
Allocation of Opportunistic Spectrum for Cognitive Radio Ad hoc Networks

Thesis submitted in partial fulfillment for the degree of

Master of Science

in

**Electrical Engineering
(Telecommunications)**

By

Vijay S Rao

Student Number: 1390074



Wireless and Mobile Communications
Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft
The Netherlands

©Vijay S Rao

Wireless and Mobile Communications

Delft University of Technology

ABSTRACT

Cognitive Radios (CRs) address the problems of spectrum scarcity and under-utilization of the spectrum. However, realizing a CR network is neither easy nor straight-forward. The link layer in CR Ad hoc Networks is responsible for allocating suitable channels, out of the currently available channels. It is also responsible for setting up communication between nodes. Further, the spectrum efficiency should be maximized in a fair way.

The problem of spectrum allocation can be modeled as a graph-theoretic problem. The selection of channels amongst the CR nodes in the network is an $\mathcal{NP}$ class problem. We prove this is, in fact, an $\mathcal{NP}$ complete problem. We propose a time-slotted system. In such a system, the schedule length needs to be kept to a minimum for higher spectrum utilization. We analyze the problem to determine conditions for an optimal allocation. We use edge coloring as a tool to analyze and propose heuristics. In ad hoc networks, distributed solutions are preferred due to the lack of infrastructure. We propose two distributed algorithms: (i) clique based, and (ii) localized heuristic algorithms. We compare the results of these heuristics with the algorithm proposed in literature. We also find the worst case bounds for these algorithms.

For efficiency purposes, it is required to have a constant number of slots per frame. In such cases, producing a valid schedule is not enough since unfairness of allocation will eventually arise. To address this issue, we modify the edge coloring and clique based heuristics to produce valid fair schedules.

Finally, we briefly consider the advantages of having a joint spectrum sensing-allocation scheme at the link layer. When the spectrum sensing scheme at the PHY layer is not completely reliable, a link layer scheme can help in reducing the false alarms and miss-detections. We, further constrain the system by limiting the number of channels that can be sensed within a frame. We present the spectrum utilization with this joint sensing-allocation policy.

Allocation of Opportunistic Spectrum for Cognitive Radio Ad hoc Networks

Thesis submitted in partial fulfillment for the degree of
Master of Science
in
Electrical Engineering
(Telecommunications)

By
Vijay S Rao
Student Number: 1390074

Jan - Oct 2009

Thesis Mentor
Dr. R Venkatesha Prasad

Thesis Supervisor
Prof.dr.ir. Ignas Niemegeers



Wireless and Mobile Communications
Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS
AND COMPUTER SCIENCE

The under-signed hereby certify that the thesis entitled “**Allocation of Opportunistic Spectrum for Cognitive Radio Ad hoc Networks**” by **Vijay Sathyanarayana Rao** fulfills the requirements for the degree of **Master of Science**, and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for the acceptance of the thesis.

Dated: October 29th, 2009

Chairman

Prof.dr.ir.I G M M Niemegeers

Advisor

Dr. R Venkatesha Prasad

Committee member

Dr. Massimiliano Simeoni

श्री योग नरसिंह स्वामि
पादारविन्दयोः समर्पयामि।

*Dedicated to
Lord Yoga Narasimha*

Acknowledgements

Almost three years after I decided to pursue my studies at Delft, I shall be completing the work I set out for. Prof. Ignas Niemegeers has been supporting me all the way through. I thank him wholeheartedly for the support and guidance he has extended to me. I thank Mr. A. C. deRidder for his kind support. I also thank Dr. Gerard Janssen who helped and instilled confidence in me during the course of my studies.

In Delft, I met Chetan Yadati (PhD candidate, Dept. of Algorithms, EWI, TUDelft) in whom I found a good friend and a guide, who constantly ensured I am on the right track. He and his wife, Shilpa Yadati, together have provided me a homely environment here. I owe them more than an acknowledgement for their warmth and support.

Dr. Venkatesha Prasad has been a true mentor to me for quite sometime now. I am happy that I could work under his thoughtful guidance. Although I owe him more, here I express my overwhelming gratitude to him for everything. These words aren't perfect, yet I say "Thank you VP!"

My heartfelt and deepest thanks to my parents for everything, and everyone in my family for their warm support. I would like to thank my friends, specially Sandeep, Hari, Amar and Dony, and my seniors at TUDelft, Madan, Shreyas and Venkat.

Finally, I would like to mention the wonderful two years I had on the 19th floor - it was nice to work and interact with all the people here. Thank you!

Contents

Abstract	1
1 Introduction	3
1.1 Cognitive Radio Ad hoc Network	5
1.1.1 Application of CRANs	5
1.2 Spectrum sharing in CRANs	6
1.2.1 Differences between MANETs and CRANs	6
1.2.2 Challenges of Spectrum sharing	7
1.3 Outline and Contributions of this thesis	8
2 Opportunistic Spectrum Access - prior art	11
2.1 Classification of CR MAC	11
2.1.1 Architecture	11
2.1.2 Allocation	12
2.1.3 Access	13
2.1.4 Control Channel	14
2.1.5 Radio Front-Ends	15
2.2 Graph-theoretic techniques for spectrum allocation	15
2.2.1 MAC Scheduling	15
2.2.2 Collaboration and fairness	16
2.3 Our approach	16
3 Allocation of Opportunistic Spectrum	19
3.1 Problem description	19
3.1.1 System model and assumptions	20
3.1.2 Validation of the model	21
3.1.3 Problem statement	22
3.1.4 $\mathcal{NP}$ completeness	23

3.1.5	Integer Linear Program formulation	25
3.2	Edge coloring basics and bounds	26
3.2.1	Algorithm bounds	28
3.3	Edge Coloring based heuristics	28
3.3.1	Color Then Assign	28
3.4	Clique based heuristics	30
3.4.1	Clique Based Heuristic	31
3.4.2	Localized heuristics	35
3.5	Simulation	36
3.5.1	Simulation environment and Experiments setup	36
3.5.2	Schedule Length	39
3.5.3	Utilization	43
3.5.4	Messages exchanged	46
3.5.5	Analysis	48
3.6	Summary	50
4	Fairness in Allocation	51
4.1	Problem description	51
4.1.1	System model and assumptions	54
4.1.2	Validation of the model	55
4.1.3	Problem statement	55
4.1.4	$\mathcal{NP}$ completeness	56
4.2	Heuristics	56
4.2.1	Fair Color Then Assign	56
4.2.2	Fair Clique based heuristic	60
4.3	Results and Discussions	64
4.3.1	Simulation environment and Experiments setup	64
4.3.2	Fairness Index	64
4.3.3	Discussions	65
4.4	Summary	65
5	Spectrum Sensing	69
5.1	System model	69
5.2	Sensing	70
5.2.1	Constrained Sensing	72

5.3	Summary	73
6	Conclusions	75
6.1	Conclusions	75
6.1.1	Allocation of opportunistic spectrum	75
6.1.2	Fairness	76
6.1.3	Imperfect sensing	76
6.2	Recommendations for future work	77
	Publications	79
	Bibliography	81

Abstract

Cognitive Radios (CRs) address the problems of spectrum scarcity and under-utilization of the spectrum. However, realizing a CR network is neither easy nor straight-forward. The link layer in CR Ad hoc Networks is responsible for allocating suitable channels, out of the currently available channels. It is also responsible for setting up communication between nodes. Further, the spectrum efficiency should be maximized in a fair way.

The problem of spectrum allocation can be modeled as a graph-theoretic problem. The selection of channels amongst the CR nodes in the network is an $\mathcal{NP}$ class problem. We prove this is, in fact, an $\mathcal{NP}$ complete problem. We propose a time-slotted system. In such a system, the schedule length needs to be kept to a minimum for higher spectrum utilization. We analyze the problem to determine conditions for an optimal allocation. We use edge coloring as a tool to analyze and propose heuristics. In ad hoc networks, distributed solutions are preferred due to the lack of infrastructure. We propose two distributed algorithms: (i) clique based, and (ii) localized heuristic algorithms. We compare the results of these heuristics with the algorithm proposed in literature. We also find the worst case bounds for these algorithms.

For efficiency purposes, it is required to have a constant number of slots per frame. In such cases, producing a valid schedule is not enough since unfairness of allocation will eventually arise. To address this issue, we modify the edge coloring and clique based heuristics to produce valid fair schedules.

Finally, we briefly consider the advantages of having a joint spectrum sensing-allocation scheme at the link layer. When the spectrum sensing scheme at the PHY layer is not completely reliable, a link layer scheme can help in reducing the false alarms and miss-detections. We, further constrain the system by limiting the number of channels that can be sensed within a frame. We present the spectrum utilization with this joint sensing-allocation policy.

CHAPTER 1

Introduction

The current regulatory practices for spectrum allocation is to license frequency bands exclusively and, usually for long periods of time over a large geographical area. This practice has been followed to minimize interference and to avoid exploitation of the spectrum. However, some bands are reserved for unlicensed usage. The important one amongst them is the Industrial, Scientific and Medical (ISM) band.

Meanwhile, recent technical advances have resulted in improvements in telecommunications, particularly in wireless networking with respect to latency, bandwidth and user friendliness. Many wireless technologies (for example, GSM, GPRS, UMTS, WiMax, 802.11, 802.15, DVB-H etc) have been developed. While some of these (GSM, GPRS, UMTS etc) have been wide deployed on an exclusively bought spectrum, many others (802.11, 802.15.1, 802.15.4, etc.) are developed for use in the ISM bands due to free spectrum access. The technical advancement and the unlicensed access have boosted the unprecedented growth of technology and wide variety of applications in these bands. This has resulted in over-crowding of the ISM bands – while the licensed bands are under-utilized – creating a “scarcity” of the spectrum. Fig. 1.1 shows the typical spectrum usage for a range of frequencies - while some frequencies are heavily used others are sparsely used. According to the Federal Communications Commission (FCC), the variations in the spectrum utilization are between 15% to 85% [2]. Defense Advanced Research Projects Agency (DARPA) in its survey reports an average of 6% usage [3]. This shows the artificial scarcity - though spectrum is available it is inefficiently used. The spectrum usage however varies spatially and temporally, which makes it hard to put the unused spectrum to use anywhere and every time. Further, the spectrum use for particular applications such as military, aviation needs to be strictly interference free. Thus there is a need to design newer architectures, and device new technology for increasing the spectrum utilization. *Cognitive Radio* (CR), *Opportunistic Spectrum Access* (OSA) and the *Dynamic Spectrum Access* (DSA) have been currently proposed to solve this artificial scarcity of spectrum. Technical advancements have also produced Software Defined Radios (SDRs). An SDR is a radio that can change its transmission parameters on the fly as it is software driven [2]. Consequently, the FCC and the research community are interested in addressing the spectrum scarcity with SDRs by building CRs [34]. With SDR, the CR can observe the complex environment in which it operates in, make choices (for example, the choice of frequency to use), and communicate with other entities with efficient use of resources (bandwidth and power), while learning throughout the process to make better choices in future. CRs can detect unused spectrum bands (spectrum

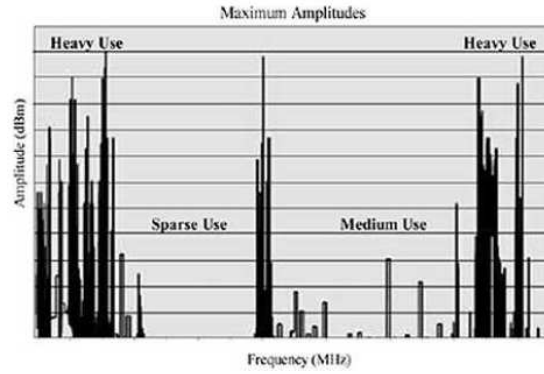


Figure 1.1: Spectrum Usage

holes), and can access these holes as shown in Fig 1.2. This *Opportunistic Spectrum Access*¹ (OSA) can address the issue of spectrum scarcity. With OSA, DARPA [3] envisages to increase the spectrum utilization by 10 times. Institute of Electrical and Electronics Engineers (IEEE) is also proposing a standard - IEEE 802.22 [13] for secondary broadband communication in the unused TV bands, while increasing the spectrum utilization. It addresses the norms the etiquettes to use the TV Bands to detect and avoid primary receivers. Further there a suite of standards dealing with CRs by IEEE SCC41.

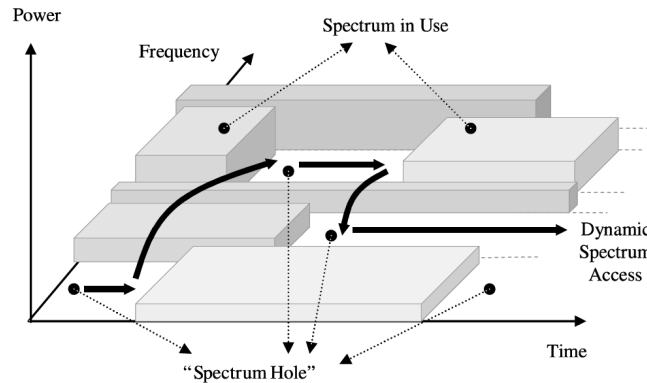


Figure 1.2: Spectrum hole concept [5]

The network architecture for CRs is presented in [5]. Fig. 1.3 shows the components of the architecture. The *Primary network* is an existing network infrastructure with a license to access certain spectrum. *Primary User* (PU) is a licensed user. Typically the primary network provides services to the PU with the help of *Primary base-stations*. The CR network (called xG network in Fig. 1.3) is unlicensed, but opportunistically accesses the spectrum band (both licensed and free bands). This CR network can be an infrastructure based network equipped with a *xG base-station* and a *Spectrum broker* to provide coordinated spectrum access, or an ad hoc network. The xG-users are called *Secondary Users* (SUs).

¹We define Dynamic Spectrum Access as accessing a spectrum band dynamically regardless of the presence of an ‘opportunity’ (spectrum hole). We define Opportunistic Spectrum Access as accessing an ‘opportunity’ dynamically.

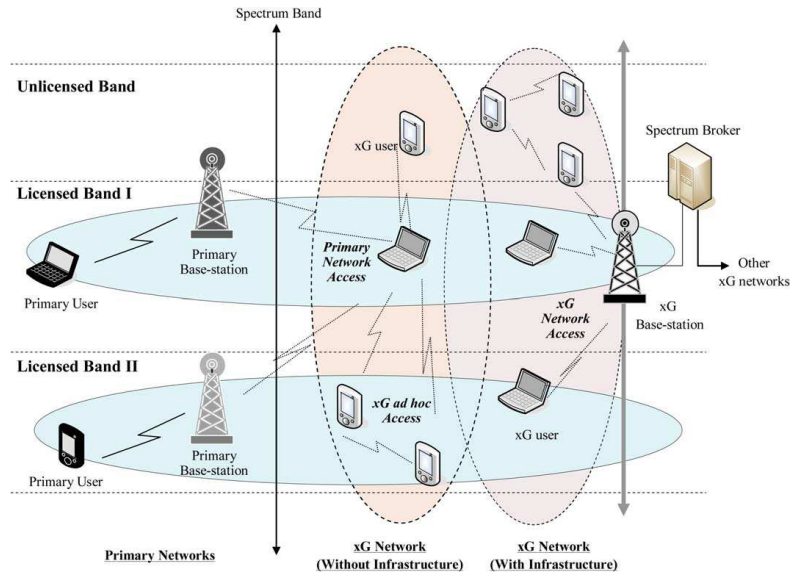


Figure 1.3: Cognitive Radio Network Architecture [5]

1.1 Cognitive Radio Ad hoc Network

An ad hoc network formed by the SUs is termed as a *Cognitive Radio Ad hoc Network* (CRAN). The nodes in a CRAN cooperate for a specific purpose or application. Few applications of CRANs are discussed below.

1.1.1 Application of CRANs

Research that helps in bringing up new technologies, which in turn provide a way for useful applications, makes greater impact on the society. One application is to use CRANs for providing textual and multimedia services, and access to IP services in a vehicular environment [37]. In this section, we look at four application areas where CRANs can be used [24].

- **Government and Regulatory interest:** The FCC and the government are interested in knowing the usability, impact and possibilities of this new technology. As stated earlier, DARPA is aiming at 10 times spectrum efficiency than what is achieved today.
- **Military communications:** With frequency agility of CRs, the ability to detect interferers and the inter-operability of radios can be used for the benefit of the military communications. It may be possible to use an unused band for secure communications, and tune to legacy systems when required.
- **Public safety:** CRs in public safety is an important application. This possibility was investigated by Adaptive Ad-Hoc Free Band Wireless Communications (AAF) project [26] in the Netherlands and FCC [14]. During a natural calamity, disaster or an accident, the licensed bands get clogged due to the heavy traffic. Due to the enhancements in public safety communications, broadband multimedia services and location-based services are needed to support the network. CR technologies

can be used to enhance public safety by utilizing the unused bands, to provide the required services, and thereby avoiding spectrum congestion.

- **Commercial use:** CRs can help alleviate the over-crowding of the ISM band, especially the 2.4GHz band. If CR becomes a commercial success, then this enhances the possibility of using off-the-shelf products for many purposes. CRs can also be used to form CR mesh networks for broadband home networking, community networking, and other applications listed in [6]. CR mesh networks can also be used as a cost-effective alternative to terrestrial connectivity in rural/sparsely populated areas.

These applications are targeted at providing better communication, especially during disasters and combats. In order to realize these applications, a coordinated access to the spectrum is required by the SUs in order to communicate and increase spectrum utilization. We look into this problem of spectrum sharing and spectrum access in the next section.

1.2 Spectrum sharing in CRANs

A seven layer framework, on the lines of the Open System Interconnection (OSI) reference model, for CRANs is proposed in [4]. It shows the functions that need to be executed at different and/or across layers to realize a CRAN. Out of the several functions, we focus on spectrum sharing issue in CRANs.

As mentioned before, there may be multiple CR users trying to access the spectrum. The access to the spectrum must be coordinated to prevent collisions on the accessed channel, and should avoid interference to the primary network. This coordination in opportunistic access and interference avoidance to primary network is termed as *Spectrum sharing*. This function necessitates a CR Medium Access Control (MAC) protocol to facilitate the sensing control to distribute the sensing task among the coordinating nodes as well as spectrum access to determine the timing for transmission.

A simplified model of the spectrum sharing function is shown in Fig. 1.4. The spectrum sharing function can be divided into two sub-functions *viz.*, Channel allocation and Spectrum access.

- *Channel allocation:* Depending on the availability of channels (the input is taken from the Spectrum sensing function) and the application requirements, a suitable channel needs to be chosen for transmission/reception from the neighboring nodes.
- *Spectrum access:* Since one channel may be accessed by several SUs, it is necessary to coordinate their access to avoid collisions. When the channel is free to access, a message is sent to the radio front-end to reconfigure the parameters - typically frequency, power and modulation - for communication.

1.2.1 Differences between MANETs and CRANs

It is important to distinguish between Mobile Ad hoc Networks (MANETs) and CRANs, since the characteristics of CRANs make it difficult to use the architecture and solutions

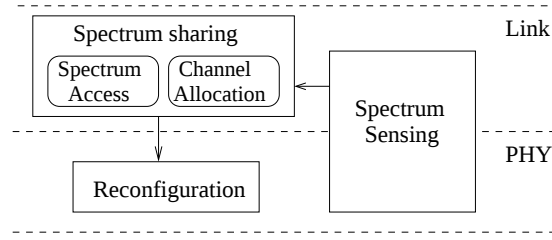


Figure 1.4: Spectrum sharing model in CRANs

of MANETs directly. Akyildiz et al. [4] point out a few differences between MANETs and CRANs. The most distinguishing aspect of CRANs from MANETs is the PU activity. Its effects can be found in all layers: channel availability, network formation and maintenance, and providing Quality of Service (QoS) to applications. Few differences with respect to spectrum sharing between MANETs and CRANs are listed here.

PU activity Each SU has a set of available channels from a wide frequency range. This set varies due to the dynamic PU activity. MANETs however, have one common channel while the multichannel ad hoc networks have all channels available at all times.

Control channel In MANETs including multichannel ad hoc networks, it is easy to exchange management frames due to the availability of a common channel. The problem in CR is that it may not have a common channel to broadcast these management frames. Instead, the CR nodes have to send beacons over all channels. This implies that network formation and maintenance process is difficult, and sometimes the topology can be incomplete.

Link reliability The stability of a link in MANET depends on the link quality and traffic load between nodes, while in CRANs PU activity also plays a major role.

1.2.2 Challenges of Spectrum sharing

Given multiple channels and CR users, the spectrum needs to be shared such that the PUs have no interference and the SU transmissions have no collisions. This is a challenging task. Further, there are policies and etiquettes being defined on portions of spectrum which the CR users should abide by. Following are a few technical challenges to be addressed to achieve coordinated sharing [5] [4].

- *Spectrum coordination and access:* A suitable channel has to be identified for communication between the transmitter and the receiver nodes. This identification process should consider the policies, the characteristics of the channels, and application requirements. Once a channel is identified, the communicating nodes have to decide (through a control channel) when to access the selected channel. This handshake is important and needs to take place before data transmission. Further, all nodes should have a fair share of spectrum depending on their requirements.
- *Spectrum changes:* Due to PU activities, a channel that was previously available may become unavailable. The communicating nodes should adapt to these dynamic

changes and should continue to operate.

- *Common Control Channel:* The above challenges may be relatively easily achieved with a common control channel; however it is difficult to identify one. Spectrum characteristics and usage vary over space and time. Hence finding one common channel available to all SUs at all times (at least most of the times) is difficult. Local common control channels may be feasible, but they are highly dependent on the topology.

The important challenge in CRANs is that all the previously listed challenges need to be realized in an ad hoc network i.e., there is no central entity/coordinator to help the nodes to synchronize, communicate and adapt. We assume that a common control channel exists for the CR nodes. This makes spectrum sharing synonymous with spectrum allocation and access, to which we attempt to analyze and provide solutions in this thesis.

1.3 Outline and Contributions of this thesis

In this thesis we address the spectrum sharing issues for CRANs. The objectives of this thesis are, to:

- analyze the problem of the spectrum coordination and access among CRAN nodes;
- propose algorithms to achieve allocation and access to the spectrum;
- propose fair share of access methods to the spectrum for all the nodes;
- propose algorithms to adapt nodes to the dynamic changes due to the PU activity;
- ensure that proposed algorithms are implementable in real-time;
- develop a simulation setup for spectrum sharing for CRANs; and
- evaluate the performances of the proposed algorithms.

The above objectives are targeted in this thesis. The following are the outline and contributions of the thesis:

Chap 2 We briefly classify the types of spectrum sharing mechanisms and provide its overview. We discuss the relevant literature with respect to this thesis.

Chap 3 We prove that the problem of spectrum allocation is an $\mathcal{NP}$ complete problem. We further find the conditions for optimal usage of resources. We use graph-theory to model the problem. We propose heuristics centralized and distributed heuristics for allocation, and find bounds for it. The use of cliques for distributed allocation has not been investigated previously. We compare and analyze the results.

Chap 4 We investigate the problem of allocation of the spectrum that is fair to all the links. This problem is $\mathcal{NP}$ complete too. This problem has received little attention before. We propose heuristics to address the fairness issue. We evaluate the algorithms.

Chap 5 We address the issue of adapting to the changes due to the PU activity. We briefly look at the joint sensing-allocation policy. We look at the spectrum efficiency of the proposed algorithm.

Chap 6 This chapter summarizes our work and provides a set of problems to address in the future.

CHAPTER 2

Opportunistic Spectrum Access - prior art

Spectrum sharing for Cognitive Radio (CR) networks has been a topic of high interest in wireless networking research. In this chapter, we present classifications of spectrum allocation and access techniques for CRANs, and provide an overview of the important milestones in the literature. We do not consider the regulations and policies since it broadens the scope of discussion of this thesis.

2.1 Classification of CR MAC

The design of MAC protocols for CRs has to consider several choices, such as, nature of channel access. These choices are usually made to suit the target application(s) or scenarios. These design choices allow the spectrum sharing techniques to be classified into many categories. The classification can be based on (a) access techniques, (b) allocation techniques, (c) availability and type of control channels, and (d) number of radio front-ends. A MAC protocol may belong to one or more of the above categories, designed to work under certain constraints for certain scenarios. Here we describe the categories, with examples from the literature.

2.1.1 Architecture

The spectrum sharing technique can either be *centralized* or *distributed*. In a centralized architecture, a centralized entity controls the spectrum allocation and access procedures. Each entity in the network forwards its state and requirements to the central entity, which decides the spectrum allocation and access schedule, and informs all participating entities about the schedule. This method is typically used for infrastructure based CR networks for example IEEE 802.22 [13]. Few other centralized solutions are proposed in [9, 8, 42]. A centralized approach for CRANs is proposed in [41], which is discussed in Sec. 2.2.2.

In a distributed architecture, the participating entities have to organize themselves and build a spectrum access map in a distributed manner. These are typically used in CRANs, where the construction of an infrastructure is not feasible. Hereafter, we restrict the discussions to CRANs. In CogMesh [11], a cluster based MAC scheme is proposed. A cluster-head is identified amongst a group of nodes, who coordinates the channel access.

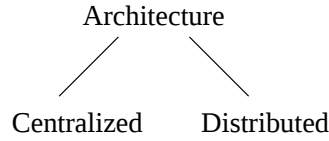


Figure 2.1: Classification based on architecture

Few more distributed approaches are proposed in [39, 35, 12, 27]. In Heterogeneous Distributed MAC (HD-MAC) [39], a clustering mechanism is proposed in which all nodes broadcast their channel set information on all the available channels to update their respective neighbors. This procedure also identifies a local common control channel for the cluster. For medium access, HD-MAC modifies the Multi-channel MAC [32] to handle the spectrum heterogeneity. The authors of Cognitive MAC (C-MAC) [12] propose a time-slotted MAC with a single transceiver to access multiple channels. One of the available channels for a set of nodes is chosen as a Rendezvous Channel that acts as a control channel, over which the coordination of medium access is handled. In Decentralized MAC (DC-MAC) [27], a node senses only one channel before a time-slot. Based on observations, a partially observable Markov decision process (POMDP) is used to model the opportunities by learning. Since this method has high computational complexity, a suboptimal strategy is used with lesser number of states. A reward is given for each successful frame transmission. The objective is to maximize the rewards. MAC Scheduling [35], another distributed scheme, is discussed in Sec. 2.2.1.

2.1.2 Allocation

The spectrum allocation technique in CRANs can either be *cooperative* or *non-cooperative* [18]. In cooperative methods, the nodes consider the presence of other nodes and their information during allocation. Each node shares its information to its one or two-hop neighbors or to a central entity. The allocation algorithm considers this information while producing an allocation map. The proposals in [41, 35, 10] use a collaborative approach to spectrum allocation. A local bargaining method for spectrum allocation is proposed in [10]. Local groups are constructed according to a poverty line that ensures a minimum allocation to nodes, and hence assuring fairness. By bargaining the spectrum utilization is increased. MAC Scheduling [35] and Zheng's collaborative methods [41] are discussed in Sec. 2.2.1 and 2.2.2.

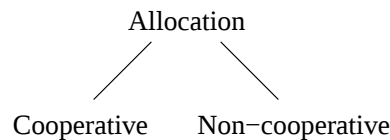


Figure 2.2: Classification based on channel allocation

The non-cooperative methods for channel allocation are proposed in [30, 27, 25]. These methods may be considered *selfish*. In non-collaborative methods, the users do not share any information. Instead the nodes access the spectrum independently according to both local observation and statistics, and use pre-determined rules, leading to minimizing communication overhead. In [25], game-theory based cooperative and non-cooperative solutions are proposed. It has been found that the non-cooperative solutions result in

lower spectrum utilization despite the gains due to low overhead [25, 41].

2.1.3 Access

Fig. 2.3(a) and 2.3(b) show two classifications based on spectrum access techniques. The idea of dynamic spectrum access is to allow secondary users to access licensed spectrum with limited interference to the primary users. This creates two possible ways [40] of accessing them - spectrum *underlay* and spectrum *overlay* (see Fig. 2.3(a)). In spectrum underlay, the SUs need to coexist with the PUs and operate below the noise floor of the PUs. In this case, the need for spectrum sensing is eliminated. One example of underlay access is to use Ultra Wide Band signals by the SUs to achieve short-range high data rate with extremely low transmission powers. In spectrum overlay technique, the white spaces in the licensed spectrum are detected and exploited. This method is termed as *Opportunistic Spectrum Access*. This method removes the restriction of low transmission powers. Other possible techniques and policies are discussed further in [40].

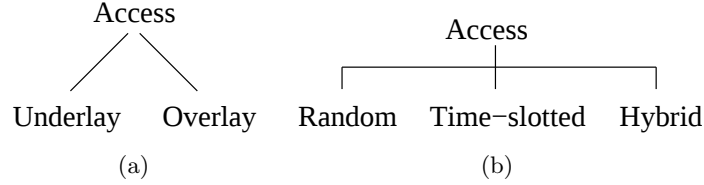


Figure 2.3: Classification based on channel access techniques

The second classification is based on the nature of channel access as shown in Fig. 2.3(b) [4]. The *random access* protocols are based on the CSMA/CA principle. These protocols do not need time-synchronization. The approaches in [23, 19] use random access method. Three radios are used in DOSS [23] for control, data and busy-tone. With dual-band signaling, this method provides a solution to the hidden and exposed terminal problems, which are prevalent in random access MAC protocols. In [19], the spectrum is divided into sensing, contention and data transmission phases. In sensing period, nodes sense for an available channel to use for data transmission. In the contention phase, the nodes reserve channel for transmission during data transmission period. In all the phases, nodes use RTS/CTS messages in the control channel.

The *time-slotted* protocols divide the time into slots for both control and data channels. Time synchronization is an important requirement here. C-MAC [12], discussed in Sec. 2.1.1, is a time-slotted MAC for CRANs. The control channel is periodically visited in order to get the neighborhood information, synchronize and maintain the network. MAC Scheduling [35], discussed in Sec. 2.2.1 also divides the time into slots and schedules the usage of the spectrum by nodes.

The *hybrid* protocols access the channels with a combination of random-access and time-slotted methods. Usually the slots are used for signaling the control information, and a random access scheme is used for data transmission. CogMesh [11], summarized in Sec. 2.1.1, uses a super frame which is divided into intervals for inter and intra-cluster operations. The super frame is divided into random access period and guaranteed access periods. In random access period, nodes contest for obtaining a slot for transmission in guaranteed access period. The guaranteed access period is divided into beacon,

neighborhood broadcast and data periods, with each period having mini-slots for transmission. DC-MAC [27], another hybrid method, was discussed in Sec. 2.1.1.

2.1.4 Control Channel

The control channel, usually referred to as *Common Control Channel* (CCC), is another parameter that plays an important role in the design of CR MAC. Unlike in IEEE 802.11 based networks, a CCC may or may not be present. If not present, it is difficult for the nodes to collaborate. The assumption of CCC in a MAC protocol is mostly motivated by the scenario or application of the network. CR MACs can be sub-divided into four types based on the type of CCC they have, as shown in Fig. 2.4. A detailed survey of CR MAC based on this classification is presented in [22].

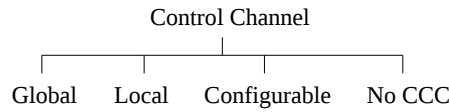


Figure 2.4: Classification based on common control channel

The CCC can be either an *in-band* or an *out-of-band* CCC. An in-band CCC is usually disrupted by PU activity, and requires setup phase every time the CCC is disrupted. In contrast, an out-of-band CCC does not suffer from these problems. Out-of-band signaling can be achieved either through a licensed channel reserved for CCC use or through an unlicensed band. The out-of-band CCC can either be *global* or *local* (includes configurable CCC), depending on the its span over the network.

- *Global control channel* Assumption of global CCC is valid since an unlicensed band such as an ISM band, can be used for control signaling by all nodes. This assumption is also not uncommon in the literature. DOSS [23], MAC Scheduling [35] and HC-MAC [19] use global CCC. The global control channel must be carefully chosen since it must not be disrupted for long periods of time. However, there are some practical issues associated with such a channel: (a) the PU activity varies from one geographical region to another, and hence, it is difficult to identify a CCC that is global, or uniformly acceptable throughout the entire network; and (b) disseminating node information for the entire network involves repeated network-wide flooding.
- *Local control channel* CR users may be grouped into clusters. This grouping of nodes may be based on their physical proximity, or the spectrum usage conditions. A local control channel is assumed to exist in such clusters and may be used for all the nodes in the same cluster. CogMesh [11] and HD-MAC [39] use local CCCs.
- *Configurable control channel* The configurable channels are similar to the local control channels except that control channel is not assumed to exist. The participating nodes configure an available channel as a temporary control channel. Also no cluster-heads or group leaders are assumed to control spectrum access in the MAC schemes implementing such a channel. C-MAC [12] uses such a control channel.

- *Without a control channel* In the worst-case, no channel can be configured as a control channel. In such a case, the nodes may use an available channel for both data and control packets. Due to lack of coordination between nodes, delay and packet collisions increase. Decentralized MAC [27] proposes MAC assuming no control channel.

2.1.5 Radio Front-Ends

The number of Radio Front-Ends (RFEs) being used by participating nodes can be sometimes more than one. Usually if multiple RFEs are used, one of them is used for control channel signaling, and others for data transmission and if required for other control signaling. For example, DOSS [23] uses 3 radios - control, busy-tone and data transmission. Two radios are used in [31]. One radio is used in [35, 12, 39, 27].

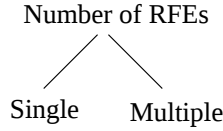


Figure 2.5: Classification based on number of radios

2.2 Graph-theoretic techniques for spectrum allocation

Graph coloring for channel allocation has been investigated extensively in cellular networks, where each base-station is to be allocated with a non-interfering channel [20] with respect to its neighboring base-stations. Graph coloring is known to be an $\mathcal{NP}$ complete problem [7]. Edge coloring has also been proposed for link scheduling in multi-hop wireless networks [33] [15]. In [33], a two hop edge coloring is proposed i.e., the adjacent edges and edges which are one hop away are assigned different colors. This coloring scheme avoids hidden terminal problem but suffers from exposed terminal problem. The distributed edge coloring heuristic proposed in [15], assigns each edge a color, where color is equivalent to a timeslot. These solutions if applied for CRANs result in degraded performance of the network, since the number of channels available at each node may vary. Before we describe our solutions in this thesis, we summarize two articles from the literature that apply graph theory for CRANs for spectrum allocation and fairness below.

2.2.1 MAC Scheduling

In [35], a directed graph is used to represent the CRAN. An Integer Linear Programming (ILP) based and a distributed heuristic based solutions are proposed. Both solutions are time-slotted and involve allocation of unique timeslot and channel to the outgoing links (edges). A valid assignment (no two nodes transmit on the same channel at the same time in interfering range) is sought. The ILP based approach has high computational complexity and does not scale well, but produces optimal allocation of the spectrum. The proposed distributed heuristic is based on edge coloring. Every node is assumed to

be aware of its two hop neighborhood. A global control channel is also assumed here. The heuristic involves two phases. In Phase-1, a $(timeslot, channel)$ pair is assigned to every link, and in Phase-2, the schedule length is propagated through the network. In Phase-1, every node is ranked based on its degree or number of channels available, or a combination of both. A node with the highest rank in its two hop neighborhood for which $(timeslot, channel)$ has not been assigned yet starts the assignment operation. Once all its edges are assigned $(timeslot, channel)$ pairs, the node is said to be covered and this assignment is distributed amongst its two hop neighborhood. Each time an arbitrary node receives a schedule it updates its knowledge of the current schedule, and it checks if it is the highest ranking uncovered node in its two-hop neighborhood to perform assignment operation. In this distributed greedy heuristic, assignment procedure is not executed in parallel at all the nodes, and each node waits for its turn until it becomes the highest ranked uncovered node to cover its edges in its two hop neighborhood. This ensures a conflict-free assignment but requires more time. In the process of conflict-free assignment, the schedule length deviates from the optimal. This delay may result in wastage of spectrum and there is a higher possibility that the set of available channels might change.

2.2.2 Collaboration and fairness

In [41], a weighted graph is used to represent the topology of the CRANs. The spectrum allocation for CRs is presented based on vertex coloring for color-sensitive graphs. A color-sensitive graph $G(U, E_C, L_B)$ is a graph with the set of vertices U , the set of colored edges E_C , and link weights L_B . The colors represent the channels. An optimization framework is proposed with the graph coloring and utility functions to optimize bandwidth and fairness. Using this model, each vertex is assigned a set of channels based on the greedy heuristic ‘Progressive Minimum Neighbor First’. Both cooperative and non-cooperative approaches are discussed. Centralized and distributed solutions are proposed that implement the utility functions. It is shown that cooperative approaches perform better than non-cooperative methods, and the distributed approaches perform almost as good as the centralized one. This work looks only at the topology-optimized allocation functions and assumes that users can utilize any number of bands at one time which seems unreasonable. It does not describe the process of transmitter-receiver handshake either.

2.3 Our approach

A spectrum sharing technique for CRANs should be able to work without a central controller, and should increase spectrum utilization while ensuring fair allocation for the links. The approach should be close to the optimal one. It should also provide a framework for the transmitter-receiver handshake, with less overhead. The approach should be implementable in real-time, unlike the ILP method. It should also try to increase with the dynamic changes in the environment due to PU activity.

MAC Scheduling [35] is the closest to our work however, MAC Scheduling looks at only producing a minimal schedule length. It does not address fairness, and does not look at the spectrum utilization at all. The schedule is not intended to fit into a frame work, and does not consider the PU activity either.

We propose a time-slotted based allocation since TDMA based methods have higher spectrum utilization than random access methods. In such time-slotted methods, if both sender and receiver are made aware of the schedule, the transmitter-receiver handshake is taken care of i.e., allocation and distribution of the schedule ensures the spectrum coordination and access. In the next chapter, we analyze the problem of spectrum allocation and access in a time-slotted frame. We explore novel graph theoretic approaches to achieve these objectives, and propose both centralized and distributed heuristics.

CHAPTER 3

Allocation of Opportunistic Spectrum

Graph theory has been widely used for modeling wireless ad hoc networks. It can be applied in CRANs too as part of solution to assigning problems. A few graph theory based solutions for spectrum allocation were discussed in Chapter 2. We too model the CRANs as a graph and thus we explore some of the possibilities to apply graph theoretic tools to solve the issues in CRNs. Since we opportunistically access the spectrum, it is necessary to have an optimal link (or MAC) layer schedule. In this chapter first we prove that the optimal allocation of opportunistic spectrum is an $\mathcal{NP}$ complete problem even for a simplified model of the problem. Later we propose heuristics for this problem. We provide three heuristics algorithms. They are (i) Edge coloring based (ii) Clique based and (iii) Localized heuristics.

3.1 Problem description

The spectrum is a shared resource amongst the nodes that are in each others' interfering range. Unlike the MAC protocols of ad hoc networks, the MAC for CRANs should handle multiple channels at the same time. Unlike the multi-channel ad hoc networks, the CRANs should be able to adapt to the dynamic PU activities. In this work we always assume that the SU nodes have only one radio front-end. Each SU node having data to communicate should get access to at least one channel for transmission. Two important conditions with which we start are:

1. *Allocation:* Every (transmitter, receiver) pair must get access to a channel.
2. *Access:* The receiver must know the channel and time at which the transmitter will transmit.

A TDMA based scheme will completely satisfy the above requirements. Moreover a TDMA allocation scheme can eliminate hidden and exposed terminal problems. Hence, we choose and propose TDMA based allocation algorithms for CRANs. However, the aim must be to use all the available spectrum, i.e., in turn all the available channels at that instant of time due to inactivity of PUs. We assume that perfect sensing is possible and the number of channels seen by all the nodes are the same to start with. We relax this condition later.

We model the CRAN as an undirected graph as shown in Fig. 3.1, with nodes denoting the vertices and all the available channels between two nodes denoting the edges between them. The allocation problem is now to compute a *schedule* i.e., find a $(timeslot, channel)$ assignment for all the edges. The schedule is a TDMA scheme i.e., the channels are divided into slots over time, and every link is assigned with a $(timeslot, channel)$ pair. To be precise, the allocation problem is to find a *valid* schedule and a schedule is termed *valid* if it satisfies the following conditions:

1. *Condition 1*: if every link is assigned with a $(timeslot, channel)$ tuple; and
2. *Condition 2*: the $(timeslot, channel)$ tuple assigned for a link L should not be assigned to any other link within the interference ranges of the end-vertices of L i.e., no hidden terminals.

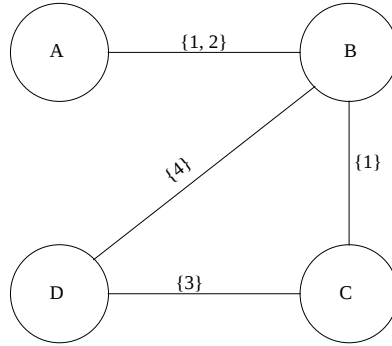


Figure 3.1: A CRAN. The available channels between nodes are indicated within brackets.

As an example, a valid schedule is shown in Fig. 3.2 for the CRAN in Fig. 3.1.

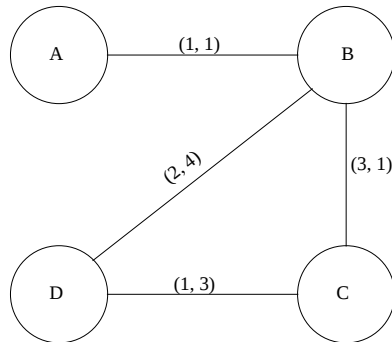


Figure 3.2: A valid assignment of $(timeslot, channel)$ tuple to links.

We now put forth the system model and assumptions, and argue their validity. With this model, we then present the problem formally.

3.1.1 System model and assumptions

We consider a multi-hop CR network formed by N nodes, each with a unique node-id $\in \{1, 2, \dots, N\}$. These nodes are equipped with a half-duplex software-defined transceiver

i.e., each node can either transmit or receive at any instant of time on a specified channel. These nodes are interested in point-to-point communication, i.e., we consider only 1-hop communication. We assume that the spectrum is divided into M orthogonal channels that are symmetric. Each channel has a unique number in $\{1, 2, \dots, M\}$. Further, we assume that a common control channel (CCC) is available to exchange the control messages amongst the nodes. The nodes are time synchronized either through the common control channel or with the help of a GPS receiver. We assume that all nodes have data to transmit to all its neighbors to ensure that we consider the worst case scenario of the problem.

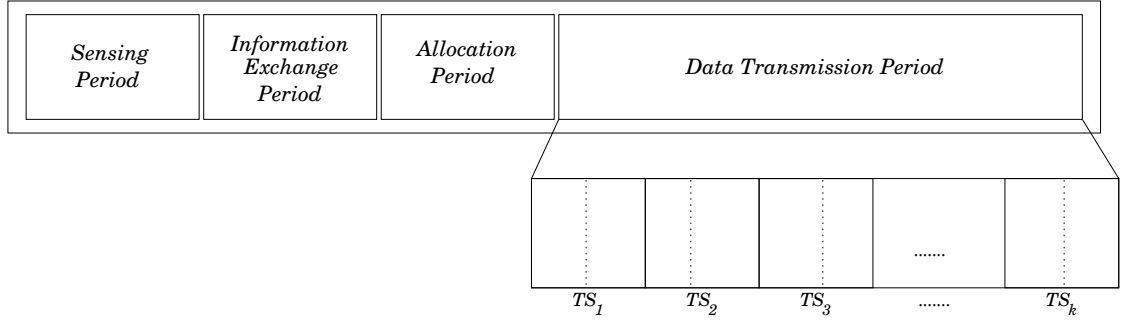


Figure 3.3: MAC super frame

Consider the MAC super frame as shown in Fig. 3.3. The super frame is divided into sensing, information exchange, allocation and data transmission periods. In the sensing period, the channels that are free of PUs are found first. We assume perfect sensing by nodes for the sake of simplicity i.e., each node can accurately determine the spectrum holes. The nodes then send the list of channels available to a central node or to their neighbors during the information exchange period. In the spectrum allocation period, an algorithm is executed to determine the schedule and the schedule is distributed.

The data transmission period is split into $\mathcal{K}$ equal length time-slots. The value of $\mathcal{K}$ is determined from the topology of the CRAN under consideration. In a time-slot TS_k both the participating nodes transmit and receive data i.e., the time-slot is considered to be consisting of two sub-slots where one node transmits and the second receives in the first sub-slot, and in the second sub-slot, the receiver in the first sub slot transmits its data to its counterpart. The sub-slots may be of unequal lengths depending on the data that each node has to transmit to the other. This assumption reduces the complexity in the design, since each link has to be assigned with one time-slot. We also note here that we have assumed in the beginning that all the nodes have data to be transmitted to all other nodes.

3.1.2 Validation of the model

While most of the assumptions in the system model in the previous section are common and reasonable, some of them need further discussion and validation. This section validates the model.

1. *Neglecting channel switch times:* In the model, the time of switching between channels has been neglected. The underlying MAC protocol should consider these

times. Moreover the protocol must also take into account of the “deaf period” during switching. However, since we are only proposing solutions for the allocation and not designing the protocol, it can be safely neglected.

2. *CCC*: We assume a CCC to simplify the implementation of the OSA network and allocation algorithms. This is a realistic assumption since an unlicensed band can be used for control information exchange. We assume a global CCC for edge-coloring heuristic and local or configurable CCCs for other heuristics. The discussion in Sec. 2.1.4 provides the arguments for validity of these assumptions.
3. *Time synchronization*: Since we propose a slotted-based scheme to access the spectrum, the nodes should have strict time synchronization. MAC Scheduling [35] uses GPS receivers for time synchronization. Another method that could be implemented using CCC with a time synchronization scheme is given in [29].

An advantage of such slotted scheme is the ability to address the issue of mobility since time synchronization and mobility are interlinked problems. It is possible to address issues of both node mobility and spectrum changes when nodes are time synchronized, since a coordinated spectrum sensing by SUs can detect these changes.

4. *Perfect sensing*: In the current state of the art, nodes cannot independently and accurately determine the spectrum holes. However with cooperation with other nodes, it is possible to achieve higher accuracy. Another aspect is the intrusion of a PU during the data transmission period affecting the schedule, which cannot be controlled. However in this chapter, to simplify the problem we assume (a) accurate sensing (with or without cooperative sensing) and (b) PU activities occur only at period T .

In each slot TS_k , the communicating nodes negotiate on the sub-slot size and the order of access. The SUs can employ fragmentation and reassembly of the data to fit the size of available sub-slot.

3.1.3 Problem statement

Let the CRAN be represented by an undirected graph $G = (V, E)$, with no self-loops, where V is the set of vertices or nodes, and E is the set of links between vertices in V . A link exists between two vertices u and v if both u and v are in communicating radii of each other and at least one common channel is available to them. The spectrum allocation problem is to find an optimal valid assignment for every link.

$$A_{(u,v),j,k} = \begin{cases} 1 & \text{if edge } (u,v) \text{ uses channel } j \text{ in time-slot } k \\ 0 & \text{otherwise} \end{cases}$$

such that,

- i. u and v are in communication radii of each other
- ii. both u and v are not involved in any other communication in TS_k
- iii. assignment of channel j does not cause interference to neighboring nodes i.e., using j between u and v does not make u and/or v a hidden terminal to other nodes.

Let $\mathbf{A}$ represent an adjacency matrix of the allocation such that,

$$\mathbf{A}_{(u,v)} = \begin{cases} (k, j) & \text{if } A_{(u,v),j,k} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Let Λ denotes the set of all possible valid assignments for $\mathbf{A}$.

The length of an allocation $\mathbf{A}$, $Len(\mathbf{A})$, is equal to the number of slots in the allocation. The allocation $\mathbf{A}_{opt}$ is optimal if the $Len(\mathbf{A}_{opt}) = \min\{Len(a_i) | \forall a_i \in \Lambda\}$.

The important symbols used are summarized in Table 3.1. It is necessary to find a minimum length MAC schedule for better channel reuse and in turn higher network throughput [28]. In the next section, we prove the $\mathcal{NP}$ completeness of this problem.

G	Graph of CRAN
N	Number of nodes in G
u or v	Represents a node in G
(u, v)	Represents the edge between u & v
M	Maximum number of channels available for CRANs
TS_k	Denotes k^{th} timeslot in data transmission period
Λ	Set of all valid assignments A
$\mathbf{A}_{opt}$	Optimal allocation in Λ
$ChSet_{(u,v)}$	The set of channels available for edge (u, v)
$NodeID_u$	A unique ID of node u
δ_u	Degree of u
$Neighbors_u$	Set of neighbors of u

Table 3.1: Definitions

3.1.4 $\mathcal{NP}$ completeness

Given N nodes and M channels, finding a valid schedule is a problem in the $\mathcal{NP}$ class. We prove this problem is in $\mathcal{NP}$ complete class. To prove the $\mathcal{NP}$ completeness of a problem, it is sufficient to show a polynomial reduction of the problem to one of the well-known $\mathcal{NP}$ complete problems [16]. One of the well-known $\mathcal{NP}$ complete problems is 3SAT. 3SAT is defined as follows. A set of clauses $C = C_1, C_2, C_3, \dots, C_r$ in variables $\alpha_1, \alpha_2, \dots, \alpha_s$ is given, each clause C_i consisting of three literals $l_{i,1}, l_{i,2}, l_{i,3}$, where a literal $l_{i,j}$ is either a variable α_k or its negation $\bar{\alpha}_k$. The problem is to determine whether C is satisfiable, that is, whether there is a truth assignment to the variables which simultaneously satisfies all the clauses in C . A clause is satisfied if one or more of its literals has value “true”.

Theorem 1. *Given a graph $G = (V, E)$ representing a CRAN, a maximum of M channels and a positive integer p , finding a valid schedule within p time-slots is $\mathcal{NP}$ complete.*

Proof. We show a polynomial reduction of the scheduling problem to the 3SAT problem to prove the former is in $\mathcal{NP}$ complete class.

Consider a $G = (V, E)$ representing a CRAN. We have defined $A_{(u,v),j,k}$ as,

$$A_{(u,v),j,k} = \begin{cases} 1 & \text{if edge } (u, v) \text{ uses channel } j \text{ in time-slot } k \\ 0 & \text{otherwise} \end{cases}$$

A valid schedule of length p should satisfy the following conditions:

$$\sum_{\forall (v) \in \text{Neighbors}_u} \sum_{j \in \text{ChSet}_{(u,v)}} A_{(u,v),j,p} \leq 1 \quad \forall p \quad \forall u \in V$$

This condition restricts two adjacent edges from being assigned the same slot. It may however be not assigned with any slot, hence the RHS can be either 0 or 1.

$$\sum_p \sum_{j \in \text{ChSet}_{(u,v)}} A_{(u,v),j,p} = 1 \quad \forall (u, v) \in E$$

This condition ensures that each edge is assigned a slot (and exactly once).

$$\begin{aligned} A_{(u,v),j,p} + \sum_e A_{(e),j,p} &\leq 1 \quad \forall (u, v) \text{ and } (e) \in E, \\ e &\in \{(x, y) | x \in \{\text{Neighbors}_u, \text{Neighbors}_v\}, \\ &y \in \text{Neighbors}_x \text{ and } y \notin \{u, v\}\} \\ &\text{and } \forall p, j \end{aligned}$$

This condition ensures that only one communication link is active in the neighborhood on a particular (*timeslot, channel*). Since G is an undirected graph, only one of (x, y) or (y, x) is considered.

We can map $\{A_{(u,v),j,p} | \forall u, v, j, p\}$ as the set of variables $\{\alpha_1, \alpha_2, \dots, \alpha_s\}$. Each of the constraints can be represented as 3 literal clauses to form the set C . The complexity of these operations is $\mathcal{O}(\text{forming boolean equations of constraints})$. Clearly we have reduced our problem in polynomial time to 3SAT problem. Hence to obtain a valid schedule, we need solve the 3SAT problem.

The second part of the proof is to show that a solution to 3SAT problem instance created above is also the solution to our problem. The solution to 3SAT consists of a truth value assignment to the variables $\{\alpha_1, \alpha_2, \dots, \alpha_s\}$, each element of which points to a $A_{(u,v),j,k}$ i.e., a truth assignment to this variable indicates that edge (u, v) can use channel j in slot k . Since all the clauses represents the conditions stated above, and the 3SAT solution obtained satisfies all the conditions, it implies the solution also represents a solution to our problem. $\square$

It should be noted that the finding a valid schedule is $\mathcal{NP}$ complete. To find a minimum length schedule from a set of valid schedules of different lengths can be accomplished in polynomial time.

3.1.5 Integer Linear Program formulation

To get an optimal schedule length, an Integer Linear Program (ILP) solution of the allocation problem is required. The ILP formulation for this problem is presented in [35], which is stated in this section. We have defined $A_{(u,v),j,k}$ as,

$$A_{(u,v),j,k} = \begin{cases} 1 & \text{if edge } (u, v) \text{ uses channel } j \text{ in time-slot } k \\ 0 & \text{otherwise} \end{cases}$$

Let Y_k be a boolean variable:

$$Y_k = \begin{cases} 1 & \text{If any transmission occurs during timeslot } k \\ 0 & \text{Otherwise} \end{cases}$$

The optimal MAC schedule can be computed by solving the following ILP:

$$\min \sum_{\forall k} k Y_k$$

Subject to

$$\sum_{\forall (v) \in \text{Neighbors}_u} \sum_{j \in \text{ChSet}_{(u,v)}} A_{(u,v),j,k} \leq 1 \quad \forall k \quad \forall u \in V \quad (3.1)$$

$$\sum_k \sum_{j \in \text{ChSet}_{(u,v)}} A_{(u,v),j,k} = 1 \quad \forall (u, v) \in E \quad (3.2)$$

$$\begin{aligned} A_{(u,v),j,p} + \sum_e A_{(e),j,p} &\leq 1 \quad \forall (u, v) \text{ and } (e) \in E, \\ e &\in \{(x, y) | x \in \{\text{Neighbors}_u, \text{Neighbors}_v\}, \\ &\quad y \in \text{Neighbors}_x \text{ and } y \notin \{u, v\}\} \\ &\forall p, j \end{aligned} \quad (3.3)$$

$$Y_k - \sum_{(u,v)} \sum_{j \in \text{ChSet}_{(u,v)}} A_{(u,v),j,k} + A_{(v,w),j,k} \leq 0 \quad \forall k \quad \forall (u, v) \in E \quad (3.4)$$

$$\sum_{(u,v)} \sum_{j \in \text{ChSet}_{(u,v)}} A_{(u,v),j,k} - |E| Y_k \leq 0 \quad \forall k \quad \forall (u, v) \in E \quad (3.5)$$

As stated before, Constraint 3.1 restricts two adjacent edges from being assigned the same slot. Constraint 3.2 ensures that each edge is assigned a slot (and exactly once) since we are interested in minimum schedule length. The third constraint ensures that only one communication link is active in the neighborhood on a particular (*timeslot, channel*). Constraints 3.4 and 3.5 ensure that Y_k is set to 1 if there is any transmission during a time-slot k , otherwise it is set to 0.

3.2 Edge coloring basics and bounds

An *edge coloring* of a graph G is an assignment of colors to the edges of G , one color to each edge. A *k-edge-coloring* of a graph $G = (V, E)$ is a mapping $c : E \rightarrow S$, where S is a set of k colors, in other words, an assignment of k colors to the edges of G . Usually, the set of colors S is taken to be $\{1, 2, \dots, k\}$. A *k-edge-coloring* can then be thought of as a partition $U = \{E_1, E_2, \dots, E_k\}$ of E , where E_i denotes the set of edges assigned color i . An edge coloring is *proper* if adjacent edges receive distinct colors. As we are concerned here only with proper edge colorings, we refer to a proper edge coloring simply as an ‘edge coloring’ now onwards.

It can be seen that if G represents a CRAN and S represents the set of timeslots $\{TS_1, TS_2, \dots, TS_k\}$, then the edge coloring of G results in a valid communication schedule when it satisfies the following condition:

Condition 3: $\forall E_m, E_n \in U$ such that any pair of the end-vertices of E_m and E_n is in interfering range, then E_m and E_n should not be assigned the same channel j .

We use colors and (time) slots interchangeably. To illustrate, we use Fig. 3.1 again. The edge colored graph is shown in Fig. 3.4. The different colors are represented by different line styles. As can be seen, the adjacent edges have different colors. Note that edges (A, B) and (C, D) have the same color. Hence, to make a valid schedule of this assignment, it is required to satisfy *Condition 3* i.e., (A, B) and (C, D) should not be assigned the same channel. One possible solution for this instance is shown in Fig. 3.2 wherein (A, B) is assigned Channel 1 and (C, D) is assigned Channel 3. Note such a communication does not cause hidden or exposed terminals.

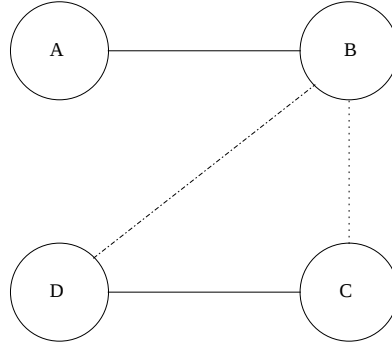


Figure 3.4: Edge coloring of Fig. 3.1 (Different line styles denote different colors)

Since we are interested in optimal length schedule, we focus our attention on *chromatic index* of G . The *edge chromatic index*, $\chi'(G)$ of graph G is the minimum positive integer k for which G is k -edge-colorable. Let $\Delta(G)$ denote the maximum degree of all vertices of G . It can immediately be seen that $\chi'(G) \geq \Delta(G)$. From Vizing’s theorem [7], it is proven that $\chi'(G) \leq \Delta(G) + 1$. We now investigate if we can set bounds on edge coloring based allocation.

We term the *(timeslot, channel)* pair as a resource. An allocation is said to be *time optimal* if it takes the least number of time-slots. We use complete graphs in the theorems since they depict the worst case scenarios.

Theorem 2. For a given graph $G = (V, E)$ with edge chromatic index $\chi'(G)$ and a set of (timeslot, channel) resources, a time optimal allocation covering all edges can be obtained if the available number of channels $N_{\text{channels}} \geq \lceil \Delta(G)/2 \rceil$ on all edges.

Proof. The total resources required for a graph G is $TS_{\text{total}} \times N_{\text{channels}}$ where TS_{total} denotes the total number of slots and N_{channels} denotes the total number of channels.

From Vizing's theorem we know that minimum number of colors or slots required to cover all the edges is $\chi'(G) \leq \Delta(G) + 1$ i.e., $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$. Therefore, $TS_{\text{total}} = \chi'(G)$.

Only complete graphs, K_n , have the maximum number of edges, which is equal to $\frac{\Delta(G)(\Delta(G)+1)}{2}$.

Case 1: When $\chi'(G) = \Delta(G)$

$$TS_{\text{total}} \times N_{\text{channels}} = \frac{\Delta(G)(\Delta(G)+1)}{2}$$

$$\text{Therefore, } N_{\text{channels}} = \frac{(\Delta(G)+1)}{2}$$

Case 2: When $\chi'(G) = \Delta(G) + 1$

$$TS_{\text{total}} \times N_{\text{channels}} = \frac{\Delta(G)(\Delta(G)+1)}{2}$$

$$\text{Therefore, } N_{\text{channels}} = \frac{\Delta(G)}{2}$$

Hence $N_{\text{channels}} = \lceil \Delta(G)/2 \rceil$. □

The minimum number of slots required to cover all edges cannot be less than $\chi'(G)$. If N_{channels} is lower than $\lceil \Delta(G)/2 \rceil$ (but at least equal to 1), it is possible to get a valid spectrum allocation by increasing the number of slots. However, the converse is not true which is stated in the following corollary.

Corollary 1. Increasing the number of channels, N_{channels} , more than $\lceil \Delta(G)/2 \rceil$ will not reduce the total number of slots required TS_{total} .

Proof. Let $N_{\text{channels}} = \lceil \Delta(G)/2 \rceil$. With this N_{channels} the maximum cardinality of set U for a complete graph $G = K_n$ is $|U_{\text{max}}| = \lceil \Delta(G)/2 \rceil$. All N_{channels} are used in U_{max} . If N_{channels} is increased by one more channel, there is no edge in U_{max} that can be assigned to this channel in the current slot. That implies, no new edge is covered with introduction of a new channel. □

Corollary 2. Optimal use of resources can only be done in a complete graph $G = K_n$ with $TS_{\text{total}} = \chi'(G)$ and $N_{\text{channels}} = \lceil \Delta(G)/2 \rceil$.

Proof. We know $|U_{\text{max}}| = \lceil \Delta(G)/2 \rceil$. For a complete graph K_n , all sets of U 's have the same cardinality, which is equal to $|U_{\text{max}}|$. That implies all N_{channels} channels are used in every slot. So we have proven that optimal use of resources is done in K_n . We now prove that such usage happens only in K_n . Let one edge be removed from the complete graph, then one set of U 's will have a cardinality of $\lceil \Delta(G)/2 \rceil - 1$. Therefore, one channel is not used in this U . Hence the proof. □

The above results apply completely to multi-channel networks. However since the number of channels may not be the same for all nodes in CRANs, these results can be applied as a special case when all edges have the same channels.

3.2.1 Algorithm bounds

From the above theorem and corollaries, we can set algorithmic performance bounds i.e., using edge-coloring greedy heuristic, how worse can the schedule length be given a graph $G = (V, E)$ of CRAN. The worst case scenario, again is a complete graph K_n with N_{channels} . It is also known that the best bound for edge coloring with greedy algorithm is $2(\Delta(G) + 1)$ [7]. From the theorem and corollaries above, we can state a loose bound:

$$2(\Delta(G) + 1) + (\lceil \Delta(G)/2 \rceil - N_{\text{channels}})(2(\Delta(G) + 1)) \quad (3.6)$$

This first term of the equation i.e., $2(\Delta(G) + 1)$, is maximum number of slots resulting from a greedy edge coloring algorithm. This number itself is insufficient to represent the bound since with one available channel, it cannot be reused without causing hidden terminals. Hence another term is added to the equation to calculate the additional slots required i.e., add the total missing resources. This provides a loose bound on the number of slots given the parameters.

Sec. 3.5.2 further discusses tighter bounds and their usage.

3.3 Edge Coloring based heuristics

In Sec. 3.1.4 we proved that the problem of spectrum allocation is $\mathcal{NP}$ complete. The best way to solve this type of problems is to use heuristics. In the edge coloring based algorithms, a central node learns the whole topology of the CRAN, performs the assignment and distributes the schedule to all the nodes. The central node can be a cluster-head [11]. In this network, the cluster-heads allocate, synchronize and maintain the network topology. Though this assumption puts extra protocol overhead, this is usually a simple method in most of the wireless networks; for example, PNC in 802.15.3c WPANs.

3.3.1 Color Then Assign

In this Color Then Assign (CTA) algorithm, we use the time-slots as colors as mentioned in Sec. 3.2. Here, we first assign the slots and then assign the channels to the slots.

Intuition

The first step of the algorithm is to assign slots to the edges by coloring. We use a greedy strategy for coloring wherein we start coloring the edges of nodes with maximum degree. The colors are assigned in a greedy method i.e., they are reused as much as possible. Each edge is colored only once. Once colors are assigned to all the edges, the next highest degree vertex is chosen and its edges are colored. This procedure repeats until all edges are colored.

The second step is to assign an available channel to each edge such that *Condition 3* is satisfied. If an edge cannot be assigned a channel, a new slot is added and then a channel is assigned. This algorithm can be better understood with an example. We again consider Fig. 3.1 for illustration, which is shown in Fig. 3.5(a).

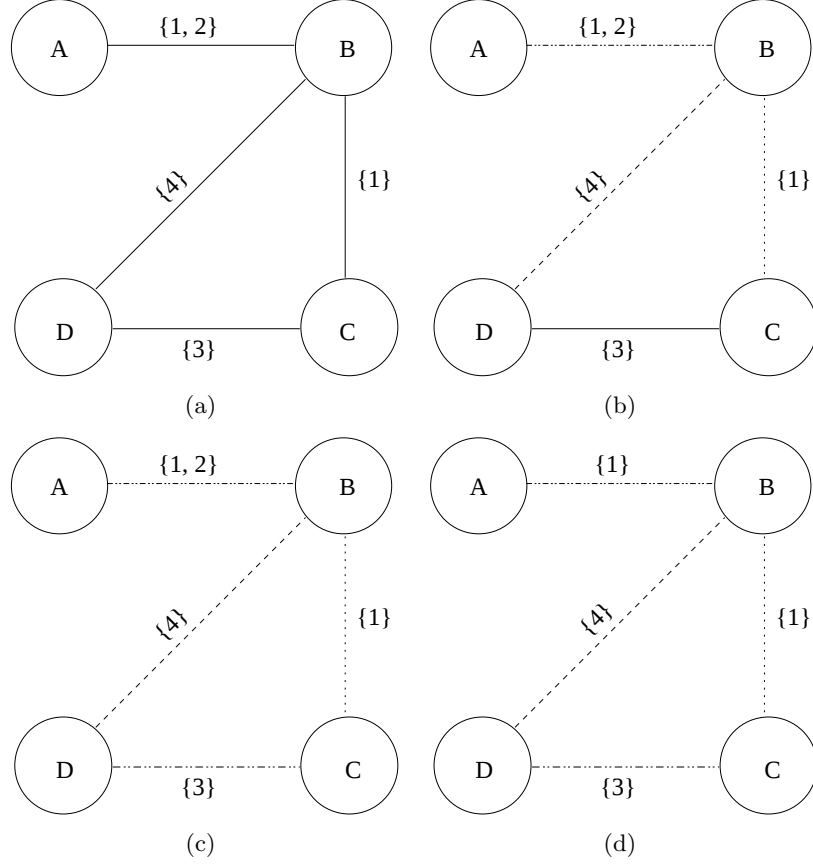


Figure 3.5: Illustration of the CTA algorithm

In the first step, the node with highest degree is chosen, namely B, and then its edges are colored (See Fig. 3.5(b)). Then the nodes with next highest degree are C and D. Since $NodeID_C$ is higher than $NodeID_D$ (lexically sorted), C is selected next. C's edges are colored now (See Fig. 3.5(c)). Since we propose a greedy method, we try to reuse colors as much as possible. After C finishes coloring, no edges are left to be colored, and there ends the first step.

In the second step, every edge is assigned a channel from its available set of channels. Since (A, B) and (C, D) have the same color, it is taken care that both of them are assigned different channels as shown in Fig. 3.5(d). In case of a conflict, (C, D) would have been assigned another color. The algorithm is formally presented in the next section.

Algorithm

The algorithm is presented in Alg. 3.1. The input to this algorithm is the graph G of CRAN. The algorithm calls two procedures viz., Slot assignment and Channel assignment, in the same order.

Algorithm 3.1 Color Then Assign (CTA) algorithm

- 1: **Input:** Connectivity graph $G = (V, E)$ and the set of available channels $ChSet_{(u,v)}$ between each pair of vertices u, v
 - 2: **Output:** *Allocation* - A valid allocation of $(timeslot, channel)$ for all edges
 - 3: Run Procedure 3.2
 - 4: Run Procedure 3.3
-

The slot assignment procedure, shown in Proc. 3.2, is a greedy algorithm to color the edges of G . It makes a copy of G and works on the copy. It first finds a maximum degree vertex u , and then for each of its edges (Line 4) it assigns a color that has not been assigned to any of the edges of u and the other end-vertex of the edge. This ensures a proper edge coloring, while trying to reduce the number of colors. Once a color is assigned, the edge is removed from the graph so that the edge is not colored again. Once all the edges of u are colored, the next vertex is chosen and the procedure repeats until all the edges are colored.

Procedure 3.2 Slot Assignment

- 1: **while** at least one node u is uncolored **do**
 - 2: Find a node u with maximum δ_u in G
 - 3: **for** v in $Neighbors_u$ **do**
 - 4: a. $Allocation_{(u,v)}.slot \leftarrow$ first unused slot to edge (u, v) such that the slot is unused in u and v
 - 5: b. Remove the edge (u, v) in G
 - 6: **end for**
 - 7: Mark u as colored
 - 8: **end while**
-

The channel assignment procedure, shown in Proc. 3.3, is a greedy algorithm to assign channels to the colored edges. We first find out the maximum slot (or color) number, $maxSlot$ assigned by Proc. 3.2 for G . The algorithm runs for all set of edges E_k in U where $1 \leq k \leq maxSlot$. For each E_k , each edge $\in E_k$ is tried to be assigned with one of its available channels such that *Condition 3* is satisfied (Lines 5 and 6). If such a channel is available, it is assigned and the edge is marked ‘assigned’. If some edge could not be assigned a channel, then it is assigned a new slot. This ensures that the edge will definitely get a slot for transmission.

Complexity

The worst-case time complexity for the slot assignment procedure is $\mathcal{O}(N^2)$ and for the channel assignment procedure is $\mathcal{O}(MNmaxSlot)$.

3.4 Clique based heuristics

A *clique* in an undirected graph $G = (V, E)$ is a subset of the vertices, $V' \subseteq V$, such that for every two vertices in V' , there exists an edge connecting the two. The complement of a clique set is an *independent set*, in the sense that every clique corresponds to an independent set in the complement graph.

Procedure 3.3 Channel Assignment

```

1:  $maxSlot \leftarrow \text{maximum of } Allocation_{(u,v)}.slot \forall u, v$ 
2: while at least one edge  $(u, v)$  is unassigned do
3:    $k \leftarrow 1$ 
4:   for  $(u, v)$  in  $G$  such that  $Allocation_{(u,v)}.slot = k$  do
5:     for  $j$  in  $ChSet_{(u,v)}$  do
6:       if  $Allocation_{(u,v)}.channel \leftarrow j$  satisfies Condition 3 then
7:         a.  $Allocation_{(u,v)}.channel \leftarrow j$ 
8:         b. Mark  $(u, v)$  as assigned
9:       end if
10:    end for
11:    if  $(u, v)$  is unassigned then
12:      a.  $maxSlot \leftarrow maxSlot + 1$ 
13:      b.  $Allocation_{(u,v)}.slot \leftarrow maxSlot$ 
14:    end if
15:  end for
16: end while

```

The clique based algorithms relax few assumptions considered previously. There are no cluster-heads and each node is responsible for allocating a $(timeslot, channel)$ between its neighbors. A local common control channel is enough for nodes to learn about their neighbors and also employ a time synchronization scheme, for example as proposed in [29]. We propose two algorithms (the second one is based on the first) for spectrum allocation based on clique formation. In these algorithms, in case of a conflict, the node-ids will be used to resolve them.

3.4.1 Clique Based Heuristic

Intuition and Algorithm

In this section, we describe the Clique Based Heuristic (CBH) algorithm and its working to get a better understanding of the same. In CBH, two hop topology information is exchanged between the nodes. Since we are attempting a distributed algorithm, we should take care that no edge is assigned more than one $(timeslot, channel)$, which creates conflicts. Each edge should be a ‘responsibility’ for one node. To determine the responsibilities, each node executes the Procedure 3.4. We introduce another symbol I , which denotes an Independent Set of G . At the end of the procedure, we have a set of nodes that form a maximal independent set (I) of the graph.

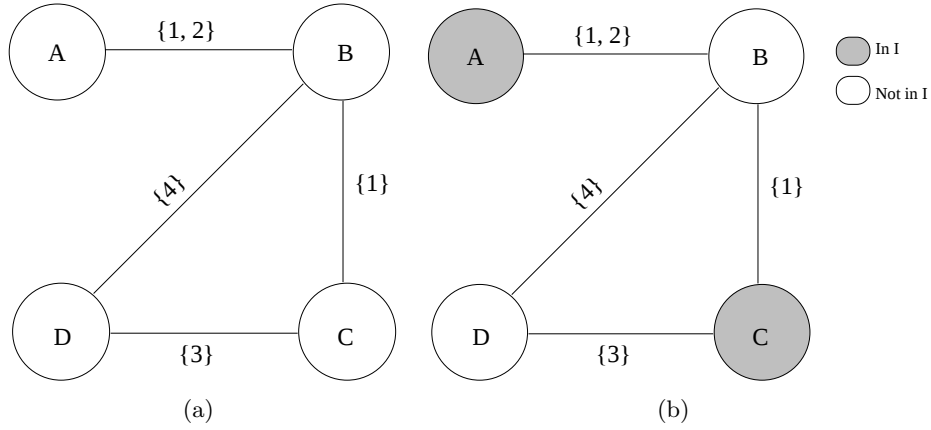
Consider the example graph shown in Fig. 3.6(a). Each node executes the Proc. 3.4. Here we shall look into the details of the working of this procedure with the example graph. When executing the procedure, node A finds out that it has a lower degree compared to its only neighbor B and hence elects itself as a member of I . The insertion of A into I is shown in Fig. 3.6(b). At B, it sees that A has the least degree of all its neighbors, and does not elect itself to I . At node C, it sees that both D and itself have the lowest degrees among its neighbors. The conflicts, as said before, are resolved with node-ids. Since node-id of C is lower than that of D (lexically ranked), C elects itself to I and D does not. The result of this procedure is shown in Fig. 3.6(b).

Procedure 3.4 Form Independent Set

```

1: At each node  $u$  do
2:  $minDegree \leftarrow \text{minimum } \delta_v \forall v \text{ in } Neighbors_u$ 
3:  $minDegreeVertex \leftarrow v \text{ such that } \delta_v = minDegree$ 
4: if  $\delta_u < minDegree$  then
5:    $I \leftarrow I \cup \{u\}$ 
6: else if  $\delta_u = minDegree$  and  $u < minDegreeVertex$  then
7:    $I \leftarrow I \cup \{u\}$ 
8: end if

```

Figure 3.6: Illustration of the *Form Independent Set* procedure

Note that each node can independently determine which node in its 1-hop neighborhood is in the I with two hop topology information. The nodes in this set do not assign $(timeslot, channel)$ for any edge. That is the nodes in complement set, $\bar{I}$ assign $(timeslot, channel)$ to those edges for which the other end is in I or if other node has higher nodeID. The assignment is described in Procedure 3.5. The maximal independent set is desired so that the number of resulting conflicts are minimized.

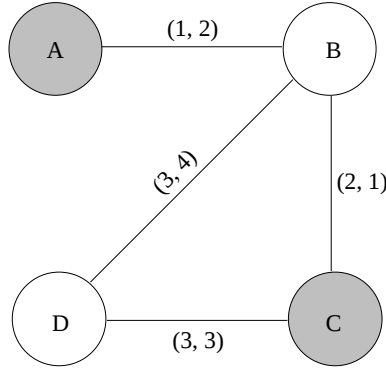
Procedure 3.5 Slot and Channel Assignment

```

1: At each node  $u \notin I$  do
2:   for  $v$  in  $Neighbors_u$  do
3:     if  $NodeID_u < NodeID_v$  then
4:       a.  $Allocation_{(u,v)}.slot \leftarrow \text{first unused slot in } u$ 
5:       b.  $Allocation_{(u,v)}.channel \leftarrow \text{random channel from } ChSet_{(u,v)}$ 
6:     else if  $v \in I$  then
7:       a.  $Allocation_{(u,v)}.slot \leftarrow \text{first unused slot } \geq (\delta_v + \text{number of } Neighbors_v \text{ having node-id less than } u)$ 
8:       b.  $Allocation_{(u,v)}.channel \leftarrow \text{random channel from } ChSet_{(u,v)}$ 
9:     end if
10:   end for

```

Each node in clique $\bar{I}$ is responsible for edges whose other end-vertex has a node-id lesser than itself, or if the other end-vertex is in I . The slots are assigned as stated in the procedure. The random channel selection and a different slot number assignment (for neighbors in I) reduce the number of conflicts. Continuing with our example, the nodes

Figure 3.7: Illustration of the *Slot and Channel Assignment* procedure

B and D are responsible for allocating (*timeslot, channels*) to edges, specifically node B is responsible for (B, A) , (B, C) and (D, B) while node D is responsible for (D, C) . Node B assigns slot numbers 1, 2 and 3 to edges (B, A) , (B, C) and (D, B) respectively, according to Line 7 of Proc. 3.5. Similarly node C assigns slot 3 to edge (D, C) . Random channels are picked for the edges and the resulting graph is shown in Fig. 3.7.

Once the assignment is done, each node exchanges the schedule in its one hop neighborhood. Conflicts may arise since each node executes the procedures locally. In the example, there is a conflict in the assignment for edges (C, D) and (B, D) since both are assigned to Slot 3. These conflicts are resolved as described in Proc. 3.6.

Procedure 3.6 Conflict Resolution

```

1: At each node  $u$  do
2:   for  $v$  in  $Neighbors_u$  do
3:     if  $u$  detects  $Allocation_{(u,v)}$  causes interference within 1-hop neighborhood and
        $NodeID_u$  is the least among all nodes that have detected the conflict then
4:       Propose another channel  $j$  for  $(u, v)$  which results in valid assignment to  $v$ 
5:       if no such  $j$  exists then
6:         Propose a new slot and channel for  $(u, v)$  that results in valid assignment
7:       end if
8:       Get the allocation from  $v$ 
9:       Distribute the allocation to 1-hop neighborhood
10:    else if  $v$  has detected  $Allocation_{(u,v)}$  causes interference within its 1-hop
        neighborhood and  $NodeID_v$  is the least among all nodes that have detected the
        conflict then
11:      Find another valid allocation
12:      Receive proposal from  $v$ 
13:      Choose the better proposal of the two
14:      Send the allocation to  $v$ 
15:      Distribute the new allocation to 1-hop neighborhood
16:    end if
17:  end for

```

If two edges have conflicting schedule, it is not required by all 4 end-vertices to find a new allocation; this only complicates the issue at hand. In some cases, unlike in the example, only 2 nodes may detect the conflict since we check for the conflict in 1-hop

neighborhood. We again resort to node-ids to determine a responsible node for conflict resolution; the least node-id of all end-vertices of the conflicting nodes is responsible to carryout this task. This node checks its allocation table, and first tries if it can solve the conflict by changing the channel assigned to the conflicting edge. If it is not possible, it finds an available slot or creates a new slot if required. This proposal for resolving conflict is sent to the other end-node of the conflicting edge. The other end-vertex also finds a possible valid allocation, compares the two proposals and chooses one of them. The decision is informed to all its 1-hop neighborhood. The conflict resolving node also notes the decision if it results in valid allocation, and broadcasts this information to 1-hop neighborhood. However, if the decision made by the other end-vertex results in a conflict, then the procedure is re-initiated by the least node-id vertex of the conflicting edges. This continues until all the edges have resolved conflicts, and a valid schedule results.

It is important to note that it is enough to propagate the information to 1-hop neighborhood since all conflicts are resolved at both end-nodes of the edge which are facing the conflict.

In our example in Fig. 3.7, the conflicting edges are (B, D) and (B, C) . The least node-id of the conflicting edges end-vertices is B. Since B knows D's allocations, it proposes to use $(4, 4)$. This proposal results in valid allocation, and D also would have proposed this two. D selects the allocation and informs A, C and B. The resulting allocation is shown in Fig 3.8.

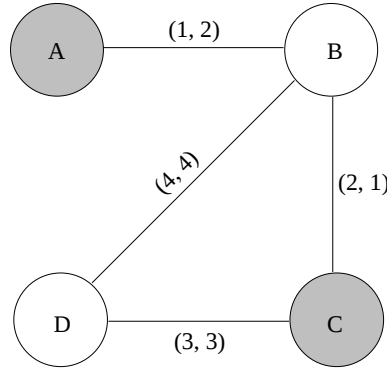


Figure 3.8: Illustration of the *Conflict Resolution* procedure

It may be noted that the conflict resolution procedure does not always result in minimum length schedule. An optimization procedure, optionally, can be executed by the nodes. In this procedure each node advertises its unused slots. If any other node can switch to one of those slots keeping the assignment valid, and the chosen slot is lower than the assigned slot, then that edge is assigned a new slot and channel. This reduces the schedule length, which is one of our goals.

The CBH algorithm is shown in Alg. 3.7.

Complexity

The worst-case time complexity for Proc. 3.4 is $\mathcal{O}(N)$, Proc. 3.5 is $\mathcal{O}(MN)$, Proc. 3.6 is $\mathcal{O}(N^2)$, and $\mathcal{O}(N)$ for the optimization procedure at each node.

Algorithm 3.7 Clique Based Heuristic

- 1: **Input:** the set of one hop neighbors for each node u and the set of available channels between each pair u, v such that v is a neighbor of u
 - 2: **Output:** $Allocation_{(u,v),j,k}$
 - 3: $I \leftarrow \emptyset$
 - 4: Run Procedure 3.4
 - 5: Run Procedure 3.5
 - 6: Distribute $Allocation_u$ over one-hop neighborhood
 - 7: Run Procedure 3.6
-

3.4.2 Localized heuristics

Intuition

The previous sections discussed heuristics that compute an allocation of spectrum at every frame. For these computations to result in valid allocation, a lot of control messages are exchanged. One way to reduce these message exchanges is by devising a new heuristic that accounts for only the changes caused by PU usage between frames.

There can be no change in PU usage that affects the current allocation, then the previous allocation can be reused. However, in another extreme, the PUs can occupy all channels cutting down the whole CRAN. Another scenario is where some SU nodes see a topology change due to PU occupying all available channels between two SU nodes, or SU nodes see a different channel available while the allocated channel gets occupied. The idea of this heuristic is to adapt only to the changes in the environment from previous allocation, and not perform a complete allocation.

Consider the example graph in Fig. 3.9(a) and a valid allocation in Fig. 3.9(b). Now due to PU activity, the graph changes to Fig. 3.9(c). Nodes A, B and C detect that Channel 1 is lost, nodes B and C detect the there is a change in topology i.e., edge (B, C) is cut.

For the edge (A, B) the previous allocation is not valid anymore since Channel 1 is not available for SU usage. Node A having the least node-id of (A, B) , proposes a possible allocation to B adhering to the *Condition 3*, in this case it will be a change of channel i.e., $(1, 2)$. Node B agrees to the proposal as long as it satisfies the *Condition 3*. In node B, there is a change in topology since the edge (B, C) was deleted. Node B now performs a reallocation for all edges whose other end vertex has a node-id greater than B i.e., (B, D) will have a reallocation. Node B, in consultation with Node D, will assign the first unused slot, so (B, D) will get an allocation of $(2, 4)$. Node C has also detected that Channel 1 is unavailable; it will re-allocate for (C, D) . The new allocation is shown in Fig. 3.9(d). To reduce the schedule length, the optimization procedure described in Sec. 3.4.1 can be executed.

Algorithm

The Localized Heuristic described in the previous section is presented in Alg. 3.8. Note the input $Allocation_{(u,v),j,k}^{\text{old}}$ can be from CBH, CTA or any other ‘parent’ algorithm. The number of message exchanges in this algorithm less than that of CBH algorithm. Firstly, there is no requirement to distribute the allocation to its neighbors (step 6 of

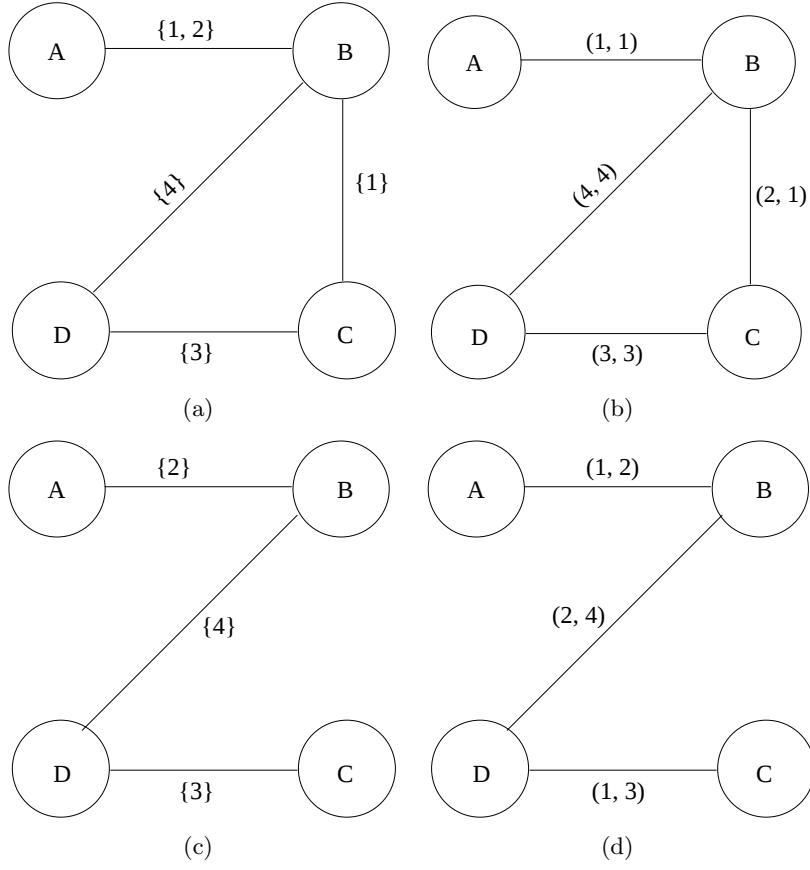


Figure 3.9: Illustration of the Localized Heuristic

Alg. 3.7). Secondly, the procedure here is similar to Conflict Resolution procedure of CBH. Moreover, the messages are exchanged only if a node is affected by PU.

Note that change in topology takes higher priority than channel change. In case a node experiences both, only topology change is taken into account since it will anyway reallocate channel, slot pair to the edges.

Complexity

The complexity of this algorithm is $\mathcal{O}(NMmaxSlot)$; where $maxSlot$ is the maximum number of slots used, at every node.

3.5 Simulation

3.5.1 Simulation environment and Experiments setup

We performed simulations to test the algorithms for various metrics. A new simulation environment was developed for this purpose in Java programming language. The discrete-event simulator runs as a single-threaded application. It simulates a 2D area with four PUs. The simulator does not simulate the wireless transmission and reception

Algorithm 3.8 Localized Heuristic

```

1: Input: Previous set of neighbors for  $x$   $Neighbors_x^{(old)}$ , the current set of neighbors
   for  $x$   $Neighbors_x$ , previous allocation  $Allocation_{(x,y),j,k}^{(old)}$ , the set of channels that
   were available for all  $(x,y)$   $ChSet_{(x,y)}^{(old)}$ , the current set of channels that are available
    $ChSet_{(x,y)}$ 
2: Output:  $Allocation_{(x,y),j,k}$ 
3: At every node  $x$  do
4:   if  $Neighbors_x^{(old)} \neq Neighbors_x$  then
5:     for  $y \in Neighbors_x$  and  $NodeId_y > NodeId_x$  do
6:       Find the least  $k$  and a  $j$  such that the assignment  $Allocation_{(x,y)} \leftarrow (k,j)$ 
       satisfies Condition 3  $\forall j$  in  $ChSet_{(x,y)}$  and for  $k$  in  $TS_k$ 
7:     end for
8:   else if  $\forall y \in Neighbors_x, Allocation_{(x,y),j,k}^{(old)} j \notin ChSet_{(x,y)}$  then
9:     Find  $j' \in ChSet_{(x,y)}$  such that Condition 3 is satisfied
10:    if  $j'$  does not exist then
11:      Find the least  $k$  and a  $j$  such that the assignment  $Allocation_{(x,y)} \leftarrow (k,j)$ 
      satisfies Condition 3  $\forall j$  in  $ChSet_{(x,y)}$  and for  $k$  in  $TS_k$ 
12:    end if
13:   else
14:      $Allocation_{(x,y),j,k} \leftarrow Allocation_{(x,y),j,k}^{(old)}$ 
15:   end if

```

since we are considering the allocation problems here, albeit wireless medium will affect the algorithm in real-world implementation. Few of the important classes of the simulator are explained below.

1. *Simulator*: This is the main object of the simulator. The simulator creates a random topology of the SU nodes. There are four PUs that almost cover the entire area. The simulator implements a M/M/1 queue for processing the events. A event can be *CALL_ARRIVAL*, *CALL_DEPARTURE* or *SU_FRAME*. The *CALL_ARRIVAL* and *CALL_DEPARTURE* events are modeled as exponential on/off distribution, while *SU_FRAME* event is periodic. The *CALL_ARRIVAL* and *CALL_DEPARTURE* events are associated with a *PrimaryUser* object. When the *SU_FRAME* event occurs, a snapshot of the system is captured in a graph, and passed on to the algorithms described above to compute schedules.
2. *Topology*: This object takes the input of the size of the area and number of SU nodes, and distributes the SU nodes over the area uniformly. It then forms the initial graph of the SU nodes by finding the neighbors for every node. Two nodes can be neighbors if they are in communicating radii of each other. To check whether if nodes are in communicating radius, the shadow fading path loss model is used.

$$Loss_{dB}(d) = PL_{dB}(d_0, f) + 10\gamma \log_{10} \left(\frac{d}{d_0} \right) + \psi_{dB} \quad (3.7)$$

where,

$PL_{dB}(d_0, f)$ is the Friis path-loss for the frequency f and reference distance for far-field of the antenna d_0

γ is the path-loss exponent

d is the distance between the transmitter and the receiver in m
 ψ_{dB} is a Gaussian random variable - $N(0, \sigma)$ in dB

The parameters used are shown in Table 3.2 and were taken from [17] for an outdoor cellular environment. We have not taken the receiver losses into consideration.

Parameter	Value
d_0	1m
f	5GHz
γ	3
σ	7

Table 3.2: Parameter values used in the simulator

3. *PrimaryUser*: A PrimaryUser object is created for every PU. A PrimaryUser object handles the events of CALL_ARRIVAL and CALL_DEPARTURE. At each CALL_ARRIVAL event it occupies one more channel (if available), and at every CALL_DEPARTURE it releases a channel for the SUs to use.
4. *SecondaryUser*: A SecondaryUser object is created for every SU. A SecondaryUser object is associated to zero or more PrimaryUser objects, depending on coverage of different PrimaryUsers at its location. At every SU_FRAME, a ‘spectrum sensing’ is performed i.e., it queries the associated PrimaryUser objects for which channels are available (note that we have assumed perfect sensing). Next, it computes who its neighbors are - two nodes in communicating ranges of each other are neighbors only if at least one channel available between them. It then computes the available channels between its neighbors.

The Simulator object passes this information to every algorithm to compute a schedule.

The simulator, with the help of the objects described above, creates scenarios as shown in Fig. 3.10. The dimensions of the area are input by the user. Four PUs are placed at four corners with their coverage as shown in Fig. 3.10. The user inputs the number of SUs to be randomly distributed over the area. The maximum number of channels available for PUs (and SUs) is input by the user, while the actual number is chosen randomly between 1 and the maximum number. Each PU has a call arrival rate of 36 calls/hour with a mean call duration of 80s. Due to the exponential on-off distribution of channel occupancy dynamic topologies of the SU nodes are created, number of channels varying between neighbors.

Four simulation scenarios are considered as shown in Table 3.3. In the first two scenarios, the number of nodes are chosen to maintain the same density for different areas. The number of channels are varied across scenarios. In the next two scenarios, the area is kept constant to create sparse and dense graphs; again, the number of channels are varied across scenarios.

The algorithms implemented are Color Then Assign (CTA), Clique Based Heuristic (CBH), and two Localized Heuristics (LH) - one that takes input from CTA (LH CTA) and the other that takes input from CBH (LH CBH). The results are compared with MAC Scheduling proposed in [35]. While CTA is centralized, LH CTA is partially

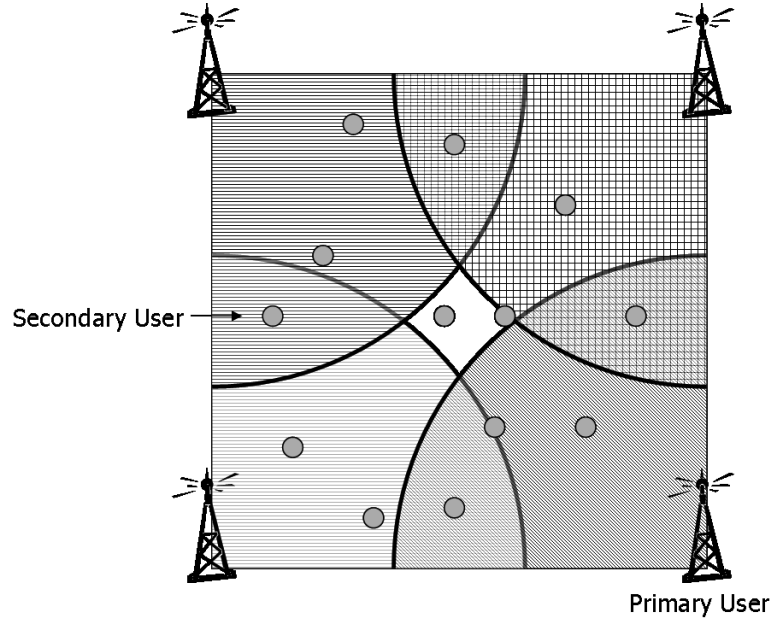


Figure 3.10: Scenario of simulations

Scenario 1			Scenario 2		
Area	Nodes	Channels	Area	Nodes	Channels
100x100	10	1 - 25	100x100	10	1 - 2
200x200	40	1 - 25	200x200	40	1 - 2
300x300	90	1 - 25	300x300	90	1 - 2
Scenario 3			Scenario 4		
Area	Nodes	Channels	Area	Nodes	Channels
200x200	10	1 - 25	200x200	10	1 - 2
200x200	40	1 - 25	200x200	40	1 - 2
200x200	90	1 - 25	200x200	90	1 - 2

Table 3.3: Simulation scenarios

decentralized i.e., the first allocation is centralized, CBH and LH CBH are completely distributed heuristics. All algorithms are run for 20 iterations with 100 SU_FRAMES in all the scenarios described before.

3.5.2 Schedule Length

The ILP formulation shown in Sec. 3.1.5 is solved using CPLEX [1] to compare different algorithms. Since CPLEX takes huge amount of time for a 90 node graph (due to the huge number of edges and channels, one constraint file was as big as 600MB), the simulation scenarios considered are modified and are shown in Table. 3.4. A new experiment consisting of 25 nodes in an area of $158 \times 158 m^2$ is added (the density of nodes is approximately the same for other experiments in scenarios 1 and 3). However, we also show the performance of other algorithms without comparing it to ILP.

Fig. 3.11 shows the mean schedule lengths of various heuristics and the optimal result from ILP in Scenario 1. For 10 nodes, there is no significant difference in the lengths.

Scenario 1			Scenario 2		
Area	Nodes	Channels	Area	Nodes	Channels
100x100	10	1 - 25	100x100	10	1 - 2
158x158	25	1 - 25	158x158	25	1 - 2
200x200	40	1 - 25	200x200	40	1 - 2
300x300	90	1 - 25	300x300	90	1 - 2
Scenario 3			Scenario 4		
Area	Nodes	Channels	Area	Nodes	Channels
158x158	10	1 - 25	158x158	10	1 - 2
158x158	25	1 - 25	158x158	25	1 - 2
158x158	40	1 - 25	158x158	40	1 - 2
200x200	90	1 - 25	200x200	90	1 - 2

Table 3.4: Simulation scenarios to compare schedule lengths

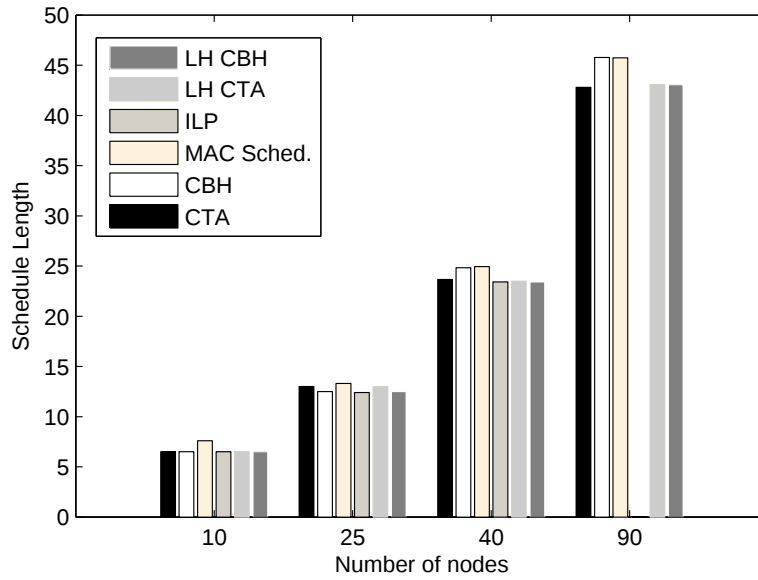


Figure 3.11: Mean schedule lengths of various algorithms in Scenario 1

When 25, 40 and 90 nodes are considered, CTA outperforms the other heuristics since it is centralized and has complete knowledge of the topology. CBH outputs slightly lower lengths than MAC scheduling algorithm. The LH based algorithms produce lengths less than or equal to their parent algorithms. This is because, LH based schemes try to adapt to the new situation by making minimal changes. However when new assignments are sought, the ‘greediness’ of the algorithms affects the lengths. In the proposed algorithms, the output depends on the decisions made during the assignment, and due to greediness of the algorithms the decisions leading upto optimal assignments may not be made. It is also seen that no matter what the parent algorithm, the mean schedule lengths are almost the same after several runs. In all the cases, the proposed heuristics are very close to the optimal result obtained from solving ILP equations, which shows the performance of the heuristics.

To compare the performance of the algorithm, all the PUs were turned off and only 1

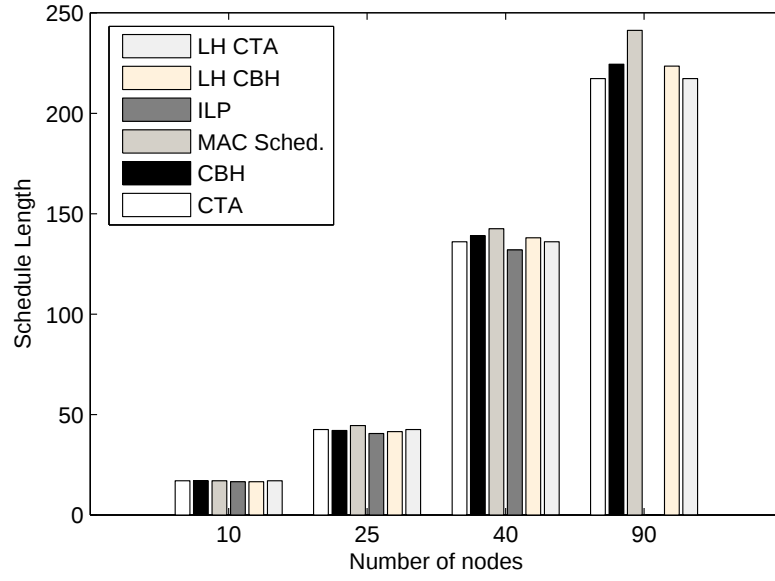


Figure 3.12: Mean schedule lengths of various algorithms in with 1 channel

channel was used by the SUs. As a result no dynamic topologies were created. The resulting schedule lengths are shown in Fig. 3.12. The performance of the LH based heuristics are the same as of their parent algorithms since there are no changes in the scenario. Again all the heuristics have a length close to the optimal one.

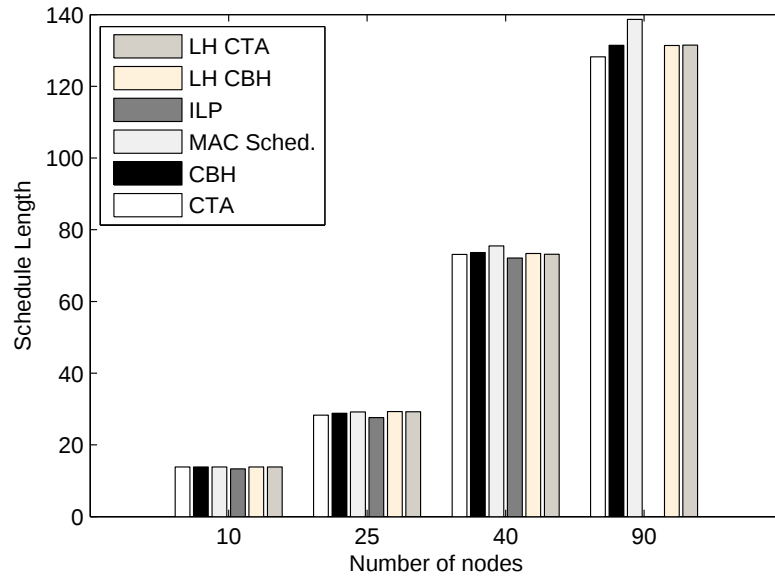


Figure 3.13: Mean schedule lengths of various algorithms in Scenario 2

In Scenario 2, a maximum of two channels are given. In the experiment, only one of the two channels was mostly available and at no point were both the channels available to the nodes. In few cases no channels were available to the SUs, and no schedule can be obtained in these cases. Dynamic topologies were created due to change in channel

availability. The result is shown in Fig. 3.13. All the proposed algorithms outperforms the MAC Scheduling [35] algorithm, and are near the optimal value.

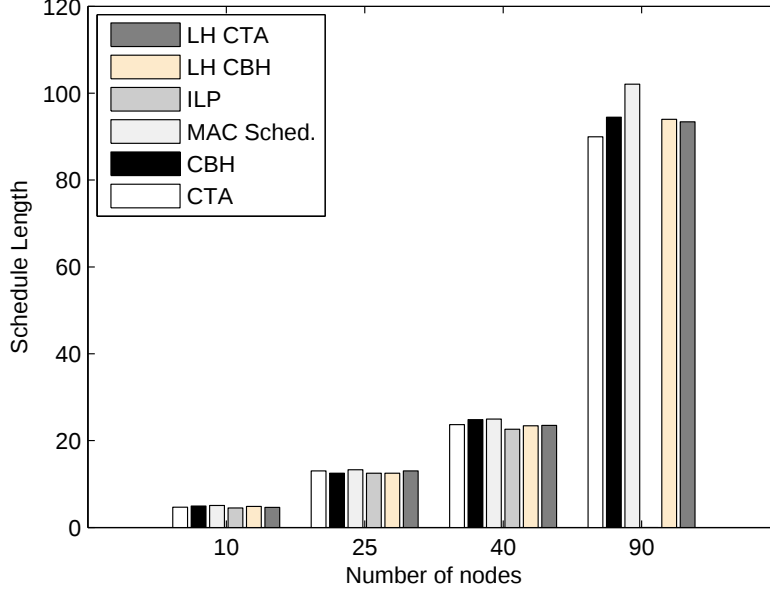


Figure 3.14: Mean schedule lengths of various algorithms in Scenario 3

In Scenario 3, the density of the nodes is changed from low to high with 25 maximum available channels. In this scenario the graphs vary from sparse to dense. Fig. 3.14 shows the schedule lengths computed by the algorithms. In sparse graphs, all algorithms have similar performance. When the density increases (90 nodes), as expected the CTA heuristic performs the best. The other proposed heuristics also have similar performances, close to optimal.

As in Scenario 3, the density varies in Scenario 4 but the number of available channels are very less. As in case of Scenario 2, in most of the cases there was only one channel available, and both channels were available for negligible number of cases. The result shown in Fig. 3.15 shows performances similar to Scenario 3.

Performance bound

A loose bound was stated in Sec. 3.2.1. A tight bound is given in the equation below. The first term again represents the maximum number of slots of 2-approximation of greedy edge coloring algorithm. In the second term $\mathcal{O}(\Delta^2(G))$ is the maximum number of edges possible in K_n ; this is divided by N_{channels} since as the number of channels increases, the number of slots required decreases.

$$2(\Delta(G) + 1) + \frac{\Delta^2(G)}{N_{\text{channels}}} \quad (3.8)$$

A tighter bound is given in Eqn. 3.9 where the denominator of the second term is changed to $N_{\text{channels}}^{3/2}$. It is confirmed from simulations that is number of slots reduce with

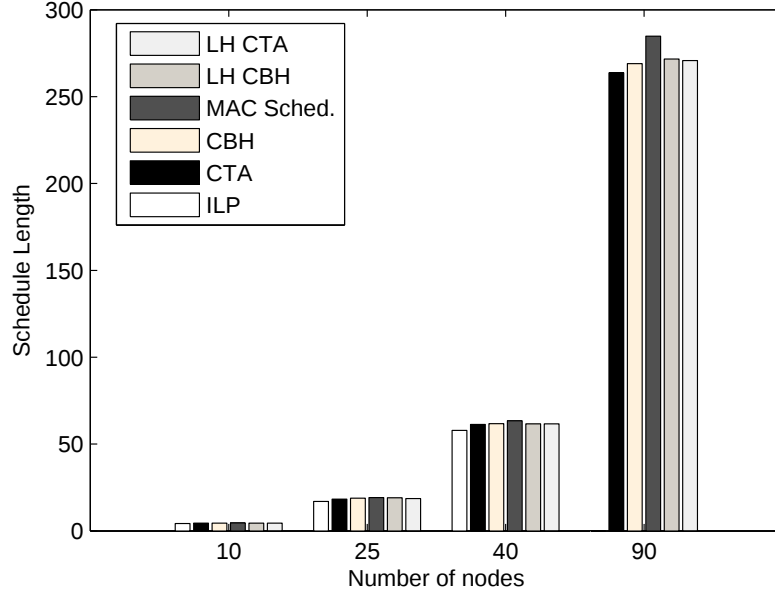


Figure 3.15: Mean schedule lengths of various algorithms in Scenario 4

$N_{\text{channels}}^{3/2}$ as number of channels increase.

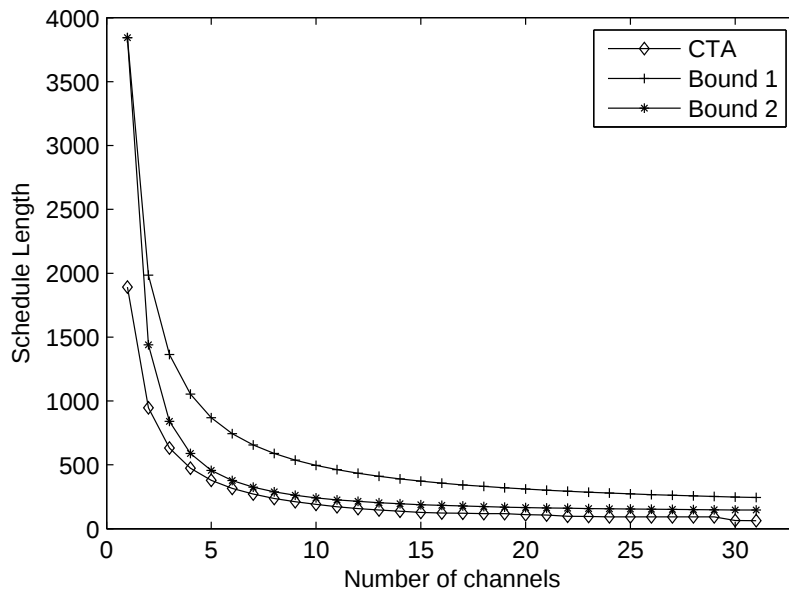
$$2(\Delta(G) + 1) + \frac{1}{N_{\text{channels}}^{3/2}} \cdot \Delta^2(G) \quad (3.9)$$

To confirm that the equation represents a bound, we simulated the worst case scenarios i.e., a complete graph with nodes $N = \{4, 5, \dots, 63\}$, and for each N , we varied number of channels from 1 to $\lceil N/2 \rceil$ (considering Corollary 1). In each case we found that the value from the equations always provides a higher value. One illustration for $N = 62$ with CTA is shown in Fig. 3.16, where Bound 1 refers to the bound given by Eqn. 3.8. Since the other heuristics have a very close performance as that of CTA, the Eqn. 3.8 is still valid. Fig. 3.16 illustrates that the Eqn. 3.9 (shown as Bound 2 in the figure) is a closer bound than the represented by Eqn. 3.8.

We use this bound to find $\mathcal{K}$, the number of slots in the data transmission period.

3.5.3 Utilization

Increasing spectrum utilization is one of the main goals in CRANs. In this section we present utilization of slots by nodes within one schedule length. A slot is *utilized* by a node if the node is allocated a channel in the slot. In addition to the allocated slots, there are can be free-slots. In the schedule it is possible that an edge not participating in the current timeslot, can be allocated with one of the available channels without disturbing the schedule and without causing interference in its two-hop neighborhood. Such a slot is counted as *free-slot* for both the end-vertices of the edge. This increases the spectrum utilization. The free-slots usage can either be pre-determined (possible in the case of CTA) and distributed with the schedule, or they may be found to be free


 Figure 3.16: Simulation of CTA and the bound for $N = 62$

by the nodes listening on control channel, and then setting up the communication with other free neighbors. Here, in simulations, we run an algorithm for a possible free-slot allocation and the averages are presented here. The simulations scenarios considered are as in Table. 3.3.

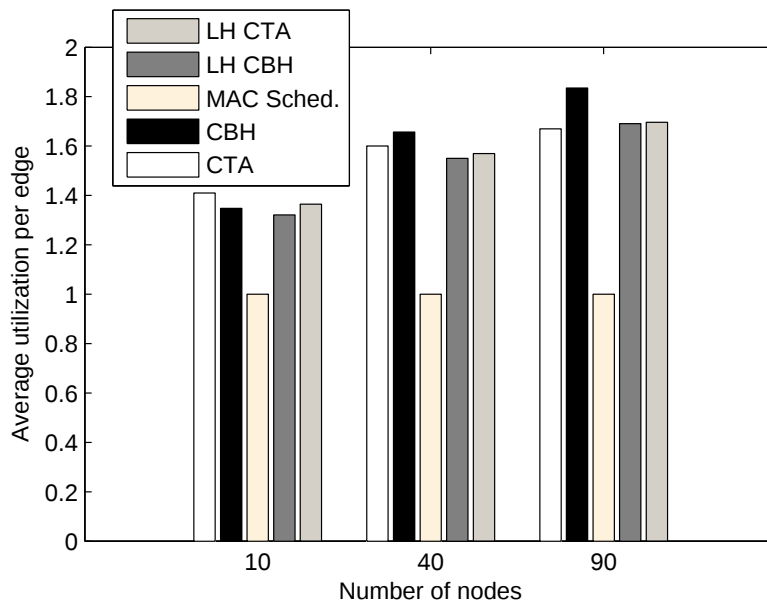


Figure 3.17: Average utilization per edge in Scenario 1

The percentage of utilization per edge for Scenario 1 is shown in Fig. 3.17. MAC scheduling proposed in [35] does not use the free-slots. The schedule, however, ensures that every edge is utilized once, hence the average utilization per edge is considered 1 for all the cases. The average utilization when using CTA is lesser since the schedule

length produced by it is lower than the other algorithms. For the same density of nodes, the average utilization reduces when the number of channels reduce. This can be seen in Fig. 3.18.

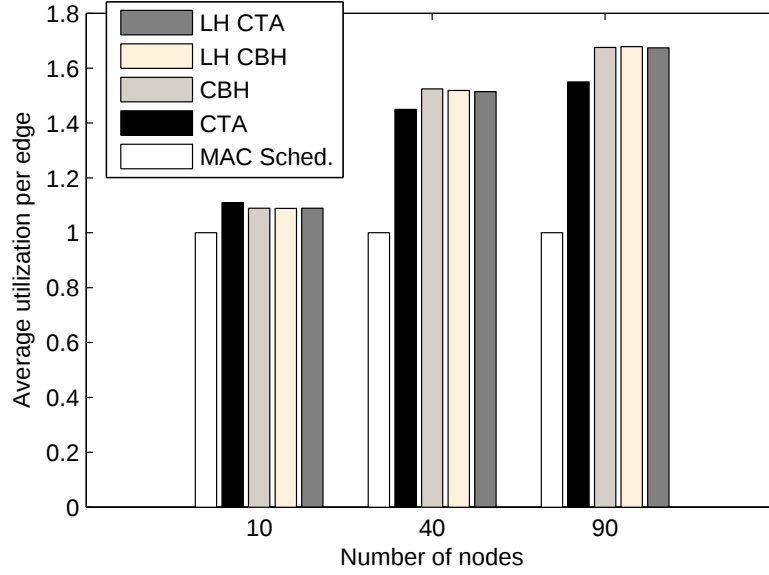


Figure 3.18: Average utilization per edge in Scenario 2

For the same density of nodes, the average utilization reduces when the number of channels reduce. This can be seen in Fig. 3.18. Similarly, the average utilization for Scenarios 3 and 4 are shown in Fig. 3.19 and Fig. 3.20.

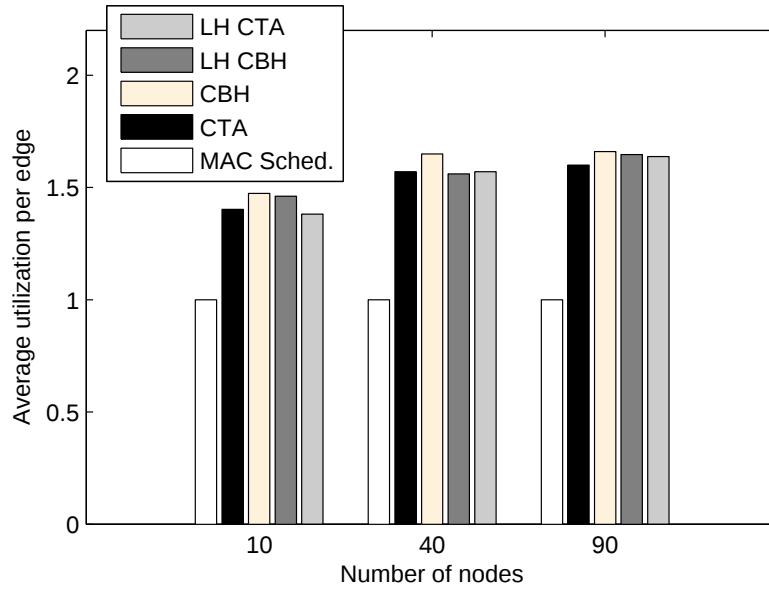


Figure 3.19: Average utilization per edge in Scenario 3

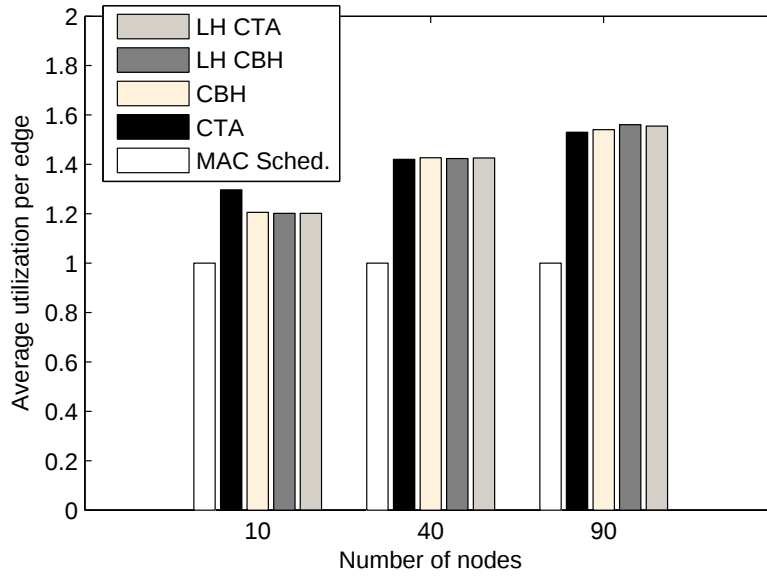


Figure 3.20: Average utilization per edge in Scenario 4

3.5.4 Messages exchanged

Another metric to compare is the number of messages exchanged by different algorithms. Here we ignore the messages exchanged in the information exchange period where each node learns about its two-hop neighbors since it is performed for all the algorithms. Further we leave out the CTA algorithm since it is a centralized algorithm. We compare the number of messages exchanged by the MAC scheduling of [35], conflict resolution phase of CBH, and the adaptation in the LH based algorithms. We assume one message contains all the required information to be sent, and there are no retransmissions involved. The messages used for broadcasting the schedule length in MAC Scheduling is also not counted. In MAC scheduling since each node broadcasts its schedule to two-hops, the number of messages exchanged of a node is equal to $1 + \text{degree of the node}$. In CBH, apart from 1-hop schedule distribution two messages are exchanged for every conflicting edge - one message for proposal by a node and a reply message by the other node with a decision of what schedule to use. It is possible that more than two messages are exchanged for the same edge since the decision could still result in a conflict. In LH based heuristics, there is no schedule distribution but only adapts the edges with invalid schedules with a valid schedule, and for each such adaptation two messages are exchanged, similar to the CBH. It is possible that more than two messages are exchanged for the same edge since the decision could still result in a conflict.

For the four scenarios described above, the average number of messages exchanged are compared. When the number of available channels are aplenty (in Scenarios 1 and 3), the LH based heuristics exchange the least number of messages. The CBH performs equally worse as that of MAC scheduling in Scenario 1, but has lower performance when the density of nodes increases in Scenario 3. Fig. 3.21 and Fig. 3.22 show the average messages exchanged per node in Scenario 1 and 3 respectively.

In Scenario 2 and 4 (see Fig. 3.23 and Fig. 3.24 respectively), the CBH again exchanges more messages than other algorithms. This is mainly because of the lower number of

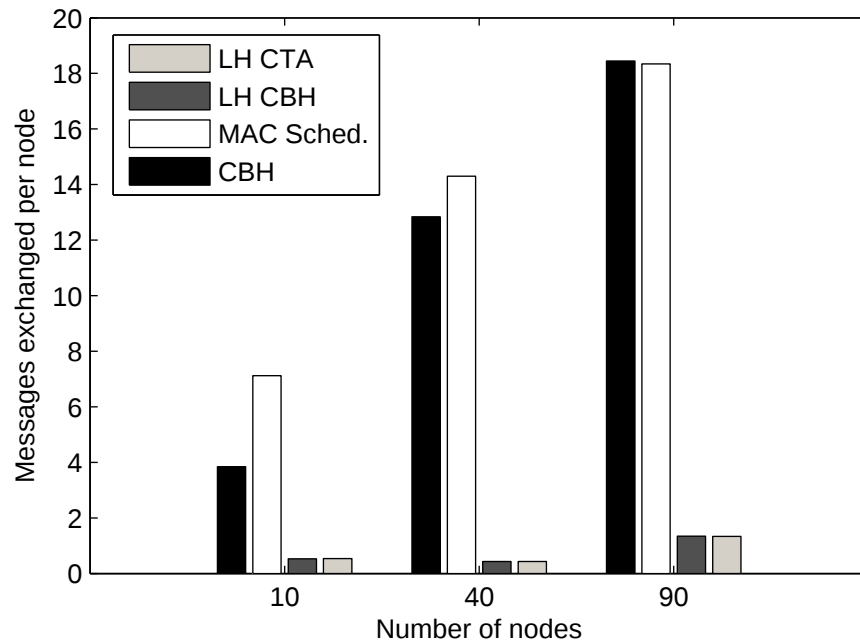


Figure 3.21: Mean number of messages exchanged in Scenario 1

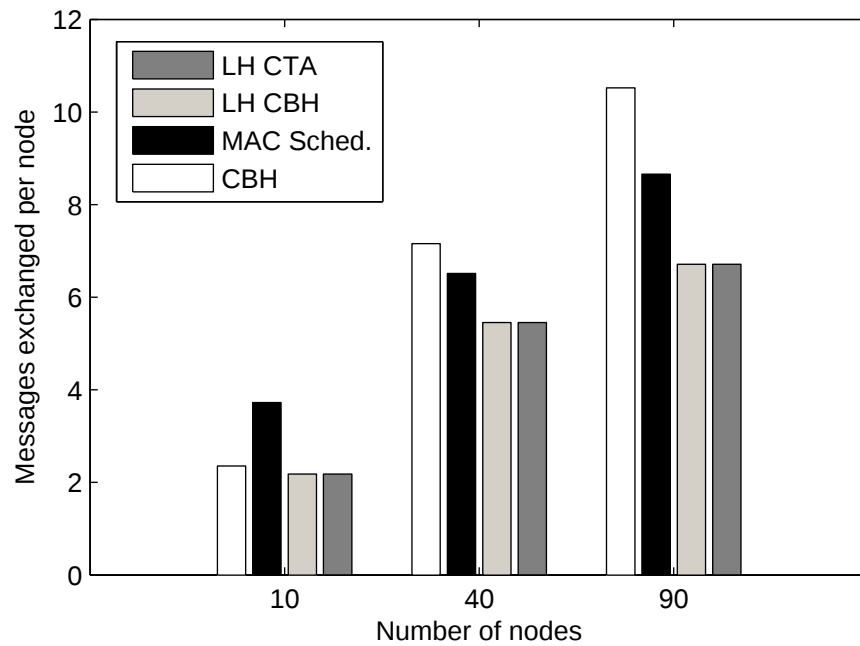


Figure 3.22: Mean number of messages exchanged in Scenario 3

available channels, and hence causing more conflicts in the drawn distributed schedule. The LH based heuristics are also affected by the lower number of available channels and the increase in density (in Scenario 4) but still outperform MAC scheduling.

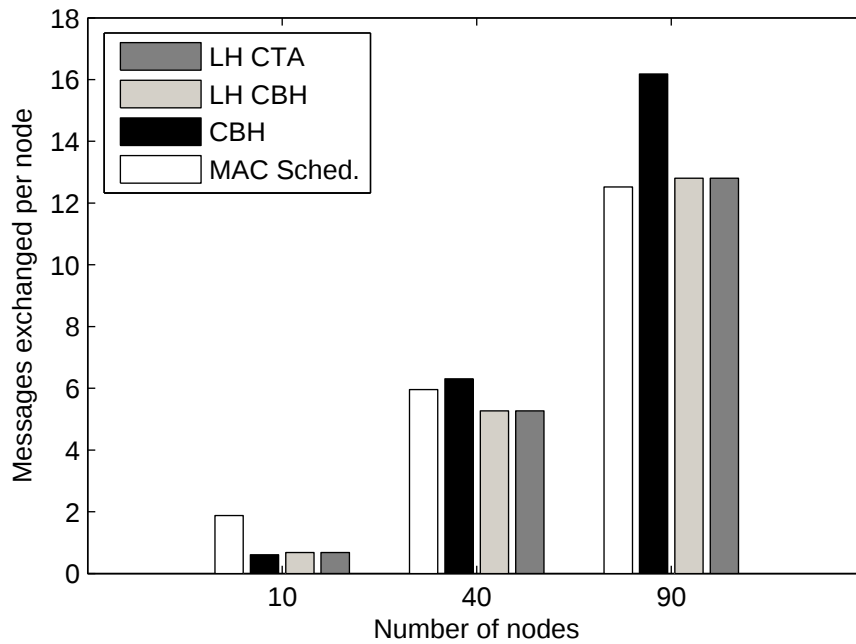


Figure 3.23: Mean number of messages exchanged in Scenario 2

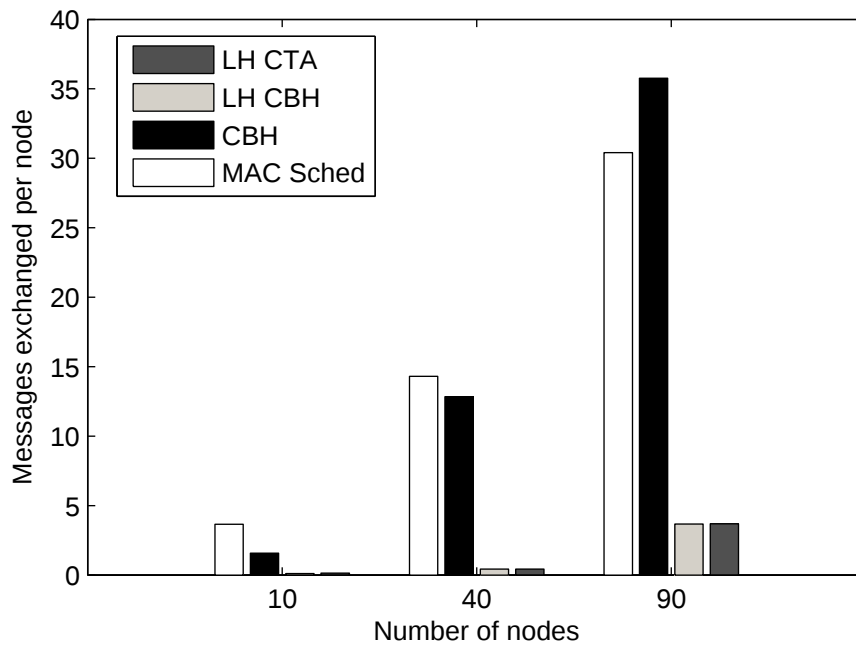


Figure 3.24: Mean number of messages exchanged in Scenario 4

3.5.5 Analysis

Based on the results in the previous sections, we can conclude the following with respect to:

- *Schedule length:* All proposed algorithms show better performances than the MAC scheduling. Out of the proposed heuristics, CTA produces the lowest schedules because of the complete knowledge of the topology. The LH based algorithms in the long run produce similar schedules regardless of the parent algorithm and outperforms CBH.
- *Utilization:* Utilization is similar for the proposed heuristics under all scenarios. CBH has the highest utilization/edge since the schedule length is higher than the other algorithms. MAC scheduling does not propose to use the free slots.
- *Number of message exchanges:* The number of message exchanges per node is lower for CBH when the number of channels are more and the density is low. However when the number of channels reduce and the density increases, the number of conflicts are high. Added to that is the distribution of schedule to neighbors which increases the number of messages exchanged. The LH based algorithms always show a good performance since there is no schedule distribution, and the adaptation takes place with fewer message exchanges. However, the LH based algorithms suffer when the density of nodes increases and the changes in the environment caused by PU increase. It should be noted that the MAC Scheduling algorithm has a Phase-2 where the schedule length is propagated through out the network. This requires a lot of message passing, which has not been used for comparison. Therefore, the number of message exchanges in the proposed heuristics have better performance than MAC Scheduling.

Though MAC scheduling, CBH and LH based algorithms are all distributed algorithms, an important differentiating factor between MAC scheduling and our proposed distributed methods is that the assignment in MAC scheduling happens sequentially whereas we perform assignment in parallel in the nodes. In MAC scheduling, every node is ranked, and waits for its turn in its two-hop neighborhood to get its chance to perform assignment. Hence the total time of assignment is much higher. Whereas in our assignments, every nodes tries to perform the assignment simultaneously. The trade-off is the number of conflicts that arise, which is reflected in the number of messages exchanged. To reduce this number, we proposed the LH algorithms which mainly cut down the message exchanges by doing away with schedule distribution every time. Hence the total time of assignment is much lower.

The number of messages exchanged for CBH and LH based algorithms increases when the density of the nodes and PU activity increases. In real-world the PU activity is not always at the peak throughout the day [36], but only during certain hours. Considering this aspect, the ill-effects of high PU activity on LH and CBH are averaged-out in the long run. For LH based algorithms in the long run, the number of message exchanges will be much lower than any other algorithm, making them highly scalable and efficient.

While MAC scheduling, CTA, and CBH perform network-wide spectrum allocation per each frame, LH based algorithms are low complexity with local adaptations which works by exploring temporal variations. Further LH based algorithms have low overhead and delay. These make LH based algorithms more desirable for implementation for the assumed system model.

3.6 Summary

In this chapter we stated the problem of communication and coordination between SU nodes in CRANs. We then presented a system model and validated it. We proposed the use of graph theoretical techniques for this problem - on per snapshot allocation. We formally presented the problem, and proved it to be an $\mathcal{NP}$ complete problem. We proposed centralized and distributed heuristics for the same, compared with another heuristic in the literature for its performance. It was shown that the proposed heuristics outperforms the heuristic in the literature. While currently available heuristics usually perform a network-wide spectrum allocation for each snapshot, we show that a locally adapting heuristic performs as well and is as efficient as the others.

CHAPTER 4

Fairness in Allocation

In the previous chapter, we considered a system model wherein the SU data transmission frame was divided into $\mathcal{K}$ slots, where $\mathcal{K}$ was calculated as a bound depending on the topology. This makes the slots have unequal length for different topologies, though the nodes may have the same amount of data to send. It is desirable to have equal slot sizes and remove the dependency on the topology. If we make the number of slots the same for all topologies, then we have a problem of finding a fair schedule i.e., some topologies may have schedule lengths greater than the number of slots we have fixed. By fairness, we mean all transmitter-receiver pair should approximately have the same percentage of access to the medium under similar traffic load. In this chapter we describe this problem, system model and propose changes to the Edge coloring and Clique based heuristics of Chap. 3 to solve this problem.

4.1 Problem description

In the previous chapter, we considered a system model where we were required to just compute valid schedules. The number of slots $\mathcal{K}$ in the data transmission period was determined by a bound. This varies across topologies. However, there are three disadvantages in doing so:

1. In dense topologies the nodes get to transmit less data due to lower slot size.
2. The information about the topology (maximum degree and number of channels) is assumed to be known by all the nodes.
3. Since $\mathcal{K}$ is an upper bound some slots are unused.

It is desirable to have fixed number of slots that does not depend on the topology of the CRAN. By doing so, we address the first two disadvantages. The third disadvantage is less of a disadvantage since nodes make use of the free-slots for transmissions. However, a new problem arises: a fair schedule is required. By fairness, we mean all transmitter-receiver pair should approximately be allocated with the same percentage of access to the medium, over a period of time.

To illustrate the problem, let us consider the CRAN in Fig. 4.1. The channels available for each link are also shown. A valid schedule for this graph is shown in Fig. 4.2, which

is of length 3.

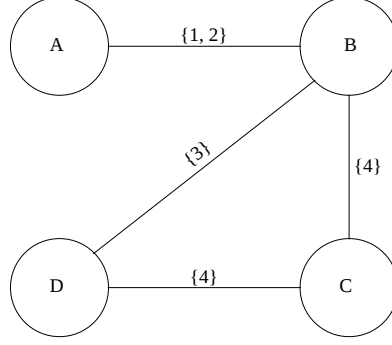


Figure 4.1: A CRAN. The available channels between nodes are indicated within brackets.

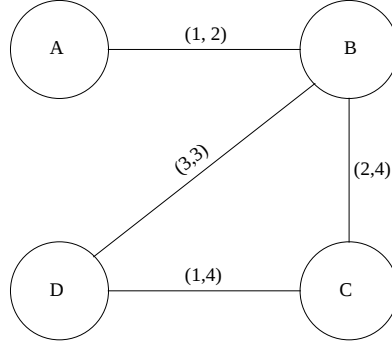


Figure 4.2: A valid assignment

Let the number of slots, $\mathcal{K}$, in data transmission period per each frame be 2. The nodes periodically sense with a period T . Let the graph at time $t = pT$, where $p = \{0, 1, 2, 3\}$, be as shown in Fig. 4.3.

Using any of the techniques described in Chap. 3, the schedule length, L_s , will be 3. They cannot be directly applied since $\mathcal{K} = 2$. However there is a possibility of computing a schedule using an algorithm of Chap. 3 every $\lceil L_s/\mathcal{K} \rceil$ frames, where the schedule is spread across multiple frames. This cannot be used because of the dynamic topologies that are created due to the PU activities, which is illustrated for the example in Fig. 4.3.

The first schedule is computed with Fig. 4.3(b) at $t = 0$ with schedule length 3. Fig. 4.4(a) shows the allocation resulting from an algorithm of Chap. 3 split into two frames. The empty slots can be used for free-slot communication. After Frame 1 i.e., $t = T$, the sensing of the spectrum results in a topology as shown in Fig. 4.3(b) where only the edge (A, B) exists. This change is unaccounted since the schedule spans across two frames. Fig. 4.4(b) shows the actual allocation due to the change in topology at $t = T$. The next schedule is computed for the graph in Fig. 4.3(c), which requires only two slots as shown in Fig. 4.4(c). This means, the next schedule is computed at $t = 3T$ for the graph in Fig. 4.3(d). Assuming the graph does not change at $t = 4T$, the actual schedule is shown in Fig. 4.4(d).

It can be clearly seen that the edge (B, D) does not get a slot until Frame 5, while other edges get more chances. If free-slot usage is also accounted, then edge (A, B) gets

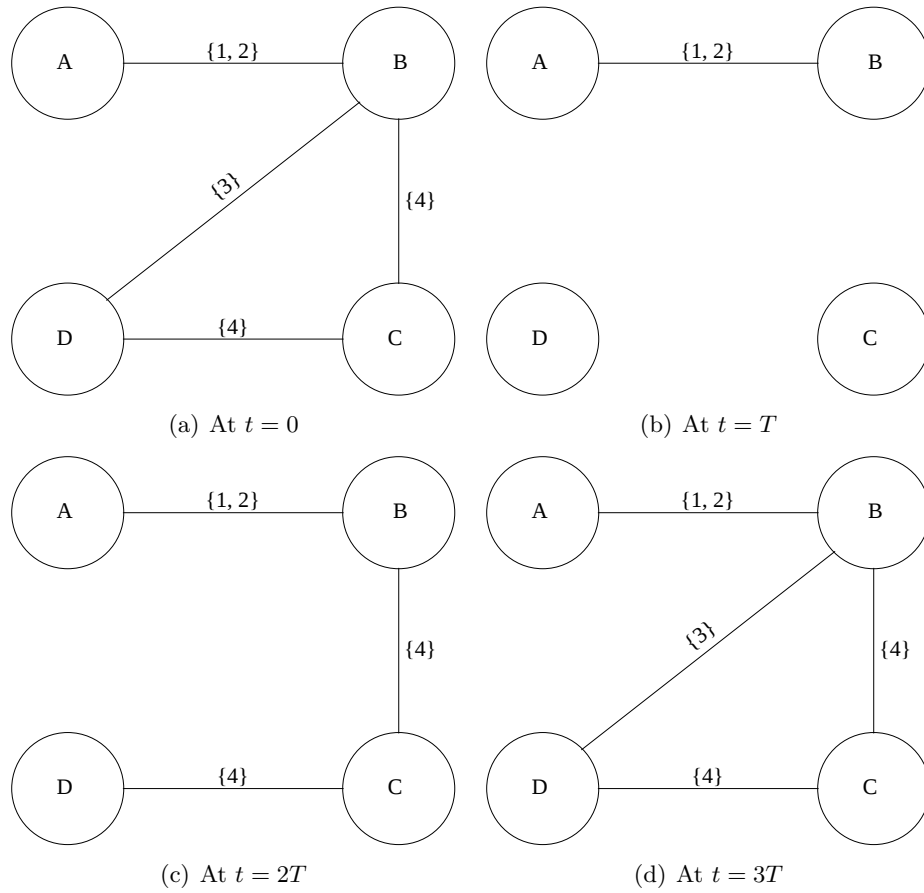


Figure 4.3: Snapshots of the graph at frame times $t = 0, T, 2T, 3T$

Frame 1		Frame 2		Frame 1		Frame 2	
Slot 1	Slot 2	Slot 1	Slot 2	Slot 1	Slot 2	Slot 1	Slot 2
(A, B)	(B, C)	(B, D)		(A, B)	(B, C)		
(C, D)				(C, D)			

(a) Allocation split into two frames

Frame 3		Frame 4		Frame 5	
Slot 1	Slot 2	Slot 1	Slot 2	Slot 1	Slot 2
(A, B)	(B, C)	(A, B)	(B, C)	(B, D)	
(C, D)		(C, D)			

(c) Possible allocation for schedule computed at $t = 2T$

(b) Actual allocation for schedule computed at $t = 0$

(d) Possible allocation for schedule computed at $t = 3T$

Figure 4.4: Inefficiency of using schedule across frames

more number of slots than any other edge. This *unfairness* should be reduced, which is addressed in this chapter.

4.1.1 System model and assumptions

The system model considered here is similar to the one described in Sec. 3.1.1 except that the total number of slots in the data transmission period are fixed. For readability purposes, the system model is repeated here with the changes.

We consider a multi-hop CR network formed by N nodes, each with a unique Node-ID $\in \{1, 2, \dots, N\}$. These nodes are equipped with a half-duplex software-defined transceiver i.e., each node can either transmit or receive at any instant of time on a specified channel from a pool of channels. These nodes are interested in point-to-point communication, i.e., we consider only 1-hop communication. We assume that the spectrum is divided into M orthogonal channels that are symmetric. Each channel has a unique number in $\{1, 2, \dots, M\}$. Further, we assume that a common control channel (CCC) is available to exchange the control messages amongst the nodes. The nodes are time synchronized either through the common control channel or with the help of a GPS receiver. We assume that all nodes have data to transmit to all its neighbors to ensure that we consider the worst case scenario of the problem.

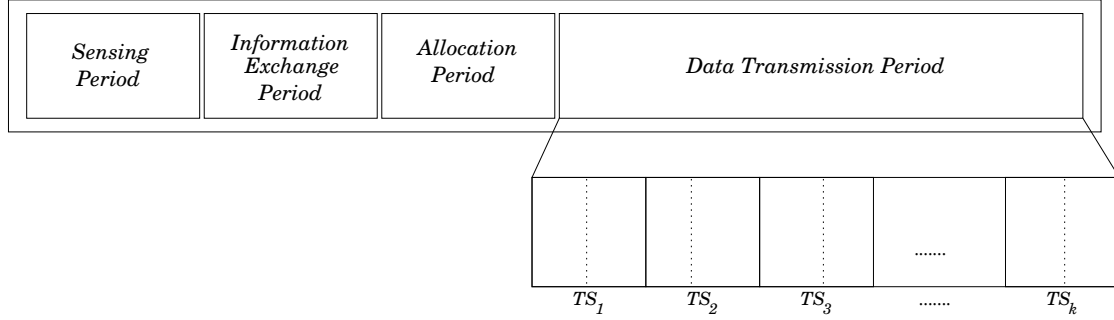


Figure 4.5: MAC super frame

Consider the MAC super frame as shown in Fig. 4.5. The super frame is divided into sensing, information exchange, allocation and data transmission periods. In the sensing period, the channels that are free of PUs are found first. We assume perfect sensing by nodes for the sake of simplicity i.e., each node can accurately determine the spectrum holes. The nodes then send the list of channels available to a central node or to their neighbors during the information exchange period. In the spectrum allocation period, an algorithm is executed to determine the schedule and the schedule is distributed.

The data transmission period is split into $\mathcal{K}$ equal length time-slots. In a time-slot TS_k both the participating nodes transmit and receive data i.e., the time-slot is considered to be consisting of two sub-slots where one node transmits and the second receives in the first sub-slot, and in the second sub-slot, the receiver in the first sub slot transmits its data to its counterpart. The sub-slots may be of unequal length depending on the data that each node has to transmit to the other. This assumption reduces the complexity in the design, since each link has to be assigned with one time-slot. We also note here that we have assumed in the beginning that all the nodes have data to transmit to all other nodes.

4.1.2 Validation of the model

The validation of the model considered here is similar to the one described in Sec. 3.1.2 of Chap. 3. For readability purposes, the validation of the model is repeated here with the changes.

1. *Neglecting channel switch times:* In the model, the time of switching between channels has been neglected. The underlying MAC protocol should consider these times. Moreover the protocol must also take into account the “deaf period” during switching. However, since we are only proposing solutions for the allocation and not designing the protocol, it can be safely neglected.
2. *CCC:* We assume a CCC to simplify the implementation of the OSA network and allocation algorithms. This is a realistic assumption since an unlicensed band can be used for control information exchange. We assume a global CCC for edge-coloring heuristic and local or configurable CCCs for other heuristics. The discussion in Sec. 2.1.4 provides the arguments for validity of these assumptions.
3. *Time synchronization:* Since we propose a slotted-based scheme to access the spectrum, the nodes should have strict time synchronization. MAC Scheduling [35] uses GPS receivers for time synchronization. Another method that could be implemented using CCC with time synchronization is given in [29].

An advantage of such slotted scheme is the ability to address the issue of mobility since time synchronization and mobility are interlinked problems. It is possible to address issues of both node mobility and spectrum changes when nodes are time synchronized since a coordinated spectrum sensing by SUs can detect these changes.

4. *Perfect sensing:* In the current state of the art, nodes cannot independently and accurately determine the spectrum holes. However with cooperation with other nodes, it is possible to achieve higher accuracy. Another aspect is the intrusion of a PU during the data transmission period affecting the schedule, which cannot be controlled. However in this chapter, to simplify the problem we assume (a) accurate sensing (with or without cooperative sensing) and (b) PUs activities occur only at the beginning of period T .

In each slot TS_k , the communicating nodes negotiate on the sub-slot size and the order of access. The SUs can employ fragmentation and reassembly of the data to fit the size of available sub-slot.

4.1.3 Problem statement

Let a graph $G = (V, E)$ represent a CRAN. Let a valid minimum length schedule of G be L_s . It is given that the data transmission period of an SU frame contains $\mathcal{K}$ slots. If $L_s > \mathcal{K}$ then some edges do not get a slot to transmit. As shown before, computing a schedule for every frame with the heuristics described in the previous chapter may result in a few edges getting more access, while some edges may not get a slot at all. Such an allocation is termed *unfair*. The problem now is to get a schedule that is *fair* to all the edges. As stated earlier, fairness here means that all (transmitter, receiver) pair

should get the same percentage of access to the medium. To quantify fairness, we define *Fairness Index* (FI) for an edge as,

$$f_{(u,v)} = \frac{\sum A_{(u,v),j,k}}{n_{(u,v)}} \quad (u,v) \in E \text{ and } \forall j,k \quad (4.1)$$

where,

$$A_{(u,v),j,k} = \begin{cases} 1 & \text{if edge } (u,v) \text{ uses channel } j \text{ in time-slot } k \\ 0 & \text{otherwise} \end{cases}$$

as defined in Chap. 3 and $n_{(u,v)}$ is the total number of times edge (u,v) existed¹.

The fairness index indicates the fraction of times an edge was given access to the medium. If an edge uses an available free-slot for communication, this also has to be taken into account to calculate the fairness index. The problem now is to ensure that all edges get the same share.

4.1.4 $\mathcal{NP}$ completeness

The problem of allocation was proven to be $\mathcal{NP}$ complete in Sec. 3.1.4 of Chap. 3. Since we are still interested in computing a schedule taking fairness into account, the complexity of the problem increases in polynomial time i.e., out of all valid schedules select the schedule that has maximum fairness index.

This makes a polynomial time exact solution to the infeasible problem. Thus we propose heuristics, based on those from Chap. 3, to solve the problem.

4.2 Heuristics

It is shown in Sec. 4.1 that we cannot compute a schedule using the algorithms of Chap. 3, since the resulting schedule may be unfair to some edges. Moreover, free-slot usage have to be accounted too. This necessitates amendments to the existing heuristics. In the following sections, we modify the edge coloring heuristic, *Color Then Assign* (CTA) (Sec. 3.3.1, Chap. 3) and the *Clique based heuristic* (CBH) (Sec. 3.4.1, Chap. 3) to provide a fair schedule. However, we do not pursue with the Localized Heuristics (LH) (Sec. 3.4.2, Chap. 3) since LH is effective only when $L_s \leq \mathcal{K}$. Otherwise, an assignment scheme has to be devised similar to CBH making it quite similar to CBH (and hence it is of less interest).

4.2.1 Fair Color Then Assign

Intuition and Algorithm

In CBH we pick a node, color its edges and then proceed to another node. Here to ensure fairness, we need to pick the *right* nodes. To select the nodes, we define Node-Fairness

¹An edge exists between u and v if there exists at least one available channel between them to communicate

F , as given below.

$$F_u = \sum_{v \in \text{Neighbors}_u} f_{(u,v)} \quad (4.2)$$

The Node-Fairness indicates the fraction of access gained by its edges until the current frame. A lower number indicates that the edges of the node have starved, and number greater than or equal to 1 indicates that all the edges have at least received one slot whenever the edge existed.

In *Fair-Color Then Assign* (F-CTA) heuristic, we make use of this metric to select the nodes to assign colors. The Proc. 4.1 shows the procedure of assigning colors to edges of the nodes. A non-colored node u with the least F_u is taken (Line 2) first. Then of all the edges in u , the edge with the least fairness index is chosen and assigned an available slot (Lines 4 and 5). The channel assignment procedure, shown in Proc. 4.2, is similar to the channel assignment procedure of CTA i.e., we try to assign a channel that does not cause collisions in two-hop neighborhood. If a channel cannot be assigned then the edge is tried to be accommodated in another slot. If it is not possible then no slot is assigned to it. Instead, it is left open as a free-slot. The algorithm of F-CTA is shown in Alg. 4.3.

Procedure 4.1 Slot Assignment

```

1: while at least one node,  $u$ , is uncolored do
2:   Find a node  $u$  with the least  $F_u$  (as in Eqn. 4.2) in  $G$ 
3:   for  $v$  in  $\text{Neighbors}_u$  do
4:     Find  $(u, v)$  with the least  $f_{(u,v)}$  (as in Eqn. 4.1) in  $u$ 
5:      $slot \leftarrow$  first unused slot to edge  $(u, v)$  such that  $slot$  is unused in  $u$  and  $v$  AND
        $slot \leq \mathcal{K}$ 
6:     if  $slot$  exists then
7:       a.  $\text{Allocation}_{(u,v)}.slot \leftarrow slot$ 
8:       b. Remove the edge  $(u, v)$  in  $G$ 
9:     end if
10:  end for
11:  Mark  $u$  as colored
12: end while

```

To better understand how the fairness is achieved across edges, we consider the previous example shown again in Fig. 4.6. Here, the changes to the graph due to the PU activities at time instances $t = 0, T, 2T, 3T$ are shown. The number of slots $\mathcal{K}$ is 2.

At $t = 0$, all the nodes have the same Node-Fairness which is equal to zero. Hence the slot assignment procedure is executed sequentially. Edges (A, B) and (C, D) receive Slot 1, and edge (B, C) receives Slot 2. Executing the Channel Assignment procedure, a valid assignment is found as shown in Fig. 4.7. Due to PU activities, at $t = T$ only the edge (A, B) exists. The Node-Fairness before allocation is as shown in Fig. 4.8(a). The first slot is allocated to (A, B) while the second is used as a free-slot again by (A, B) . This is shown in Fig. 4.8. The allocation obtained from this procedure for $t = 2T, 3T$ and $4T$ is shown in Fig. 4.9, Fig. 4.10 and Fig. 4.11 respectively. It is assumed that the graph does not change after $3T$.

Procedure 4.2 Channel Assignment

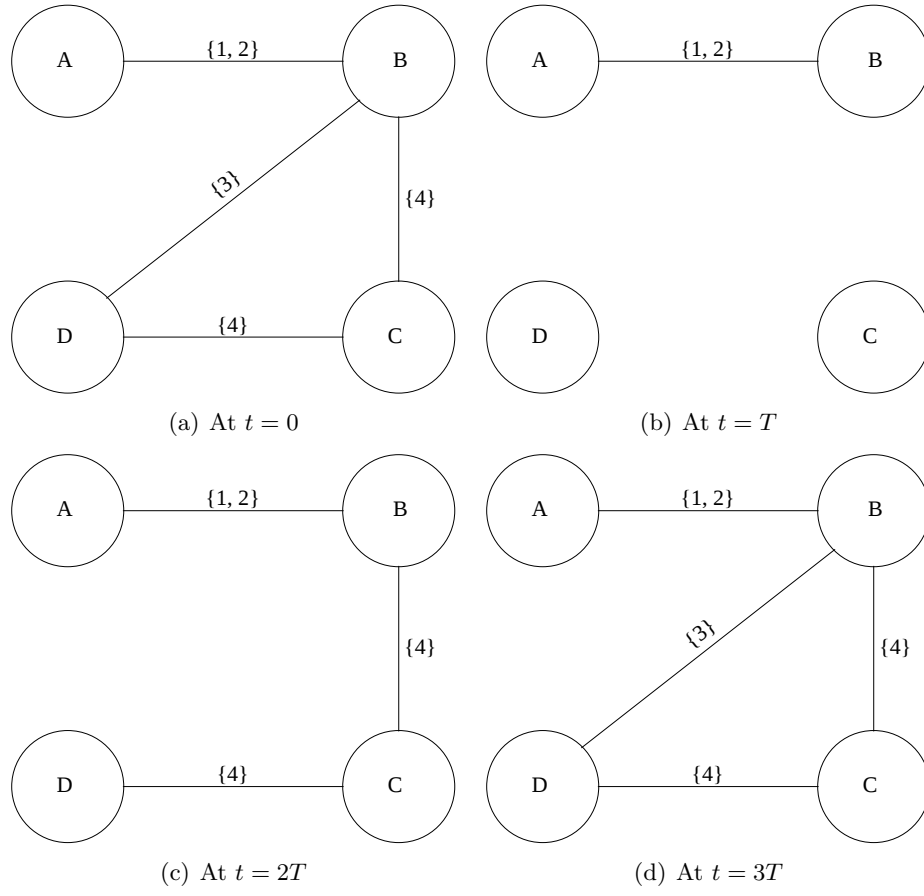
```

1:  $maxSlot \leftarrow$  maximum of  $Allocation_{(u,v)}.slot \forall u, v$ 
2:  $k \leftarrow 1$ 
3: while  $k \leq \mathcal{K}$  do
4:   for  $(u, v)$  in  $G$  such that  $Allocation_{(u,v)}.slot = k$  do
5:     for  $j$  in  $ChSet_{(u,v)}$  do
6:       if  $Allocation_{(u,v)}.channel \leftarrow j$  such that using  $j$  does not interfere in its 2-hop
         neighborhood then
7:          $Allocation_{(u,v)}.channel \leftarrow j$ 
8:       end if
9:     end for
10:    if no channel could be assigned then
11:      if if  $u$  and  $v$  are unused in slot  $k + 1$  AND  $Allocation_{(u,v)}.slot \leftarrow k + 1$  does
        not cause interference in 2-hop neighborhood then
12:         $Allocation_{(u,v)}.slot \leftarrow k + 1$ 
13:      end if
14:    end if
15:  end for
16:   $k \leftarrow k + 1$ 
17: end while

```

Algorithm 4.3 Fair Color Then Assign (F-CTA) algorithm

- 1: **Input:** Connectivity graph $G = (V, E)$, maximum number of slots K and the set of available channels $ChSet_{(u,v)}$ between each pair of vertices u, v
 - 2: **Output:** $Allocation$ - A valid allocation of $(timeslot, channel)$ for all edges
 - 3: Run Procedure 4.1
 - 4: Run Procedure 4.2
-


Figure 4.6: Snapshots of the graph at frame times $t = 0, T, 2T, 3T$

Node	Value	Slot 1		Slot 2	
A	0	Edge	Allocation	Edge	Allocation
B	0	(A, B)	(1, 1)	(B, C)	(2, 4)
C	0	(C, D)	(1, 4)		
D	0				

(a) Node-Fairness (b) Allocation

Figure 4.7: Allocation for Frame 1

Node	Value	Slot 1		Slot 2	
A	1	Edge	Allocation	Edge	Allocation
B	2/3	(A, B)	(1, 1)	(A, B)	(2, 1)
C	1				
D	1/2				

(a) Node-Fairness (b) Allocation

Figure 4.8: Allocation for Frame 2

Complexity

The worst-case time complexity for the slot assignment procedure $\mathcal{O}(N^2)$ and for the channel assignment procedure is $\mathcal{O}(MNK)$.

Node	Value	Slot 1		Slot 2	
A	$3/2$	Edge	Allocation	Edge	Allocation
B	1	(C, D)	$(1, 4)$	(B, C)	$(2, 4)$
C	1	(A, B)	$(1, 1)$		
D	$1/2$				
(a) Node-Fairness		(b) Allocation			

Figure 4.9: Allocation for Frame 3

Node	Value	Slot 1		Slot 2	
A	$4/3$	Edge	Allocation	Edge	Allocation
B	1	(B, D)	$(1, 3)$	(B, C)	$(2, 4)$
C	1				
D	$2/3$				
(a) Node-Fairness		(b) Allocation			

Figure 4.10: Allocation for Frame 4

Node	Value	Slot 1		Slot 2	
A	1	Edge	Allocation	Edge	Allocation
B	$8/9$	(A, B)	$(1, 1)$	(B, C)	$(2, 4)$
C	$2/3$	(C, D)	$(1, 4)$		
D	$4/5$				
(a) Node-Fairness		(b) Allocation			

Figure 4.11: Allocation for Frame 5

4.2.2 Fair Clique based heuristic

Intuition and Algorithm

In CBH, we formed a clique set, and the members of this set were made responsible for some of their edges. The members then allocated a slot and a channel to these edges. Conflicts can arise since the allocation was distributed. The conflicts are resolved either by changing a channel or by adding a new slot.

Here, we propose Fair-Clique based heuristic (F-CBH) based on CBH. The formation of the clique set is retained as before, which is shown in Proc. 4.4. The responsibility of the edges to the nodes also remains the same. However the Slot and Channel Assignment procedure, shown in Proc. 4.5, has the following change - the set of responsible edges for a node are sorted in ascending order of their fairness indices, and first $\mathcal{K}$ edges of the sorted list are assigned with a slot and a channel.

As in CBH, conflicts are detected when the schedules are distributed. Here the two-hop neighborhood schedule is learnt at every node. The procedure for conflict resolution is also the same - the node with least Node-ID proposes to find a new non-interfering channel if possible, or proposes to use a new slot. In F-CBH, if a new slot is being proposed, it is taken care that the new slot is less than $\mathcal{K}$. The other end-vertex receives the proposal, and also computes another valid allocation for this edge. In CBH, there are always two proposals possible, however in F-CBH there can be zero, one or two proposals. In case there are two, the best one i.e., one which has lower slot, is chosen. If

Procedure 4.4 Form Independent Set

```

1: At each node  $u$  do
2:  $minDegree \leftarrow$  minimum  $\delta_v \forall v$  in  $Neighbors_u$ 
3:  $minDegreeVertex \leftarrow v$  such that  $\delta_v = minDegree$ 
4: if  $\delta_u < minDegree$  then
5:    $I \leftarrow I \cup \{u\}$ 
6: else if  $\delta_u = minDegree$  and  $u < minDegreeVertex$  then
7:    $I \leftarrow I \cup \{u\}$ 
8: end if

```

Procedure 4.5 Slot and Channel Assignment

```

1: At each node  $u \notin I$  do
2:  $slot \leftarrow 1$ 
3: while at least one edge  $(u, v) \forall v \in Neighbors_u$  is unmarked do
4:   if  $NodeID_u < NodeID_v$  OR  $v \in I$  then
5:     Find  $f_{(u,w)}^{min}$  as in Eqn. 4.1 from the set of unmarked edges
6:      $slot \leftarrow$  first unused slot to edge  $(u, v)$  such that  $slot$  is unused in  $u$  and  $v$  AND
        $slot \leq K$ 
7:     if  $slot$  exists then
8:       a.  $Allocation_{(u,v)}.slot \leftarrow slot$ 
9:       b.  $Allocation_{(u,v)}.channel \leftarrow$  random channel from  $ChSet_{(u,v)}$ 
10:    end if
11:    Mark  $(u, w)$ 
12:  end if
13: end while

```

there is one proposal, it is chosen to resolve the conflict. In case there are none, then out of the interfering edges, the edge with lower fairness index retains the allocation and the other one loses. In case both have same fairness index, then the edge having the least node-id retains the allocation and the other edge loses. In this way, priority is given to the starving edges to ensure fairness.

Procedure 4.6 Conflict Resolution

```

1: At each node  $u$  do
2:   for  $v$  in  $Neighbors_u$  do
3:     if  $Allocation_{(u,v)}$  causes interference within 2-hop neighborhood and  $NodeID_u$  is
       the least among all end-vertices of interfering edges then
4:       Propose another channel  $j$  for  $(u, v)$  which results in valid assignment to  $v$ 
5:       if no such  $j$  exists then
6:         Propose (if possible) a new slot  $\leq K$  and channel for  $(u, v)$  that results in
           valid assignment
7:       end if
8:       Get the allocation from  $v$ 
9:       Distribute the allocation to 2-hop neighborhood
10:    else if  $Allocation_{(u,v)}$  causes interference within 2-hop neighborhood and
       $NodeID_v$  is the least among all end-vertices of interfering edges then
11:      Find another valid allocation (if possible)
12:      Receive proposals from  $v$ 
13:      if there is at least one alternative valid allocation proposal then
14:        Choose the better proposal of the two
15:      else
16:        Of the interfering edges, the edge having lower fairness index gets the priority -
          it retains the previous allocation and the other edge is divested of its allocation
17:      end if
18:      Send the allocation to  $v$ 
19:      Distribute the new allocation to 2-hop neighborhood
20:    end if
21:  end for

```

The algorithm of F-CBH is shown in Alg. 4.7.

Algorithm 4.7 Fair Clique Based Heuristic (F-CBH)

```

1: Input: the set of one hop neighbors for each node  $u$  and the set of available channels
   between each pair  $u, v$  such that  $v$  is a neighbor of  $u$ 
2: Output:  $Allocation_{(u,v),j,k}$ 
3:  $I \leftarrow \emptyset$ 
4: Run Procedure 4.4
5: Run Procedure 4.5
6: Distribute  $Allocation_u$  over two-hop neighborhood
7: Run Procedure 4.6

```

To understand better how the fairness is achieved across edges, we consider the previous example shown again in Fig. 4.6. Here, the changes to the graph due to the PU activities at time instances $t = 0, T, 2T, 3T$ are shown. The number of slots $\mathcal{K}$ is 2.

At $t = 0$, all the edges have the same fairness index equal to zero. The Nodes B and D

are in clique set after the execution of Proc. 4.4. Node B is responsible for edges (B, A) , (B, C) and (B, D) , and Node D is responsible for (D, C) . After assigning slots and channels, (B, A) is assigned (1,2), (B, C) is assigned (2,4), and (D, C) is assigned (1,4). Note there are no conflicts and the assignment is retained. This is shown in Fig. 4.12.

Node	Value	Slot 1		Slot 2	
(B, A)	0	Edge	Allocation	Edge	Allocation
(B, C)	0	(A, B)	(1, 2)	(B, C)	(2, 4)
(B, D)	0	(C, D)	(1, 4)		
(D, C)	0				
(a) Edge-Fairness		(b) Allocation			

Figure 4.12: Allocation for Frame 1

The fairness index is calculated at every node at $t = T$; the values are shown in Fig. 4.13(a). The clique set consists of only Node B, and the first slot is given to (A, B) . The second slot is used by (A, B) again but as a free-slot assignment. This is shown in Fig. 4.13.

Node	Value	Slot 1		Slot 2	
(B, A)	1	Edge	Allocation	Edge	Allocation
(B, C)	1	(A, B)	(1, 2)	(B, A)	(2, 1)
(B, D)	0				
(D, C)	1				
(a) Edge-Fairness		(b) Allocation			

Figure 4.13: Allocation for Frame 2

This procedure is repeated for further frames. The values of fairness-index and the assignment for the next three frames are shown in Fig. 4.14, Fig. 4.15 and Fig. 4.16.

Node	Value	Slot 1		Slot 2	
(B, A)	3/2	Edge	Allocation	Edge	Allocation
(B, C)	1	(B, C)	(1, 4)	(B, A)	(2, 1)
(B, D)	0			(C, D)	(2, 4)
(D, C)	1				
(a) Edge-Fairness		(b) Allocation			

Figure 4.14: Allocation for Frame 3

Node	Value	Slot 1		Slot 2	
(B, A)	4/3	Edge	Allocation	Edge	Allocation
(B, C)	1	(B, D)	(1, 3)	(B, C)	(2, 4)
(B, D)	0				
(D, C)	1				
(a) Edge-Fairness		(b) Allocation			

Figure 4.15: Allocation for Frame 4

Node	Value	Slot 1		Slot 2	
(B, A)	1	Edge	Allocation	Edge	Allocation
(B, C)	1	(B, D)	$(1, 3)$	(B, C)	$(2, 4)$
(B, D)	$1/2$	(b) Allocation			
(D, C)	$2/3$				
(a) Edge-Fairness					

Figure 4.16: Allocation for Frame 5

Complexity

The worst-case time complexity for Proc. 4.4 (Form Independent Set) is $\mathcal{O}(N)$, Proc. 4.5 (Slot and Channel Assignment) is $\mathcal{O}(MNK)$, and Proc. 4.6 (Conflict resolution) is $\mathcal{O}(N^2)$.

4.3 Results and Discussions

4.3.1 Simulation environment and Experiments setup

For simulations, we used the similar simulation environment described in Sec. 3.5.1. We implemented the two heuristics into the simulator, and verified its operations. The parameters for the simulation is the same as described in Sec. 3.5.1.

For comparison purposes, we use the scenarios as shown in Table. 4.1. These correspond to Scenarios 2 and 4 of Chap. 3. We considered only these two scenarios since there is no need to evaluate fairness when there are less number of nodes and fairly large number of available channels.

Scenario 1			Scenario 2		
Area	Nodes	Channels	Area	Nodes	Channels
100x100	10	1 - 3	200x200	10	1 - 3
200x200	40	1 - 3	200x200	40	1 - 3
300x300	90	1 - 3	200x200	90	1 - 3

Table 4.1: Simulation scenarios

In Scenario 1, the number of nodes is chosen to maintain the same density for different areas. In Scenario 2, the area is kept constant to create sparse and dense graphs. In both scenarios, the number of channels is very less to create a worst case condition for the evaluation of heuristics.

Both algorithms viz., Fair Color Then Assign (F-CTA) and Fair Clique Based Heuristic (F-CBH) are implemented. The algorithms are run for 20 iterations with 100 SU_FRAMES in the two scenarios described before. The average values are reported.

4.3.2 Fairness Index

Fig. 4.17 and Fig. 4.18 show the average fairness index per edge for the two scenarios. For 10, 40 and 90 nodes, it is seen that both the algorithms have similar performances.

In addition to the fairness index, the figures show the minimum fairness index computed by both algorithms. The minimum value of fairness index indicates how worse an edge of the graph is affected. Both algorithms show similar worst-hit edge for 40 and 90 nodes, but for 10 nodes the F-CTA performs slightly better. The maximum degree Δ (averaged over all iterations) is also marked in all the figures. The fairness index is approximately 1 when number of slots is equal to Δ . The variations in the graphs are due to the dynamic changes that occur due to PU activity.

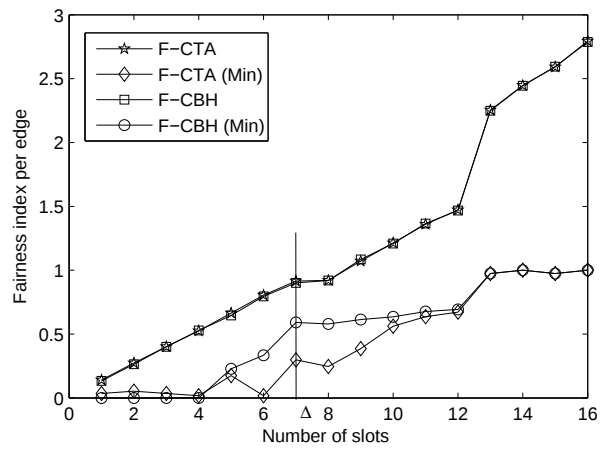
4.3.3 Discussions

We have compared the performances of the two heuristics: F-CTA and F-CBH over two scenarios where number of channels is less. Both of them show similar performances. However, it must be noted that F-CBH being a distributed heuristic performs as good as the centralized F-CTA.

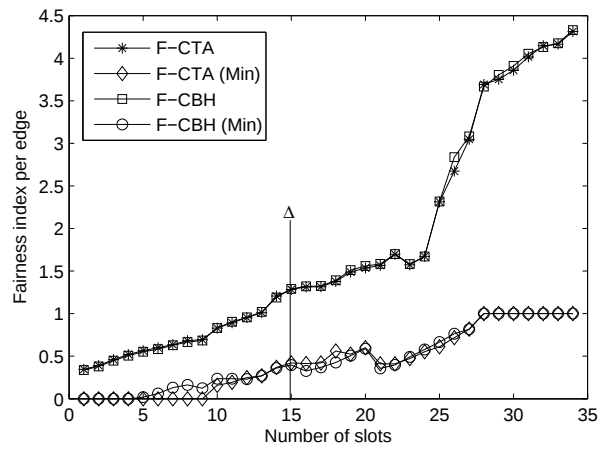
Fixing the number of slots in a frame depends on various parameters such as, length of the frame, number of nodes to accommodate per frame, amount of data to be transferred per link, data rate on the link, etc. To fix the number of slots in a frame, we can use the data presented in the previous section. If it is possible to know the maximum degree Δ of the topology, then we may fix the number of slots $\mathcal{K} \geq \Delta$. This can be seen from the Fig. 4.17 and Fig. 4.18. In these figures, it can be seen that having number of slots $\mathcal{K} = \Delta$, the fairness index approaches 1 and the minimum fairness index is not zero i.e., all the edges get at least 1 slot to transmit.

4.4 Summary

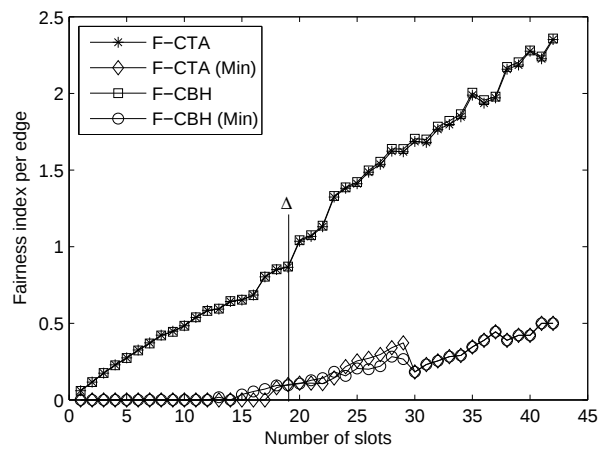
In this chapter we presented the problem of computing a fair schedule when the number of slots is fixed in a frame. Based on the heuristics of the previous chapter, we then proposed two heuristics F-CTA and F-CBH to compute fair schedules. Their performances are evaluated. It was also shown that the minimum number of slots in a frame should be at least the maximum degree, so that the edges can have fair amount of access to the medium.



(a) 10 nodes

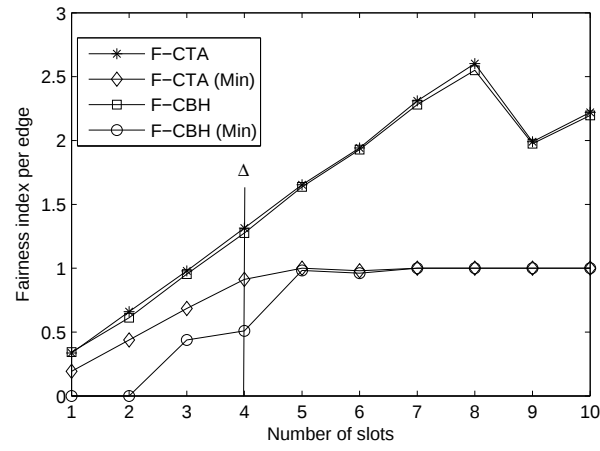


(b) 40 nodes

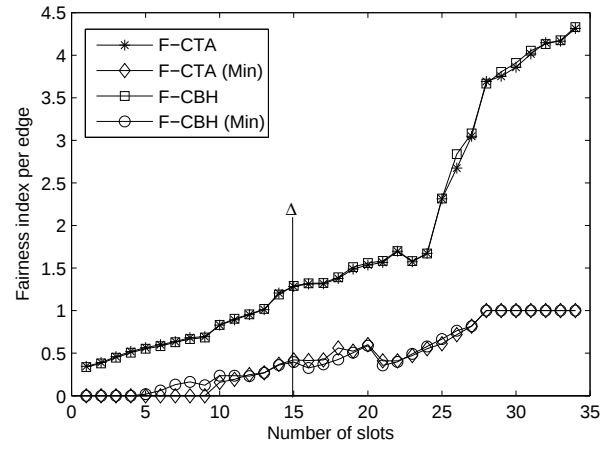


(c) 90 nodes

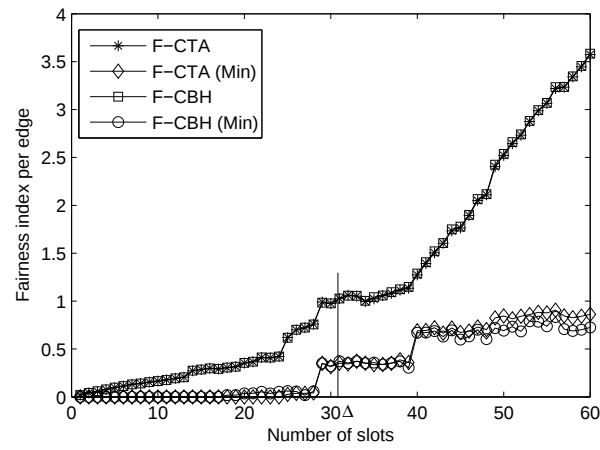
Figure 4.17: Scenario 1



(a) 10 nodes



(b) 40 nodes



(c) 90 nodes

Figure 4.18: Scenario 2

CHAPTER 5

Spectrum Sensing

In the previous chapters, we assumed the nodes can detect the spectrum holes accurately. However, in reality it is not the case. Nodes may be unable to detect PU transmissions due to fading, and shadowing. This error reduces the chances of detecting a transmission and also may misdetect a channel being vacant. This will cause interference to the PU transmissions. These errors are undesirable. Further, it must be noted that the nodes cannot sense all the channels for opportunity in every frame due to time limitations and power constraints. Hence few channels must be selected, whose probability of being available are seen to be higher to increase the opportunities. In this chapter, we investigate the Markov model proposed in [27] for improvements that can be obtained from a joint spectrum-sensing allocation policy. We use this model with our allocation scheme to compare the improvements.

5.1 System model

The system model is similar to the one in Sec. 4.1.1, Chap. 4.

We consider a multi-hop CR network formed by N nodes, each with a unique node-id $\in \{1, 2, \dots, N\}$. These nodes are equipped with a half-duplex software defined transceiver i.e., each node can either transmit or receive at any instant of time on a specified channel. These nodes are interested in point-to-point communication, i.e., we consider only 1-hop communication. We assume that the spectrum is divided into M orthogonal channels that are symmetric. Each channel has a unique number in $\{1, 2, \dots, M\}$. Further, we assume that a common control channel (CCC) is available to exchange the control messages amongst the nodes. The nodes are time synchronized either through the common control channel or with the help of a GPS receiver. We assume that all nodes have data to transmit to all its neighbors to ensure that we consider the worst case scenario of the problem.

Consider the MAC super frame as shown in Fig. 5.1. The super frame is divided into sensing, information exchange, allocation and data transmission periods. In the sensing period, the channels that are free of PUs, are found first.

We assume that the sensing by nodes has an error of p_{se} i.e., each node cannot accurately determine the spectrum holes. We do not assume a cooperative method to improve the sensing. Further, the short sensing period constrains the number of channels that can be

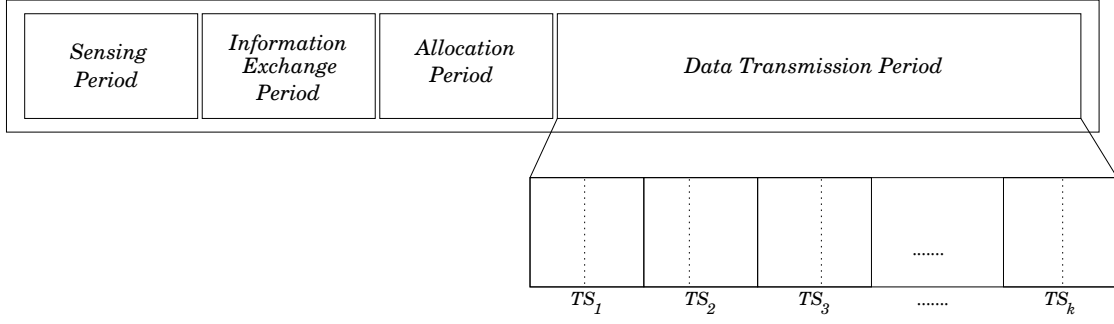


Figure 5.1: MAC super frame

sensed. The nodes then send the list of channels available to a central node or to their neighbors during the information exchange period. In the spectrum allocation period, an algorithm is executed to determine the schedule and the schedule is distributed.

The data transmission period is split into $\mathcal{K}$ equal length time-slots. In a time-slot TS_k both the participating nodes transmit and receive data i.e., the time-slot is considered to be consisting of two sub-slots where one node transmits and the second receives in the first sub-slot, and in the second sub-slot, the receiver in the first sub slot transmits its data to its counterpart. The sub-slots may be of unequal lengths depending on the data that each node has to transmit to the other. This assumption reduces the complexity in the design, since each link has to be assigned with one time-slot. We also note here that we have assumed in the beginning that all the nodes have data to be transmitted to all the other nodes.

5.2 Sensing

Spectrum sensing has been mostly addressed from the PHY layer (for ex: [27, 5, 4]) perspective. In detection of available channels, the decision is taken from the binary hypotheses test:

$\mathcal{H}_0$ (null hypothesis indicating that the sensed channel is available)

$\mathcal{H}_1$ (otherwise)

If the sensor misjudges a $\mathcal{H}_0$ for $\mathcal{H}_1$, it raises a false alarm, thereby wasting the opportunity. On the other hand, if the detector mistakes $\mathcal{H}_1$ for $\mathcal{H}_0$, a mis-detection results, causing the SU transmission collide with the PU transmissions.

In DC-MAC [27], a Partially Observed Markov Decision Process (POMDP) framework is defined to reduce the sensing errors based on previous data, wherein for M channels, 2^M states are defined with their transition probabilities. This framework is computationally inefficient. A greedy method is also described in [27] based on a simple Markov chain. The Markov chain representation is shown in Fig. 5.2.

The probability that a channel is available is calculated as shown in Eqn. 5.1.

$$(\omega_a \beta_a + (1 - \omega_a) \alpha_a) B_a, \quad (5.1)$$

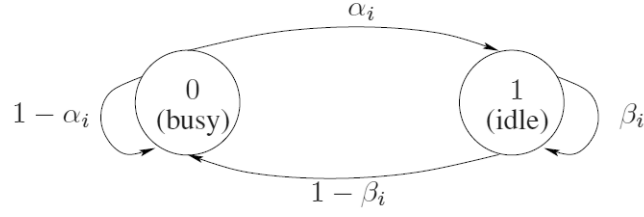


Figure 5.2: Markov Chain model

where ω_a is the probability (conditioned on the sensing and history) that channel a is available at the beginning of a slot. B_a is the reward given to the channel if the channel was successfully used for transmission. The greedy approach tries to maximize the immediate reward for a time-slot t . This can be represented as,

$$\arg \max_M X$$

We propose a simple modification to this method - we try to predict. At every node, a history of decisions for every channel is maintained. Let the probability of correct decisions made earlier, for the channel m , be p_m^{corr} . The probability that channel m is available is

$$X = (\beta_m + \alpha_m),$$

We predict the availability of a channel as $D_m^{\text{curr}} = zX + (1 - z)p_m^{\text{corr}}$. Here, z is the smoothing factor. If the probability D_m^{curr} is greater than a threshold Υ , then the channel is said to be available. This prediction method takes into account of the previous decisions made.

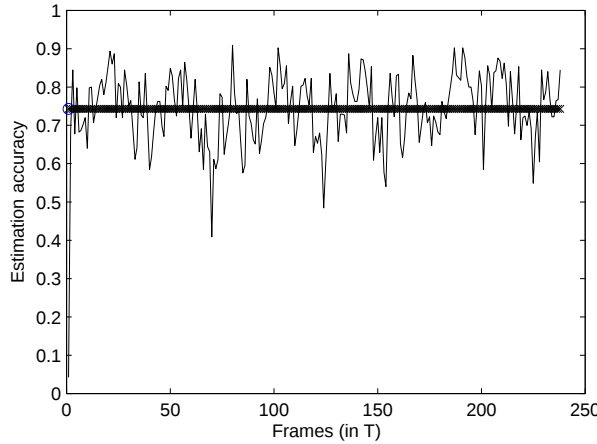


Figure 5.3: Estimation accuracy

We simulated our CRANs for a sensing error of $p_{se} = 0.1$ and using the above scheme to find the performance. The simulator and its parameters are the same as described in Sec. 4.3.1, Chap 4. Fig 5.3 shows the estimation accuracy of this scheme over many frames. Initial frames make incorrect decision due to lack of history. As the pr The

mean is indicated by the horizontal line, which is around 0.75. This shows the prediction works reasonably well.

5.2.1 Constrained Sensing

In a frame, it is not possible to sense all the channels for opportunity detection, due to the channel switching times, time is limited for sensing. It is possible to sense only w ($1 \leq w \leq M$) channels depending on the hardware technology. It is therefore necessary to pick the channels with the highest probability of availability. Using the prediction scheme it is possible to pick the best channels w to detect the opportunity. However, even if we sense w channels and find they are available, we are ignoring the other $M - w$ channels out of which few may be available.

To improve on this aspect, we propose a channel selection strategy. The procedure for selection is shown in Proc. 5.1. We define another threshold $\Upsilon_1 > \Upsilon$. If $D_m^{\text{curr}} \geq \Upsilon_1$ then we can say with high confidence that a channel is available. We mark those channels as available. If $X \geq \Upsilon$ then we mark these as potential channels for sensing; w out of these are selected for sensing. In the Proc. 5.1, If no channel was marked, then w channels are selected randomly for sensing.

Procedure 5.1 Channel selection procedure.

- 1: Define $\Upsilon_1 > \Upsilon$ as a threshold above which the channel has high probability of correct estimation
 - 2: For all M channels, compute D_m^{curr}
 - 3: **for** $\forall m \in M$ **do**
 - 4: **if** $D_m^{\text{curr}} \geq \Upsilon_1$ **then**
 - 5: Mark state of m as known
 - 6: If $d_m^{\text{prev}} = 1$ then m is used for transmission
 - 7: **else if** $D_m^{\text{curr}} \geq \Upsilon$ AND $d_m^{\text{prev}} = 1$ **then**
 - 8: mark m for sensing
 - 9: **end if**
 - 10: **end for**
-

Using the same simulation environment, we simulated random topologies of 40 nodes in an area of $200 \times 200 m^2$. M was fixed at 10 and w at 2. We computed the utilization per edge from the CTA algorithm of Chap. 3 to see the performance.

Fig. 5.4 shows the utilization per edge from the CTA algorithm of Chap. 3. Without a selection strategy, the utilization is less as expected. When the Proc. 5.1 was executed, the average number of available channels was $\tilde{3}.2$, in contrast to 2 when no strategy is used. This increased the performance of the system. Of course, the utilization is by far less than the utilization shown in Chap. 3.

This shows that a joint spectrum sensing-allocation policy can be very beneficial. However deeper analysis is required to quantify the benefits and the cases where this scheme fails. Also, more experiments are required to confirm the operation of the scheme.

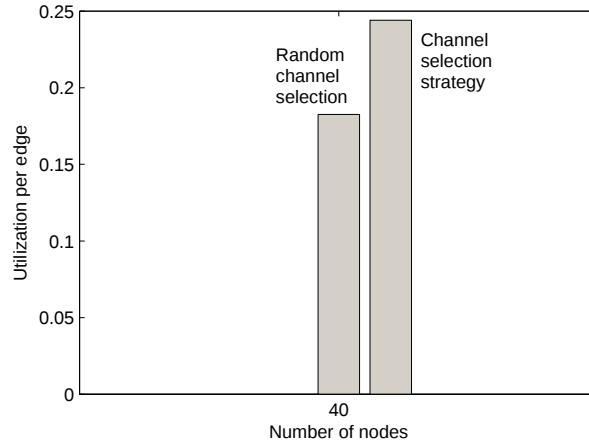


Figure 5.4: Utilization with channel selection strategy

5.3 Summary

In this chapter, we looked into the joint spectrum sensing-allocation scheme based on the proposal in DC-MAC [27]. We modified the scheme of DC-MAC and presented the estimation accuracy of the system. We also devised a simple channel-selection strategy for constrained sensing. It is shown that the spectrum utilization increases with such a strategy. However, more insights are required for better understanding and for proposing better schemes.

CHAPTER 6

Conclusions

6.1 Conclusions

The artificial scarcity of the spectrum has been the object of research in the recent past. Cognitive Radio, Dynamic spectrum access and Opportunistic spectrum access have all been proposed. However, in an ad hoc setting, there are many issues to be solved. One among them is to negotiate and share the available spectrum. This needs cooperation amongst the nodes and some intelligent ways of sharing the resources so as to increase the usage of spectrum that is available. Indeed the spectrum is scarce (though artificial) but the methods to share them should not be inefficient.

We addressed the allocation of the spectrum in a Cognitive Radio Ad hoc Networks (CRANs) setting. We first investigated into the various issues associated in such a setting. We addressed them in the order of their complexity to provide a better roadmap for CRANs to share and efficiently use the spectrum. We list the main contributions of our work here in sequel as it appears in this thesis.

6.1.1 Allocation of opportunistic spectrum

The first issue we were challenged with was to analyze the problem of spectrum allocation from Cognitive Radio Ad hoc Networks perspective. In Chap. 3, we analyzed, proposed and evaluated the heuristics against the optimal and the one in literature. Following are the conclusions from Chap. 3.

1. The problem of allocation of spectrum for CRANs is $\mathcal{NP}$ complete.
2. The number of channels required for a time-optimal allocation is $\lceil N/2 \rceil$ where N is the number of nodes.
3. The minimum number of slots required in any graph G with any number of channels is $\chi'(G)$.
4. Optimal use of resources can only be done in a complete graph $G = K_n$.
5. Distributed heuristics, with parallel allocation were proposed. The Clique based heuristic is novel and is proposed for the first time for link scheduling (in wireless sensor networks or in multichannel MAC).

6. Devising a schedule offline, and then applying Localized Heuristics will have better performance than computing a schedule every frame.
7. The proposed heuristics produce schedule lengths close to the optimal ILP results.
8. A tight bound on the length of schedules produced by the heuristics was determined.
9. The number of message exchanges by the distributed heuristics are higher than a centralized scheme. It is also shown that the proposed distributed heuristics use lesser number of messages than the one in the literature.

6.1.2 Fairness

The next challenge was to constrain the system with fixed number of slots, to get the system closer to real implementation. The issue of fairness arises. This made the system challenging. We proposed heuristics for this problem in Chap. 4. Following are the conclusions from the chapter.

1. If the number of slots are fixed in a frame, then the heuristics have to be adapted to ensure fair access to every link.
2. With respect to sharing the resources distributed heuristic has same performance as that of the centralized one.
3. It is advisable to fix the number of slots in a frame to at least the maximum possible degree of the topology, to ensure fair access.

6.1.3 Imperfect sensing

In a real-life scenario, individual nodes cannot sense the opportunities perfectly. We add this constraint to the previous system. In Chap. 5, to get the system closer to implementation, we also add the constraint of limiting the number of channels that can be sensed. Following are the conclusions from the chapter.

1. A Markov Chain based link layer sensing-allocation policy is investigated.
2. A joint sensing-allocation policy increases the spectrum utilization.
3. Given a constraint on number of channels that can be sensed in a frame, a good channel state prediction or estimation strategy is required to enhance the spectrum efficiency.

We started from an allocation problem where we find a minimum length valid schedule. We show that the heuristics can perform almost as good as the optimal ones. We assumed perfect sensing and that the number of slots in a frame is determined based on the topology. However, it is preferable to have fixed number of slots per frame. When we chose to do so, the scheduling algorithms resulted in unfair schedules. Hence, we devised algorithms for fair scheduling. However, the perfect sensing of opportunities was

retained. To move closer to the practical scenarios, we relaxed this constraint and further added a constraint of limited number of channels that can be sensed (indeed a node can not sense all the frequency bands). A simple modification to an existing scheme was proposed. A channel selection strategy based on this modification was also proposed. While we move closer to the end of this thesis, we can fairly claim that we have addressed almost the near real-life scenarios of CRANs and the assumptions we have retained are that of a real system. We have devised time-slotted heuristics that perform close to the optimal values, which are fair and work in conditions where spectrum sensing is not perfect.

6.2 Recommendations for future work

As we draw curtain to this thesis, we are still challenged with many issues. Of course we do not claim to have solved all the problems associated with allocation problems in CRANs. Thus we list some of the immediate tasks that are necessarily to be addressed in CRANs.

The work on joint sensing-allocation policy is an interesting problem to solve. Efficient prediction of channel state is important for detecting opportunities. However, existing schemes [21] are computationally complex and cannot be applied in real-time. Cooperative and non-cooperative methods can be looked into to increase the spectrum efficiency. Auctioning and game-theoretic approaches are interesting and yet challenging to be used here.

While we looked at unweighted graphs, it is required to consider weighted graphs for allocation. Weights on the links are usually an indicator to the Quality-of-Service (QoS) that can be obtained on that particular link [41]. In these graphs, allocation is again $\mathcal{NP}$ complete. Distributed heuristics are required to ensure fairness and maximum utilization of the spectrum. In the literature Zheng [41] discusses some of the utility functions. Their scheme is distributed however it is not computationally efficient. A distributed heuristic with QoS constraints is, therefore, an interesting problem and helps in analyzing the possible QoS guarantees in CRANs.

We assumed logical channels. However, in reality, the spectrum it is not so. An important allocation problem thus is, how wide a spectrum-band a particular node should use, and at what center-frequency? For how long it should be used [38] ?. This question has several real-world connotations and problems, that can be divided into sub-problems. This needs a deeper research to realize Cognitive Radio Ad hoc Networks of future.

Publications

Following are the publications from this thesis work:

1. Vijay Sathyanarayana Rao, R Venkatesha Prasad, Chetan Yadati and I.G.M.M. Niemegeers. Allocation of Opportunistic Spectrum in Cognitive Radio Ad hoc Networks. *Accepted to appear in 2010 IEEE Consumer Communications and Networking Conference - Wireless Home Communication and Networking*, Jan 2010.
2. Vijay Sathyanarayana Rao, R Venkatesha Prasad, Chetan Yadati and I.G.M.M. Niemegeers. Algorithms for distributed spectrum allocation for opportunistic ad hoc networks. *Submitted to IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN) 2010*.
3. Vijay Sathyanarayana Rao, R Venkatesha Prasad, and I.G.M.M. Niemegeers. Cognitive Radio Ad hoc Networks - a survey. *In preparation*.

Bibliography

- [1] ILOG CPLEX Solver 11, IBM Inc. <http://www.ilog.com/products/cplex/>.
- [2] FCC, Notice of proposed rule making and order, FCC ET Docket No 03-222, December 2003.
- [3] DARPA xG Program. http://www.darpa.mil/STO/Solicitations/WAND/pdf/XG_overview_for_WAND.pdf, 2007.
- [4] Ian F. Akyildiz, Won-Yeol Lee, and Kaushik R. Chowdhury. CRAHNs: Cognitive radio ad hoc networks. *Ad Hoc Networks*, vol. 7(5):pp. 810 – 836, 2009.
- [5] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, vol. 50(13):pp. 2127 – 2159, 2006.
- [6] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks*, vol. 47(4):pp. 445–487, 2005.
- [7] J. A. Bondy and U. S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer London, 2008.
- [8] V. Brik, E. Rozner, S. Banerjee, and P. Bahl. DSAP: a protocol for coordinated spectrum access. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005*, pages 611–614, Nov. 2005.
- [9] M.M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. DIMSUMnet: new directions in wireless networking using coordinated dynamic spectrum. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005*, pages 78–85, June 2005.
- [10] Lili Cao and Haitao Zheng. Distributed spectrum allocation via local bargaining. In *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks. IEEE SECON 2005.*, pages 475–486, Sept., 2005.
- [11] Tao Chen, Honggang Zhang, G.M. Maggio, and I. Chlamtac. Topology Management in CogMesh: A Cluster-Based Cognitive Radio Mesh Network. In *IEEE International Conference on Communications, ICC '07.*, pages 6516–6521, June 2007.
- [12] C. Cordeiro and K. Challapali. C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks. In *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007.*, pages 147–157, 2007.

-
- [13] Carlos Cordeiro, Kiran Challapali, Dagnachew Birru, and Sai Shankar N. IEEE 802.22: An Introduction to the First Wireless Standard based on Cognitive Radios. *Journal of Communications*, vol. 1(1), 2006.
 - [14] FCC. Cognitive radio for public safety. <http://www.fcc.gov/pshs/techttopics/techtopic8.html>.
 - [15] S. Gandham, M. Dawande, and R. Prakash. Link scheduling in sensor networks: distributed edge coloring revisited. In *Proceedings of IEEE INFOCOM 2005.*, volume 4, pages 2492–2501 vol. 4, March 2005.
 - [16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP Completeness*. W. H. Freeman and Co., 1979.
 - [17] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
 - [18] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, Feb. 2005.
 - [19] Juncheng Jia, Qian Zhang, and Xuemin Shen. HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management. *IEEE Journal on Selected Areas in Communications*, vol. 26(1):106–117, Jan. 2008.
 - [20] I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey. *IEEE Personal Communications*, 3(3):10–31, Jun 1996.
 - [21] Hyoil Kim and Kang G. Shin. Efficient discovery of spectrum opportunities with mac-layer sensing in cognitive radio networks. *IEEE Transactions on Mobile Computing*, 7(5):533–545, 2008.
 - [22] T. Vamsi Krishna and Amitabha Das. A survey on MAC protocols in OSA networks. *Computer Networks*, vol. 53(9):pp. 1377 – 1394, 2009.
 - [23] L. Ma, X. Han, and C. C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005*, pages 203–213, 2005.
 - [24] D. Maldonado, B. Le, A. Hugine, T.W. Rondeau, and C.W. Bostian. Cognitive radio applications to dynamic spectrum allocation: a discussion and an illustrative example. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005.*, pages 597–600, Nov. 2005.
 - [25] N. Nie and C. Comaniciu. Adaptive channel allocation spectrum etiquette for cognitive radio networks. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks. DySPAN 2005*, pages 269–278, Nov. 2005.
 - [26] P. Pawelczak, R. Venkatesha Prasad, L. Xia, and I.G.M.M. Niemegeers. Cognitive radio emergency networks - requirements and design. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005.*, pages 601–606, Nov. 2005.

- [27] Zhao Qing, Tong Lang, Swami Ananthram, and Chen Yunxia. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. *IEEE Journal on Selected Areas in Communications*, 25(3):589–600, 2007.
- [28] S. Ramanathan and E.L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking*, 1(2):166–177, Apr 1993.
- [29] Kay Römer. Time synchronization in ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 173–182, New York, NY, USA, 2001.
- [30] S. Sankaranarayanan, P. Papadimitratos, A. Mishra, and S. Hershey. A bandwidth sharing approach to improve licensed spectrum utilization. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks. DySPAN 2005*, pages 279–288, Nov. 2005.
- [31] Tao Shu, Shuguang Cui, and M. Krunz. Medium Access Control for Multi-Channel Parallel Transmission in Cognitive Radio Networks. In *IEEE Global Telecommunications Conference, GLOBECOM '06.*, pages 1–5, Dec. 2006.
- [32] Jungmin So and Nitin H. Vaidya. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 222–233, New York, NY, USA, 2004. ACM.
- [33] H. Tamura, K. Watanabe, M. Sengoku, and S. Shinoda. On a new edge coloring related to multihop wireless networks. In *Asia-Pacific Conference on Circuits and Systems, 2002. APCCAS '02. 2002*, volume 2, pages 357–360 vol.2, 2002.
- [34] R.W. Thomas, L.A. DaSilva, and A.B. MacKenzie. Cognitive networks. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005.*, pages 352–360, Nov. 2005.
- [35] Mansi Thoppian, S. Venkatesan, Ravi Prakash, and R. Chandrasekaran. MAC-Layer Scheduling in Cognitive Radio based Multi-Hop Wireless Networks. In *WOWMOM '06: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 191–202, Washington, DC, USA, 2006. IEEE Computer Society.
- [36] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. Primary user behavior in cellular networks and implications for dynamic spectrum access - [Topics in radio communications]. *IEEE Communications Magazine*, 47(3):88–95, 2009.
- [37] Lin Xu, R. Tonjes, T. Paila, W. Hansmann, M. Frank, and M. Albrecht. Driving to the internet: Dynamic radio for ip services in vehicular environments. In *Proceedings of 25th Annual IEEE Conference on Local Computer Networks, LCN 2000.*, pages 281–289, 2000.
- [38] Yuan Yuan, Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, and Yunnan Wu. Allocating dynamic time-spectrum blocks in cognitive radio networks. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 130–139, New York, NY, USA, 2007. ACM.

- [39] Jun Zhao, Haitao Zheng, and Guang-Hua Yang. Distributed coordination in dynamic spectrum allocation networks. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pages 259–268, Nov. 2005.
- [40] Qing Zhao and B.M. Sadler. A Survey of Dynamic Spectrum Access. *IEEE Signal Processing Magazine*, vol. 24(3):79–89, May 2007.
- [41] Haitao Zheng and Chunyi Peng. Collaboration and fairness in opportunistic spectrum access. In *IEEE International Conference on Communications, ICC 2005*, volume 5, pages 3132–3136 Vol. 5, May 2005.
- [42] Chao Zou and Chunxiao Chigan. A game theoretic dsa-driven mac framework for cognitive radio networks. In *IEEE International Conference on Communications. ICC '08.*, pages 4165–4169, May 2008.