for the remaining scheduled links need to be recomputed. The procedure is repeated until all the computed TPs are feasible, i.e., satisfy (13).

It should be noted that the fixed lower bound \(P_{\text{min}}\) is derived from the coupled (SNR) constraint, and the derivation is valid only when the TPs are set to the best response function. Such a bound would no longer be valid if we were to set the TP of some infeasible link \(j\) (with \(p_j < P_{\text{min}}\) before power adjustment) to \(P_{\text{min}}\), since in this case the lower bounds on the TPs of other feasible links (given by \(\frac{\sum_{i\neq i} h_{ij}p_j + \sigma^2}{h_{ii}}\) SNR\(_{th}\) for link \(i\)) are now functions of the various TPs, and any of these power-dependent bounds may be violated as a consequence of increasing \(p_j\) to \(P_{\text{min}}\). The following example illustrates the situation. Consider a scenario with two links. Suppose that the computed TP of link 2 (\(p_2\)) was found to be less than \(P_{\text{min}}\). If we set \(p_2\) to \(P_{\text{min}}\) and recompute the NE for both links, we will likely end up with a higher value for \(p_1\) than in the first NE computation (because link 1 will need to increase its power to overcome the extra interference due to increasing \(p_2\)). The newly computed \(p_1\) may not make it possible for link 2 (with \(p_2P_{\text{min}}\)) to satisfy its SNR constraint. In other words, setting \(p_2\) to \(P_{\text{min}}\) and recomputing the NE does not guarantee that the resulting NE will satisfy the SNR constraint (which depends on the TPs of all links). For this reason, if the first NE computation produces \(p_2 < P_{\text{min}}\), we simply drop link 2 from the considered links.

Note that our approach to find NE is different from that proposed by Rosen [68] due to two primary limitations. One limitation is that Rosen’s framework requires the strategy set to be nonempty, which is not always satisfied in our setting. This is why we are sometimes forced to drop some links to be able to find a solution for a subset of the competing ones. Another limitation is that Rosen’s algorithm runs a

\(^3\)An alternative to sending the NCTS is not to send back a regular CTS. However, in common CSMA/CA protocols, not receiving the CTS usually triggers a retransmission of the RTS, which would be unnecessary in our case.
number of iterations to reach the NE, which typically requires execution time that exceeds the milli-second granularity required for decisions at the MAC layer. We also note that uniqueness of NE in our formulation is not a concern because we are interested in finding any feasible solution that allows the competing links to proceed simultaneously.

2.3.3 Selecting the Pricing Factor

Our previous analysis was carried out with \( \alpha_i \) set to \( 1/P_{\text{max}} \) for all \( i \). In this section, we give insight into the rationale behind this choice of the pricing factor. We later study the impact of other choices on the achievable network throughput and energy consumption.

In principle, the pricing factor is intended to drive the NE solution towards a Pareto optimum. Computing the “optimal” value for the pricing factor on a per-link basis requires global information about channel gains, which is not available in our distributed setup. Instead, we derive a range of values for \( \alpha_i \) (upper and lower bounds) that is necessary for the existence of a feasible NE and that at the same time limits the energy consumption at the given terminal. We show that \( \alpha_i = 1/P_{\text{max}} \) falls within this range; at the same time, such a simple, node-independent setting enables us to convert the coupled (SNR) constraint into an easier-to-handle power-bounding constraint.

**Proposition 1.** If there exists a feasible power assignment (i.e., one that satisfies both \( C_1 \) and \( C_2 \)) for the transmitters to proceed concurrently, then there exists an upper bound on \( \alpha_i \), above which no feasible solution can be found for the game.

**Proof.** Intuitively, if the pricing term \( \alpha_ip_i \) is too high, then terminals will prefer not to transmit. We consider the best response of player (terminal) \( i \) of the original game together with the SNR constraint of link \( i \), given the powers of all the other players (\( p_{-i} \)). Let \( y_i \triangleq \frac{h_i}{\sum_{j \neq i} h_j p_j + \sigma^2} \). Accordingly, the SNR constraint can be expressed as \( p_i y_i \geq \text{SNR}_{th} \), or \( p_i \geq \frac{\text{SNR}_{th}}{y_i} \). Combining the above equation with the constraint \( C_1 \), we get the following inequality:
\[ y_i \geq \frac{\text{SNR}_{th}}{P_{max}}. \quad (2.14) \]

Taking \( \frac{\partial u_i}{\partial p_i} = 0 \), we get \( p_i = \frac{1}{\alpha_i} - \frac{1}{y_i} \). By considering the SNR constraint, the best response of player \( i \), \( p^*_i \), becomes:

\[
p^*_i = \begin{cases} 
0, & \text{if } p_i \leq 0 \text{ or } (0 < p_i < \frac{1}{y_i} \text{SNR}_{th} \text{ and } u_i(\text{SNR}_{th} y_i) \leq 0) \\
\frac{\text{SNR}_{th}}{y_i}, & \text{if } 0 < p_i < \frac{1}{y_i} \text{SNR}_{th} \text{ and } u_i(\text{SNR}_{th} y_i) > 0 \\
\frac{1}{\alpha_i} - \frac{1}{y_i}, & \text{if } \frac{1}{y_i} \text{SNR}_{th} \leq p_i \leq P_{max} \\
P_{max}, & \text{if } p_i \geq P_{max}
\end{cases} \quad (2.15)
\]

The case when \( 0 < p_i < \frac{1}{y_i} \text{SNR}_{th} \) needs to be carefully considered. Link \( i \) should decide whether to stay silent (\( p_i = 0 \)) or set \( p_i = \frac{1}{y_i} \text{SNR}_{th} \) by comparing the utility functions \( u_i(p_i = 0) \) and \( u_i(p_i = \frac{\text{SNR}_{th}}{y_i}) \). It is obvious that \( u_i(p_i = 0) = 0 \) and \( u_i(p_i = \frac{\text{SNR}_{th}}{y_i}) = \text{log}(1+\text{SNR}_{th}) - \alpha_i p_i \). If \( u_i(\frac{\text{SNR}_{th}}{y_i}) > 0 \), the terminal will use the power \( \frac{\text{SNR}_{th}}{y_i} \) to transmit; otherwise, there is no feasible solution and the sender of link \( i \) should stay silent and wait until the next contention period.

Since \( p_i \) is essentially a function of \( \alpha_i \), we can express the best response of link \( i \) as a function of \( \alpha_i \), as follows:

\[
p^*_i = \begin{cases} 
0, & \text{if } \alpha_i > \frac{\log(1+\text{SNR}_{th}) y_i}{\text{SNR}_{th}} \\
\frac{\text{SNR}_{th}}{y_i}, & \text{if } \frac{1}{1+\text{SNR}_{th}} < \alpha_i < \frac{\log(1+\text{SNR}_{th}) y_i}{\text{SNR}_{th}} \\
\frac{1}{\alpha_i} - \frac{1}{y_i}, & \text{if } \frac{1}{P_{max} + \frac{1}{y_i}} < \alpha_i < \frac{y_i}{1+\text{SNR}_{th}} \\
P_{max}, & \text{if } \alpha_i < \frac{1}{P_{max} + \frac{1}{y_i}}
\end{cases} \quad (2.16)
\]

Fig. 2.3 illustrates the best response of link \( i \) as a function of \( \alpha_i \), where \( \text{SNR}_{th} = 6 \text{ dB} \), \( P_{max} 0.03 \text{ W} \), and \( y_i = 300 \text{ W}^{-1} \). The figure shows that if the pricing factor \( \alpha_i \) is set too large, then link \( i \) will not be activated. The upper bound on \( \alpha_i \) is therefore \( \frac{\log(1+\text{SNR}_{th}) y_i}{\text{SNR}_{th}} \).

The goal of the pricing term in the utility function is to prevent terminals from
using excessive powers. If the pricing factor is too small, then every terminal will use the largest possible power for transmitting. This will eventually result in high interference on other transmissions. Therefore, $\alpha_i$ will also need to be lower bounded to reduce interference and energy consumption.

**Proposition 2.** To prevent terminals from using excessive powers, $\alpha_i$ has to be lower bounded by $\frac{\text{SNR}_{th}}{1 + \text{SNR}_{th} P_{\text{max}}}$. 

**Proof.** From Fig. 2.3, we can see that when $\alpha_i$ is less than $\frac{y_i}{1 + y_i P_{\text{max}}}$, node $i$ will always try to transmit at power $P_{\text{max}}$. Since $y_i \geq \frac{\text{SNR}_{th}}{P_{\text{max}}}$, $\frac{y_i}{1 + y_i P_{\text{max}}} \geq \frac{\text{SNR}_{th}}{1 + \text{SNR}_{th} P_{\text{max}}}$. Thus, $\alpha_i$ needs to be lower bounded by $\frac{\text{SNR}_{th}}{1 + \text{SNR}_{th} P_{\text{max}}}$ regardless of $y_i$. Otherwise, node $i$ will always use $P_{\text{max}}$ to transmit. 

Note that Proposition 2 provides a loose lower bound on $\alpha_i$. If $\frac{\text{SNR}_{th}}{1 + \text{SNR}_{th} P_{\text{max}}} \leq \alpha_i \leq \frac{y_i}{1 + y_i P_{\text{max}}}$, then $p_i^* = P_{\text{max}}$. From the previous two propositions, we select $\alpha_i$ as follows.

**Proposition 3.** Taking $\alpha_i = \frac{1}{P_{\text{max}}}$ satisfies the bounds in Proposition 1 and Proposition 2, and hence facilitates achieving the NE.
Proof. For the lower bound, \( \frac{\text{SNR}_{th}}{1 + \text{SNR}_{th}} \frac{1}{P_{max}} < \frac{1}{P_{max}} \). For the upper bound, since \( y_i \geq \frac{\text{SNR}_{th}}{P_{max}} \), \( \log(1 + \text{SNR}_{th}) y_i \geq \log(1 + \text{SNR}_{th}) \). In order to correctly decode a received packet, \( \text{SNR}_{th} \) must be \( > 0 \) dB, i.e., \( \log(1 + \text{SNR}_{th}) > 1 \). Therefore, \( \frac{\log(1 + \text{SNR}_{th})}{P_{max}} > \frac{1}{P_{max}} \).

The effect of the pricing factor on the system throughput and energy consumption will be demonstrated in Section 2.5.

2.4 Proposed GMAC Protocol

In this section, we design our distributed MAC protocol (GMAC) for maximizing the spatial throughput in a MANET. Terminals within a neighborhood use the previously presented game-theoretic approach to contend for the channel and compute their transmission powers that achieve the NE (note that multiple neighborhoods may exist simultaneously). We first describe our system model and assumptions, and then provide details of the GMAC protocol.

2.4.1 Protocol Overview

Unlike the IEEE 802.11 scheme, GMAC does not use RTS/CTS control packets to silence neighboring terminals. Instead, these packets are used to broadcast interference and channel-gain information that can be used by overhearing terminals to decide the feasibility of concurrent transmissions and their corresponding transmission powers. To assert their intentions to transmit, terminals exchange control packets over a certain time duration, referred to as the access window (AW). The AW allows several pairs of neighboring terminals to exchange their control packets so that data transmissions can proceed concurrently (see Fig. 2.4). Using an AW for contention was originally proposed in the MACA-P protocol [5] and was later integrated into the design of POWMAC [58]. In GMAC, we exploit the AW differently, as will be explained in Section 2.4.2.

The AW consists of several fixed-duration access slots, whose number is adjusted dynamically according to the network load. The transmission powers of contende-
ing terminals are not assigned until the AW is over. This mitigates the effect of heuristically presetting the tolerable interference (as in POWMAC [58]) and results in better use of the network capacity.

GMAC protects ACK packets by sending them sequentially at power $P_{\text{max}}$ after data transmissions are completed. The order in which ACKs are transmitted corresponds to the order in which the corresponding terminals appear in the AW.

![Figure 2.4: An example of three concurrent transmissions.](image)

The protocol allows several clusters (regions where multiple links contend for channel access) to be formed dynamically in a multi-hop network. The transmitters in the cluster will not be aware of their data packet transmission powers until announced by the “cluster head” (see Section 2.4.2 for more details). The purpose of this design is to maximize the spatial throughput in each cluster in order to improve the throughput over the entire network.

### 2.4.2 Operational Details

In GMAC, a potential transmitter that senses a free channel and is not aware of any scheduled transmissions is considered a master sender. The intended receiver of a master sender is called a master receiver. We refer to the link that involves the master sender and receiver as the master link. All other communicating terminals in the same vicinity of a master link are slave terminals. The region formed by the transmission range of the master receiver constitutes a cluster in which the

---

4Note that no actual clustering takes place.
master receiver is its head. All terminals inside this cluster other than the master sender are called *in-cluster* slave terminals. Such terminals can hear and correctly decode the master receiver’s CTS packet. Slave terminals that are outside the master receiver’s cluster but are within the master sender’s transmission range are called *out-cluster* slave terminals. Below, we describe the operation of each type of terminals. Throughout, we assume that $\alpha_i$ is fixed to $\frac{1}{P_{\text{max}}}$.

**Master sender**

Consider a master sender $A$ that has a data packet to transmit to another terminal $B$. If $A$ does not sense the carrier for a random duration of time, it sends an RTS at power $P_{\text{max}}$. This RTS packet includes the backoff duration that preceded the sending of the RTS and the remaining number of slots ($N_{AW}$) in the AW (how $N_{AW}$ is determined will be explained in Section 2.4.3). Upon receiving the RTS, the master receiver $B$ estimates the channel gain $h_{AB}$ based on the received power. Terminal $B$ then calculates the power $p^*_A$ that maximizes the utility function in (2.1) assuming that only one transmission ($A \rightarrow B$) will take place in $B$’s cluster. Accordingly,

$$p^*_A = \frac{1}{\alpha_i} - \frac{\sigma^2}{h_{AB}}. \tag{2.17}$$

If $p^*_A < P_{\text{min}}$, then the existing interference at receiver $B$ is too high and the transmission should not be allowed to proceed. In this case, terminal $B$ will respond with an NCTS, informing $A$ that it cannot proceed with its transmission. On the other hand, if $p^*_A \geq P_{\text{min}}$, $B$ will send back a CTS containing the values of $h_{AB}$ and $N_{AW}$. It should be noted that the computed $p^*_A$ at this point is not necessarily the transmission power that $A$ will eventually use to send its data packet. The final transmission power used by $A$ will not be decided until the end of all negotiations in the AW.

Upon receiving $B$’s CTS, terminal $A$ replies back with a DTS (decide-to-send) packet that includes the channel gain $h_{AB}$. The DTS is needed to inform out-cluster slave terminals about the success of the RTS/CTS exchange between $A$ and $B$. The
3-way (RTS/CTS/DTS) handshake is depicted in Fig. 2.5.

**In-cluster slave terminals**

Consider an in-cluster slave terminal, say terminal $C$ in Fig. 2.4. Suppose that $C$ overhears $B$’s CTS packet and has a data packet to send. It first backs off for a random duration of time. If no carrier is sensed, it sends an RTS packet at power $P_{\text{max}}$. Terminal $C$’s RTS will include $h_{AB}$, obtained from the previous RTS/CTS/DTS exchange between $A$ and $B$. It will also include $h_{CB}$ to be used by receiver $D$. When $D$ receives the RTS, it first calculates $h_{CD}$. If $D$ previously overheard $A$’s DTS, it may have already computed $h_{AD}$; otherwise, $h_{AD}$ is set to zero (this means that $D$ is out of the maximum transmission range of $A$). Terminal $D$ then calculates the NE power vector $P^* = [p^*_{AB} \ p^*_{CD}]$, assuming that two transmissions $A \rightarrow B$ and $C \rightarrow D$ will take place simultaneously. If the computed powers are feasible, i.e., satisfy the feasibility condition in (2.13), $D$ sends back a CTS that includes $h_{CD}$ and $h_{AD}$. Otherwise, if either $p^*_{AB}$ or $p^*_{CD}$ is infeasible, or if the matrix $H$ is non-invertible, $D$ sends back an NCTS. Upon receiving a CTS from $D$, terminal $C$ sends a DTS that includes $h_{CD}$ and $h_{AD}$. This channel gain information will be later used by the master receiver to compute the final transmission powers for all transmitters within its cluster. If more transmissions are to be scheduled following the RTS/CTS/DTS exchange between $C$ and $D$, the same procedure is repeated.

In general, the RTS of any transmitter contains the channel gains between that transmitter and all receivers previously scheduled in the same AW. Each receiver needs to make an admission decision by calculating the NE power vector $P^*$, assuming that all previously scheduled links and its own link will proceed concurrently. This receiver will then send back a CTS if a feasible solution exists. The CTS contains all the channel gains between the current receiver and the transmitters of all previously scheduled links. Finally, the transmitter sends a DTS that announces the channel gains included in the CTS to be used by the master receiver.

The above serialized admission phase is necessary to allow nodes that are in the vicinities of slave nodes but are not in the vicinity of the master receiver to determine
whether they can create new clusters or not. Note that it is natural to adopt a non-cooperative game-theoretic approach during this phase, as each contending link operates separately from other links using only local information about links in its receiver’s vicinity. Iterative power control, typically used in game-theoretic approaches, is not used here due to its complexity, which involves performing intra-packet power adjustments.

Final computation of transmission powers
After the master receiver (cluster head) receives information about all in-cluster transmissions, it computes the NE power values for all transmitters using (2.11). In the unlikely event that \( \mathbf{H} \) becomes non-invertible, the master receiver drops the link that causes this non-invertibility from the set of links to be concurrently activated. Because of the un likeliness of such a scenario, the impact of dropping a link on the overall spatial reuse is negligible (in fact, we have not seen this scenario in any of our simulation experiments). The cluster head will then broadcast a power-to-send
(PTS) packet, informing all in-cluster transmitters (including the master sender) of the powers to be used for transmitting their data packets.

It is worth noting that by the time the final TPs are determined, the master receiver will have enough information about in-cluster terminals to enable it to optimize their TPs with respect to a global utility function (e.g., cooperatively maximize the sum of their utilities). Yet, we do not pursue such a strategy, primarily because the resulting optimization problem is non-convex (the non-convexity stems from the TP variables in the denominator of the log term in the utility function), making it infeasible to determine the optimal solution. Instead of determining a heuristic (sub-optimal) solution for a non-convex global utility function, we opt to compute an optimal solution for the convex local utilities. Note that in our setup any feasible solution (i.e., one that admits all contending links) is acceptable, so a globally optimal solution is not actually needed.

Out-cluster slave terminals

The data transmissions of out-cluster slave transmitters do not add significant interference to the master receiver. Therefore, the strategy that we propose for such terminals is different from the one for in-cluster slave terminals. An out-cluster transmitter, say terminal $E$ in Fig. 2.4, only needs to compute its own utility without worrying about the interference it causes to the master receiver. Terminal $E$ assumes that transmitters of previously scheduled links will use the maximum transmission power ($P_{\text{max}}$) for their transmissions. It computes its transmission power accordingly. Terminal $E$ first sends an RTS packet if no carrier is sensed for a short duration. The RTS does not need to include the channel gain information as before. However, it needs to specify that $A$’s transmission has been scheduled. Receiver $F$ assumes that transmitter $A$ will transmit at power $P_{\text{max}}$ and the received power $p_{AF}$ is considered background noise. Thus, the transmission power that $E$ should use can be computed as follows:

\[
p^*_\EF = \frac{1}{\alpha_i} - \frac{\sigma^2 + h_{AF}P_{\text{max}}}{h_{EF}}. \tag{2.18}
\]
If the computed $p_{EF}^*$ is feasible, terminal $F$ will send back a CTS. The power $p_{EF}^*$ should be included in this CTS. After receiving this CTS, terminal $E$ will send a DTS that includes $p_{EF}^*$. This $p_{EF}^*$ is the final data transmission power of terminal $E$ in this scenario.

**ACK transmissions**

By overhearing the CTS/DTS packets, each terminal in the vicinity of scheduled transmissions will have knowledge of the time required for the schedule to be completed. Such a terminal will not contend for the channel until the data and ACK packets of the overheard schedule are completely transmitted. For scheduled transmissions, ACK packets will be serially transmitted using $P_{\text{max}}$ after all data transmissions are completed. As previously mentioned, the order of the ACK transmissions will conform to the order of their schedules in the AW. Therefore, ACK transmissions within a virtual cluster do not interfere with each other.

**Contention resolution**

We adopt a similar approach to the Proportionally Fair Contention Resolution (PFCR) algorithm proposed in [59]. The PFCR algorithm uses a persistent mechanism for contention resolution instead of the backoff mechanism used in the IEEE 802.11 scheme. It operates as follows. Each terminal can be in one of three possible states: NO-CONTEND, CONTEND and TRANSMIT. When a terminal $i$ has a packet to send, it has to wait until the start of the next available slot in the AW. It then senses the channel. If no carrier is sensed, terminal $i$ will switch its state from NO-CONTEND to CONTEND with probability $p_i$ (how $p_i$ is updated is explained below). Terminal $i$ then chooses a random wait time $B_i$ that is uniformly distributed in the interval $[0, B]$, where $B$ is a fixed system parameter. If terminal $i$ senses a busy channel during this wait time $B_i$, it switches its state from CONTEND to NO-CONTEND, and $p_i$ is updated as $p_i = (1 - \beta)p_i + \gamma$, where $\beta$ and $\gamma$ are system parameters. Otherwise, the state of terminal $i$ is switched to TRANSMIT, which means that $i$ can now send RTS packets to try to acquire the channel. In the mean
time, $p_i$ is updated as $p_i = p_i + \gamma$. The persistent approach was shown to be more robust and efficient than the backoff approach [59]. It also ensures proportional fairness among users.

Since $B$ is a system parameter and is known by all the terminals in the network, using the PFCR algorithm ensures that the size of the access slot (AS) is fixed. As shown in Fig. 2.6, the size of each AS is $B + \|\text{RTS}\| + \|\text{CTS}\| + \|\text{DTS}\| + 3\text{ SIFS}$, where SIFS denotes the short interframe spacing between successive control packets.

Since the actual backoff duration $B_i$ is uniformly distributed in $[0, B]$, transmitter $i$ will have to include this $B_i$ in its RTS packet. As a result, all $i$’s neighbors will be aware of the start and duration of the AS. In other words, all terminals receiving this RTS will be locally synchronized. Similarly, receiver $j$ will also need to include this $B_i$ in its CTS packet.

### Figure 2.6: Access slot in the GMAC protocol.

Since the actual backoff duration $B_i$ is uniformly distributed in $[0, B]$, transmitter $i$ will have to include this $B_i$ in its RTS packet. As a result, all $i$’s neighbors will be aware of the start and duration of the AS. In other words, all terminals receiving this RTS will be locally synchronized. Similarly, receiver $j$ will also need to include this $B_i$ in its CTS packet.

#### 2.4.3 Computing the AW Size

Each terminal $i$ will store an initial value for the AW, which is to be used if $i$ acts as a master sender. As described below, the AW size of $i$ will be updated (increased or decreased) according to the load in $i$’s vicinity. The choice of the AW size should aim at maximizing the number of concurrent transmissions, without
wasting network resources. GMAC adopts an AW adaptation approach similar to that of POWMAC [58]. After each data transmission, every terminal checks the number of concurrent transmissions in its vicinity. If this number is less than a prespecified fraction (say $\delta\%$) of the AW size, then the current AW size is too large compared to the load. In this case, the AW size is decreased by one. Similarly, if this number is larger than $\delta\%$ of the current AW size, then AW size will be increased by one to possibly allow for more concurrent transmissions.

### 2.4.4 Terminals Within Two Master Clusters

In a multi-hop network, there may be cases where a slave terminal, say $C$, resides within two different clusters. This situation is exemplified in Fig. 2.7, where the transmissions $A \rightarrow B$ and $A' \rightarrow B'$ are scheduled master links that have not started their data transmissions yet. The AWs of the two master receivers may not be synchronized, as shown in Fig. 2.8. If terminal $C$ wishes to transmit, it has two options. One option is to wait until the two master links finish their transmissions. Another option is to determine whether the misalignment between the two AWs is less than the maximum backoff window ($B$). If so, then terminal $C$ can send its RTS and compete for the channel. The problem with the second option is that even if the misalignment is smaller than $B$, allowing $C \rightarrow D$ to proceed will reduce the possibility of admitting future transmissions in both of these two clusters. So this option is only beneficial under low traffic conditions, where the possibility of having future transmissions is low so that the situation depicted in Fig. 2.7 plays a big role in system throughput. We adopt the first option in this work.

### 2.4.5 Fairness Issues

At first, one might think that the non-uniqueness of our NE solution leads to unfairness in the operation of the protocol. However, this is not the case, because a prospective link automatically drops itself out if no feasible solution is found when considering links that have previously contended for the channel during the cur-
rent access window. So the earlier a link captures an access slot, the better are its chances of being admitted. Because each contending sender must back off randomly before sending its RTS packet in a given access slot of a given access window (this also includes the master sender), nodes with packets to transmit have the same likelihood of capturing that slot. Whichever transmitting node captures the slot is guaranteed to proceed if its NE solution (considering itself and previously scheduled transmissions) is feasible. Accordingly, the protocol is as fair as the classic CSMA/CA scheme.

Figure 2.7: Example of a slave terminal that is within two clusters.

Figure 2.8: Example of two unsynchronized schedules.
We now analyze the overhead of the GMAC protocol and compare it with the 802.11 scheme. Let $S_c$ and $S_d$ denote the sizes of the control and data packet, respectively. We ignore the size of the small guarding interval (SIFS) in the following calculation. In 802.11, one RTS and one CTS packets are exchanged ahead of each data packet transmission. Therefore, the total time needed to transmit one data packet is $2S_c + S_d$. Assume that the size of the AW in GMAC is $N$ slots and the average number of concurrent transmissions is $n$, where $1 < n \leq N$. The entire control overhead is thus $N(3S_c + B) + S_c$ seconds per scheduling period. The average time needed to transmit one data packet is $[N(3S_c + B) + S_c + S_d]/n$. Thus, GMAC outperforms the 802.11 protocol if $[N(3S_c + B) + S_c + S_d]/n < (2S_c + S_d)$. By ignoring the small backoff duration $B$, the throughput improvement ($K$) of GMAC over 802.11 protocol is given by:

$$K = \frac{n}{1 + (3N - 1)/(2 + S_d/S_c)}.$$  

(2.19)

$K$ becomes larger as the ratio between the size of data packets and control packets ($S_d/S_c$) grows. To illustrate, suppose that GMAC allows on average two concurrent transmissions ($n = 2$). To achieve throughput improvement ($K > 1$), we need $|S_d/S_c| > |3N - 3|$. If $S_c$ is 20 bytes and $N = 5$, the size of a data packet needs to be larger than 240 bytes. Otherwise, GMAC will perform worse than the 802.11 protocol. We will further illustrate the relationship between $K$ and the ratio $(S_d/S_c)$ in our simulations in Section 2.5.

The overhead of GMAC is similar to that of POWMAC [58]. In order for GMAC to outperform POWMAC, we need the average number of concurrent transmissions in GMAC to be larger than that of POWMAC. It was shown through the example in Section 2.1 that this can happen. We will further investigate the throughput improvements of GMAC and POWMAC through simulations.
2.5 Performance Evaluation

We now evaluate the performance of the GMAC protocol and compare it with two single-channel protocols: POWMAC [58] and IEEE 802.11. Our performance metrics are the network goodput (number of successfully received bytes per time unit), the number of concurrent transmissions, and the total energy consumption per delivered packet (accounting for both data and control packet transmissions). We show the performance under different node densities, traffic models, packet sizes, and pricing factor ($\alpha_i$).

2.5.1 Simulation Setup

We conduct simulations using the CSIM simulation package [54]. Our simulator captures the behavior of the physical and MAC layers of a wireless network. At the physical layer, each node estimates the interference and checks the received SNR to determine whether the packet is correctly received or a collision has occurred. For the MAC layer, the simulator incorporates the 802.11 scheme, POWMAC, and our proposed GMAC.

The simulation parameters are provided in Table 2.1. These parameters correspond to the Cisco Aironet Series 350 hardware specifications [21]. We use the two-ray propagation model with a path loss factor of 4. The network area is divided into $10 \times 10$ equal-sized squares, one for each terminal. The location of each terminal is randomly assigned within its square. Each terminal generates packets according to a Poisson process with rate $\lambda$. We conduct experiments for single-neighborhood and multi-neighborhood configurations, as described below.

2.5.2 Single-Neighborhood Network Configuration

In some scenarios, such as computers in a conference room, each terminal is within the maximum transmission range of all other terminals. In other words, all the exchanged control packets, which are sent using $P_{\text{max}}$, can be received by each terminal. We model this scenario in our simulations by taking the field size to be...
Table 2.1: Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Packet Size</td>
<td>2 KB</td>
</tr>
<tr>
<td>Transmission rate</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>SINR threshold</td>
<td>6 dB</td>
</tr>
<tr>
<td>Maximum transmission range</td>
<td>750 meters</td>
</tr>
<tr>
<td>Maximum carrier-sense range</td>
<td>1500 meters</td>
</tr>
<tr>
<td>Path loss factor</td>
<td>4</td>
</tr>
<tr>
<td>Number of terminals</td>
<td>100</td>
</tr>
</tbody>
</table>

$500m \times 500m$. For each generated packet, the destination is randomly selected.

Fig. 2.9 depicts the network goodput versus $\lambda$ for the three examined protocols. The figure shows that GMAC achieves about 80% improvement in network goodput over the 802.11 standard, and about 40% improvement over POWMAC. Fig. 2.10 depicts a histogram of the number of concurrent transmissions ($m$) for both POWMAC and GMAC. It is clear that GMAC achieves a larger number of concurrent transmissions than POWMAC for all values of $m \geq 2$. This explains the goodput differences between the two protocols. Note that in this configuration, the 802.11 standard only allows one transmission to proceed at one time.

2.5.3 Multi-Neighborhood Network Configuration with a Fixed Field Size

We now examine a more general ad hoc network scenario, whereby terminals can be out of range from each other, leading to hidden terminal problems. Specifically, we place 100 terminals within a square area of length 1500 meters. The square is split into 100 smaller squares, one for each terminal. The location of a terminal within each small square is randomized. For each generated packet, the destination is randomly selected from the one-hop neighbors of the source. Fig. 2.11(a) depicts the network goodput versus $\lambda$. It shows that GMAC can achieve up to 70% increase in goodput over 802.11 and up to 25% increase over POWMAC. This improvement is due to the increase in the number of concurrent transmissions. Fig. 2.11(b) depicts
the energy consumption versus $\lambda$ for the three protocols. It is clear that the energy consumption associated with GMAC is comparable to that of POWMAC and 802.11.

### 2.5.4 Multi-Neighborhood Network Configuration with a Variable Field Size

In this scenario, we vary the length of the square field while fixing the number of terminals (i.e., we vary the node density). The packet generation rate is fixed at 40 packets/sec. The achieved goodput is shown in Fig. 2.12(a). In this scenario, GMAC shows consistent goodput improvement over both 802.11 scheme and POWMAC, especially under high densities. At the same time, the energy consumption of GMAC is comparable to both the 802.11 scheme and POWMAC, as shown in Fig. 2.12(b).

### 2.5.5 Effect of the Pricing Factor $\alpha_i$

Optimal selection of $\alpha_i$ relies on the channel conditions and the transmission powers of all the links. For optimal performance results, every link $i$ needs to compute the
optimal value of its \( \alpha_i \) on-line (i.e., on a per packet basis). This is difficult since the information about other transmission powers is not available to link \( i \) during the contention phase. Our analysis in Section 2.3 showed that \( \alpha_i = \frac{1}{P_{\text{max}}} \) facilitates achieving NE. However, using a fixed value for \( \alpha_i \) may not result in optimal system throughput. To demonstrate the effect of \( \alpha_i \) on the performance, we vary \( \alpha_i \) and report the achieved goodput and total energy consumption. For this experiment, we set \( \lambda = 40 \) packets/sec and take the field size to be 500m×500m.

Fig. 2.13(a) indicates that the overall system goodput is maximized when \( \alpha_i \) is slightly smaller than \( 1/P_{\text{max}} \). However, as \( \alpha_i \) decreases, the number of collisions increases, i.e., the protocol becomes more aggressive. This consequently results in an increase in the total energy consumption, as shown in Fig. 2.13(b). Intuitively, a smaller \( \alpha_i \) allows more links to proceed concurrently, but this does not necessarily guarantee the maximum network goodput because more collisions may occur. Although \( \alpha_i = 1/P_{\text{max}} \) does not maximize the system goodput, it provides a reasonable tradeoff between throughput and energy consumption. Adaptive setting of \( \alpha_i \) is left for future work.
2.5.6 Effect of Packet Size

We now study the effect of the data packet size on the network goodput. In this experiment, the field size is fixed to 1500m×1500m, and the packet generation rate is taken as $\lambda = 20$ packets/sec. We fix the size of the control packet to 20 bytes. Fig. 2.14(a) shows the improvement in network goodput for GMAC as the data packet size increases. Although both POWMAC and GMAC depict the same trend, GMAC outperforms POWMAC by allowing more concurrent transmissions to take place. Fig. 2.14(b) shows the total energy consumption for different packet sizes and illustrates that more energy per byte is consumed for smaller packet sizes. This is not surprising since the percentage of energy spent in transmitting control packets increases when data packet sizes decrease.

2.5.7 Comparison with Centralized Scheduling

In this section, we show that even though GMAC can significantly improve goodput, its scheduling of transmissions is still suboptimal when compared to an optimal scheduling performed by a centralized controller. The controller is assumed to be aware of all the terminals and requests, and can completely regulate all the traffic. It
Figure 2.12: Performance under a fixed number of nodes.

uses exhaustive search to select the largest possible number of feasible transmissions and allow them to proceed concurrently.

In this simulation experiment, 10 terminals are randomly placed in a single-neighborhood network configuration of size 500m × 500m. The resulting goodput is shown in Fig. 2.15.

The difference in goodput between the optimal solution and GMAC is due to three main reasons. First, GMAC does not have global knowledge of the entire network (e.g., channel gains between all contending terminals) in order to schedule the transmissions accordingly. Second, instead of using a complex optimization method, GMAC uses a low-complexity, game-theory-motivated heuristic algorithm to achieve a feasible solution. The heuristic nature of the algorithm lies in how terminals are admitted into the AW and when the final power decisions are made. Finally, the exchange of control packets wastes a proportion of system resources and consequently reduces goodput. Nonetheless, for a distributed solution, GMAC performs reasonably well in dense networks and can improve the throughput over conventional schemes by up to 80%.
2.5.8 Effect of Traffic Model

In the previous simulations, packet arrivals were modelled by a Poisson process with a packet generation rate $\lambda$. To study the effect of the traffic model, we replace the Poisson traffic generator by an Interrupted Poisson Process (IPP) that is characterized by an alternating ON/OFF pattern. Each terminal has two states: ON and OFF. The time spent in the ON state is exponentially distributed with parameter $t_{ON}$ and the time in the OFF state is exponentially distributed with parameter $t_{OFF}$. During the ON period, packets are generated according to a Poisson process with parameter $\lambda$. We fix $\lambda = 40$ packets/second and $t_{OFF} = 0.1$ sec, and vary the value of $t_{ON}$. The field size is 1500m×1500m, and the number of nodes is set to 25. The system goodput is shown in Fig. 2.16(a). The figure shows that GMAC outperforms 802.11 protocol and POWMAC in all scenarios. Fig. 2.16(b) shows the total energy consumption and demonstrates that the three protocols consume comparable amounts of energy.
2.6 Conclusions

In this chapter, we proposed a game-theoretic power control MAC protocol (GMAC) for improving throughput in MANETs. GMAC uses a single channel for both data and control packets. It allows each user to determine whether or not it is feasible to transmit concurrently with previously scheduled transmissions. GMAC enables multiple transmissions to proceed concurrently by computing NE powers for all contending transmitters.

We compared the performance of GMAC with the IEEE 802.11 standard and the POWMAC scheme. Our simulation results show that GMAC significantly improves the network goodput over both schemes. In some scenarios, the network goodput under GMAC was 80% (40%) larger than that of the 802.11 (POWMAC) scheme. GMAC also maintains comparable energy consumption to both POWMAC and the 802.11 scheme. As a future work, we will extend the analysis and the protocol design to support variable transmission rates. We will also extend the protocol by incorporating terminal priorities.
Figure 2.15: Comparison with optimal centralized scheduling.

Figure 2.16: Performance under IPP traffic generation model.
In this chapter, we design a distributed MAC protocol for operating spectrum-agile radios in a multi-hop ad hoc networks. Our protocol differs from previous designs in that it exploits the dual-receive capability of radios, thus overcoming various channel access problems that are common to multi-channel designs, including multi-channel hidden-terminal, transmitter deafness, and control channel bottleneck problems. We conduct theoretical analysis of the protocol, and study its performance via simulations. We also propose a cross-layer framework for joint adaptive load and medium access controls. In this framework, the traffic loads of individual nodes are adapted based on the values of local MAC parameters.

3.1 Introduction

The concept of spectrum-agile radios, also known as CRs has triggered great interest within the research community (see [31] for a comprehensive survey). Recently, FCC suggested referring to any radio with adaptive spectrum awareness as a CR. Specifically, a CR should be able to adapt its transmission parameters to the surrounding environment.

There exist a number of challenges in operating a cognitive radio network (CRN), namely, the opportunistic spectrum sensing capacities (e.g., [28]), the fundamental performance limit analysis (e.g., [24] [41]), and the spectrum sharing and coordinations of operating CRs (e.g., [102], [98], [13], [85], [49], [33] and [22]). In this chapter, we focus on designing an efficient and adaptive access control scheme that supports dynamic channel selection and power/rate allocation in a multi-hop CRN.
Existing work on this topic can be classified according to their architectures (centralized or decentralized). The IEEE 802.22 working group [2] is in the process of standardizing a centralized MAC protocol to negotiate the spectrum reuse of CRs operating on the TV broadcast bands. Also, in [13] [98], the authors proposed centralized protocols for coordinating spectrum access. According to these protocols, a central entity controls spectrum coordination and access. For an ad hoc network, it is desirable to have a distributed MAC protocol that allows CR users to individually sense and control their spectrum. A number of decentralized MAC protocol was proposed in the context of CRN (e.g., [49], [102], [22], and [85]). Most of these protocols assumed that each CR is equipped with multiple radios and is capable of multi-channel transmissions and receptions. This assumption comes with the cost of extra hardware, but greatly simplifies the task of multi-channel MAC design. Issues such that hidden terminals, exposed terminals and connectivity can be overcome easily. In our design, we assume that only a single radio is equipped at each CR.

Although not designed specifically for CRN, there have been a large number of multi-channel MAC protocols addressing the medium access problem with a single half-duplex radio (e.g., [89], [80], [10] and [22]). All of these protocols, however, have similar limitations and do not deal with the new challenges with CR. We list them here for completeness. For example, MMAC [80] is one of the most prominent multi-channel MAC protocols. It defines a default control channel which all terminals must periodically switch to and synchronize to for a specific time duration. MMAC overcomes the multi-channel hidden-terminal problem and provides connectivity comparable to single-channel networks, but only when terminals are within the transmission range of each other. SSCH [10] uses a channel hopping approach. If a node wants to communicate with another node, it follows the other node’s channel hopping schedule. After two nodes successfully exchange control information, they stay on that channel to complete data transfer. However, switching amongst channels may take considerable time and hence may increase delay and decrease throughput.

Given the above, we propose a distributed multi-channel MAC protocol for multi-
hop ad hoc networks based on a single half-duplex radio. One major difference of our design from the above single-transceiver MAC protocols is that we assume each radio is capable of receiving over two channels simultaneously. This is referred to as the “dual receive” capacity of the radio. This dual-receive capacity has already been supported by some recent devices. For example, QUALCOMM’s RFR6500 device [4] supports “simultaneous hybrid dual-receive operation, which allows for 1X paging signal monitoring during a 1xEV-DO connection, while monitoring other frequency bands for hand-off”. Another example is Kenwood’s TH-D7A Dual-Band Handheld Transceiver [3], which supports simultaneous reception over both data and voice channels using a single antenna. Though a simple enhancement in physical layer design, the dual-receive capacity of the radios makes the MAC design much easier. To elaborate, if we assume a common (or coordinated) control channel, the terminals can tune one of their two receiving branches on the control channel while not transmitting. In this way, the multi-channel hidden-terminal problem can be greatly alleviated. Other issues, such as the transmitter deafness and the control channel crowdness problems can also be overcome easily. One contribution of this chapter is to provide a complete contention-based MAC design to overcome the above issues. To the best of our knowledge, there was no MAC design that exploits the radio’s dual-receive capacity in multi-channel ad hoc networks.

Due to the contention nature of the MAC protocol, a wireless link is prone to be a bottleneck because of the limited spectrum, channel contention delays, and possible collisions. Transport layer protocols, such as TCP, have been used to address these issues via flow control. However, it has been shown in many papers that wireless multi-hop ad hoc networks perform poorly with TCP traffic (e.g., [52], [26], and [99]). This is because with heavy traffic loads, severe MAC contention and network congestion will occur, which will lead to decreased end-to-end throughput and increased end-to-end delay. A number of schemes have been proposed to improve TCP performance over multi-hop ad hoc networks (see [19] for a complete survey). Many of these schemes ignore the impact of MAC layer contentions on traffic flow. Most recently, the authors in [99] studied the coupling between medium contention
and network congestion. They also proposed a novel cross-layer flow control and medium access scheme. This proposed scheme utilizes the information from MAC control frames to conduct flow control functions. However, this scheme is limited to single-channel ad hoc networks and only considers the IEEE 802.11 DCF MAC function.

In the second part of this chapter, we propose a cross-layer framework for joint adaptive load control and medium access control operating in multi-channel multi-hop ad hoc networks. In this framework, the traffic loads of individual nodes are adapted using local MAC parameters. Two alternatives are provided: one is used when the control channel is the bottleneck; and the other one is applied when the data channels are the bottleneck. Simulation results show that the proposed scheme achieves more than 90% of the maximum system throughput, with low data collision rate and end-to-end delay.

The rest of the chapter is organized as follows. Section 3.2 introduces our system model, including the radio capacities and the network model. The MAC protocol design and capacity analysis are presented in Section 3.3. In Section 3.4, we present the adaptive load control mechanism. The performance of the proposed MAC protocol and the adaptive load control algorithm are evaluated in Section 4.6. Finally, Section 3.6 gives concluding remarks.

### 3.2 System Model

We consider an ad hoc network consisting of \( N \) CR nodes. CRs may not be necessarily within the transmission range of each other, i.e., hidden-terminal problems can occur, both due to distance and to channel heterogeneity. Via spectrum sensing, nodes obtain a list of available channels\(^1\). We assume that these channels are orthogonal and let \( K \) be the number of the available channels. Transmissions can be done over any one of the \( K \) channels.

The CRs have the following main features that are relevant to the MAC design:

---

\(^1\)The spectrum sensing and its hardware implementations is out of the scope of this dissertation (see [28] for a reference).
1. Dual-receive: Each radio can simultaneously receive on two of the $K$ channels, and can transmit on one channel only. The operation is half-duplex, i.e., while transmitting, the radio cannot receive/listen, even over other channels.

2. Spectrum sensing: When not transmitting, CR can measure the levels of interference (e.g., $E_b/N_0$) of all channels simultaneously.

3. Rate Adaptation: CR is capable of implementing a variety of modulation schemes and waveforms. Altogether, the system can provide $N_R$ different information rates, each with its corresponding $\text{SNR}_{th}$.

4. Rate-$\text{SNR}_{th}$ relationship: The radios use forward error correction (FEC) together with spreading to reduce the impact of interference. The combined impact of FEC, spreading, modulation, and etc., is reflected in the relationship between the transmission rate and $\text{SNR}_{th}$. In practice, this relationship takes the shape of a staircase, as shown in Fig. 3.1.

![Figure 3.1: Rate/SNR$_{th}$ relationship.](image)

5. Power Control: CR supports a fine-grain power control. The transmission power is limited to $P_{max}$.
3.3 MAC Protocol Design

In this section, we describe a distributed MAC protocol that exploits the above radio capacities in a multi-hop ad hoc CRN. Our protocol adaptability consists of joint channel/rate/power control, i.e., selecting the most appropriate channel along with the most appropriate transmission rate and transmission power for the given data packet.

3.3.1 Protocol Overview

In general, the parameters of a MAC protocol can be optimized for different objectives (e.g., minimize energy consumption, maximize network throughput and etc.). In our work, the MAC is optimized primarily to maximize the network’s throughput. This is done by allowing as many simultaneous transmissions as possible, with each transmission performed at the highest possible rate. Because of the discrete nature of rate adaptation (see Fig. 3.1), the maximum throughput can be achieved by using different transmission powers. Hence, in our design, we first target throughput maximization as the primary objective, followed by energy minimization (via power control) as a secondary objective. Energy reduction, which is reflected in Fig. 3.1 by a “power gain,” will be discussed in detail in Section 3.3.3.

To achieve a completely distributed design, we adopt a random channel access approach that is a variant of CSMA/CA. Fundamentally, such an approach requires nodes to exchange control information. We let the most reliable channel among the $K$ channels be the control channel (CC). The control information are exchanged on CC using $P_{max}$ and a fixed rate $R_{ctrl}$ (a similar approach was also used in many multi-channel MAC protocols, such as [60] and [85]). We will later discuss how to decide $R_{ctrl}$ in section 3.3.4. We assume no data packets can be transmitted over the CC and no multiple concurrent transmissions overlap over the same data channel.
3.3.2 Operational details

To reduce CR collisions in a multi-hop ad hoc CRN, each node $i$ maintains a list of available channels ($AC_i$) and a list of busy nodes ($BN_i$). The $AC_i$ list consists of channels whose network allocation vectors (NAVs) are zero according to node $i$. The $BN_i$ list consists of the IDs of nodes that are currently busy transmitting/receiving data packets in $i$th neighborhood. When node $i$ has a data packet to transmit, it checks its $AC_i$ and $BN_i$ lists. If the intended receiver of this data packet is not in $BN_i$ and if $AC_i$ is nonempty, node $i$ contends over the CC using a variant of CSMA/CA. Specifically, for its first transmission or following a successful data transmission, node $i$ selects a random backoff duration $B_i$ that is uniformly distributed between $B_{min}$ and $B_{max}$, where $B_{min}$ ($B_{max}$) is the minimum (maximum) backoff duration. Node $i$ periodically decrements the backoff timer when the CC is idle (i.e., no carrier is sensed), and freezes the timer when the CC is busy. When the timer reaches 0 and the channel is still idle, the radio is permitted to transmit its request-to-send (RTS) control packet, i.e., the node starts contending over the CC.

While the backoff timer is frozen (CC is busy), node $i$ continues to listen to, and possibly receive over the CC. If node $i$ overhears a clear-to-send (CTS) packet from any other node, it updates $AC_i$ and $BN_i$. If $AC_i$ is still nonempty and the intended receiver is not in $BN_i$, node $i$ continues to decrease its timer; otherwise, it freezes its timer until a new data channel becomes available or until the intended receiver finishes its data transmission/reception. If node $i$ overhears an RTS packet, it freezes its timer for the duration of that RTS plus $t_{SIFS}$, where $t_{SIFS}$ denotes a small duration between any two control packets. This ensures that node $i$ will not attempt to transmit a control packet in the period between the RTS and the subsequent CTS (which node $i$ may not be able to hear).

If node $i$ captures the CC, it starts an RTS/CTS exchange with the intended receiver. The RTS contains the $AC$ list of the transmitting node along with the packet size in bytes of the ensuing data packet (see Fig. 3.2). The packet size is used along with the data-channel rate (specified by the channel index that is
indicated in the CTS) to determine the duration of the ensuing data packet.

<table>
<thead>
<tr>
<th>RTS</th>
<th>Transmitter (TX) ID</th>
<th>Receiver (RX) ID</th>
<th>Packet size</th>
<th>Available channel list</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTS</td>
<td>TX ID</td>
<td>RX ID</td>
<td>Packet size</td>
<td>Data channel index</td>
</tr>
</tbody>
</table>

Nodes that are not transmitting will always have one of their two receive "branches" listen to the CC. So upon receiving an RTS packet from node \( i \), the intended receiver, say node \( j \), uses \( AC_i \) along with its own \( AC_j \) to determine the appropriate channel and rate to be used for the subsequent data packet (see details in Section 3.3.3). The selected channel must belong to both \( AC_i \) and \( AC_j \). The receiver will then send a CTS packet, containing the indices of the selected channel and rate. If no channel is available, the receiver will respond with a negative CTS (NCTS) packet.

If a transmitting node \( i \) does not hear back a CTS or NCTS from the intended receiver within a specified duration, it concludes that its RTS must have collided with another control packet or that the intended receiver itself is busy transmitting a control or a data packet. In this case, node \( i \) backs off following the same backoff procedure described before (i.e., selecting a new value for \( B_i \) between \( B_{\text{min}} \) and \( B_{\text{max}} \), and etc.), and retransmits the RTS. Even if the RTS/CTS exchange was conducted successfully and a data-packet transmission ensued over some chosen data channel, a collision over the data channel is still possible due to inconsistencies between the \( AC \) tables of different nodes and also due to unaccounted for interference from outside the network. If that happens, the receiver will not send the ACK packet, triggering a retransmission of the data packet. It should be noted that the conventional exponential backoff approach (i.e., doubling the value of \( B_{\text{max}} \) after each collision), which is commonly used in single-channel protocols such as the 802.11 is not adopted in our design, as such an approach is deemed conservative for
a multi-channel environment.

If the source node successfully receives the CTS packet, it will start its data transmission over the specified data channel. Upon hearing the CTS packet, all neighboring nodes that are listening to the CC, i.e., not in the process of transmitting packets, will update their $BN$ lists by adding the IDs of the transmitter and receiver nodes of the upcoming transmission (obtained from the overheard CTS packet), along with the remaining time for the transmission (which includes the ACK duration). These nodes will also update the NAV entry that corresponds to the assigned channel.

While receiving a data packet over a given data channel, a node still listens to other RTS/CTS exchanges taking place over the CC, and can update its $AC$ and $BN$ lists accordingly. However, a node that is transmitting a data packet will not be able to listen to the CC, so its $AC$ and $BN$ tables may become outdated. We refer to this problem as transmitter deafness, which is primarily caused by the half-duplex nature of the radios. To remedy this problem, when the receiver sends its ACK, it includes in this ACK any changes in the $AC$ and $BN$ lists that may have occurred during the transmission of the data packet. The transmitter uses this information to update its own tables.

Just before completing the receipt of a data packet, a node may start receiving a control packet over the CC (see Fig. 3.3). To avoid interrupting the reception of the control packet, the receiving node defers the transmission of its ACK packet, i.e., a time gap is inserted between the end of the data packet and the start of its associated ACK. Note that sending the ACK packet directly after the data packet may result in an inaccurate NAV tables at both the receiver and transmitter, and may lead to subsequent data collisions. The time gap depends on the type of the control packet. If the control packet is a CTS, the node will start its ACK transmission on the data channel right after completely receiving that CTS. If the control packet is an RTS, the receiver will wait until the end of the next control packet (potentially, a CTS) and then send the ACK. This design significantly reduces the number of data collisions due to incomplete NAV information. Accordingly, following the transmission of the
data packet, the transmitting node sets its ACK timer to the duration of two control packets plus an ACK duration. The RTS-CTS-DATA-ACK exchange is depicted in Fig. 3.4.

Figure 3.3: Two scenarios that motivate delaying the ACK packet.

Figure 3.4: RTS-CTS-DATA-ACK exchange.

### 3.3.3 Channel, Rate and Power Assignment

After receiving an RTS packet that includes the sender’s AC list, the receiver needs to choose an appropriate data channel, a transmission rate, and a transmission power.
Channel and Rate Selection

Based on the received power value of the RTS $P_r$, the receiver estimates the transmitter-receiver gain ($h_c$) over the CC: $h_c = P_r / P_{\text{max}}$. From $h_c$, the receiver estimates the channel gains over all data channels. Let $h_m$ denote the channel gain over the $m$th data channel. Then,

$$h_m = h_c \ast (f_c / f_m)^2$$  \hspace{1cm} (3.1)

where $f_c$ denotes the carrier frequency of the CC and $f_m$ denotes the carrier frequency of the $m$th data channel. Equation 3.1 is derived assuming a simplified path-loss model of the form [29]:

$$P_r = P_t K (\frac{d_0}{d})^\gamma$$  \hspace{1cm} (3.2)

where $d_0$ is the close-in distance, $\gamma$ is the path-loss exponent (between 2 and 4), and $K$ is a unit-less constant that can be determined via measurements or estimated assuming omni-directional antennas and a free-space propagation environment:

$$K = \left(\frac{\lambda_m}{4\pi d_0}\right)^2$$  \hspace{1cm} (3.3)

where $\lambda_m$ is the carrier wavelength for channel $m$. Given the above, the channel gain over channel $m$ is given by:

$$h_m = \frac{P_r}{P_t} = \frac{1}{f_m^2} \left(\frac{c}{4\pi d_0}\right)^2 (\frac{d_0}{d})^\gamma$$  \hspace{1cm} (3.4)

So $h_m$ is inversely proportional to $f_m^2$.

Note that the above derivation is generally valid when the transmission distance $d > d_0$, where $d_0$ is typically is the range 1-10 meters for indoor environments and 10-100 meters for outdoor environments.

Another approach for estimating $h_m$ is to rely on direct channel-gain measurements, taken over channel $m$. Specifically, the receiving node may use the trans-
mission and reception powers of previously received data packets over channel $m$ to estimate the channel gain between itself and a given transmitter. This approach, however, does not work well in highly mobile environments (i.e., when nodes change their positions frequently relative to the inter-packet times of packets sent over the same channel).

The receiver then estimates the interference-plus-noise power $N_0(m)$ over each data channel $m$. It is assumed that the hardware is capable of providing such estimates for all channels.

For a given data packet, let $S$ be the intersection of the $AC$ lists of the transmitter and receiver. The goal of the receiver is to select the channel that provides the highest possible data rate. The data rate, however, is dependent on the received power. So, the receiver first assumes that the transmitter will use its maximum transmission power $P_{\text{max}}$, and accordingly determines the optimal channel $m^*$:

$$m^* = \arg\max_{m \in S} R(m)$$

where

$$R(m) = f(SNR(m))$$

$$SNR(m) = \frac{h_m P_{\text{max}}}{N_0(m)}$$

In (3.5), $f$ is the step function that represents the rate-SNR$_{th}$ relationship (see Fig. 3.1). The solution to the above problem gives the channel $m^*$ to use and the maximum data rate $R(m^*)$ associated with that channel.

**Power Control**

The above channel/rate assignment is done assuming the transmitter uses $P_{\text{max}}$. It is possible to reduce this power while maintaining the same channel/rate assignment. This is done by solving the following problem:

$$\min P(m^*)$$

s.t. $$f\left(\frac{h_m^* P(m^*)}{N_0(m)}\right) = R(m^*)$$

(3.6)
Essentially, the above problem amounts to eliminating the slack between $P_{\text{max}}$ and the minimum power needed to support the rate $R(m^*)$. In practice, the minimum required transmission power needs to be inflated by a small fraction, say $\delta$, to account for out-of-network interference (this fraction is often called a link margin). As a result, the final transmission power for the data packet is set to $\min \left( (1 + P(m^*), P_{\text{max}}) \right)$. The receiver includes in the CTS packet the index of the selected channel $m^*$, the selected rate (or its index), and the selected transmission power.

### 3.3.4 Channel Capacity Constraints

Assuming that the packet generation rate of every node is $\lambda$ (packet/sec), and the data packets have a fixed size $D$ (bits), the traffic load generated by each node is $T = \lambda D$ (bits/sec). By considering the stability of the transmitter queue at each node along with the theoretical capacities of the control and data channels, the following two constraints can be established on the relationship among $T$, $D$ and $N$. The first constraint comes from the capacity limit of the CC as follows. The average packet generation rate of the whole network is $N\lambda = NT/D$. Each data packet transmission requires at least two control packets (RTS and CTS), hence occupying the CC for $(\|\text{RTS}\| + \|\text{CTS}\|)/R_{\text{ctrl}} + t_{\text{SIFS}}$ seconds. The maximum number of packets that the CC can accommodate in one second is $1/(\|\text{RTS}\| + \|\text{CTS}\|)/R_{\text{ctrl}} + t_{\text{SIFS}}$. To guarantee a stable system, we have:

$$C_1 : \frac{TN}{D} < \frac{1}{\frac{\|\text{RTS}\| + \|\text{CTS}\|}{R_{\text{ctrl}}} + t_{\text{SIFS}}} \quad (3.7)$$

The second constraint is due to the capacity of the $K - 1$ data channels. On average, this capacity is $(K - 1)R_{\text{avg}}/D$, where $R_{\text{avg}}$ is the average transmission rate (in bps) per data channel. To guarantee a stable system, we have:

$$C_2 : \frac{TN}{D} < \frac{9R_{\text{avg}}}{D} \quad (3.8)$$
Intuitively, in some cases, $C_1$ is tighter than $C_2$ when, for example, $R_{ctrl}$ is small; in some other cases, the data channels dominate the performance when, for example, $D$ is large and the CC is not fully utilized. The optimal $R_{ctrl}$ can be found by equating the right sides of equations (3.7) and (3.8). By ignoring the small $t_{SIFS}$, the optimal control rate is given by:

$$R_{opt}^{ctrl} = \frac{(K - 1)R_{avg}(\|RTS\| + \|CTS\|)}{D}$$

(3.9)

When each radio joins the network, it calculates $R_{opt}^{ctrl}$ according to (3.9). In Section 4.6, we will compare the system performance with different $R_{ctrl}$ values.

### 3.4 Adaptive Load Control

In the previous section, we discussed the MAC design and derived the optimal control rate. However, these may not guarantee the effectiveness of the MAC. For instance, in the case of high traffic loads, the system may still experience high network congestions and medium contentions. This has been well studied as a flow control problem. As we mentioned before, most of the existing flow-control schemes ignore the impact of MAC layer contentions on traffic flow. In this section, we propose a cross-layer framework for joint adaptive load control and medium access control.

First assume that nodes have the flexibility and intelligence to adjust their traffic loads (i.e., elastic traffic). For example, video/voice users may adjust their codec rates. If nodes keep increasing the value of $T$, one of the two constraints $C_1$ and $C_2$ will reach its capacity eventually, which is referred to as the system saturation point, in the sense that system throughput saturates or even degrades after this point, as later shown in Section 4.6.

To optimize the system performance, we expect that the system operates close to the saturation point. However, the traffic load $T$ corresponding to the saturation point depends on various factors, such as the network density and the channel conditions, which are generally not available in a distributed ad hoc network. This
motivates us to propose a distributed adaptive load control scheme, in which nodes
adjust their source rates based on local MAC information. Before the load control,
nodes in the network determine the bottleneck of the system by comparing $R_{ctrl}$
with $R_{ctrl}^{opt}$. If $R_{ctrl} < R_{ctrl}^{th}$, then CC is the bottleneck; otherwise, data channels are
the bottleneck.

3.4.1 Control Channel Bottleneck

In the case of CC bottleneck, if the system is saturated, the nodes with packets to
transmit (or relay) would experience long delay in contention on CC, as illustrated in
Fig. 3.5. When node $i$ has a packet to transmit, it senses the CC and starts backoff
when the channel becomes idle. The backoff timer $B_i$ keeps decreasing when the CC
is idle and freezes when the CC is busy (because of RTS/CTS transmissions). When
$B_i$ reaches 0, node $i$ initializes a RTS/CTS handshaking. We define the control
contention delay (CCD) $\tau_i$ as the duration between the time CTS is successfully
received and the time node $i$ starts its backoff. CCD can be estimated by every
individual node for each of its transmitted RTS packet.

![Figure 3.5: Control contention delay.](image)

Note that without contentions, the expected value of CCD, $\bar{\tau}_i = t_{RTS} + t_{CTS} +
t_{SIFS} + B$, where $\bar{B} = (B_{max} + B_{min})/2$ denotes the average backoff duration of a
given node. If we have a large number of contending nodes in $i$th neighborhood,
each with its own packet to transmit, then $\tau_i$ is expected to be large. This is because
on average, a large number of RTS/CTS pairs is to be observed before $B_i$ reaches
0. Thus \( \tau_i \) is a measure of the CC crowding in the neighborhood.

The above analysis is analogous to a queueing system where the queueing delay is large if we have large number of arriving customers while other system parameters remain the same. We can then model the system using a \( G/M/1 \) queue. The CC is modeled as a virtual single-server queue (with no “actual” buffer), with fixed service rate \( (R_{ctrl}) \). The virtual queueing system is described in Fig. 3.6. Note that this system is only an approximate analogy to the actual system due to the random backoff.

Given the model in Fig. 3.6, the system is stable when the total arrival rate is smaller than the service rate, and the saturation point corresponds to the situation when the total arrival rate is close to the service rate. However, there is no way for the individual node to estimate the total arrival rate, the only estimate available is the CCD \( \tau_i \) for each packet, which can be interpreted as the queuing time plus the service time. From \( \tau_i \), node \( i \) can estimate the approximate number of customers that are serviced ahead of itself by \( q_i = \tau_i / \bar{\tau}_i - 1 \), where \( q_i \) can be regarded as the queueing length.

![Figure 3.6: Control channel virtual queue.](image)

Note that \( q_i \) is the queueing length for the current packet and is a measure of the CC crowding. Node \( i \) also needs to predict the channel conditions for the next packet. In the prediction, recent data should be get more weight than past data. The simple exponential smoothing (SES) model is suitable for this situation. Let \( \alpha \) denote the smoothing constant (a value between 0 and 1), and let \( \hat{q}_i(t) \) denote the value of the predicted series at time \( t \). The following formula is used to update the smoothed series recursively as new observations are recorded:

\[
\hat{q}_i(t) = \alpha q_i(t) + (1 - \alpha)\hat{q}_i(t-1)
\]  

(3.10)
Thus, the current smoothed value is an interpolation between the previous smoothed value and the current observation, $\alpha$ controls the closeness of the predicted value to the most recent observation. If $\alpha = 1$, the SES model is equivalent to a random walk model; if $\alpha = 0$, the SES model is equivalent to the mean model. The SES model is superior than the mean model or the moving average model because it places relatively more weight on the most recent observation (i.e., it’s more responsive to recent changes). Moreover, the smoothing constant $\alpha$ in the SES model can be easily optimized by minimizing the mean square error (MSE) between the predicted value and the observations.

Intuitively, a small $\tilde{q}_i$ indicates an under-utilized queue, so node $i$ should increase its traffic load $T_i$. On the other hand, if $\tilde{q}_i$ is large, node $i$ should decrease $T_i$. The thresholds can be set similar to the DECBit protocol, i.e., if $q_i < 1$, node $i$ will regard the queue as under-utilized and increase its source rate; if $1 \leq q_i \leq 1 + \delta$, node $i$ will keep its current source rate; if $q_i \geq 1 + \delta$, node $i$ will decrease its load. $\delta$ is selected according to the traffic requirements, if the traffic is delay-tolerant (e.g., FTP traffic), then $\delta$ can be set as a large value, otherwise, $\delta$ is set small.

The load control algorithm in the case of CC bottleneck is summarized in Algorithm 1.

### 3.4.2 Data Contention Delay

In the case of data channel bottleneck, the data channels can be modelled as a G/M/(K-1) queueing system, and a data contention delay can be used to adjust the traffic loads. An alternative to this is to use the number of occupied data channels (ODC) ($o_i$) directly as a measure of the data-channel crowdness. Intuitively, if $o_i$ is close to $K - 1$, it means most of the data channels are busy, and node $i$ should decrease $T_i$, and vice versa. The SES model is also used to predict the data channel occupancy:

$$
\hat{o}_i(t) = \alpha o_i(t) + (1 - \alpha)\hat{o}_i(t - 1) \tag{3.11}
$$

After $\hat{o}_i(t)$ is computed, we set three thresholds $\delta_1$, $\delta_2$ and $\delta_3$, with $0 < \delta_1 < \delta_2 < \delta_3$. 

\( \delta_3 < (K - 1) \). These thresholds are selected according to the QoS requirements of the traffic. Each node adjusts its traffic load by comparing \( \tilde{o}_i(t) \) with these thresholds. The adaptive load control algorithm with data channel bottleneck is summarized in Algorithm 1.

3.4.3 Load Control Algorithm

Each node initializes its source rate \( T_i \), which is randomly selected between \([T_i^{\text{min}}, T_i^{\text{max}}]\), where \( T_i^{\text{min}} \) (\( T_i^{\text{max}} \)) denotes the minimum (maximum) required source rate. Nodes may have different \( T_i^{\text{min}} \) and \( T_i^{\text{max}} \) depending on their applications. Whenever there is a packet generated from node \( i \), in the case of CC bottleneck, node \( i \) measures \( \tau_i \) and computes the queue length \( q_i(t) \) for this packet, with which it predicts the queue length \( \tilde{q}_i^D(t) \) using (3.10). In the case of data channel bottleneck, node \( i \) estimates \( o_i \) and predicts \( \tilde{o}_i(t) \) using (3.11).

The packet generation rate \( \lambda_i \) is adjusted similar to the TCP congestion control mechanism, which is known as a stable mechanism. To elaborate, when the CC is the bottleneck, for any attempted packet with \( \tilde{q}_i < 1 - \delta \), node \( i \) regards the queue as greatly under-utilized and increases \( \lambda_i \) exponentially by \( \lambda_i = \lambda_i + 1 \ (T_i = T_i + D) \), similar to the “slow start” phase in TCP; if \( \tilde{q}_i \in [1 - \delta, 1] \), node \( i \) increases \( \lambda_i \) linearly by \( \lambda_i = \lambda_i + 1/\lambda_i \ (T_i = T_i + D^2/T_i) \), similar to the “congestion avoidance” phase in TCP; if \( \tilde{q}_i \in [1, 1+\delta] \), node \( i \) keeps the current \( \lambda_i \); finally, if \( \tilde{q}_i > 1+\delta \), node \( i \) regards the network as congested, and cuts \( \lambda_i \) by half, i.e., \( \lambda_i = \lambda_i/2 \ (T_i = T_i/2) \). After \( T_i \) is updated, it is Euclidean projected to the interval \([T_i^{\text{min}}, T_i^{\text{max}}]\) as \( T_i = [T_i^{\text{min}} + T_i^{\text{max}} - T_i]/2 \). When data channels are the bottleneck, a similar procedure is applied by comparing \( \tilde{o}_i(t) \) with \( \delta_1, \delta_2 \) and \( \delta_3 \).

In Fig. 3.7, we give an example to graphically illustrate the adaptive load control mechanism in the case of CC bottleneck.
Algorithm 1: Adaptive load control for node $i$

Initialize $T_i(0) \in [T_i^{min}, T_i^{max}]$, and iteration count $l = 0$;
Compute $R_{ctrl}^{opt}$ using (3.9).

while (time $<$ SIMTIME):

1: if $R_{ctrl} < R_{ctrl}^{opt}$ and node $i$ receives its own CTS then
2: // Control channel bottleneck
3: $l = l + 1$;
4: Estimate the current CCD $\tau_i(l)$;
5: Compute the queue length $q_i(l) = \frac{\tau_i(l)}{\bar{\tau}_i(l)} - 1$;
6: Predict the queue length $\tilde{q}_i(l)$ using (3.10);
7: if $\tilde{q}_i(l) < 1 - \delta$ then
8: $T_i(l + 1) = T_i(l) + D$;
9: else if $\tilde{q}_i(l) \in [1 - \delta, 1]$ then
10: $T_i(l + 1) = T_i(l) + D^2/T_i(l)$;
11: else if $\tilde{q}_i(l) \in [1, 1 + \delta]$ then
12: $T_i(l + 1) = T_i(l)$;
13: else
14: $T_i(l + 1) = T_i(l)/2$;
15: end if
16: $T_i(l + 1) = [T_i(l + 1)]^{T_i^{max}}_{T_i^{min}}$
17: else if $R_{ctrl} \geq R_{ctrl}^{opt}$ and node $i$ receives an ACK then
18: // Data channel bottleneck
19: $l = l + 1$;
20: Estimate the current ODC $o_i(l)$;
21: Predict the $\tilde{o}_i(l)$ using (3.11);
22: if $\tilde{o}_i(l) < \delta_1$ then
23: $T_i(l + 1) = T_i(l) + D$;
24: else if $\tilde{o}_i(l) \in [\delta_1, \delta_2]$ then
25: $T_i(l + 1) = T_i(l) + D^2/T_i(l)$;
26: else if $\tilde{o}_i(l) \in [\delta_2, \delta_3]$ then
27: $T_i(l + 1) = T_i(l)$;
28: else
29: $T_i(l + 1) = T_i(l)/2$;
30: end if
31: $T_i(l + 1) = [T_i(l + 1)]^{T_i^{max}}_{T_i^{min}}$
32: end if
3.5 Performance Evaluation

In this section, we evaluate the performance of the proposed MAC protocol and the adaptive load control scheme.

3.5.1 Simulation Setup

The performance results are based on simulations conducted using CSIM, a C-based process-oriented discrete-event simulation package. The simulator captures the behaviors of the physical and MAC layers of radios. At the physical layer, each radio measures the interference and checks the SNR of the received signal against the corresponding $\text{SNR}_{th}$ to determine whether the packet can be correctly received. At the MAC layer, the simulator incorporates the MAC design described in the previous sections.

The main performance metrics of interest are the end-to-end network throughput, the data/control collision rate, and the average energy consumption per successfully received data bit ($E_b$). Here $E_b$ includes the energy consumed in control packets. Note that energy consumption per bit is a more meaningful metric than the average transmission power per bit, as the latter does not account for the time duration of the bit (i.e., rate adaptation).

In the simulated network, $N$ nodes are randomly distributed in a square area of length 2000 meters. Each node generates traffic according to a Poisson process with

Figure 3.7: An example of load control mechanism in the case of CC bottleneck.
Table 3.1: Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes (N)</td>
<td>40</td>
</tr>
<tr>
<td>Number of channels (K)</td>
<td>10</td>
</tr>
<tr>
<td>Data packet size (D)</td>
<td>8000 bits</td>
</tr>
<tr>
<td>$P_{\text{max}}$</td>
<td>17 dBm</td>
</tr>
<tr>
<td>$N_0$</td>
<td>-111 dBm</td>
</tr>
<tr>
<td>Control packet size</td>
<td>120 bits</td>
</tr>
<tr>
<td>SIFS duration</td>
<td>10 $\mu$s</td>
</tr>
<tr>
<td>$B_{\text{min}}$</td>
<td>20 $\mu$s</td>
</tr>
<tr>
<td>$B_{\text{max}}$</td>
<td>160 $\mu$s</td>
</tr>
<tr>
<td>Power inflation factor ($\delta$)</td>
<td>0.2</td>
</tr>
</tbody>
</table>

rate $\lambda$ (packets/sec). Accordingly, the traffic load generated by each node is $T = \lambda D$ (bits/second). The packet destination is randomly chosen with equal probabilities from all the other nodes in the network. Note that at high transmission rates, the destination of a packet may be outside the transmission range of the source node. In this case, multi-hop transmission (using the shortest path algorithm) is used. We implement a simple min-hop Bellman-Ford routing algorithm and use it as a basis for multi-hop communications (for both data and control packets). For simplicity, we ignore the routing overhead. Note that we do not attempt to optimize the performance of the routing protocol, which is an important issue by itself, and will be dealt with in a future work. The parameters used in the simulations are summarized in Table 3.1.

3.5.2 Performance under different control rates

We first compare the system performance under different $R_{\text{ctrl}}$. Let $N_R = 5$, and $R1$ to $R5$ be the control rates of increasing order. The values of different control rates and their corresponding SNR$_{th}$ are summarized in Table 3.2. Note that these values represent the joint effects of the spreading, modulation, and FEC.

From the parameters in Table 3.1 and 3.2, we conclude that all nodes can hear from each other when $R_{\text{ctrl}} = R1$, thus simulating a single-hop network. Increasing
Table 3.2: Control Rates

<table>
<thead>
<tr>
<th>Rate Index</th>
<th>Chips/Symbols</th>
<th>Rate</th>
<th>SNR_{th}</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>64</td>
<td>78 Kbps</td>
<td>0.26</td>
</tr>
<tr>
<td>R2</td>
<td>32</td>
<td>156 Kbps</td>
<td>0.52</td>
</tr>
<tr>
<td>R3</td>
<td>25</td>
<td>200 Kbps</td>
<td>0.67</td>
</tr>
<tr>
<td>R4</td>
<td>1</td>
<td>2.5 Mbps</td>
<td>5.90</td>
</tr>
<tr>
<td>R5</td>
<td>1</td>
<td>5 Mbps</td>
<td>15.73</td>
</tr>
</tbody>
</table>

$R_{ctrl}$ implies a smaller transmission range for control packets, and may result in a multi-hop scenario, where hidden-terminal problems can occur. From the analysis in Section 3.3.4, $R_{opt}^{ctrl}$ is a value between $R3$ and $R4$. So when $R_{ctrl} = R1$ or $R2$ or $R3$, the CC is the bottleneck; when $R_{ctrl} = R4$ or $R5$, the data channels are the bottleneck.

Fig. 3.8(a) depicts the end-to-end throughput versus $T$ under different $R_{ctrl}$. For all cases, when $T$ is small, the system is able to serve all incoming traffic, and the throughput grows almost linearly with $T$. As $T$ increases, the performance becomes bounded by $C1$ (or $C2$), i.e., the control (or data) channel(s) is (are) no longer capable of coping with the traffic demands, and the system eventually reaches the saturation points. For the lowest three control rates, increasing $R_{ctrl}$ alleviates the CC crowding, and thus delays the process of reaching the saturation points. For the largest two control rates, the throughput reaches its peak value at some $T$, and then starts to drop with a further increase in $T$. This behavior can be explained by the fact that at large values of $R_{ctrl}$ and $T$, the network starts to experience a large number of data collisions due to the contention nature of the MAC protocol.

Fig. 3.8(b) depicts the data-packet collision fractions under different $R_{ctrl}$. It shows that before the saturation points, the data collision fraction is relatively small (less than 1% for $R1$ to $R3$ and 4% for $R4$ and $R5$). After the saturation points, the data collision fraction becomes high for $R4$ and $R5$, because the data channels are saturated in such cases. This also explains the drop in throughput after saturation point in Fig. 3.8(a). The control-packet collision fraction follows a similar trend,
Fig. 3.8(c) shows the average energy consumption for successfully transmitting one data bit. More energy are needed for low $R_{ctrl}$. In other words, using high $R_{ctrl}$ incurs short-range and multi-hop transmissions, and is more energy-efficient, as verified in many energy-oriented MAC protocols. Also note that the average energy consumption slightly increases after the saturation points because of the increased collisions.

Fig. 3.8(d) shows the end-to-end delay versus $T$. As expected, the end-to-end delay is relatively low (less than 100 ms) before the system saturates, which is generally sufficient for most real-time applications. After the saturation points, the end-to-end delay greatly increases. Note that although higher $R_{ctrl}$ results in more hops to reach the same destination, the cumulative end-to-end delay is still smaller than that of the low $R_{ctrl}$ cases.

### 3.5.3 Adaptive Load Control Performance

In Fig. 3.8, we showed that the performance of our proposed MAC is optimized at the saturation points with certain traffic load $T$. However, the optimal value of $T$ is generally unknown, especially in a distributed environment. We show in this section the effectiveness of our adaptive load control mechanism by comparing its performance with that of the saturation points.

Fig. 3.9(a) depicts the system end-to-end throughput using adaptive load control and the throughput achieved at the saturation points ($\bar{T}^s$). Note that $\bar{T}^s$ may not be the maximum system throughput ($\bar{T}^{opt}$), because we assumed $T$ to be the same for all nodes in the last section. When the nodes vary their loads, the problem of finding $\bar{T}^{opt}$ is intractable even with a central entity. In fact, $\bar{T}^s$ is a good approximation of $\bar{T}^{opt}$ for $R1$ to $R3$, but is smaller than $\bar{T}^{opt}$ for $R4$ and $R5$ because of the dropping effect at high loads as shown in Fig. 3.8(a). This is the reason that the throughput using load control under $R_{ctrl} = R5$ can perform slightly better that $\bar{T}^s$. Fig. 3.9(a) shows that the system throughput using load control is more than 90% of $\bar{T}^s$. Finally, Fig. 3.9(b) shows that the data collision fractions using adaptive
load control mechanism are relatively low, thus proves the efficiency of the adaptive load control mechanism.

3.6 Conclusions

In this chapter, we proposed a distributed multi-channel MAC protocol for multi-hop ad hoc networks. This MAC exploits the dual-receive capacity of the radios to solve the multi-channel hidden-terminal problem. It also solves the transmitter deafness problem and selects an appropriate control rate. From the simulation results, we conclude that prior to the saturation points, the system throughput increases with the traffic load, and the system has low end-to-end delay and collision rates. After the saturation points, the system throughput may be decreased or stabilized, together with higher end-to-end delay and collision rates. A distributed adaptive load control mechanism was proposed for the system to operate efficiently and autonomously, where each node uses the local MAC parameters to adapt its traffic load. Simulation results show that the load control mechanism can achieve more than 90% of the throughput achieved at the saturation points, together with low collision rate.
Figure 3.8: Performance under different control rates.
Figure 3.9: Performance of the adaptive load control mechanism.
In the chapter, we relax the “exclusive channel reservation” requirement and allow neighboring CR transmissions to overlap in the frequency spectrum. We propose a novel joint power/channel allocation scheme that improves the performance through a distributed pricing approach. In this scheme, the spectrum allocation problem is modeled as a non-cooperative game, with each CR pair acting as a player. A price-based iterative water-filling algorithm is proposed, which enables CR users to reach a good NE. This PIWF algorithm can be implemented distributively with CRs repeatedly negotiating their best transmission powers and spectrum. Simulation results show that the social optimality of the NE solution is dramatically improved through pricing. Depending on the different orders according to which CRs take actions, we study sequential and parallel versions of the PIWF algorithm. We show that the parallel version converges faster than the sequential version. We then propose a corresponding MAC protocol to implement our resource management schemes. The proposed MAC allows multiple CR pairs to be first involved in an admission phase, then iteratively negotiate their transmission powers and spectrum via control-packet exchanges. Following the negotiation phase, CRs proceed concurrently with their data transmissions. Simulations are used to study the performance of our protocol and demonstrate its effectiveness in terms of improving the overall network throughput and reducing the average power consumption.
4.1 Introduction

Several scenarios can be found for operating a CRN. In this dissertation, we focus on an opportunistic CRN where the CRs are secondary users that coexist with PRs. The PRs are licensed to operate over certain frequency bands. They do not cooperate with or even provide feedback to the CRs. CRs continuously sense the channel and exploit spectrum “holes” for their transmissions. One of the main challenges in an opportunistic CRN is how to design an efficient and adaptive channel access scheme that supports dynamic channel selection and power/rate allocation in a distributed (ad hoc) CRN environment. An efficient design is one that tries to maximize the CRN’s performance without disturbing PR transmissions. A typical measure of efficiency is the achievable sum-rate across all CR pairs. Unfortunately, the problem of maximizing the sum-rate over a multi-user, interference channel subject to individual power constraints is a non-convex optimization problem [81]. Such a problem becomes even more intractable when we allow multiple CRs to share the same channel, as we now have to consider CR-to-CR interferences in addition to PR-to-CR and CR-to-PR interferences.

Several attempts have been made to solve the aforementioned interference channel problem. One well-known resource allocation scheme, called iterative water-filling (IWF), was first proposed in [94], where a non-cooperative game was used to model the spectrum management problem with each user iteratively maximizing its own rate. This per-user optimization problem is convex and leads to a water-filling solution. For the two-user case, it was proven that the NE exists and the IWF algorithm converges to the NE under certain conditions. However, this NE is generally not Pareto optimal [63] and may be quite inefficient in term of sum-rate [16]. This is because in a non-cooperative game, each user only has the incentive to maximize its own utility function without considering the overall system performance. A centralized spectrum management scheme was proposed in [16], which greatly improves the system performance over the IWF scheme by utilizing a centralized spectrum management center (SMC). However, such an approach cannot be implemented in
an ad hoc opportunistic CRN, where none of the CRs has global knowledge of the entire CRN to function as the SMC.

Given the above, we are motivated to design a channel/power/rate allocation scheme that overcomes the inefficiency of IWF and yet can be implemented in a distributed fashion. Specifically, we provide incentives to CR users such that they can reach a more socially efficient NE. A commonly used incentive technique in game theory is *pricing* (a thorough review is provided in [32]). Previously, pricing techniques have been implemented in various wireless networks such as cellular networks, ad hoc networks, and peer-to-peer networks (e.g., [70], [18], [87], [9], [35]). In this chapter, we apply pricing techniques to CRNs. We propose a *price-based iterative water-filling* (PIWF) algorithm, and show that this algorithm maintains the simplicity and distributed operation of the IWF algorithm while achieving better bandwidth efficiency (i.e., higher sum-rate). The effectiveness of the pricing technique depends on the selection of the “pricing functions,” which is a challenging problem by itself. Although there may exist an “optimal” pricing function that allows the NE to converge to a Pareto-optimum solution, the search for such a pricing function generally requires a central controller and is hard to be implemented in a distributed manner. Some sub-optimal pricing functions have been proposed in the literature. For example, the authors in [18] proposed an auction-like pricing scheme for MANETs. The unit price in this scheme (uniform across all users) is gradually increased until the system settles down at a feasible NE. A similar approach was taken in [70], where the users of a wireless data network keep increasing their prices in a uniform fashion until one user begins to receive a decreasing utility. Both of the previously mentioned pricing schemes achieve feasible NEs and improve the system performance. However, the achieved NEs are not guaranteed to be globally optimal, which is partially caused by the fact that both of the two approaches take a uniform unit price for all players in the game. In our work, we determine a *user-dependent* pricing function, which not only improves the NE, but also achieves globally or locally optimal NE after a few iterations. Such a pricing function can be determined by allowing each CR user to distributively acquire its neighborhood
information via control-packet exchanges. Note that a similar pricing analysis was recently conducted in [35] in the context of interference management. Much of the emphasis and convergence results were related to a different utility function and network settings than the ones in our chapter.

Another problem of applying the IWF algorithm in [94] to CRNs is that this algorithm only considers a total power constraint for each user. In a CRN, PRs impose strict power constraints over each frequency band, so CR users have to abide by frequency-dependent power constraints. Such constraints will affect the response of each CR user and thus the achieved NE. In this chapter, we incorporate a frequency-dependent power mask constraint into the optimization problem.

In our proposed algorithm, each user maximizes its own utility function (which includes a pricing function) by performing a single-user price-based water-filling, while treating the interference from other CR users at each sub-band as additive white Gaussian noise (AWGN). The same procedure iterates sequentially, eventually converging to the NE. If the number of users in the network is large, sequential updating may suffer from slow convergence. Therefore, we also discuss a parallel PIWF algorithm (the parallel concept was introduced in [72]), which is an instance of the Jacobi scheme: At each iteration, CRs update their strategies simultaneously, based on the interference measured in the previous iteration. Simulations indicate that this parallel version converges faster than the sequential PIWF algorithm. Both the sequential and parallel PIWF algorithms require CRs to be synchronized and the system parameters to be correctly estimated for each CR. These conditions may not be satisfied in practical systems. To overcome this problem, a “relaxed” update scheme can be used (as in [11], [53], and [73]) and is studied in our work. For the “relaxed” version of the PIWF algorithm, each CR is required to remember its most recent policy choices together with the choices of other users. The relaxed algorithm is more robust to inaccurate estimates and channel oscillations, but it may impact the convergence speed.

Our PIWF algorithms are then integrated into the design of a distributed medium access (MAC) protocol for CRNs. This protocol allows CRs to dynam-
ically select channels and adapt to different transmission powers and rates. We discuss how the various versions of PIWF impact the MAC design. Simulations are conducted to compare the performance of the proposed protocol against other adaptive protocols.

The rest of this chapter is organized as follows. The system model is described in Section 4.2. Section 4.3 formulates the non-cooperative game and introduces the pricing techniques. We then discuss the PIWF algorithms for solving the NE in Section 4.4 and design the corresponding MAC protocol in Section 4.5. In Section 4.6, we provide simulation results of the PIWF algorithms and compare them with the classic IWF algorithm. Finally, we draw conclusions and discuss future extensions in Section 4.7.

4.2 System Model

We consider a hybrid network consisting of several PRNs and one CRN. The CRN contains $N$ CR pairs. The total spectrum consists of $K$ orthogonal frequency channels with central frequencies $f_1, f_2, \ldots, f_K$, where $K < N$. Each PR in a PRN may operate over one or multiple channels. Let $\Omega_N = \{1, 2, \ldots, N\}$ and $\Omega_K = \{1, 2, \ldots, K\}$ denote the sets of CR links and channels, respectively.

Each CR may simultaneously transmit over multiple channels. It can also receive over multiple channels (from the same transmitter) at the same time. However, we require that each CR operates in a half-duplex manner, meaning that it cannot receive while transmitting, and vice versa. Let $M_i(f_k)$ denote the total noise-plus-interference level measured by CR user $i$ over channel $k$. This quantity includes the PR-to-CR interference, the CR-to-CR interference, and the thermal noise. We assume that when not transmitting, CR $i$ is capable of measuring $M_i(f_k)$ over all channels $k \in \Omega_K$. Let $M_i \triangleq [M_i(f_1), M_i(f_2), \ldots, M_i(f_K)]$, which is used by CR $i$ to perform channel selection, power control, and rate allocation, as described later.

The motivation of using CR technology is to enhance the spectrum utilization by allowing CR users to share the spectrum with PRs. Some previous work [78] assumed
that CR transmissions do not interfere with each other, i.e., only one CR user can operate over a given channel in a given neighborhood (along with the PRs). In this way, there is no spectrum sharing among CR users. Such schemes limit the number of admitted CR links, especially when the number of channels is small. In our work, we allow multiple CR users to share a particular channel. Figure 4.1 depicts a channel allocation example for a CRN with \( K = 3 \) and \( N = 4 \). The dark square indicates that a channel is utilized by a CR. For example, link 1 uses channels 1 and 2, while link 4 uses channel 1 only. We denote the set of utilized channels for CR link \( i \) as \( S_i \). In the above example, \( S_1 = \{1, 2\} \) and \( S_4 = \{1\} \). The transmission power vector of CR link \( i \) over all channels is denoted by \( P_i = [P_i(f_1), P_i(f_2), \ldots, P_i(f_K)] \), where \( P_i(f_k) \) is the transmission power of CR \( i \) on channel \( k \). If channel \( k \) belongs to \( S_i \), \( P_i(f_k) > 0 \); otherwise, \( P_i(f_k) = 0 \).

Figure 4.1: Example of channel allocation for 4 CR links.

To ensure feasible spectrum sharing, we impose the following constraints:

1. Maximum transmission power constraint: The total transmission power of a CR over the selected channels should not exceed \( P_{\text{max}} \), i.e., \( \sum_{k \in S_i} P_i(f_k) \leq P_{\text{max}} \). Here, we assume that the total power constraint is the same for all users. It is easy to extend the treatment to the case where \( P_{\text{max}} \) is user-dependent.

2. CR-to-PR power mask constraint: The transmission power of CR \( i \) on channel
94

$k$ is constrained by $P_t(f_k) \leq P_{\text{mask}}(f_k)$, where $P_{\text{mask}}(f_k)$ is the power mask on channel $k$. Such a per-device power mask is easier to verify at the design stage from a practical point of view. For example, the power mask is often specified by FCC regulations. In this case, CR vendors need to design the radio while ensuring its RF transmission power meets the FCC power mask. Such a philosophy is often used in various wireless technologies (e.g., UWB, Wi-Fi, Walkie-Talkies, etc.). Note that because the number of active CR links that share a given frequency band varies in time and space, it is impractical to design the hardware to account for a “neighborhood-dependent” power mask.

We use the vector $P_{\text{mask}} \triangleq [P_{\text{mask}}(f_1), P_{\text{mask}}(f_2), \ldots, P_{\text{mask}}(f_K)]$ to denote the power mask on all channels. In the following analysis, we assume that $P_{\text{mask}}$ is given a priori.

3. Minimum SINR constraint: If the received SINR over a given channel is below the SINR threshold ($\text{SINR}_{\text{th}}$), the CR will not use that channel.

We assume that the CRs are either static or are moving slowly compared to the convergence time of the resource assignment algorithms. This assumption is generally acceptable because our iterative algorithms operate on the time scale of few milliseconds, whereas pedestrian and vehicular mobility impacts the network topology on the time scale of seconds. In addition, CRs are homogeneous, meaning that they follow the same operation rules and have the same system constraints.

4.3 Problem Formulation

In a “non-cooperative” CRN, each CR user is interested in maximizing its own achievable rate. Such a greedy behavior can be modeled using game theory. Game theory analyzes the interactions of players in decision-making processes. It can be used to identify distributed optimal strategies for the players [61]. A normal game $\mathcal{G}$ is expressed as: $\mathcal{G} = \{\Omega; \mathcal{P}, \{U_i\}\}$, where $\Omega = \{1, 2, \ldots, N\}$ is a finite set of rational players; $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \ldots \times \mathcal{P}_N$ is the action space with $\mathcal{P}_i$ being the action set for player $i$; and $U_i : \mathcal{P} \rightarrow \mathcal{R}$ is the utility (payoff) function of player $i$,.
which depends on the strategies of all players. We can model the channel/power allocation problem in a CRN as a non-cooperative game, in which the players are the CR users; their actions are the transmission power vector (i.e., the action for user $i$ is given by $P_i = [P_i(f_1), P_i(f_2), \ldots, P_i(f_K)]$); and their utility functions are associated with their actions and the quality of the channels. Note that a CR user in the game denotes a CR link consisting of a pair of CRs.

### 4.3.1 Utility Function

In our game, the utility function of user $i$ can be considered as the reward received by this user from the network. This reward should depend on the user’s action $P_i$ and the union set of all other users’ actions $P_{-i}$, where $P_{-i} \overset{\text{def}}{=} [P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_N]^T$. While the selection of the utility function is not unique, the selected utility function must have physical meaning for the particular application. A natural selection of the utility function for CR link $i$ (also used in [94; 20; 73]) is its transmission rate, given by:

$$U_i(P_i, P_{-i}) = \sum_{k \in \Omega_K} u_i(P_i(f_k)) =$$

$$\sum_{k \in \Omega_K} \left[ \log_2 \left( 1 + \frac{h_{ii}(f_k)P_i(f_k)}{\sum_{j \in \Omega_N, j \neq i} h_{ji}(f_k)P_j(f_k) + M_i^{(PR)}(f_k) + N_i(f_k)} \right) \right]$$

(4.1)

where $h_{ji}(f_k)$ denotes the channel gain between the transmitter of link $j$ and the receiver of link $i$ over channel $k$, $M_i^{(PR)}(f_k)$ denotes the PR-to-CR interference at the receiver of CR link $i$ over channel $k$, and $N_i(f_k)$ denotes the received thermal noise power on channel $k$. In the chapter, this relationship is taken as Shannon’s capacity formula. In a practical multi-rate wireless system, the power-rate relationship takes the form of a staircase, and the user sets the transmission rate to the maximum possible rate (among a finite set of rates) that satisfies the SNR threshold at the given transmission power value. It is straightforward to extend our design to accommodate such a power-rate relationship.
Given the utility function in (4.1), users select their transmission powers to maximize their own utility functions, and under certain conditions, they eventually reach at a NE after several iterations. As discussed in Section 4.1, because of the non-cooperative nature of the game, each CR user behaves selfishly. Thus, the resulting NE may be far from the Pareto optimum [63]. In practice, we are interested in maximizing a weighted sum of the utilities of all users, defined as:

$$\max_{\mathbf{P}} \sum_{i \in \Omega_N} w_i U_i(\mathbf{P}_i, \mathbf{P}_{-i}) = \max_{\mathbf{P}} \sum_{i \in \Omega_N} w_i \sum_{k \in \Omega_K} u_i(P_i(f_k))$$

(4.2)

where $w_i$ denotes the weight assigned to CR user $i$, which may be interpreted in different ways (e.g., priority factor of user $i$). Note that the power assignment that solves (4.2) is a Pareto-optimum solution.

To drive the NE towards the Pareto optimum boundary, we use pricing as an incentive for each selfish CR user to work in a cooperative manner. A new utility function with pricing is then defined as follows:

$$\tilde{U}_i(\mathbf{P}_i, \mathbf{P}_{-i}) = \sum_{k \in \Omega_K} \tilde{u}_i(P_i(f_k))$$

(4.3)

with

$$\tilde{u}_i(P_i(f_k)) \overset{\text{def}}{=} -c_i(f_k) + \log_2 \left( 1 + \frac{h_{ii}(f_k)P_i(f_k)}{(\sum_{j \in \Omega_N, j \neq i} h_{ji}(f_k)P_j(f_k)) + M_i^{(PR)}(f_k) + N_i(f_k)} \right)$$

(4.4)

where $c_i(f_k)$ represents the pricing function for user $i$ on channel $k$. As discussed in Section 4.1, our goal is to choose a user-dependent pricing function that can drive the CR users to converge to an efficient NE. How to define this pricing function will be discussed in Section 4.3.3.
4.3.2 Game Formulation

Given the price-based utility function in (4.4), each CR user $i$ iteratively selects its power vector $P_i$ to maximize $\tilde{U}_i(P_i, P_{-i})$ subject to the constraints listed in Section 4.2. This results in the following non-cooperative game $G$:

$$\max_{P_i} \tilde{U}_i(P_i, P_{-i}), \; \forall i \in \Omega_N$$

s.t.

C1: $P_i(f_k) \geq 0, \; \forall i \in \Omega_N$ and $k \in \Omega_K$ \hspace{1cm} (4.5)

C2: $\sum_{k \in \Omega_K} P_i(f_k) \leq P_{\text{max}}, \; \forall i \in \Omega_N$

C3: $P_i(f_k) \leq P_{\text{mask}}(f_k), \; \forall i \in \Omega_N$ and $k \in \Omega_K$

If there is a solution to the above game, then it would be the one that achieves the NE. Note that the above game differs from the game studied in [94] in the form of the utility function and the additional power mask constraint. Thus, the existence proofs in [20] and [94] cannot be directly applied.

The following proposition show that a NE solution always exists for the above game.

**Proposition 1.** For any given $P_{\text{max}}$ and $P_{\text{mask}}$ values, there is at least one NE for the game $G$ in (5.1).

**Proof.** The game in our setup can be shown to be a concave game if the following two properties are satisfied:

1. The action space $P$ is a closed and bounded convex set;

2. The utility function $\tilde{U}_i(P_i, P_{-i})$ is concave over its strategy set.

It is straightforward to show that the two properties are satisfied by the game $G$. Because a concave game always admits at least one NE [68], the proposition follows immediately. \qed
Given the existence of a NE solution, we need to design an algorithm for CR users to reach the NE. However, before we do that, we first investigate the form of the optimal pricing function.

4.3.3 Optimal Pricing Function

Pricing is an idea that originated from economics (e.g., [32]). It denotes the cost of commodities for individual decision makers. In the power control context (e.g., [70] and [87]), pricing is used as an incentive mechanism to improve the efficiency of the NE. To illustrate, in Figure 4.2, we depict an example of the Pareto-optimal frontier and the NE for a two-user game. In general, the NE is not Pareto optimal. Previous pricing techniques usually improve the achieved NE (i.e., moving it closer to the Pareto-optimal frontier) using heuristic pricing functions, since an optimal pricing function generally requires global information and could hardly be deployed in a distributed manner. For example, in [70] and [87], the pricing function is a suboptimal linear function with a fixed linear pricing factor for all players.

![Figure 4.2: Nash equilibrium and Pareto-optimal Frontier.](image-url)

The pricing function can take various forms. A linear pricing function is commonly used because of its implementation simplicity. In this chapter, we use a user-dependent linear pricing function that drives the NE close to the Pareto optimal frontier with each player having only local and certain neighborhood information. The neighborhood information is acquired via control packets that are exchanged...
during the channel access process (see Section 4.5 for details). Note that a similar pricing analysis was recently conducted in [35] in the context of interference management.

**Proposition 2.** If there exists a NE for the game $G$ and if this NE is Pareto optimal, then the linear pricing function factor for user $i$ should be:

$$
\lambda_i(f_k)^{\text{opt}} = \frac{1}{w_i} \sum_{j \in NBR_i} w_j \frac{h_{jj}(f_k)P_j(f_k)h_{ij}(f_k)}{M_j(f_k)(M_j(f_k) + h_{jj}(f_k)P_j(f_k))}
$$

(4.6)

where $NBR_i$ denotes the set of neighbors for user $i$.

**Proof.** By definition, a NE is the solution to the individual utility optimization problem for each user given all other users’ actions. In our formulation, each individual optimization problem is a convex problem with the linear constraints C1-C3 in (5.1). So the Lagrangian function for user $i$ can be written as:

$$
J_i = w_i \sum_{k \in \Omega_K} \hat{u}_i(P_i(f_k)) + \sum_{k \in \Omega_K} \alpha_{i,k}P_i(f_k)
- \beta_i(\sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}}) - \sum_{k \in \Omega_K} \gamma_{i,k}(P_i(f_k) - P_{\text{mask}}(f_k))
- \beta_i(\sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}}) - \sum_{k \in \Omega_K} \gamma_{i,k}(P_i(f_k) - P_{\text{mask}}(f_k))
$$

(4.7)

where $\alpha_{i,k}$, $\beta_i$, and $\gamma_{i,k}$ are the Lagrangian multipliers (non-negative real numbers). The K.K.T. conditions [12] are given by:

$$
\frac{\partial J_i}{\partial P_i(f_k)} = w_i \frac{\partial \hat{u}_i(P_i(f_k))}{\partial P_i(f_k)} - w_i \lambda_i(f_k) + \alpha_{i,k} - \beta_i - \gamma_{i,k} = 0, \forall k \in \Omega_K
$$
\[ p_i(f_k) \geq 0, \forall k \in \Omega_K \]
\[ \alpha_{i,k} p_i(f_k) = 0, \forall k \in \Omega_K \]
\[ \sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}} \leq 0 \]
\[ \beta_i \left( \sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}} \right) = 0 \]
\[ \sum_{k \in \Omega_K} P_i(f_k) - P_{\text{mask}(f_k)} \leq 0, \forall k \in \Omega_K \]
\[ \gamma_{i,k}(P_i(f_k) - P_{\text{mask}(f_k)}) = 0, \forall k \in \Omega_K \]

In contrast, to solve the social optimization problem (4.2) with constraints C1-C3, the Lagrangian function can be written as:

\[ J = \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} u_i(P_i(f_k)) + \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} \alpha_{i,k} P_i(f_k) \]
\[ - \sum_{i \in \Omega_N} \beta_i \left( \sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}} \right) \]
\[ - \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} \gamma_{i,k}(P_i(f_k) - P_{\text{mask}(f_k)}) \]
\[ = w_i \sum_{k \in \Omega_K} u_i(P_i(f_k)) + \sum_{j \in \Omega_N, j \neq i} w_j \sum_{k \in \Omega_K} u_j(P_j(f_k)) \]
\[ + \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} \alpha_{i,k} P_i(f_k) - \beta_i \sum_{k \in \Omega_K} (P_i(f_k) - P_{\text{max}}) \]
\[ - \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} \gamma_{i,k}(P_i(f_k) - P_{\text{mask}(f_k)}) \]  

The K.K.T. conditions for the optimization problem in (4.2) are given by:

\[ \frac{\partial J}{\partial P_i(f_k)} = w_i \frac{\partial u_i(P_i(f_k))}{\partial P_i(f_k)} + \sum_{j \in \Omega_N, j \neq i} w_j \frac{\partial u_j(P_j(f_k))}{P_i(f_k)} \]
\[ + \alpha_{i,k} - \beta_i - \gamma_{i,k} = 0, \forall i \in \Omega_N \text{ and } k \in \Omega_K \]
\[ p_i(f_k) \geq 0, \forall i \in \Omega_N \text{ and } k \in \Omega_K \]
\[ \alpha_{i,k}p_i(f_k) = 0, \forall i \in \Omega_N \text{ and } k \in \Omega_K \]
\[ \sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}} \leq 0, \forall i \in \Omega_N \]
\[ \beta_i \left( \sum_{k \in \Omega_K} P_i(f_k) - P_{\text{max}} \right) = 0, \forall i \in \Omega_N \]
\[ P_i(f_k) - P_{\text{mask}}(f_k) \leq 0, \forall i \in \Omega_N \text{ and } k \in \Omega_K \]
\[ \gamma_{i,k}(P_i(f_k) - P_{\text{mask}}(f_k)) = 0, \forall i \in \Omega_N \text{ and } k \in \Omega_K \]

By comparing K.K.T. conditions in (4.8) and (4.10), to obtain the same solution, we must have:

\[ \lambda_i(f_k) = -\frac{1}{w_i} \sum_{j \in \Omega_N, j \neq i} w_j \frac{\partial u_j(P_j(f_k))}{\partial P_i(f_k)} \]

By substituting \( u_j(P_j(f_k)) \) into (4.11), we have:

\[ \lambda_i(f_k) = \frac{1}{w_i} \sum_{j \in \Omega_N, j \neq i} w_j \frac{h_{jj}(f_k)P_j(f_k)h_{ij}(f_k)}{M_j(f_k)(M_j(f_k) + h_{jj}(f_k)P_j(f_k))} \]

If the transmitter of link \( i \) and the receiver of link \( j \) are not neighbors, i.e., the transmission of link \( i \) at the maximum power cannot reach the receiver of link \( j \), the channel gain \( h_{ij}(f_k) \) is set to zero. Thus, the optimal pricing factor for link \( i \) only depends on its neighborhood information. We then have the result in Proposition 2. \( \square \)

Intuitively, a higher pricing factor \( \lambda_i(f_k) \) will prevent user \( i \) from using a large transmission power on channel \( k \). In view of (4.6), for link \( i \) to determine its optimal pricing factor, the following procedure is needed: If a neighbor \( j \) is to transmit over channel \( k \), it needs to broadcast its transmission power \( P_j(f_k) \), the measured total noise and interference \( M_j(f_k) \), and the channel gain \( h_{jj}(f_k) \) between the transmitter and the receiver of link \( j \). The above information can be incorporated into the control packets of the MAC protocol (details in Section 4.5). In addition, \( h_{ij}(f_k) \) can be measured from the received signal power of the control packet.
4.4 Iterative Algorithms

From the propositions in the previous section, we can use the following iterative procedure to reach the NE. Each individual CR user, say $i$, first adjusts its linear pricing factor $\lambda_i(f_k)$ over all channels according to (4.6), and then determines its best action [63], i.e., the optimal channel/power/rate combination, by measuring the total noise-plus-interference level $M_i$ over all channels. The best response of user $i$ is to maximize its individual utility function (4.4) subject to the constraints C1-C3. The same procedure is repeated for all users in the network. If such a procedure converges, then by definition, it has to converge to a NE of the game in (5.1).

Note that the utility function in (4.1) is monotonically increasing with $P_i(f_k)$ given that the other users’ powers are fixed, and the only condition that prevents user $i$ from choosing infinitely large transmission power is the total power constraint C2. In our work, after adding the linear pricing function, the utility function (4.4) now leads to finite optimal power settings even without the constraint C2.

**Proposition 3.** Treating other users’ transmissions as interference, the best response of user $i$ is given by:

$$P_i = BR_i(P_{-i}) = [BR_i(P_{-i})(f_1), \ldots, BR_i(P_{-i})(f_K)]$$

with

$$BR_i(P_{-i})(f_k) = \left[ \frac{1}{\beta + \lambda_i(f_k)} - \frac{M_i(f_k)}{h_{ii}(f_k)} \right]_{\text{mask}(f_k)}$$

where $[x]^b_a$, with $b > a$, denotes the Euclidean projection of $x$ onto the interval $[a, b]$, i.e., $[x]^b_a = a$ if $x < a$, $[x]^b_a = x$ if $a \leq x \leq b$, and $[x]^b_a = b$ if $x > b$. The water level $\beta$ is chosen to satisfy the total power constraint C2.

A similar result for the IWF algorithm is provided in [73]. Although we have an additional pricing function, a similar analysis can be used to reach the result in Proposition 3. We also provide an alternative proof in the Appendix using the sequential optimization technique as discussed in [23] and [75].
Note that without the power mask constraint and without the pricing function (i.e., \( \lambda_i(f_k) = 0 \) for all \( k \) and \( i \)), (4.13) becomes the classical water-filling solution. Figure 4.3 graphically illustrates the difference between the traditional water-filling [94] and the price-based water-filling solution (4.13). The variable water level in the right-hand side of Figure 4.3 is because of the addition of the pricing factor in (4.14).

![Figure 4.3: IWF versus PIWF.](image)

Several approaches are available for CR users to converge to the NE according to the best response function defined in (4.13). Naturally, CR users may make their decisions one after another or in parallel, which corresponds to sequential and parallel update procedures. The specific algorithms will be described next along with their convergence properties.

### 4.4.1 Sequential Price-based Iterative Water-filling

If CR users are to make their best-response decisions sequentially according to a fixed order, we have a sequential PIWF algorithm, and the algorithm can be generalized as follows:

The condition \( \frac{\|P_i^{(0)} - P_i^{(l-1)}\|}{\|P_i^{(l-1)}\|} \leq \varepsilon \) is the stopping criteria for the PIWF algorithm. Normally, \( \varepsilon \) is set to a small value, such as 5%. If that condition is not satisfied after \( L_{\text{max}} \) iterations, the algorithm terminates. The above algorithm is akin to the Gauss-Seidel procedure [46], where the players take their turns sequentially and act...
Algorithm 2 Sequential PIWF

0: Initialize $P_i(f_k) = 0, \forall i \in \Omega_N$ and $k \in \Omega_K$.
0: Initialize iteration count $l = 0$.
0: Repeat iterations:
1: $l = l + 1$;
2: for $i = 1$ to $N$ users do
3:   for $k = 1$ to $K$ channels do
4:     Estimate the total interference plus noise level $M_i(f_k)$;
5:     Compute the pricing factor $\lambda_i(f_k)$ using (4.6);
6:     Estimate the channel gain $h_{ii}(f_k)$ using the received signal power of the control packet.
7:   end for
8:   $P_i^{(l)} = \text{BR}_i(P_1^{(l)}, \ldots, P_{i-1}^{(l)}, P_{i+1}^{(l-1)}, \ldots, P_N^{(l-1)})$;
9: Transmit on selected channels using $P_i^{(l)}$.
10: end for
11: until $l > L_{\text{max}}$ or $\|P_i^{(l)} - P_i^{(l-1)}\| \leq \varepsilon$ for all $i \in \Omega_N$.

on the most recent policy information obtained from the other players. In a two-user scenario, the $(l+1)$th iteration for user 1 can be expressed as:

$$P_1^{(l+1)} = \text{BR}_1(\text{BR}_2(P_1^{(l)})) = (\text{BR}_1 \ast \text{BR}_2)(P_1^{(l)}) = T(P_1^{(l)})$$ (4.15)

The Nash equilibrium is thus a fixed point [74] under the mapping $T(.)$. For the $N$-user case, the expression is more complicated, but we will keep the notation $T$ as the mapping between the previous power vector and the current power vector. For the IWF algorithm, to ensure convergence to the NE, several sufficient conditions were proposed in the literature. The convergence condition was first provided in [94] for two-user cases and in [20] for $N$-user cases. More recently, the convergence conditions have been further relaxed in [36] and [73].

Since the utility function (4.4) in our formulation possesses an additional pricing function, the previous convergence proofs may not be applicable any more. In fact, since we are using a time-varying pricing factor in Algorithm 1, the mapping function $T(.)$ is also time-varying. Thus, the fixed point theorem that underlies the proofs in [36] and [73] can no longer be used. The convergence proof with a time-
The varying mapping function is challenging and will be left for a future work. However, convergence is always observed in our simulations for various network conditions. Figure 4.4 depicts the convergence behavior over several iterations with $N = 10$ and $K = 5$. In the test network, ten CR pairs are randomly placed in a square area. The figure shows the average sum-rate improvement of the sequential PIWF over the classic IWF algorithm for 1000 runs, with the starting sum-rate of the IWF algorithm normalized to one. The two algorithms converge at comparable speeds, but the NE solution for the sequential PIWF algorithm is much better than the NE of the classic IWF algorithm.

![Figure 4.4](image)

Figure 4.4: Normalized system sum-rate versus iterations.

Although the convergence proof for the time-varying pricing function is difficult to establish, if the pricing factor remains fixed over a few iterations, the convergence proof in [73] is still applicable. This is because adding a linear pricing function with a fixed pricing factor to the utility function in (4.1) has no impact on the convergence proof in [73]. If we take the result in [73] and apply it to our CRN setting, we have the following proposition:

**Proposition 4.** Given a linear pricing function with a fixed pricing factor over a
few iterations, the sequential update procedure converges to the unique NE if one of
the two following sets of conditions is satisfied:

\[
\sum_{j \in \Omega_N, j \neq i} \max_{k \in S_i \cap S_j} \frac{h_{ji}(f_k)}{h_{ii}(f_k)} < 1, \forall i \in \Omega_N \tag{4.16}
\]

\[
\sum_{i \in \Omega_N, i \neq j} \max_{k \in S_i \cap S_j} \frac{h_{ji}(f_k)}{h_{ii}(f_k)} < 1, \forall j \in \Omega_N \tag{4.17}
\]

From (4.16) and (4.17), the convergence and the uniqueness of the NE are ensured if the CRs that share the same channel are far apart from each other, and thus cause weak interference on one another. Figure 4.5 graphically illustrates these two conditions in a CRN that consists of three CR links. Each link is allocated two channels (e.g., nodes A and B are allocated channels 1 and 3). Condition (4.16) indicates that for each CR receiver, the summation of the normalized channel gains between that receiver and all interfering CR transmitters that share the same channel (normalized by the channel gain of the intended packet) is less than 1. For example, for node B, this condition reduces to \( \frac{h_{CD}(f_1)}{h_{AB}(f_1)} + \frac{h_{EB}(f_3)}{h_{AB}(f_3)} < 1 \). Similarly, this condition is applied to receivers D and F. If this condition is satisfied at all CR receivers, then according to Proposition 4 the sequential update procedure converges to the unique NE. The second condition (4.17) indicates that for each CR transmitter, the summation of the normalized channel gains between that transmitter and unintended CR receivers sharing the same channel is less than 1. For example, for transmitter A (4.17) reduces to \( \frac{h_{AD}(f_1)}{h_{CD}(f_1)} + \frac{h_{AF}(f_3)}{h_{EF}(f_3)} < 1 \). Similarly, we can derive this condition for transmitters C and E. If this condition is satisfied at all CR transmitters, it is also sufficient to assert that the sequential update procedure converges to the unique NE.

If the number of users in the network is large, the sequential update procedure may suffer from slow convergence. Therefore, we now discuss a parallel version of the PIWF algorithm.
4.4.2 Parallel Price-based Iterative Water-filling

Algorithm 2 describes a parallel update version of the PIWF algorithm (the parallel IWF concept was first introduced in [72]). The stopping criteria for the parallel PIWF are the same as those of the sequential PIWF. The parallel PIWF algorithm is related to the Jacobi computational procedure [30], where in each iteration CR users simultaneously perform price-based water-filling based on the interference generated by the other users in the previous iterations. In a two-user case, the counterpart of (4.15) is:

$$P_1^{(l+2)} = BR_1(P_2^{(l+1)}) = BR_1(BR_2(P_1^{(l)})) = T(P_1^{(l)})$$  \hspace{1cm} (4.18)

In [73], it was proved that the convergence conditions for the parallel IWF and the sequential IWF are the same. For a time-varying PIWF, the proof is not applicable. But if the pricing factor of the linear pricing function remains fixed over a few iterations, we can apply the corresponding proof and arrive at the following corollary of Proposition 4.

**Corollary 1.** If the conditions of Proposition 4 are satisfied, the parallel update procedure converges to the unique NE of the game.

Corollary 1 shows that stability under the Gauss-Seidel procedure coincides with stability under the Jacobi iteration. Furthermore, following the argument in [73], one can prove that any asynchronous computation in which the players act at random
Algorithm 3 Parallel PIWF

0: Initialize $P_i(f_k) = 0, \forall i \in \Omega_N$ and $k \in \Omega_K$.
0: Initialize iteration count $l = 0$.
0: Repeat iterations:
1: $l = l + 1$;
2: for $i = 1$ to $N$ users do
3: for $k = 1$ to $K$ channels do
4: Estimate the total interference plus noise level $M_i(f_k)$;
5: Compute the pricing factor $\lambda_i(f_k)$ using (4.6);
6: Estimate the channel gain $h_{ii}(f_k)$ using the received signal power of the control packet.
7: end for
8: $P_i^{(l)} = BR_i(P_1^{(l-1)}, \ldots, P_{i-1}^{(l-1)}, P_{i+1}^{(l-1)}, \ldots, P_N^{(l-1)})$;
9: end for
10: for $i = 1$ to $N$ users do
11: Transmit using $P_i^{(l)}$.
12: end for
13: until $l > L_{max}$ or $\frac{||P_i^{(l)} - P_i^{(l-1)}||}{||P_i^{(l-1)}||} \leq \varepsilon$ for all $i \in \Omega_N$.

and use the most recent available policy from other players converges to a NE, as long as no players remain idle for an infinite duration. Hence, the achieved NE based on asynchronous updates coincides with the NE achieved with parallel or sequential updates.

The parallel and sequential PIWF procedures are distributed algorithms that maximize the total achievable data rate. Both have the same implementation complexity of the traditional IWF. As shown in Figure 4.4, these two algorithms greatly outperform IWF. In Figure 4.6, we can see that the parallel PIWF converges faster than the sequential PIWF, especially for a large number of users. In this simulation, we assume that CRs are randomly located in a square area and 5 channels are available for their transmissions. Whether the players act sequentially or in parallel makes a difference in the MAC design. In Section 4.5, we discuss the impact of the update procedure on the MAC design.
Figure 4.6: Convergence speed of the sequential/parallel PIWF.

4.4.3 Relaxation Algorithms

Both the sequential and parallel PIWF algorithms require the system parameters to be correctly estimated for each CR. This condition may not be satisfied in practical systems. To overcome this problem, a “relaxed” update scheme can be used (as in [11], [53], and [73]), and will be discussed here for completeness. In such a “relaxed” version, each CR is required to remember its most recent policy choices together with the choices of other users. The relaxed algorithms are more robust to occasional estimation errors and channel oscillations, but lead to certain degradation in the convergence speed.

More specifically, we can achieve a relaxed version of the sequential PIWF algorithm if the best response function in Algorithm

\[
P_{i}^{(l)} = \omega P_{i}^{(l-1)} \\
+ (1 - \omega) BR_i(P_1^{(l)}, \ldots, P_{i-1}^{(l)}, P_{i+1}^{(l-1)}, \ldots, P_N^{(l-1)})
\]  (4.19)
where the factor $\omega \in [0,1)$ can be interpreted as the memory factor. The larger the value of $\omega$, the longer the memory of the algorithm. With a larger $\omega$, the algorithm is more robust to estimation errors at the cost of slower convergence.

Similarly, we can arrive at a relaxed version of the parallel PIWF algorithm if the best response in replaced by:

$$
P_i^{(l)} = \omega P_i^{(l-1)} + (1 - \omega)BR_i(P_i^{(l-1)}, \ldots, P_i^{(l-1)}, P_{i-1}^{(l-1)}, \ldots, P_{i+1}^{(l-1)}, \ldots, P_N^{(l-1)}) \tag{4.20}
$$

As proved in [73], the relaxation algorithms converge to the unique NE of the game for any $\omega \in [0,1)$ under the conditions in Proposition 4.

All the above proposed iterative algorithms are rate-adaptive (RA), where the data rates of users are maximized under transmission power constraints. Similarly, we can design “margin-adaptive” (MA) algorithms where users attempt to minimize their transmission powers while satisfying a target data rate. Both RA and MA algorithms follow similar mechanisms. Due to space limitations, we do not discuss MA algorithms in this chapter.

4.5 MAC Protocol Design

In this section, we describe a MAC protocol that allows CR users to operate efficiently in an opportunistic CRN. This protocol implements the distributed channel/power allocation strategies discussed in the previous sections. It should be noted that a number of multi-channel MAC protocols have been proposed in the context of CRNs (e.g., [49], [102], [100], and [78]). Most of them do not allow multiple CR transmissions within the same neighborhood to overlap in frequency channels, so there is no interference among CR users. Such a restriction simplifies the MAC design, but limits its spectrum efficiency. A natural extension (analogous to the improvement offered by the POWMAC protocol [58] over the classic CSMA/CA) is to allow CR users to overlap in spectrum, provided that their mutual interference does not lead to collisions. The IWF algorithm [94] and the no-regret algorithm [62]
were proposed as two possible enabling techniques. However, the works in [94] and [62] provide only channel/power allocation algorithms and do not offer a practical MAC design. In this section, we incorporate our price-based channel/power allocation algorithms into an operational MAC protocol. Since the IWF algorithm is a special case of the proposed PIWF algorithm, our MAC protocol can be simplified to accommodate the classic IWF algorithm, thus complementing the work in [94].

4.5.1 Assumptions

We consider a CRN setting with the following features:

- A dedicated control channel or a coordinated control channel [100] is used to support a community of CR users. Control packets are transmitted over the control channel using a pre-assigned power value $P_{cont}$.
- Channel gains between any two terminals are symmetric.
- The channel gain is static for the duration of several control packets and a flow of data packets.

4.5.2 Protocol Overview

Our MAC protocol uses three types of control packets for the handshaking between a CR transmitter and a CR receiver: Request-to-Send (RTS), Clear-to-Send (CTS), and Decide-to-Send (DTS). Unlike in the classic CSMA/CA scheme and other multi-channel MAC protocols for CRNs, these control packets are not used to exclusively reserve channels (i.e., prevent neighboring CRs from accessing the reserved channels), but rather to exchange some information within the neighborhood. Such information is used by terminals to determine their transmission parameters.

The control packets are exchanged within a certain duration, referred to as the contention window (CW). A CW can be initiated asynchronously by any CR user
that has packets to transmit and that is not aware of any active CWs in its neighborhood. Such a user is referred to as a *master user*. Other CR users that follow the schedule of an ongoing CW are called *slave users*. Note that the master/slave designation of a user is dynamic, i.e., it changes with traffic and mobility conditions. The objective of the CW is to allow several pairs of CR nodes to repeatedly negotiate their transmission channels and powers. As shown in Figure 4.7, the CW is divided into two parts. The first part, referred to as the *access window* (AW), is used by CR nodes to compete for admission to the CW and initialize their transmission policies. The second part, referred to as the *training window* (TW), is used by the CR nodes to repeatedly negotiate their channel/power policies (as explained later). Note that the AW can be considered as the first iteration of the training process. CR nodes that have been successfully admitted during the CW transmit a flow of data packets over one or multiple data channels (as determined during the CW) within a *data window* (DW). The durations for the AW and DW are changed adaptively, similar to the single-channel POWMAC protocol [58]. As for TW, its size (in slots) is dictated by the convergence speed of the iterative resource allocation algorithm. In general, an unnecessarily large value increases the overhead, but does not necessarily improve the throughput (as shown in Figure 4.4). On the other hand, a small value may give sub-optimal results. In Section 4.6, we study the performance of the MAC protocol under various TW sizes.

Figure 4.7: Overview of the MAC operation with two CR transmissions (*A* → *B* and *C* → *D*).
4.5.3 Operation Details

Access Window

When a CR node $A$ intends to communicate with another node $B$, it first needs to contend during the AW. If node $A$ is not aware of any ongoing AW in its neighborhood, it initiates a new AW (i.e., it becomes a master user). Otherwise, node $A$ contends during one of the slots of the ongoing AW. In either case, node $A$ first backs off by a random amount of time, selected from $[T_{\text{min}}, T_{\text{max}}]$, before accessing the channel.

The AW consists of a number of fixed-size slots. The size of each slot is $T_{\text{max}}$ plus the durations of the RTS, CTS and DTS packets, plus 3 SIFS durations (SIFS denotes the short interframe spacing between successive control packets). In each slot, CR nodes compete for admission following a standard CSMA approach.

If CR $B$ successfully receives the RTS packet from $A$, it needs to decide the initial channel/power policy for the link $A \rightarrow B$. This is done as follows:

- First, node $B$ estimates the channel gain between itself and node $A$ (denoted by $h_{AB}(f_0)$). This is facilitated by knowledge of the RTS’s transmission power ($P_{\text{cont}}$) and the received power of the RTS. From $h_{AB}(f_0)$, CR $B$ computes $h_{AB}(f_k)$ for all $k \in \Omega_K$. The determination of $h_{AB}(f_k)$ from $h_{AB}(f_0)$ is made possible by knowing the carrier frequencies and by assuming a certain path-loss model. For example, under the two-ray model [67] and for a given transmission power, $h_{AB}(f_k) = h_{AB}(f_0) \times (f_0/f_k)^2$, where $f_0$ is the carrier frequency of the control channel.

- Next, node $B$ measures $M_B = [M_B(f_1), M_B(f_2), \ldots, M_B(f_K)]$ over all data channels. Note that for the sequential PIWF algorithm, if there are previous CTS/DTS packets that have been received in the same AW, $M_B$ is computed as the sum of the current $M_B$ and the predicted CR-to-CR interference, which is obtained by assuming that the neighboring links transmit using the channels/powers specified in their CTS/DTS.
• Then, node $B$ determines the pricing factor $\lambda_B(f_k)$ for all data channels $k$. For the sequential PIWF algorithm, $\lambda_B(f_k)$ is computed using (4.6), where the neighborhood information is obtained from previously received CTS/DTS packets in the same AW. For the parallel PIWF algorithm, $\lambda_B(f_k)$ is initialized to 0.

• Finally, based on the above information, node $B$ decides its best-response transmission policy according to Proposition 3.

After the above procedures have been executed, node $B$ will send a CTS, announcing its channel/power allocation. The CTS includes $M_B(f_k)$ and $h_{AB}(f_k)$ for all $k \in S_B$, which are used by neighboring CRs to update their best responses. Note that even if the set of selected channels $S_B$ is empty (i.e., the computed transmission power is zero for all channels), the link $A \rightarrow B$ will still be admitted in the AW. This is because the data transmission $A \rightarrow B$ may later be allowed to proceed after several iterations in the TW.

If node $A$ receives the CTS from $B$, it will respond with a DTS packet, repeating the information included in the CTS. This DTS is used to alleviate the hidden terminal problem as in [58].

The above steps are repeated by CR pairs in every AW slot.

**Training Window**

CR nodes that are admitted in the AW iteratively negotiate their transmission parameters in the TW, following the same order of their admissions in the AW. In contrast to the AW, the TW is accessed in a TDMA manner. The TW consists of a number of slots (the TW size), where each slot is used to conduct one iteration of the channel/power allocation algorithm, using CTS and DTS packets. Note that there is no need for the RTS during the TW, since new admissions are not allowed.

In each iteration, the receiver of a CR link updates the transmission policies based on the policies of its neighbors. The updates are made based on either the sequential or the parallel scheme. Specifically, if the sequential PIWF algorithm is
applied, the transmission policy of each CR user is made based on the policies of all previous users in the same iteration (obtained from CTS/DTS packets) and those of the other users in the previous iteration, as described in parallel PIWF algorithm is applied, the policy of each CR user is made based on the policies of other CR users in the previous iteration, as described in the AW is regarded as the initial iteration of the training process. After each computation, the receiver sends a CTS, announcing its transmission policy. Upon receiving the CTS, the transmitter will send a DTS, repeating the information in the CTS.

**Data Window**

The last negotiated transmission policies in the TW are used by the CR nodes for data transmissions in the DW. In the DW, a flow of data packets is transmitted from each CR transmitter. The length of the flow is selected such that the channel conditions remain static over the entire flow. Obviously, the DW size needs to be selected according to the channel’s coherence time.

### 4.5.4 Simplified Packet-based MAC Design

The above MAC design can be used for flow-based channel access, where a sequence of data packets are transmitted using converged channel/power policies agreed upon during the TW. Thus, the sum-rate of all competing CRs is likely to be maximized if the channels remain static over the duration of the data flow. However, if sum-rate optimality is not critical, we can simplify the protocol by removing the TW and only allow for a single data-packet transmission in the DW. This design then becomes packet-based, and the convergence is now achieved after several sessions of CW and DW (provided that channel conditions remain static within this period).

Note that in the previous section, all CR nodes contend in the AW with equal probability. In contrast, in the packet-based MAC design, the admitted users in the previous AW have priorities in accessing the control channel over other CR users. Specifically, the admitted links in the previous AW will contend in the current AW without backoff, according to their order in the previous AW, as long as they still
have packets to transmit. After these links have been admitted, other links compete for the remaining slots, following the backoff mechanism that was discussed in the previous section. Such a design is meant to facilitate the convergence behavior. To ensure fairness among users, we set a limit $(\theta)$ on the maximum number of continuously prioritized packets. Specifically, if one CR user acquires the channel in one session, it will have priorities in accessing the control channel for the next $\theta - 1$ packet transmission durations (as long as the user has packets to transmit). The parameter $\theta$ is selected to be larger than the convergence time of the algorithm.

The channel/power policies are updated in the AW following similar procedures to the flow-based MAC. The only difference is that the interference-plus-noise level is now estimated from the previous DW, instead of the previous iteration in the TW. In the next section, we compare the performance of this design with that of the flow-based MAC.

4.5.5 Implementation of Relaxed Algorithms

The implementation of the “relaxed” version algorithms to the MAC design is straightforward. Each CR receiver memorizes its most recent policy, and makes decisions based on (4.19) for the sequential algorithm (or (4.20) for the parallel algorithm). Since the convergence speed of the “relaxed” PIWF is slower, a larger TW size is needed for the flow-based MAC protocol. Similarly, for the packet-based MAC, a larger number of data packets are transmitted before the convergence can be achieved.

4.6 Performance Evaluation

To evaluate the effectiveness of the proposed MAC, we conduct MATLAB-based simulations of a hybrid network with one PRN and one CRN. Nodes in these networks are uniformly distributed over a square area of length 100 meters. The PRN operates in the 300 MHz band, occupying five non-overlapping 1-MHz channels, with 10 PRs in each channel. The time is divided into slots, each of length 10 ms.
In each slot, each PR attempts to transmit with a probability of $\alpha$, the PR’s activity factor. The transmission power of each PR is 1 Watt when it is on, and the antenna length is 5 cm.

We use the following signal propagation model to simulate the PR-to-CR and CR-to-CR interference over channel $k$ [67]:

$$P_r(f_k) = P_{d_0}(f_k) \left( \frac{d}{d_0(f_k)} \right)^{-\gamma}, \text{ for } d > d_0(f_k)$$

(4.21)

where $P_r(f_k)$ is the received power over channel $k$, $d$ is the distance between the transmitter and the receiver, $d_0(f_k)$ is the reference distance, $P_{d_0}(f_k)$ is the reference received power at distance $d_0(f_k)$ over channel $k$, and $\gamma$ is the path loss exponent. We set $d_0(f_k) = 1$ meter, $\gamma = 4$, and we compute $P_{d_0}(f_k)$ as follows [67]:

$$P_{d_0}(f_k) = P_t(f_k)G_t(f_k)G_r(f_k)(\frac{\nu(f_k)}{4\pi d_0(f_k)})^2$$

(4.22)

where $P_t(f_k)$ is the transmission power on channel $k$, $G_t(f_k)$ and $G_r(f_k)$ are the transmitter and receiver antenna gains on channel $k$, and $\nu(f_k)$ is the wavelength of the carrier frequency of channel $k$. For simplicity, we set $G_t(f_k)G_r(f_k) = 1$ for all channels.

We simulate 10 pairs of CRs. The maximum transmission power for CR is 1 Watt, which ensures that CRs are within the maximum transmission range of each other. The AWGN noise level $N_0$ is set to $-70$ dBm over all channels. Each CR transmitter generates burst of packets according to a Poisson process with parameter $\Lambda$ burst/second. Each burst has an exponentially distributed duration with mean $1/\mu$ second. The traffic rate for CR is defined as $\Lambda/\mu$. We set the CR-to-PR power mask to 0.5 Watt for all channels.

We compare the performance of the proposed flow-based PIWF-MAC protocol with the packet-based PIWF-MAC protocol, against the flow-based IWF-MAC protocol. Since the IWF algorithm is a special case of the PIWF algorithm, the proposed MAC protocols are also applicable to the IWF algorithm. The DW for the flow-based MAC protocol allows 10 data packets to be transmitted in a row.
The comparison is in terms of the system throughput and the average power consumption. The system throughput is defined as the average volume of CR traffic bits that are transmitted in one second, and the power consumption is calculated as average power consumption by all CRs.

The resulting performance is depicted in Figure 4.8 through Figure 4.10. Figure 4.8(a) shows the system throughput versus the traffic rate. As expected, the flow-based PIWF-MAC protocol gives the highest throughput. This throughput improvement over IWF-MAC becomes more significant at higher traffic rates. It is interesting to see that the simplified packet-based PIWF-MAC protocol exhibits comparable system throughput with the flow-based PIWF-MAC protocol. Besides achieving a higher system throughput, the PIWF-MAC protocols also save transmission power, as shown in Figure 4.8(b). This is because in IWF, users greedily maximize their own rates using the maximum transmission power, while such greedy behaviors are overcome by the pricing technique used in PIWF. Note that although the packet-based PIWF-MAC consumes less energy in control overhead than the flow-based PIWF-MAC, it consumes more energy in data transmissions. This is because in the packet-based PIWF-MAC, the optimal power assignment is achieved after several packet transmissions (due to the absence of a control-based training phase). The confluence of the two energy-consumption factors still favors the flow-based PIWF-MAC.

Figure 4.9(a) depicts the network throughput versus the PR’s activity factor $\alpha$. As expected, a higher $\alpha$ results in a higher PR-to-CR interference, which negatively affects the throughput. Figure 4.9(b) shows the corresponding average power consumption. In all cases, PIWF-MAC protocols consume less power.

Finally, Figure 4.10(a) shows the throughput versus the training window size. Since the simplified packet-based PIWF-MAC does not employ a TW, we only compare the flow-based PIWF-MAC and flow-based IWF-MAC. Intuitively, a larger TW size will ensure that CR users will converge to the NE. However, as seen in Figure 4.4, 2-3 iterations are normally sufficient to reach a near-optimal sum-rate. The same behavior is observed from the MAC simulations. As seen in Figure 4.10(a),
Figure 4.8: Performance when $\alpha = 0.1$. 

(a) System throughput vs. traffic rate.

(b) Average power consumption vs. traffic rate.
Figure 4.9: Performance under traffic rate $\Lambda/\mu = 0.5$. 

(a) System throughput vs. PR activity factor.

(b) Average power consumption vs. PR activity factor.
taking a TW of size 2 is enough to achieve 95% of the maximum system throughput in the simulation setup. Figure 4.10(b) shows the corresponding average power consumption.

4.7 Conclusions

In this chapter, we proposed a PIWF algorithm for spectrum sharing in cognitive radio networks. Our PIWF algorithm can be implemented distributively with CRs repeatedly negotiating their transmission powers and spectrum. Simulation results showed that the proposed algorithm greatly improves the NE compared with the one achieved using the IWF approach. Based on the order by which CR nodes make their resource allocation decisions, we studied sequential and parallel versions of the PIWF algorithm. The parallel update scheme was shown to converge faster than the sequential update scheme, especially for a large number of users. We also presented “relaxed” versions of the PIWF algorithms, which are more robust to estimation errors and channel oscillations at the cost of slower convergence. Based on the PIWF algorithms, flow-based and packet-based MAC protocols were designed. Our simulation results showed that the PIWF-MAC protocol achieves considerably higher system throughput compared with the IWF-MAC, with less energy consumption.
Figure 4.10: Performance under traffic rate $\Lambda/\mu = 0.7$ and $\alpha = 0.1$. 

(a) System throughput vs. TW size.

(b) Average power consumption vs. TW size.
CHAPTER 5

ENERGY-ORIENTED CHANNEL ACCESS DESIGN FOR INTERFERENCE-BASED MULTI-CHANNEL WIRELESS NETWORKS

In this chapter, our goal is to minimize the network-wide energy utilization while satisfying rate demand and power mask constraints at each CR. Same as Chapter 4, we allow neighboring CRs to share the spectrum. The resource allocation problem is modeled as a non-cooperative game, where CRs repeatedly negotiate their best powers and spectrum selection to reach a NE. Due to the selfishness of the CR users, this NE is generally socially inefficient. Accordingly, we propose two incentive mechanisms to improve the social efficiency of the NE. Simulations are used to demonstrate the effectiveness of our models.

5.1 Introduction

In this chapter, we focus on resource allocation and channel access design for a multi-user (ad hoc) opportunistic CRN. A distributed radio resource allocation (RRA) framework is proposed to achieve high network-wide resource utilization for the CRN. Conventionally, such an RRA problem is formulated as either a centralized optimization problem or a distributed control problem. A centralized formulation (e.g., [88; 104; 15; 100; 33; 76; 92; 97]), requires the global knowledge of the channel and transmission conditions at each node, incurring high overhead and complexity for implementation. The existing distributed schemes are mostly based on heuristic control logics (e.g., various stepped-adjustment of power/rate based on local observations or local bargains, see [42; 103; 14; 77]). Although such schemes simplify implementations, their oversimplified control mechanism usually undermines the effectiveness of their control, leading to poor network performance. The RRA
framework proposed in this work fills an important gap between the complexity and the performance. Specifically, our algorithm is a distributed scheme and only requires local information of a CR. However, it implements more advanced optimization considerations in its operation than the existing distributed schemes and provide highly efficient resource utilization from a network’s standpoint.

We emphasize that our RRA framework is in definite contrast to the related works that optimize the resource utilization of a single CR user. For example, assuming a semi-Markov process for the PR traffic, Kim and Shin [48] proposed a sensing-period adaptation algorithm to maximize the throughput of a CR link by minimizing the delay in finding an available channel. Based on a similar PR traffic model, the authors in [37] studied a throughput-optimal dynamic access scheme subject to a constraint on the CR-to-PR violation rate, but only for a system of one PR network and one CR link. Other works that only optimize for a single CR link include the formulations based on the partially observable Markov decision process (POMDP) model [102], the constrained Markov decision processes (CMDPs) model [101], and the optimal stopping-rule models [17; 44].

In this chapter we relax the “exclusive channel reservation” requirement and allow neighboring CR transmissions to overlap in the frequency spectrum. The resource allocation problem is then modeled as a non-cooperative game, where CRs repeatedly negotiate their best powers and spectrum until they converge to a NE for the game. Depending on the performance and overhead of the exchanged information (e.g., channel gain), we consider three update strategies. The selfish update strategy requires a small amount of overhead but achieves modest performance. The optimal update strategy achieves the best performance at the cost of exchanging a large amount of information among neighbors and losing the convexity of the optimization problem. Finally, the simple incentive-based strategy balances the performance and overhead while preserving convexity. All of the above strategies do not require the interference statistics.

Different from the information-theory-based optimization methodology used in existing literature, our work under the interference-channel setup takes an opera-
tional view in studying the problem. In particular, we are interested in achieving reasonably good performance at the expense of light to moderate implementation overhead, while existing literature are focused on finding the optimal achievable rate-region for the interference channel. Implementation overhead is not a major concern in their studies. More specifically, focusing on one CR link, the authors in [41] investigated the capacity of opportunistic communications in the presence of dynamic spectral activity. For the simple but nontrivial case of two non-cooperative users, the author in [94] proposed an iterative water-filling solution for the joint power allocation and spectrum sharing problem. Under mild conditions, this solution converges to a NE. For more than two users, the convergence behavior of this solution is still an open issue. Even for the two-user case, the achievable rate region has not been obtained yet. If full collaboration is allowed among receiving nodes, the convergence of the sum-rate-maximizing iterative water-filling solution was demonstrated in [96] for the multi-user case. In [65], the authors proposed a complex centralized spectrum allocation scheme for CRs, which aims at maximizing the aggregate throughput. Game theoretic approaches to spectrum allocation via a spectrum server were proposed in [38], where the competitive allocation solution was shown to be the NE of the game. As a distributed mechanism, the work in [85] modeled the resource allocation problem in CRN as a noncooperative game and proposed a price-based iterative water-filling algorithm to achieve good CRN throughput performance.

Simulations are used to study and compare the performance of our formulations and demonstrate their effectiveness in improving the network throughput and reducing the connection blocking probability.

The rest of the chapter is organized as follows. The system model and the problem statements are described in Section 5.2. Optimal resource assignments under different models are discussed in Section 5.3. Simulation results are presented in Section 5.4, followed by conclusions in Section 5.5.
5.2 Problem Formulation

In this section, we discuss the system model and resource constraints for joint spectrum/power/rate adaptation in an opportunistic CRN.

5.2.1 System Model

We consider a decentralized (ad hoc) CRN that coexists geographically with $K$ legacy PRNs, each licensed to operate over its own frequency band/channel. Let $\Omega_K$ denote the set of $K$ non-overlapping channels. For all $k \in \Omega_K$, let $W_k$ and $f_k$ be the bandwidth and carrier frequency associated with the $k$th PRN. In reality, a PRN may occupy more than one frequency band. Such a network can be easily represented as multiple (virtual) PRNs that operate over different bands. The total spectrum $W = \sum_k W_k$ may be used opportunistically by CR users, subject to constraints that will be discussed shortly. The CRN consists of $N$ CR links. Let $\Omega_N$ denote such a set. Each CR link $i$ can use multiple channels for its data transmission. We denote the set of utilized channels for CR link $i$ as $S_i$. Let $P_{ik}$ and $R_{ik}$, $k \in S_i$, be the transmission power and rate for link $i$ over channel $k$, respectively.

In principle, there are three types of interference that the CRN has to consider: CR-to-PR, PR-to-CR, and CR-to-CR interference. Our model in the chapter takes into account the CR-to-CR interference by allowing CR links in the same neighborhood to operate over the same channels. We assume that each CR link, say $i$, periodically senses the $K$ data channels and estimates the overall instantaneous interference vector $I_i = (I_{i1}, \ldots, I_{iK})$ at its receiver. This interference vector is used in the channel selection, rate allocation, and power control algorithms, as described later. We further assume that the channel gain is constant within a decision period, which is appropriate for low to moderate mobility scenarios. As such, the randomness in the channel’s SINR is solely attributed to $I_i$. For channel $k$, let $\gamma_{ik} \overset{\text{def}}{=} \frac{g_{ik} P_{ik}}{I_{ik} + N_0}$ be the received SINR for CR link $i$ over channel $k$, where $g_{ik}$ is the channel gain between the transmitter and receiver of link $i$ over channel $k$, and $N_0$ is the AWGN power, which is assumed to be the same across all channels.
Besides the $K$ data channels, we assume the availability of a control channel of bandwidth $W_{K+1}$. This channel is needed to enable the exchange of control messages between CR nodes. Although in a pure opportunistic CR environment, the feasibility of dedicating a control channel for the CRN may be questionable, we argue that because of the short length of control packets, the bandwidth of the control channel is typically negligible compared with that of the data channels, i.e., $W_{K+1} \ll \sum_{k=1}^{K} W_k$. Thus, allocating such a channel does not significantly affect the overall spectrum utilization. Moreover, the control channel may be shared with other networks (e.g., it could be one of the unlicensed ISM bands).

### 5.2.2 Resource Constraints

The following constraints need to be satisfied by any CR transmission, say $i$:

**C1: Rate Demand ($R^*$):** For all CR links $i \in \Omega_N$, we require $\sum_{k \in S_i} R_{ik} \geq R^*$. Note that this constraint may be applied at the packet level or at the flow level. More details will be provided in Section 5.3.

**C2: CR-to-PR Power Mask:** For all $i \in \Omega_N$ and all $k \in S_i$, the transmission power $P_{ik}$ must satisfy $E[P_{ik}] \leq \Pi_k$. The vector $\Pi = (\Pi_1, \ldots, \Pi_K)$ is the CR-to-PR power mask. This mask is needed to ensure that CR transmissions do not cause unacceptable interference to neighboring PRs. The determination of an appropriate $\Pi$ is an important issue and may accommodate various considerations.

### 5.3 Resource Allocation Strategies

In this section, we focus on the resource assignment problem at the physical/link layers, which amounts to determining the appropriate frequency bands for various CR transmissions and their corresponding powers and rates subject to various constraints. A key challenge in optimizing the operation of a CRN is how to deal with PR interference. Even if the receiver of a CR link $i$ is capable of estimating the
instantaneous interference vector $I_i$, this vector is likely to vary with time, according to the traffic and mobility dynamics of the PR users. In addition, a feedback channel from the CR receiver to the transmitter may not always exist.

As discussed before, in our model, we allow channel sharing among neighboring CR links. For a single channel, this problem is known in the literature as the “interference channel problem” [71]. Game theory has been used to study the dynamics of such systems (e.g., [94], [85], and [72]).

For the channel sharing model, we analyze the distributed resource assignment problem as a non-cooperative game and allow the CR links to iteratively negotiate their channel/power/rate parameters. More specifically, each CR link, say $i$, measures its instantaneous interference vector $I_i$ in each packet. Note that $I_i$ now includes both PR-to-CR and CR-to-CR interferences. According to the measured value of $I_i$, the CR user individually selects its spectrum and transmission powers to satisfy the power mask and rate constraints. This procedure is repeated for all CR links. Under certain conditions, the power/rate solution eventually converges to the NE, as discussed later.

In making their resource assignment decisions, CR users may adopt different strategies. In the following, we study three such strategies. The first is referred to as the selfish update strategy, where each CR user minimizes its own transmission powers without considering how this impacts other users. This strategy is simple to implement but may result in poor system performance. The second strategy is the optimal update strategy, where each CR user ensures that its rate constraint and those of its neighbors are satisfied when selecting its own transmission powers. We prove that this game has the same K.K.T. conditions as those of the global power minimization problem, and thus it converges at least to a local optimum. However, in the formulation of this strategy, we lose the convexity property. Therefore, we propose a third simplified update strategy, where each CR user is assigned additional frequency-dependent weights in the individual power minimization problem. The weight here represents the impact of a particular CR user on neighboring CR links.
This strategy greatly improves the performance compared with the first strategy, while keeping the computational complexity reasonable.

5.3.1 Selfish Update Strategy

In here, the power minimization problem is equivalent to the margin adaptive (MA) problem, and is known in the literature as “the dual form” of the rate-adaptive (RA) problem [12]. Therefore, RA algorithms for the interference channel, such as iterative water-filling (IWF) [94], can be applied. More specifically, we define the following non-cooperative game, where each CR link, say $i$, minimizes its total transmission power over all channels subject to the rate and power mask constraints:

\[
\begin{align*}
\text{minimize} & \quad \mathbf{P}_i = [P_{i1}, \ldots, P_{iK}] \\
\text{s.t.} & \quad C1: \quad P_{ik} \geq 0, \; \forall k \in \Omega_K \\
& \quad C2: \quad \sum_{k \in \Omega_K} R_{ik} \geq R^* \\
& \quad C3: \quad P_{ik} \leq \Pi_k, \; \forall k \in \Omega_K.
\end{align*}
\]

This optimization procedure is executed by all CR users, and the solution, if exists, is a NE. We note that this approach is fully distributed, and each optimization step leads to a water-filling solution. However, because each user only minimizes its own power without regard to other users, the resulting NE is likely to be socially inefficient when compared with the following global power minimization problem:

\[
\begin{align*}
\text{minimize} & \quad \mathbf{P} = [P_{11}, \ldots, P_{1K}, P_{21}, \ldots, P_{2K}, \ldots, P_{N1}, \ldots, P_{NK}] \\
\text{s.t.} & \quad C1': \quad P_{ik} \geq 0, \; \forall i \in \Omega_N \text{ and } \forall k \in \Omega_K \\
& \quad C2': \quad \sum_k R_{ik} \geq R^*, \; \forall i \in \Omega_K \\
& \quad C3': \quad P_{ik} \leq \Pi_k, \; \forall i \in \Omega_N \text{ and } \forall k \in \Omega_K.
\end{align*}
\]

To drive the game towards a more socially efficient solution, we apply the incentives technique, as shown in the next section.
5.3.2 Optimal Update Strategy

In [85], we studied the multiuser RA problem and imposed incentives on the utility function (sum-rate) in the form of a pricing term. This corresponds to changes in the constraints in its dual (sum-power) problem. More specifically, we define the following game for each CR user \(i\):

\[
\begin{align*}
\text{minimize} & \quad \sum_{k \in \Omega_K} P_{ik} \\
\text{s.t.} & \quad P_{ik} \geq 0, \quad \forall k \in \Omega_K \\
& \quad \sum_{k \in \Omega_K} R_{ik} \geq R^* \\
& \quad P_{ik} \leq \Pi_k, \forall k \in \Omega_K. \\
& \quad \sum_k R_{jk} \geq R^*, \quad \forall j \in \text{NBR}_i \text{ and } \forall k \in \Omega_K \\
\end{align*}
\tag{5.3}
\]

where \(\text{NBR}_i\) denotes the set of CR neighbors for user \(i\).

The game in (5.3) differs from the one in (5.1) in the additional constraint C4. When CR link \(i\) chooses its transmission power, it needs to ensure that the rate demands of all of its neighbors are also satisfied. One can prove that the K.K.T. conditions for (5.3) are the same as those of (5.2). Therefore, the NE achieved in (5.3) is at least locally optimal. If the iterative procedure converges to a unique NE, it must be the global optimum. Intuitively, the new constraints let users take into account the interference they cause to their neighbors.

However, this optimal local optimization problem (5.3) is difficult to solve for two reasons. First, due to the non-convex nature of constraint C4, the problem cannot be easily solved using convex optimization methods. Of course, complex numerical methods can be used to solve such non-convex problem (e.g., [95]), but the solution set is out of the scope of this chapter. Second and more importantly, this localized optimization problem requires each CR to obtain all the information (channel gain, interference and etc) from its neighbors. This requires excessive message exchanges and is infeasible in practice. Thus, we suggest below an incentive-based approach for each CR link, with the purpose of maintaining the simplicity and distributed
nature of the selfish iterative approach and at the same time improving the quality of the NE.

5.3.3 Simplified Incentive-based Strategy

In this section, we first provide the game formulations for the incentive-based strategy, and then compare it with the optimal update strategy. For user \( i \), this game is given as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{k \in \Omega_K} (1 + \lambda_{ik}) P_{ik} \\
\text{s.t.} & \quad C1: P_{ik} \geq 0, \quad \forall k \in \Omega_K \\
& \quad C2: \quad \sum_{k \in \Omega_K} R_{ik} \geq R^* \\
& \quad C3: \quad P_{ik} \leq \Pi_k, \quad \forall k \in \Omega_K
\end{align*}
\] (5.4)

where \( \lambda_{ik} \) is a user- and channel-dependent weight. Its value is computed at each iteration \( l \) by using the power values from the previous iteration \( l - 1 \):

\[
\lambda_{ik} = -\sum_{j \in \text{NBR}_i} \frac{\partial R_{jk}}{\partial P_{ik}}|_{P = P^{(l-1)}}. 
\] (5.5)

Intuitively, \( R_{jk} \) is a decreasing function of \( P_{ik} \) if link \( j \) is in the neighborhood of link \( i \). So a large \( \lambda_{ik} \) means that increasing of \( P_{ik} \) will have a large negative impact on the rate of other links. As a result, more weight should be put on \( P_{ik} \) when minimizing the sum power of link \( i \).

Note that the above game is an approximation of the optimal update approach. The approximation can be observed by comparing the K.K.T. conditions of the game (5.4) and the global minimization problem (5.2). More specifically, the Lagrangian function of the global optimization problem (5.2) is given by:

\[
J = \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} P_{ik} - \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} \alpha_{ik} P_{ik} - \sum_{i \in \Omega_N} \beta_i (\sum_{k \in \Omega_K} R_{ik} - R^*) + \sum_{i \in \Omega_N} \sum_{k \in \Omega_K} \zeta_{ik} (P_{ik} - \Pi_k)
\] (5.6)

where \( \alpha_{ik}, \beta_i, \) and \( \zeta_{ik} \) are non-negative Lagrangian multipliers. The derivative of \( J \)
with respect to $P_{ik}$ is:

$$
\frac{\partial J}{\partial P_{ik}} = 1 - \alpha_{ik} - \beta_i \frac{\partial R_{ik}}{\partial P_{ik}} - \sum_{j \neq i} \beta_j \frac{\partial R_{jk}}{\partial P_{ik}} + \zeta_{ik}.
$$

(5.7)

In contrast, the Lagrangian function for the game (5.4) of user $i$ is given by:

$$
J_i = \sum_{k \in \Omega_K} (1 + \lambda_{ik}) P_{ik} - \sum_{k \in \Omega_K} \alpha_{ik} P_{ik} - \beta_i (\sum_{k \in \Omega_K} R_{ik} - R^*) + \sum_{k \in \Omega_K} \zeta_{ik} (P_{ik} - \Pi_k).
$$

(5.8)

Therefore,

$$
\frac{\partial J_i}{\partial P_{ik}} = 1 + \lambda_{ik} - \alpha_{ik} - \beta_i \frac{\partial R_{ik}}{\partial P_{ik}} + \zeta_{ik}.
$$

(5.9)

The derivatives (5.7) and (5.9) are equivalent if $\lambda_{ik} = -\sum_{j \neq i} \beta_j \frac{\partial R_{jk}}{\partial P_{ik}}$. If link $j$ is not in the neighborhood of link $i$, then $R_{jk}$ is not a function of $P_{ik}$ and $\frac{\partial R_{jk}}{\partial P_{ik}} = 0$. Also, if we take $\beta_j = 1$ for all $j \in \text{NBR}_i$, we get the expression in (5.5).

We can further reduce (5.5) to:

$$
\lambda_{ik} = \frac{g_{ij,k} g_{jj,k} P_{jk}^{(l-1)}}{I_{jk}^{(l-1)} (I_{jk}^{(l-1)} + g_{jj,k} P_{jk}^{(l-1)})}
$$

(5.10)

where $I_{jk}^{(l-1)}$ represents the interference measured at the receiver of link $j$ on channel $k$ in the $(l-1)$th iteration, and $g_{ij,k}$ is the channel gain between the transmitter of link $i$ and the receiver of link $j$ over channel $k$. From (5.10), we see that in order for CR user $i$ to update its power, it needs the transmission power of its neighbors and their measured interference in the previous iteration. This information can be conveyed using control packets. Node $i$ can easily measure the channel gain between itself and its neighbors.

The existence of the NE for the above games can be proved using Nikaido-Isoda Theorem [63]. The convergence conditions of the iterative update procedures can be derived and proved using similar procedures as in [85] and [73].
5.4 Performance Evaluation

To evaluate the effectiveness of our algorithms, we conduct numerical experiments using MATLAB and also simulate a hybrid system that consists of 2 PRNs and 1 CRN. Nodes in these networks are uniformly distributed over a 100-meter-radius circle. The first PRN operates in the 900 MHz band, occupying five non-overlapping 1-MHz channels that are labeled as channels 1 to 5 in the simulation. The numbers of PRs operating on these channels are 100, 200, 300, 400, and 500, respectively. The second PRN operates in the 2.4 GHz band, also occupying five non-overlapping 1-MHz channels. These channels are numbered 6 to 10 in the simulation. The numbers of PRs over these channels are 100, 200, 300, 400, and 500, respectively.

We divide the time into slots, each of length 10 ms. At any given slot, each PR in the first and the second PRNs attempts to transmit with a probability of 0.1 and 0.4, respectively. The transmission power of each PR is 1 Watt. The signal power attenuates with distance $d$ as $d^{-4}$.

We simulate 10 CR links that are within the transmission range of each other, so a control packet sent from any CR can be heard by all CRs. Given the large number of PRs, the total PR-to-CR interference can be approximated by a lognormal distribution [69], whose mean and variance are determined by the number of PRs and their activity factors. Thus, the instantaneous interference is sampled from this lognormal distribution. We set $N_0$ to 10 dB lower than the lowest mean interference level over all frequency bands. The flow generation at each source CR follows a Poisson process with parameter $\lambda$ flows/second. Each flow has an exponentially distributed duration of mean $1/\mu$ second. The traffic rate of a source CR is defined as $\lambda/\mu$. The rate demand for the $i$th CR link is $0.5i/\ln 2$ Mbits/second. We set the CR-to-PR power mask to $\Pi_1 = \ldots = \Pi_{10} = 10$ mW.

We compare the performance of our proposed resource allocation schemes with the multi-channel RBCS scheme [42] and three exclusive-channel-occupancy resource allocation schemes (including deterministic flow-control, deterministic packet-control, and stochastic control) proposed in [79]. In contrast to our multi-
channel parallel transmission strategy, the multi-channel RBCS scheme selects the best available channel for data transmission (i.e., a node uses only one channel at a time). Although it is not originally designed for a CRN, we adapt this scheme to the CRN application by modifying the channel selection condition, as follows. If the average transmission power associated with the best available channel satisfies the CR-to-PR interference mask, then that channel will be selected; otherwise no channel will be assigned and the request will be blocked. Our performance metrics include the power efficiency of the CRN (i.e., total power consumption per delivered bit) (Figure 5.1), the connection blocking probability (Figure 5.2), and the total system throughput (Figure 5.3). All results are based on the average of 50 runs.

From Figure 5.1, we observe that the simple incentive-based update scheme consumes the least amount of power per bit. This is because the simple incentive-based approach can take advantage of collaboration between neighbors to reduce power consumption. The deterministic packet-control scheme consumes slightly more power than the stochastic control scheme, but consumes much less power (more than 50% less) than the packet-based multi-channel RBCS scheme. The selfish-update scheme also has a poor power efficiency due to the socially non-efficient NE of the non-cooperative game. This is especially true under heavy traffic load, when the conflict between neighboring CRs becomes more acute.

Figures 5.2 and 5.3 indicate that significant improvements in the blocking probability and system throughput are achieved by the proposed strategies compared with the multi-channel RBCS. Under the exclusive-channel-occupancy model, the stochastic control formulation achieves the best throughput performance. This is not surprising given that the stochastic scheme utilizes more channel information than others (it uses both the interference distribution and the instantaneous interference). In addition, the instantaneous rate constraint is relaxed to an average rate constraint in the stochastic approach. As such, power can be optimally distributed over time (or equivalently, over distribution) to minimize the average power consumption.
Figure 5.1: Power efficiency vs. traffic rate (power mask = 10 mW).

Figure 5.2: Connection blocking probability vs. traffic rate (power mask = 10 mW).
It is interesting to compare the results of the exclusive-channel-occupancy model with those of the channel-sharing model. We observe that channel sharing may not always improve the system performance. For example, the selfish-update strategy achieves low connection blocking probability and high system throughput at the cost of high power consumption when the traffic is light, but its performance deteriorates at high traffic rates. This is because the selfish behavior of CR links is more harmful to the overall system performance when more users are competing for the spectrum. The simplified incentive-based strategy successfully overcomes this selfishness and achieves much higher throughput and lower connection blocking rate, especially at high traffic rates. Among our proposed strategies, the simplified incentive-based strategy achieves the best performance at the cost of more overhead exchanges among neighbors and thus a more complicated MAC protocol design.

Finally, we study the call blocking probability and the total power consumption as functions of the power mask in Figure 5.4 and Figure 5.5, respectively. Not surprisingly, when the power mask increases, the call blocking probability decreases and the total power consumption increases. The channel sharing model is less
sensitive to the power mask than the exclusive channel occupancy model, because CR users have more freedom to access and share the spectrum. Figure 5.4 and Figure 5.5 justify the importance of selecting an appropriate power mask.

![Graph showing connection blocking probability vs. power mask](image)

Figure 5.4: Connection blocking probability vs. power mask ($\frac{\lambda}{\mu} = 0.5$).

### 5.5 Conclusions

In this chapter, we proposed a collection of power/rate control and channel assignment algorithms to improve the energy efficiency of the CRN. We showed that, in general, by adopting cooperation between nodes, the system performance can be significantly improved. However, this performance gain comes at the cost of more information exchange between nodes and a more complicated control mechanism at each node. Among the studied formulations, the simple incentive-based update strategy showed the best performance at the cost of high protocol overhead. Our future work will incorporate the power mask into the optimization framework, such that it is used not only as a shield for PRs but also as a tool to improve the CRN’s performance.
Figure 5.5: Average power vs. power mask ($\lambda/\mu = 0.5$).
CONCLUSIONS AND FUTURE RESEARCH

In this dissertation, we study the resource allocation problem within various wireless network environments. For single-channel MANETs, we proposed a game-theoretic power control MAC protocol (GMAC) for improving throughput in MANETs. GMAC uses a single channel for both data and control packets. It allows each user to determine whether or not it is feasible to transmit concurrently with previously scheduled transmissions. GMAC enables multiple transmissions to proceed concurrently by computing NE powers for all contending transmitters. Simulation results show that GMAC significantly improves the network goodput over both POWMAC and IEEE 802.11 schemes. In some scenarios, the network goodput under GMAC was 80% (40%) larger than that of the 802.11 (POWMAC) scheme. GMAC also maintains comparable energy consumption to both POWMAC and the 802.11 scheme. As a future work, we may extend the analysis and the protocol design to support variable transmission rates. We may also extend the protocol by incorporating terminal priorities.

For multi-channel exclusive-usage wireless networks, we proposed a distributed MAC protocol to exploit the dual-receive capacity of the radios to solve the multi-channel hidden-terminal problem. It also solves the transmitter deafness problem and selects an appropriate control rate. From the simulation results, we conclude that prior to the saturation points, the system throughput increases with the traffic load, and the system has low end-to-end delay and collision rates. After the saturation points, the system throughput may be decreased or stabilized, together with higher end-to-end delay and collision rates. A distributed adaptive load control mechanism was proposed for the system to operate efficiently and autonomously, where each node uses the local MAC parameters to adapt its traffic load. Simulation results show that the load control mechanism can achieve more than 90% of
the throughput achieved at the saturation points, together with low collision rate. As of the future work, we plan to incorporate the routing design into our cross-layer framework.

For throughput-oriented multi-channel wireless networks, we proposed a PIWF algorithm for spectrum sharing in cognitive radio networks. Our PIWF algorithm can be implemented distributively with CRs repeatedly negotiating their transmission powers and spectrum. Simulation results showed that the proposed algorithm greatly improves the NE compared with the one achieved using the IWF approach. Based on the order by which CR nodes make their resource allocation decisions, we studied sequential and parallel versions of the PIWF algorithm. The parallel update scheme was shown to converge faster than the sequential update scheme, especially for a large number of users. We also presented “relaxed” versions of the PIWF algorithms, which are more robust to estimation errors and channel oscillations at the cost of slower convergence. Based on the PIWF algorithms, flow-based and packet-based MAC protocols were designed. Our simulation results showed that the PIWF-MAC protocol achieves considerably higher system throughput compared with the IWF-MAC, with less energy consumption.

For energy-oriented multi-channel wireless networks, we proposed a collection of power/rate control and channel assignment algorithms to improve the energy efficiency of the CRN, with each algorithm targeting a particular system setup. We showed that, in general, by adopting cooperation between nodes, the system performance can be significantly improved. However, this performance gain comes at the cost of more information exchange between nodes and a more complicated control mechanism at each node. Among the studied formulations, the simple incentive-based update strategy showed the best performance at the cost of high protocol overhead. Our future work will incorporate the power mask into the optimization framework, such that it is used not only as a shield for PRs but also as a tool to improve the CRN’s performance.
APPENDIX A

Proof. We first solve the optimization problem without the power mask constraint C3, using the method of Lagrange multipliers. This leads to a water-filling solution of the form [94]:

\[ P_i^*(f_k) = \left[ \frac{1}{\beta + \lambda_i(f_k)} - \frac{M_i(f_k)}{h_{ii}(f_k)} \right]^+ \]  \hspace{1cm} (A.1)

If \( P_i^*(f_k) \) is the optimal power allocation over channel \( k \), then the slope of the utility function \( u_i(P_i(f_k)) \) must be positive at the point \( P_i^*(f_k) \). Otherwise, a power vector \( \mathbf{P}_i \) with a smaller \( P_i(f_k) \) could reach a higher utility \( U_i(\mathbf{P}_i) \), with all the constraints satisfied. Thus, the utility function \( u_i(P_i(f_k)) \) is monotonically increasing between 0 and \( P_i^*(f_k) \). Accordingly, if any of the \( P_i^*(f_k) \) in (A.1) violates the upper bound C3, then the corresponding bounded optimal solution must be the upper bound \( P_{\text{mask}}(f_k) \) itself (a similar approach was also adopted in [23]). After bounding \( P_i^*(f_k) \) by \( P_{\text{mask}}(f_k) \), the remaining power will be further water-filled over other channels, thus reaching the result in (4.13). \[ \square \]
REFERENCES


(JSTSP) - Special Issue on Signal Processing and Networking for Dynamic


[87] F. Wang, O. Younis, and M. Krunz. GMAC: A game-theoretic MAC protocol
on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks
(WiOpt06), April 2006.

[88] W. Wang and X. Liu. List-coloring based channel allocation for open-spectrum
pages 690–694, Fall 2005.

MAC protocol with on-demand channel assignment for multi-hop mobile ad

[90] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu. Intelligent medium access for mobile
ad hoc networks with busy tones and power control. IEEE Journal on Selected

cellular wireless systems. In Proceedings of the IEEE INFOCOM Conference,

[92] Y. Xing, C. N. Mathur, M. A. Haleem, R. Chandramouli, and K. P. Sub-
balakshmi. Dynamic spectrum access with QoS and interference temperature
2007.

[93] M. Yao, im Dong In, and L. Alex. Weighted sum rate optimization of multicell
cognitive radio networks. In Proceedings of the IEEE GLOBECOM Conference,
Nov 2008.

[94] W. Yu. Competition and cooperation in multi-user communication environ-

[95] W. Yu and R. Lui. Dual methods for nonconvex spectrum optimization of multi-
carrier systems. IEEE Transaction on Communications, 54(7):1310–1322,
July 2006.


