IMPROVING SPECTRUM EFFICIENCY IN HETEROGENEOUS WIRELESS NETWORKS

By

Chin-Jung Liu

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2018

ABSTRACT

IMPROVING SPECTRUM EFFICIENCY IN HETEROGENEOUS WIRELESS NETWORKS

By

Chin-Jung Liu

Over the past decades, the bandwidth-intensive applications that are previously confined to wired networks are now migrating to wireless networks. This trend has brought unprecedented high demand for wireless bandwidth. The wireless traffic is destined to dominate the Internet traffic in the future, but many of the popular wireless spectrum bands, especially the cellular and ISM bands, are already congested. On the other hand, some other wireless technologies, such as TV bands, often do not fully utilize their spectrum. However, the spectrum allocation is tightly regulated by the authority and adjusting the allocation is extremely difficult. The uneven utilization and the rigid regulation have led to the proposal of heterogeneous wireless networks, including cognitive radio networks (CRN) and heterogeneous cellular networks (HetNet). The CRNs that usually operate on different technologies from the spectrum owner attempt to reuse the idle spectrum (i.e., white space) from the owner, while HetNets attempt to improve spectrum utilization by smallcells. This dissertation addresses some of the challenging problems in these heterogeneous wireless networks.

In CRNs, the secondary users (SU) are allowed to access the white spaces opportunistically as long as the SUs do not interfere with the primary users (PU, i.e., the spectrum owner). The CRN provides a promising means to improve spectral efficiency, which also introduces a set of new research challenges. We identify and discuss two problems in CRNs, namely non-contiguous control channel establishment and k-protected routing protocol design. The first problem deals with the need from SUs for a channel to transfer control information. Most existing approaches are channel-hopping (CH) based, which is inapplicable to NC-OFDM. We propose an efficient method for guaranteed NC-OFDM-based control channel establishment by utilizing short pulses on OFDM subcarriers. The results show that the time needed for establishing control channel is lower than that of CH-based approaches. The second problem deals with the interruption to a routing path

in a CRN when a PU becomes active again. Existing reactive approaches that try to seek for an alternative route after PU returns suffer from potential long delay and possible interruption if an alternative cannot be found. We propose a k-protected routing protocol that builds routing paths with preassigned backups that are guaranteed to sustain from k returning PUs without being interrupted. Our result shows that the k-protected routing paths are never interrupted even when k PUs return, and have significantly shorter backup activation delays.

HetNets formed by smallcells with different sizes of coverage and macrocells have been proposed to satisfy increased bandwidth demand with the limited and crowded wireless spectrum. Since the smallcells and macrocells operate on the same frequency, interference becomes a critical issue. Detecting and mitigating interference are two of the challenges introduced by HetNets. We first study the interference identification problem. Existing interference identification approaches often regard more cells as interferers than necessary. We propose to identify interference by analyzing the received patterns observed by the mobile stations. The result shows that our approach identifies all true interferers and excludes most non-interfering cells. The second research problem in HetNets is to provide effective solutions to mitigate the interference. The interference mitigation approaches in the literature mainly try to avoid interference, such as resource isolation that leads to significantly fewer resources, or power control that sacrifices signal quality and coverage. Instead of conservatively avoiding interference, we propose to mitigate the interference by precanceling the interfering signals from known interferers. With precancellation, the same set of resources can be shared between cells and thus throughput is improved.

This dissertation addresses several challenges in heterogeneous wireless networks, including CRNs and HetNets. The proposed non-contiguous control channel protocol and k-protected routing protocol for CRNs can significantly improve the feasibility of CRNs in future wireless network applications. The proposed interference identification and interference precancellation approaches can effectively mitigate the interference and improve the throughput and spectrum utilization in HetNets. This dissertation aims at breaking the barriers for supporting heterogeneous wireless networks to improve the utilization of the precious and limited wireless spectrum.

Copyright by
CHIN-JUNG LIU
2018

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude and respect to Prof. Li Xiao, who believed in me from my application credentials and accepted to become my advisor. Dr. Xiao guided my research, often with extra patience, helped me to overcome challenges, and encouraged me to become an independent researcher. Without Dr. Xiao's guidance and support, I would not be able to complete this dissertation. My gratitude is also devoted to my review committee members Prof. Abdol-Hossein Esfahanian, Prof. Guoliang Xing and Prof. Tien-Yien Li for their support and guidance.

My deepest appreciation goes to my parents, Mr. Jung-Hua Liu and Mrs. Su-Chu Chen, who supported me unconditionally. I owe my parents everything. Thank you for encouraging me in the pursuits of my dream and thank you for your sacrifices.

I am also grateful to every member of the ELANS lab, especially Dr. Pei Huang, Dr. Chen Qiu, Dr. Kanthakumar Pongaliur, Dr. Sayeed Hyder, Mrs. Xi Yang and Mr. Masoud Zarifnashat. Particularly, I must thank Dr. Pei Huang for being an excellent labmate, an even better friend, and to some extent, a mentor. Dr. Huang has spent a significant amount of time to discuss my research with me and has contributed to this dissertation with an impact.

Finally, I would like to thank everyone who directly or indirectly offered his or her help to this dissertation. My sincerest thanks are extended to all faculty members of the Department of Computer Science and Engineering at Michigan State University (MSU), for enriching my knowledge in the field. Also, I would like to give my special thanks to Prof. Chung-Ta King at National Tsing Hua University. Prof. King inspired me to explore the world of computer science, and he supported the start of my journey here at MSU. Special thanks should also be given to my dear friends: Chien-Wei Chang, Yu-Kai Huang, and Shih-Wen Su. I appreciate them for sharing my happiness, sadness, and my growth, even when I am thousands of miles away. As for any remaining errors or deficiencies, the responsibility for this work rests entirely upon the author.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

With the rapid advancement of wireless technologies in the past few decades, the bandwidth-intensive applications that are previously confined to wired networks are migrating to wireless networks [81, 91]. These bandwidth-intensive applications have brought unprecedented high demand for wireless bandwidth. The wireless traffic is destined to dominate the Internet traffic in the future, but many of the popular wireless spectrum bands, especially the cellular bands and the industrial, scientific and medical radio bands (ISM bands), are already congested. Various techniques have been proposed and have pushed the capacity of a spectrum band close to the theoretical upper bound according to Shannon-Hartley Theorem [106]. Increasing the spectrum bandwidth is the last resort to improve the capacity. However, the wireless spectrum is a finite and increasingly precious resource that is carefully regulated by the administrative authority.

The administrative authority partitions the wireless spectrum into various spectrum bands. Each licensed spectrum band is allocated to a specific usage, such as the cellular bands and TV bands. In general, there are mainly two spectrum access policies: unlicensed and licensed. The owner technology of the licensed spectrum band has the exclusive access to the band. Since different technologies cannot access the same licensed spectrum at the same time in the same area, the interference is minimized. Contrary to licensed spectrum bands, the access of unlicensed spectrum bands, such as the ISM bands, is less restrictive. Any device can access the spectrum in a contention manner. However, the static spectrum allocation does not consider spatial and temporal variations of spectrum utilization. The unlicensed spectrum bands have become more congested [5] than ever before, while many of the licensed spectrum bands are severely underutilized. Many measurement reports [31, 124] have revealed that a significant amount of idle spectrum (i.e., white spaces, spectrum holes) found in licensed spectrum bands, such as TV bands. The uneven utilization and the rigid regulation have led to the proposal of heterogeneous wireless networks, including cognitive radio networks (CRN) and heterogeneous cellular networks (HetNet).

1.1 Cognitive Radio Networks

In CRNs, the secondary users (SU) that operate on different technologies and protocols from the primary users (PU, i.e., the spectrum owner), are allowed to access the white spaces opportunistically as long as the SUs do not interfere with the PUs. The dynamic spectrum access (DSA) is proposed to let unlicensed users exploit the white spaces in the licensed spectrum opportunistically [29, 30, 38]. The DSA is realized by the cognitive radio (CR) technology, where the devices can adjust operating parameters autonomously. The CRNs provide a promising means to improve spectral efficiency, which also introduces a set of new research challenges. The fifth generation (5G) cellular networks also envision DSA as a possible resort for augmenting cellular service [56, 119]. We identify and discuss two problems in CRNs, namely non-contiguous control channel establishment and k-protected routing protocol design.

1.1.1 Non-contiguous Control Channel Establishment in CRNs

The first problem we study in this dissertation is finding control channels for CRNs. In any wireless network, a pair of nodes needs to agree on necessary communication parameters (e.g., center frequency, bandwidth) before commencing data communication, so are SUs in CRNs. The SUs need to perform specific handshaking procedures on a common channel for obtaining these communication parameters. The process of two or more SUs attempting to meet on a channel is known as *rendezvous*. A control channel is needed not only for handshaking but also for transferring control information for CR and network functionalities (e.g., neighbor discovery, cooperative channel sensing, routing information.) An *always-on* control channel between any pair of SUs is needed for a CRN to function correctly. However, since the PU activities can be highly dynamic and are hard to predict, preselecting a control channel in the licensed spectrum is difficult and lacks flexibility. On the other hand, using unlicensed spectrum bands for control purpose not only deteriorates the congestion in the already crowded unlicensed spectrum bands but also deviates from the original purpose of the CRNs.

Most existing approaches are channel-hopping (CH) based, where the SUs hop on predefined channel hopping-sequences (CH-sequences). However, the main drawback of CH-based approaches is that the size of the CH-sequences and the time it takes to find a control channel increases drastically along with the number of channels. On the other hand, the non-contiguous orthogonal frequency-division multiplexing (NC-OFDM) control interfaces are no longer limited to access fixed-width channels only and can aggregate non-contiguous subcarriers for communication. The CH-based approaches cannot be applied to the NC-OFDM based interfaces because the number of subcarriers in the spectrum is ten times or even a hundred times to the number of channels.

To this end, we propose an efficient method for guaranteed NC-OFDM-based control channel (NCCC) establishment by utilizing short pulses on OFDM subcarriers. The results show that the time needed for establishing control channel is lower than that of CH-based approaches. Moreover, our approach can establish NCCC even if there is no common channel in the CRN and can significantly improve the success rate of forming CRNs.

1.1.2 k-Protected Routing Protocol Design

The second problem deals with the interruption to a routing path in the CRNs when the PUs become active again. In CRNs, the SUs have to ensure that their communications do not interfere with the PUs. When the PU who owns the spectrum band starts to be active again, the SUs that are currently using the spectrum band must stop using it as soon as possible. However, stop using the band might break the routing path between of the SUs and interrupt the ongoing communication session. The potential interruption is undesirable for quality-of-service (QoS) sensitive applications, such as audio or video conferencing and multimedia streaming, which motivates us to study the routing protocol designs in CRNs that build routing paths that will not be interrupted when PUs return to be active.

Existing reactive approaches that try to seek for an alternative route after PUs become active suffer from potential long delay and possible interruption if an alternative path cannot be found. Therefore, we propose the k-protected routing protocol for the CRNs. The k-protected routing

protocol attempts to find a set of main links with main spectrum resource plus k sets of preassigned backup spectrum resource and backup paths, that is guaranteed to sustain from k PU returns without being interrupted. When an application requests communication to another SU, depend on the characteristic of the application, it requests a different level of protection (different k). Our result shows that with the preallocated resources, the k-protected routing paths are never interrupted even when k PU returns, and have significantly shorter backup activation delays.

1.2 Heterogeneous Cellular Networks

On the other hand, we have also witnessed the blistering rate of advances in cellular technology in recent two decades. Significant amounts of research efforts have been devoted to the fourth generation (4G) cellular networks, to meet the overwhelming demands for bandwidth. The mobile operators are constantly buying or bidding spectrum bands to provide higher capacity and to support more mobile users. However, as aforementioned, the rigid regulation makes the process both difficult and time-consuming. It is critical to maximizing the efficiency of the limited spectrum currently available to the mobile operators. It is well known that there is a tradeoff between cellular station (cells) coverage and capacity [49, 52]. Since it is difficult to acquire more spectrum, *network densification* by deploying denser smaller cells (smallcells) in the same geographical area would significantly improve the capacity of the same amount of spectrum.

Network densification has been an important evolution direction since Long Term Evolution Advanced (LTE Advanced) release 10[89]. The smallcells with different sizes of coverage including microcell, picocells, and femtocells, are deployed with or without the macrocellular station (macrocell) coverage and form heterogeneous cellular networks (HetNets). The cells in HetNets operate on the same licensed spectrum band and use the same technology as macrocellular infrastructures. Hence no modification to existing user equipment (UE, e.g., mobile phones and tablets) is required. The smallcells connect to the core cellular network through wired IP-based broadband backhauls such as cables or the digital subscriber line (DSL). The smallcells are expected to provide seamless voice coverage and local area networks (LAN) comparable throughput. [7, 20, 35, 89, 111, 129].

Some researchers' vision of HetNet also considers augmenting cellular service by including other wireless technologies such as Wi-Fi [3] and LTE-U [95]. In this dissertation, we focus on HetNets formed by the smallcells and macrocells with same wireless technology, but with different coverage size, which is defined similarly as in [35, 51, 111]. In contrast to SUs in CRNs that attempt to utilize the licensed spectrum bands when the spectrum bands are idle, the smallcells are cooperating with the macrocellular infrastructures to augment the cellular service quality, to improve the capacity, and to reduce the load of the macrocellular infrastructures. However, since the smallcells and macrocells operate on the same frequency, interference becomes a critical issue and must be handled with care. In this dissertation, we study the new challenges in detecting and mitigating interference introduced by HetNets.

1.2.1 Interference Detection

The first research challenge for HetNets addressed in this dissertation is deciphering the interference relations. A typical HetNet can compose of several types of smallcells, including femtocells, picocells, and microcells. The femtocells and picocells aim at improving indoor signal quality for homes or offices, while microcells are beneficial for enterprise users. However, femtocells are deployed by end-users in an ad hoc fashion, and several characteristics of femtocells make the interference identification challenging. In densely populated urban areas, large-scale deployment of femtocells is expected to be realized in the future, and inter-femtocell interference must be handled with care. Existing interference identification approaches often regard more cells as interferers than necessary, such as considering neighboring femtocells with signal strength higher than certain thresholds as interferers. Misidentified interferers bring unnecessary interference mitigations, which result in a reduction of throughput to each cell. It is critical to identify true interferers accurately and to avoid misidentifying non-interfering femtocells as interferers.

We propose a method to improve the interference identification by taking advantage of the availability of multiple subchannels in an orthogonal frequency-division multiple access (OFDMA) [90] system. We let each femtocell transmit data using a subset of subchannels in the downlink to the

mobile stations (MS, e.g., phones and tablets). The different combinations of subchannels are called *patterns* in this dissertation. The femtocells transmit with different transmission patterns, which impact the subchannels the MSs can receive data. The subchannels from which an MS receives data constitute a received pattern. We propose an efficient algorithm that detects the potential interfering femtocells and the interference relations in a femtocell network by examining these patterns intelligently. After the interference relations are detected, isolated (non-overlapping) wireless resources are assigned to the femtocells to avoid inter-femtocell interference. The evaluation results show that our method successfully identifies all real interferers and eliminated most of the non-interfering femtocells compared with approaches that use received signal strength (RSS). With improved interference relation, we propose an efficient weighted vertex-coloring based resource assignment algorithm that allocates resources with better fairness and achieves higher throughput than existing approaches.

1.2.2 Interference Precancellation

The next research problem in HetNets is to mitigate the interference in HetNets, which means avoiding interference from happening, dealing with interfering signals, and ensuring the delivery of signals to the intended receiver. Most existing interference mitigation techniques handled inter-cell interference by assigning isolated resources to interfering cells [8, 73, 112, 120, 120, 126] or power control [21, 28, 80]. However, resource isolation leads to significantly fewer resources and power control sacrifices signal quality and coverage. Both approaches are mainly avoiding the interference from all potential interferers, which is overly conservative if the exact interferers are identified and known. In this dissertation, we propose an interference mitigation mechanism that deals with known exact interferers, called interference precancellation.

In cellular networks, the cellular base stations are connected to the cellular core network through high speed wired backhauls. The core cellular network disseminates the data for each MS to the cellular base station (cells, either macrocells or smallcells) via the backhauls. Then, cells send the data to the MSs via the wireless downlink. Since the data to all cells are originated from the

core cellular network, the interfered signal and interfering signal are known by the core cellular network. Therefore, instead of conservatively avoiding interference, we propose a mechanism that let the core cellular network *precancel* the interfering signal from the interfered signal before the interfered signal is sent to the cells. With precancellation, the cells do not need to avoid interference by conservatively allocating isolated resources or reducing transmission power and sacrificing their coverages. Instead, the same set of resources can be shared between MSs through interference precancellation and thus throughput is improved.

1.3 Dissertation Organization

The rest of the dissertation is organized as follows. Chapter 2 discusses the control channel establishment challenges in CRNs and our NC-OFDM-based control channel (NCCC) establishment algorithm. We discuss the k-protected routing protocol that deals with the potential interruption when the PUs become active in Chapter 3. Chapter 4 presents our interference detection and interferer identification in HetNets using received pattern. We present the interference precancellation that mitigates interference from known interferers in Chapter 5. Finally, we conclude this dissertation and discuss future research work in Chapter 6.

CHAPTER 2

EFFICIENT NC-OFDM-BASED CONTROL CHANNEL ESTABLISHMENT PROTOCOL IN COGNITIVE RADIO NETWORKS

The first problem we study in this dissertation is finding control channels for cognitive radio networks (CRN). In any wireless communication system, a pair of devices needs to agree on some necessary communication parameters (e.g., central frequency, bandwidth) before commencing data communication. Similarly, in CRNs, for two or more SUs to agree on the communication parameters, the SUs need to perform specific handshaking procedures on a common channel. The process of two or more SUs attempting to meet on a channel is known as *rendezvous*. A *control channel* is needed not only for handshaking but also for exchanging various control messages for CR functionalities (e.g., cooperative channel sensing) and network functionalities (e.g., neighbor discovery, routing information). Therefore, an *always-on* common control channel (CCC) between any pair of SUs is needed for a CR network (CRN) to function correctly. Previous CRN studies either rely on a dedicated CCC or dynamically find CCCs through *channel-hopping based (CH-based)* schemes. These methods, however, have some limitations.

The majority of dynamic CCC studies are CH-based, where the SUs hop on predefined *channel hopping-sequences (CH-sequences)*. The predefined CH-sequences are designed to have a certain degree of overlapping. When the SUs that wish to establish communication hop on the same channel, the SUs perform a handshake. The intentional overlapping of CH-sequences ensures that, by hopping on the CH-sequences, two SUs are guaranteed to meet on a channel in a finite time. The time consumption for all SUs in a CRN to rendezvous is called *Time-to-Rendezvous* (TTR).

The CH-based schemes take different channel availabilities into account and can adapt to the primary network dynamics. However, the main drawback of CH-based schemes is that the size of the CH-sequences and the TTR increases drastically along with the number of channels. Even for the state-of-the-art CH-based algorithm, Enhanced Jump-Stay algorithm (EJS algorithm) [71], the upper bound of the TTR is as high as $O(P^2)$, where P is the smallest prime number that is greater

than the number of channels.

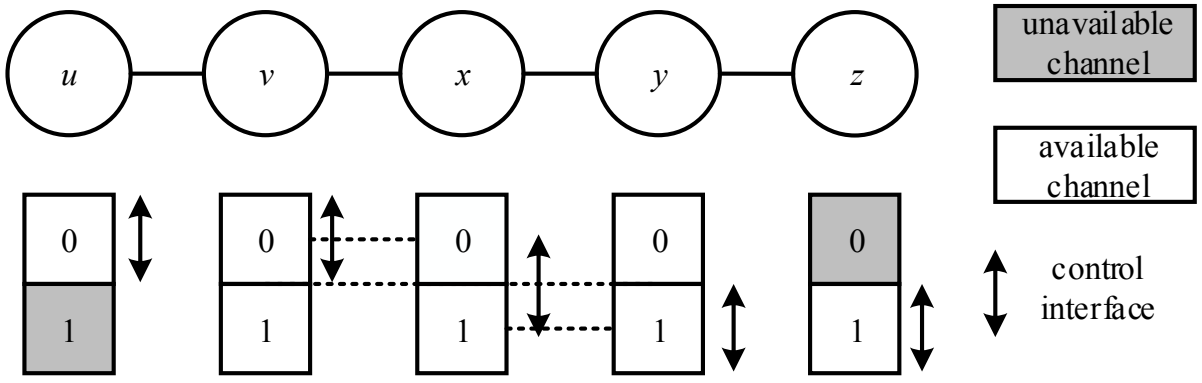


Figure 2.1: An NCCC establishment example. Traditional approaches for fixed-width channels systems cannot find a control channel for this example.

Additionally, traditional approaches require that at least a common channel must be available at all SUs across the whole network so that the SUs can all utilize the channel as the CCC [71, 114]. However, it is possible that such a channel does not exist and the SUs fail to form an effective CRN because the SUs are unable to exchange necessary control messages. In contrast to traditional channel-based wireless transceivers, several spectrum-agile systems [18, 122] propose the use of the *non-contiguous orthogonal frequency division multiplexing (NC-OFDM)* technology that is able to transmit and to receive using spectrum fragments. Utilizing spectrum fragments not only improves spectral efficiency, but also enables highly flexible spectrum allocation. By using the NC-OFDM interfaces for control purposes, SUs can operate on multiple spectrum fragments or to access spectrum that spans across multiple channels, which allows an NC-OFDM-based control channel (NCCC) to be established.

For a network shown in Figure 2.1, channel 0 is not available to node z, and channel 1 is not available to node u. Traditional channel-based approaches cannot establish control channel in this network because a network-wide CCC does not exist. A possible solution to this network is to let some nodes toggle between the two channels. For example, let node u and v use channel 0 as

the control channel and let node y and z use channel 1 as their control channel. Node x needs a mechanism to switch between channel 0 and 1 so that x can still exchange messages with node v and y. However, this channel switching mechanism of node x is not easy to implement and requires a certain amount of coordination and some overhead.

In contrast, the NC-OFDM enables flexible spectrum access, and an NC-OFDM interface is no longer restricted to access fixed-width channels. A feasible NCCC allocation for SU x exists by tuning x's control interface to access half of channel 0 and half of channel 1. A set of feasible NCCC allocations is found and can be established for the network. However, since the control interface no longer accesses fixed-width channels, The CH-based approaches cannot be applied to the establishment of the NCCCs. It is also unrealistic to apply a hopping-based approach to subcarrier level because the number of subcarriers is significantly larger than the number of channels. Although the NC-OFDM technology enables highly flexible spectrum usage, the establishment of the NCCC is challenging and is seldom discussed.

Instead of finding a globally available channel, this chapter proposes to take advantage of the NC-OFDM interfaces that can access and allocate partial channels for control purpose. We propose an efficient method for establishing control channels in CRNs that coexist with OFDM-based primary networks by utilizing short pulses on the OFDM subcarriers [27]. When a pulse is added to an OFDM subcarrier, only a small number of symbols in primary data transmissions experience interference. For a pulse with a duration of k symbols, at most $k + 1$ symbols in the time domain are affected. Instead of letting the SUs hop on their available channels, we exploit the pulses to design a control channel establishment strategy, called *probe and rendezvous*.

The main idea of *probe and rendezvous* is that an SU *probes* its neighbors about *one* available channel using *one* pulse. Considering two neighboring SUs u and v that are trying to handshake. The transmitting SU u transmits a pulse on a specific subcarrier, indicating a specific channel c is available. On detecting the pulse, if channel c is unavailable at the neighboring SU v, v ignores this pulse and waits for future pulses. When u times out and gets no response, u knows that channel c is unavailable at its neighbors and u will probe another channel in future rounds. On the other hand,

if channel c is available at SU v, v tunes its control interface to channel c and handshakes with u on channel c. Intuitively, one may wonder how an SU should handle multiple pulses detected at the same time. We will discuss these scenarios in Section 2.2.4. The major contributions of this chapter are summarized as follows.

- A CRN can establish an NCCC even if there is no common available channel across the whole CRN. Our approach is able to establish *at least 10% more* NCCCs compared with CH-based approaches.
- By exploiting the OFDM subcarrier pulses, we propose an NCCC establishment method called *probe and rendezvous* that does not require time synchronization. Each SU probes its neighboring SUs of their available resources with *minimized* interference to PU activities.
- The simulation results show that our approach is able to establish NCCCs with *at least 19.60% lower time consumption* than CH-based approaches.
- Our approach is able to find out that there is no feasible allocation in two rounds of probing.

We organize the rest of this chapter as follows. Section 2.1 provides discussions on related work and a brief introduction to the background of OFDM systems. We give a problem formulation and introduce our control channel establishment in Section 2.2. Section 2.3 is devoted to evaluating our algorithm. Finally, we summarize this chapter in Section 2.4.

2.1 Related Work and Background

Designing and finding control channels [4, 79] is an essential fundamental component to form feasible CRNs. In this section, we discuss the most representative existing control channel establishment studies and give an overview of OFDM systems.

2.1.1 Related Work

We categorize existing CRN studies into two categories: centralized and decentralized systems. Decentralized systems can be classified into dedicated CCC and channel-hopping schemes. The

majority of previous decentralized control channel establishment studies are based on channel-hopping.

Centralized Systems: In most centralized CRN systems [15, 16], the SUs report their local channel availability information and other CR-related information to a central entity through a pre-selected CCC known by all SUs. The central entity is responsible for coordinating and maintaining the CRN. The data communications between the SUs are also determined by this central entity. Simplicity is the main advantage of centralized systems. Due to the dynamic PU activities, the licensed spectrum availability varies not only temporally, but also spatially. The SUs at different geographical locations are very likely to have different spectrum holes. An SU might have different spectrum availability at a different time as well. As a result, preselecting a dedicated CCC on target licensed spectrum band is impractical and it makes the CRN lacks adaptivity to primary network dynamics. On the other hand, the CCCs in unlicensed spectrum deteriorate the congestion in unlicensed spectrum and deviates the purpose of the CRNs. Moreover, aside from the problems of CCC mentioned, the preselected CCC is also vulnerable to jamming attacks and the central entity is a single point of failure.

Dedicated CCC for Decentralized Systems: Several decentralized studies adopt a dedicated CCC [33]. In [32, 36, 82, 110], the SUs need to monitor an out-of-band CCC in unlicensed bands. The SUs exchange control messages through the out-of-band CCC, while the actual data transmissions are performed in licensed spectrum bands. Again, the use of out-of-band control channel deviates from the purpose of CRN and also deteriorates the congestion in unlicensed bands. HC-MAC [55] considers the hardware limited SUs that have a single radio interface. The goal of HC-MAC is to optimize the proportion of the time SUs on spectrum sensing and spectrum access.

Instead of occupying channels in the licensed spectrum for control purposes, O-CCC [26] proposed to create control channels by utilizing guard band subcarriers between OFDM-based primary networks. The SUs sense primary activities and the SUs determine the set of subcarriers as the CCC. When an SU detects a channel to be idle, the SU attempts to utilize the subcarriers in the idle channel and combine them into the CCC. This approach creates CCC in licensed bands

that coexist with primary activities. However, the number of guard band subcarriers is constant and thus the achievable rate for the CCC is also limited.

Dynamic CCC for Decentralized Systems: Previous dynamic and decentralized control channel establishment studies either establish a local control channel through negotiations or adopt CH-based schemes for finding a network-wide CCC.

The studies in [65, 132] focus on negotiating a local control channel, where the SUs form clusters according to their channel availability or physical locations. Each group has its local control channel, instead of a network-wide CCC. However, the negotiating the local control channel takes multiple rounds of negotiation through PU channels, which might result in interference with PU activities. Moreover, the overhead of forming and maintaining clusters and determining the local control channel is considerable.

Several dynamic and decentralized CH-based schemes are proposed for SUs to perform handshaking without dedicated CCCs. The SUs continuously hop on channels in a predefined CH-sequences. When the SUs hop on the same channel in the same time slot, the SUs perform handshaking. This approach is known as *blind rendezvous*, and the channel is known as the *rendezvous channel*. The rendezvous channel serves as the *local* control channel for the SUs. The SUs keep hopping on the CH-sequences until a *global* CCC is found. The most straightforward CH-sequence algorithms are random-based. Each SU simply hops on its available channel and hopes to rendezvous with another SU. AMRCC [34] improves the purely random algorithms by letting SUs have a higher probability to hop to channels with lower interference to the primary network. However, random-based CH algorithms cannot guarantee rendezvous in a finite time.

As a result, several CH scheme studies [12, 62] for synchronized networks focus on guaranteed rendezvous in a finite time by generating deterministic CH-sequences that have a certain degree of overlapping. The QCH [12] generates CH-sequences that are guaranteed to rendezvous by taking advantage of the property of quorum systems. A quorum system is composed of a set of quorums. Each quorum is a subset of elements. The intersection of any two quorums is designed to be a non-empty set. The QCH formulates channels as the elements in a quorum system, and each SU

hops on a quorum (a subset of the elements). The property of the quorums guarantees the SUs to rendezvous in a finite time. From the simulation results, the QCH outperforms other CH-based algorithms regarding TTR and throughput. However, also due to the property of the quorums, time synchronization is required for the QCH to function correctly, but global time synchronization may not be easily achieved in practice, especially in CRNs where control channels have not been established. An extended asynchronous quorum-based A-QCH [12] algorithm is proposed, but the A-QCH only works for networks consist of two channels. Such limitation makes it inapplicable to most of the CRNs.

Several CH-based algorithms [71, 76, 77, 114] that do not require time synchronization are proposed. In the MC/MMC [114], each SU picks a proper prime number and a rate smaller than the prime number. Based on these two parameters, the MC/MMC generates CH-sequences using a predefined modulo operation. Although the MC/MMC achieve rendezvous efficiently, the MC/MMC cannot guarantee rendezvous if two SUs happen to construct their CH-sequences with identical prime number and rate. RW [77] provides guaranteed rendezvous for asymmetric models and asynchronous environment. Each SU in RW formulates its available channels as vertices in a ring. The SUs traverse the vertices in the ring with different speeds. Rendezvous is guaranteed because the SU with higher speed would eventually rendezvous with lower speed SU on a vertex. However, theoretical analysis shows that the maximum TTR for the asymmetric model is $O(M^2 \cdot N)$, where M is the total number of channels and N is the number of SUs.

The Jump-Stay (JS) [76] and the Enhanced JS algorithm (EJS algorithm) [71] also utilize modulo operations to generate CH-sequences. JS and EJS produces CH-sequences consist of two repetitive periods, jumping period and staying period. The length of jumping and staying period varies across SUs due to the randomly selected step-length and starting index, and thus rendezvous is guaranteed. Although CH-based algorithms guarantee rendezvous, their TTR increases drastically along with the number of channels. The upper bound of the EJS algorithm's TTR is $O(P^2)$, where P is the smallest prime number that is greater than the number of channels. Instead of hoping the whole network to rendezvous on the same channel, ETCH [130] proposes an efficient rendezvous

algorithm that aims at utilizing multiple rendezvous channels as control channels. ETCH needs to pre-construct schedules with guaranteed rendezvous, but it is not likely to pre-construct overlapping schedules. However, since the NC-OFDM control interface is no longer limited to access fixed-width channels, CH-based approaches cannot be applied to the establishment of NCCCs.

2.1.2 OFDM Overview and NC-OFDM

The orthogonal frequency-division multiplexing (OFDM) is a digital modulation technique that encodes digital data into multiple carrier frequencies. In OFDM systems, instead of transmitting data using the whole spectrum band as a single carrier, the spectrum band is divided into evenly spaced orthogonal sub-bands, known as *subcarriers*. The digital data are encoded into parallel bit streams with a certain degree of redundancy, to recover the data/symbol losses due to interference and distortion. Each stream is transmitted by a subcarrier in parallel. The OFDM is effective in dealing with multi-path fading and is robust against narrow-band interference.

Instead of rigidly utilizing contiguous spectrum for wireless communication, Jello [122] and Papyrus [123] proposes the NC-OFDM that enables data communications over multiple spectrum fragments. The NC-OFDM transmitter transmits using multiple available spectrum fragments. The receiver filters out unwanted signals and only receives signals on those spectrum fragments used by the transmitter. However, a mechanism is needed for the transmitter and receiver to achieve a consensus on which spectrum fragments to be included for communication. RODIN [18] proposes a technique called spectrum-shaping for general wireless devices so that NC-OFDM is not required for DSA. By using filters, each spectrum fragment can be used as an independent channel. In each channel, cross-correlation is used to inform the receiver of the used spectrum fragments. Because each channel must be isolated through filters and guard bands, the spectral efficiency is reduced.

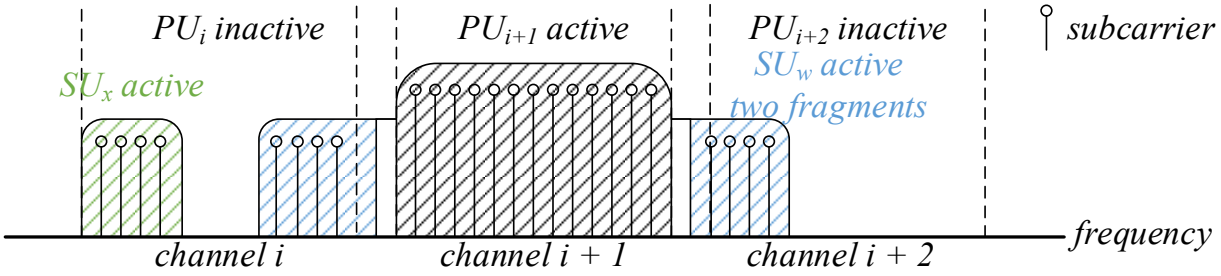


Figure 2.2: NC-OFDM interface enables transmission on non-contiguous spectrum fragments simultaneously. The taller/shorter subcarriers denote PU/SU is active.

Unlike channels, the spectrum fragments do not have fixed subcarrier composition. Neither do the spectrum fragments have a fixed spectrum width. As shown in Figure 2.2, instead of allocating whole channel i to SU x, with NC-OFDM, we can allocate a spectrum fragment (the leftmost block) in channel i to SU x. Meanwhile, NC-OFDM enables the allocation of multiple non-contiguous spectrum fragments in different PU channels, such as SU w shown in Figure 2.2, which is accessing the two spectrum fragments surrounding channel i + 1. Therefore, CH-based algorithms for control channel cannot be applied on NC-OFDMs directly. Even if a receiver has exactly the same spectrum fragment used in a transmission, the receiver does not know how to filter out unwanted signals. These issues have made NCCC establishment a challenging problem. In this chapter, we propose an efficient NCCC establishment algorithm for this problem.

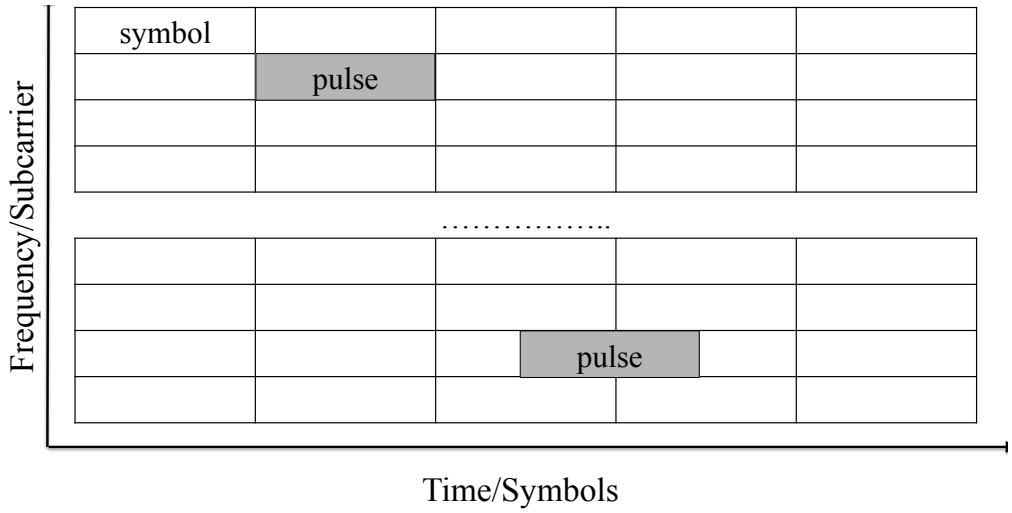


Figure 2.3: Pulses on OFDM subcarriers [27].

2.1.3 Pulses on OFDM Subcarriers

In [27], a novel technique that conveys messages by transmitting short and high power pulses over OFDM subcarriers is proposed. Pulses are simple sinusoid signals transmitted over a specific OFDM subcarrier that creates a discernible high power pulse on the subcarrier. As shown in Figure 2.3[27], the pulses are only located in one subcarrier for a short period of time and will only affect a small number of symbols of the same subcarrier. For a pulse with a duration of k symbols, at most $k + 1$ consecutive symbols in the time domain are affected.

As discussed in Section 2.1.2, based on current channel condition, most OFDM systems perform rate adaptation to maximize throughput. Usually, the receiver side performs the channel condition estimation on a per packet basis. The receiver estimates the channel condition (e.g., bit error rate (BER), signal-to-noise ratio (SNR)) on receiving a packet. The channel condition is then reported back to the packet sender in ACKs. Based on the channel condition received in ACKs, the transmitter selects the highest data rate modulation and coding schemes (MCS) that can ensure data delivery. For poor channel conditions, the transmitter uses a lower rate MCS such as QPSK with 1/2 coding rate, which means that 1/2 of the data are redundant for forward error correction. In contrast, for good channel conditions, the transmitter uses a higher rate MCS such as 64-QAM.

However, even the state-of-the-art rate adaptation protocol SoftRate [118] tends to select modulation and coding scheme conservatively, because even the latest channel condition estimation technique is known to be imperfect and error-prone. Therefore, each successful transmission has some *link margin* that can sustain the use of pulses.

Transmitting pulses does not consume extra energy on the transmitter. When transmitting regular signals, the transmitter has to allocate transmission power to all subcarriers to transmit signals. For pulses, the transmitter concentrates all the transmission power to a specific subcarrier and maps zeros to all other subcarriers. Therefore, the power on the subcarrier is significantly higher than other subcarriers. The receiver can detect a discernible power spike, and thus the pulse detection can be done by examining the received power level. If a subcarrier has a significantly higher power level than the other subcarriers, it may be carrying a pulse.

Pulse and FECs Experiment: We conducted experiments on GNU Radio [13]/USRP [37] to gain insight into pulses coexisting with OFDM-based transmissions. The channel bandwidth is 6 MHz, the central frequency is located at 4.0 GHz and the channel is divided into 64 subcarriers. The pulse transmission gain is set to 10 dB. A pulse is transmitted in two OFDM symbols (each of 64 samples) with a total duration of $2 \times 10.67\ \mu s$. The duration ensures the receiver can obtain the OFDM symbol without requiring the receiver to align with the first OFDM symbol. When the neighboring nodes monitor the channel by performing the FFT, the nodes can detect a discernible power spike, indicating there is a pulse.

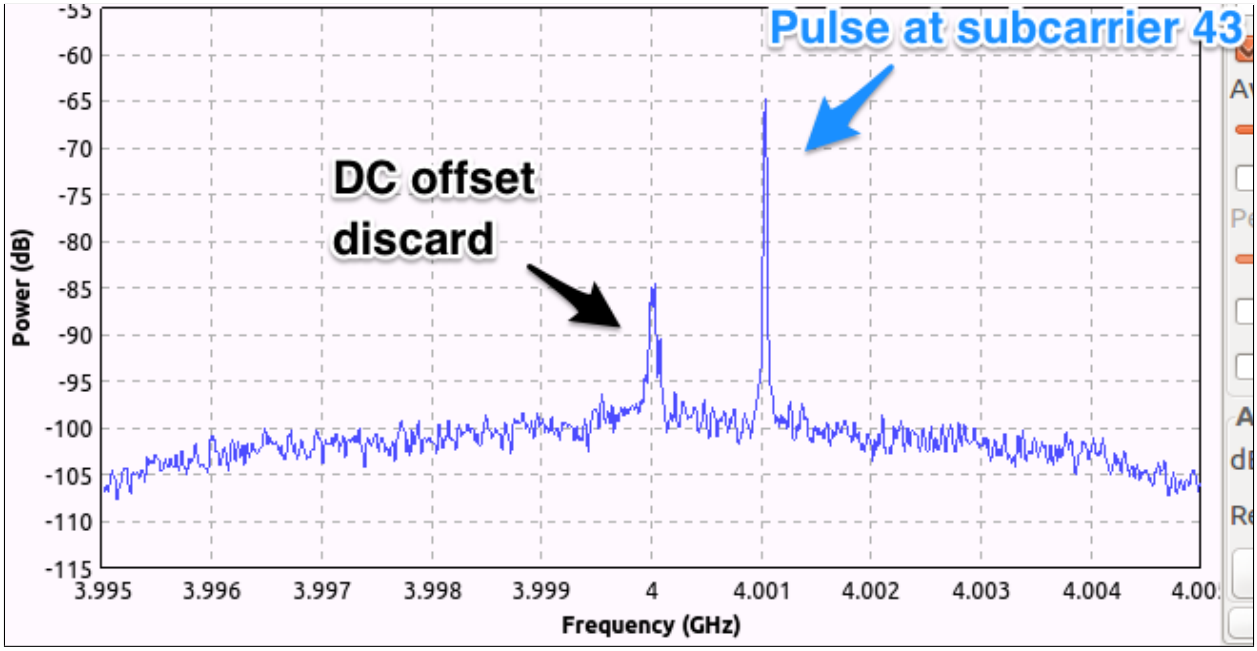


Figure 2.4: The power spike on subcarrier 43 with transmission gain 10 dB detected by a receiver.

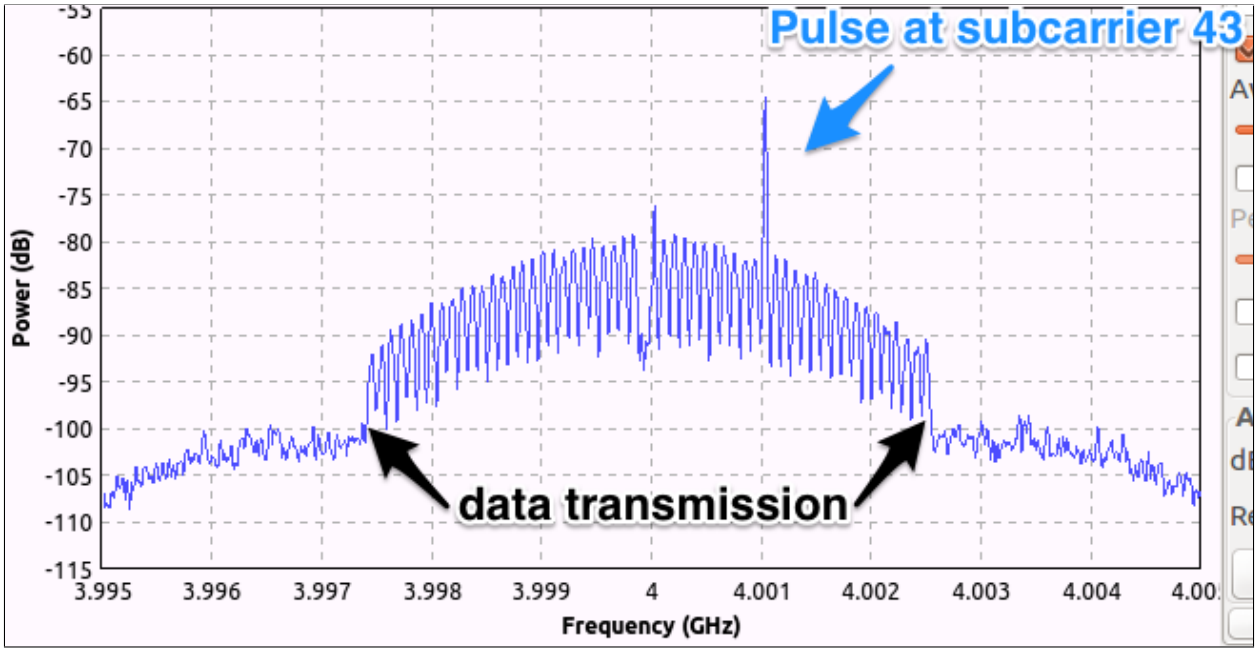


Figure 2.5: The power spike on subcarrier 43 and 6MHz data transmission detected by a receiver. Both transmission gains are 10 dB.

In Figure 2.4, by concentrating transmission power to subcarrier 43, the power spike is evident, and the detection can be done by checking the energy level on the subcarriers. In Figure 2.5, even with another data transmission of transmission gain 10 dB, the pulse power level is also higher

compared with other subcarriers carrying data. The pulse detection can be done by checking the power level on the subcarriers. A pulse is detected on subcarrier s if the power level on subcarrier is δ dB higher than the average power on all subcarriers, and the power level is higher than that on subcarrier s - *1* and s + *1*. δ is set to 10 in our experiment. To gain some insight into the side effect of the pulses to data transmissions, we also conducted experiments that transmit pulses on subcarriers simultaneously with OFDM transmissions encoded with forward error correction codes (FECs).

We conduct experiments by alternating transmission gain to see how pulses impact the FEC protected OFDM transmissions. The channel bandwidth is 10 MHz, and the channel is composed of 64 subcarriers. The central frequency is located at 4.0 GHz. We simulate OFDM-based primary activities by letting a USRP transmit OFDM signals to another USRP. The transmitter and the receiver are one meter away from each other. The transmission gain is set to 20 dB and is decreased by 1 dB after each experiment. The modulation is QPSK and the transmission time is about 60 seconds. Another USRP transmits two OFDM symbols carrying a pulse on one subcarrier every 10 milliseconds. We measure the bit error rate (BER) by comparing the received file on the receiver and the original file on the transmitter.

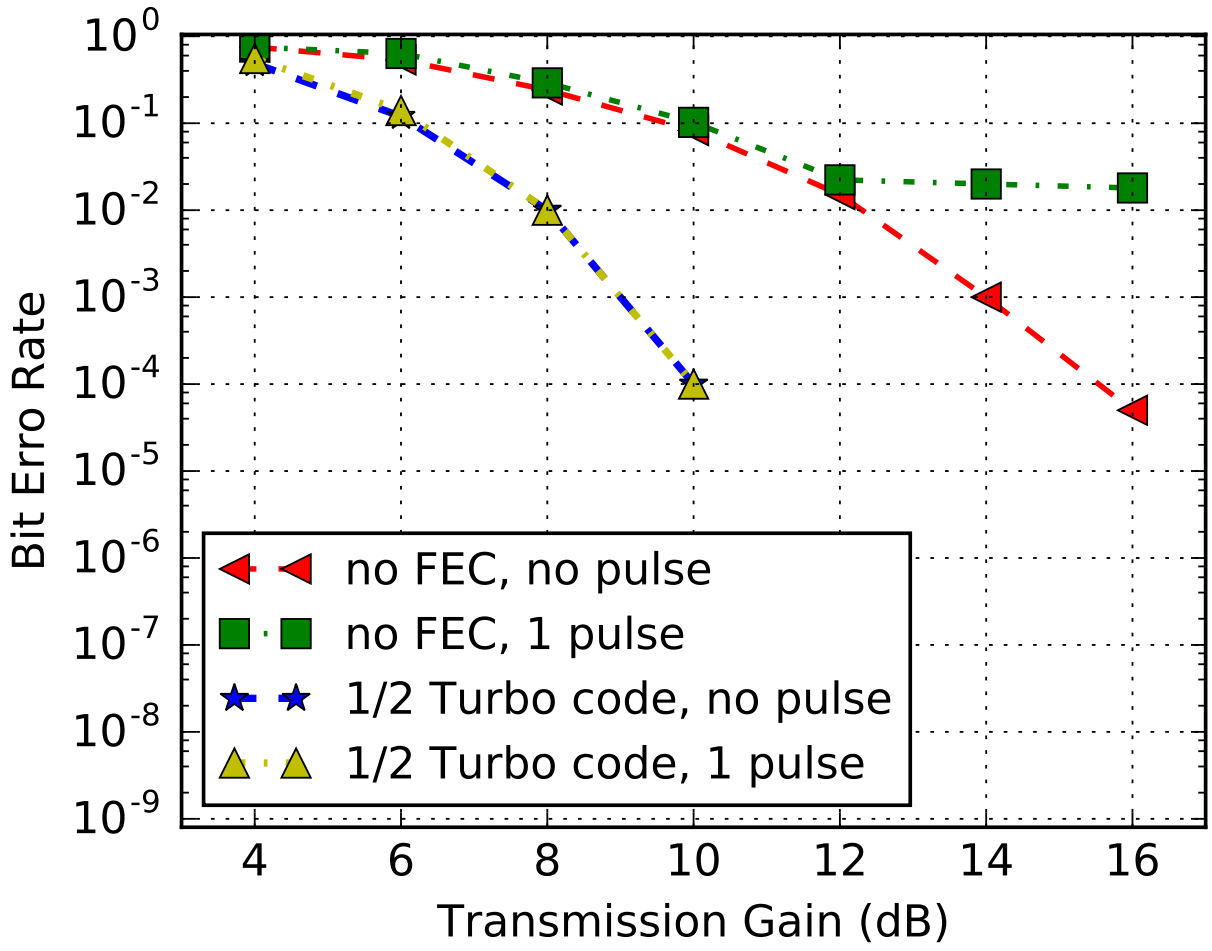


Figure 2.6: Bit error rate with or without FEC. The modulation is QPSK.

The BERs of different transmission gains are shown in Figure 2.6. Without the presence of pulses, the sender still needs 10 dB transmission gain so that the BER at the receiver is 10.23%. With 1/2 Turbo code, the sender with 6 dB transmission gain results in 12.71% BER at the receiver. With the presence of pulses and no FECs, the receiver cannot receive data correctly whenever there is a pulse. The BER on the receiver remains over 2% regardless of the transmission gain at the transmitter. It is because the pulses destroy one symbol located on one of the subcarriers. The transmission is not protected by the FEC to recover the symbols destroyed by the pulses. With 1/2 Turbo code, the destroyed data carried by the destroyed symbol can be corrected and the BER is not significantly different from that of 1/2 Turbo code without the presence of pulses. The experiment shows that the transmission with the FECs can correct data from pulse damages as long as there is

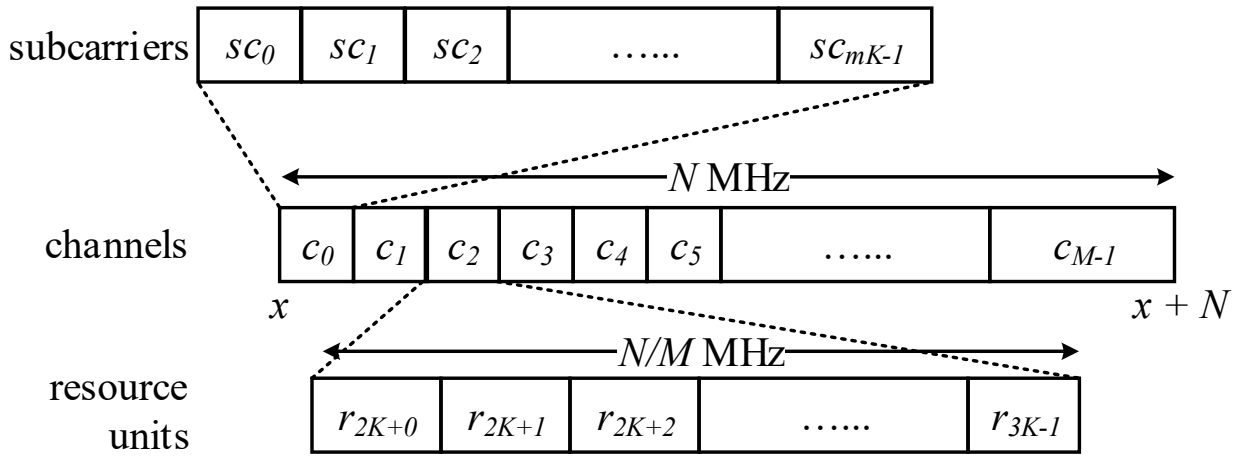


Figure 2.7: An illustration of channels and subcarriers. x is the starting frequency.

some *link margin*, which is very common in wireless systems [27]. Therefore, we propose to use pulses to establish NCCC for OFDM-based primary networks.

2.2 Problem Formulation and Solution

In this section, we present the problem formulation and our proposed solution by utilizing pulses on OFDM subcarriers.

2.2.1 Problem Formulation

Consider a CRN consists of S, $S \geq 2$, SUs and one or more OFDM-based PUs in the same geographical area. The licensed spectrum bandwidth is N MHz and is divided into M channels. The channels are denoted as $C = \{c_0, c_1, ..., c_{M-1}\}$, where c_i denotes the ith channel. A CRN is said to be symmetric if for all SUs u and $v \in S$, their channel availability $C(u) = C(v)$. Otherwise, the network is considered to be asymmetric. Obviously, algorithms for asymmetric models can be applied to symmetric models naturally. In the proposed NCCC establishment, we consider asymmetric model and the SUs are not synchronized.

Each channel is further divided into K non-overlapping *resource units (RUs)*, as shown in

Figure 2.7. Each RU is composed of m non-overlapping OFDM subcarriers, and a channel is thus composed of $(m \cdot K)$ subcarriers. An RU is the smallest unit for a pair of SUs to form NCCC allocations. The goal of NCCC establishment algorithm is to efficiently find an RU allocation as the NCCC for each SU's.

The number of subcarriers in a channel is greater than the total number of channels M and $m \cdot K \geq M$[1]. Subcarrier j in channel i is denoted as $sc_{(i \cdot m \cdot K + j)}$. Although allocating control channels in subcarrier-level (i.e., $m = 1$, one RU is one subcarrier) is theoretically possible, but the overhead is too high and is unnecessary. In the NCCC establishment algorithm, we group m subcarrier as an RU and allocate RUs to form NCCC allocations. The RU availability of channel c_i are denoted as $R_i = \{r_{(i \cdot K+0)}, r_{(i \cdot K+1)}, ..., r_{(i \cdot K+K-1)}\}$. The RU availability in the licensed spectrum are denoted as $R = \{r_0, r_1, ..., r_{(M \cdot K-1)}\}$. In the following context, the word *resource* denotes *resource unit*.

Each SU is equipped with an NC-OFDM-based control interface that can to access non-contiguous spectrum and perform CR functionalities. We assume that the control interface is able to perform channel sensing, and to tune itself within the spectrum with negligible overhead. With this control interface, each SU u is able to sense local channel availability, but the channel availabilities of other SUs are unknown to SU u. The channel availability of an SU u is denoted as $C(u) = \{c(u)_0, c(u)_1, ..., c(u)_{M-1}\}$. A channel i is said to be available to SU u if u can access channel i without interfering with any PU activity. The RU availability of SU u is represented by a several bits $R(u) = r(u)_0 r(u)_1 ... r(u)_{M \cdot K-1}$, where $r(u)_i$ is 1 if this RU is available to u, otherwise $r(u)_i$ is 0. The length of $R(u)$ is denoted as $|R(u)|$. If channel i is available to u, all RUs in channel i are available and $r(u)_{i \cdot K+0}, r(u)_{i \cdot K+1}, ..., r(u)_{i \cdot K+K-1}$ are all 1s. If $c(u)_i$ is unavailable, all RUs in channel i are unavailable and corresponding $r(u)$s are marked as 0s. NCCC allocates RUs, instead of channels for control purposes.

[1]This statement is true because, for any communication system, a channel is composed of at least one carrier. For OFDM-based systems, there are more subcarriers than channels. For example, in 802.11a/g/n, there are 15 channels and 64 subcarriers in a 20 MHz channel. For LTE-A, even the narrowest bandwidth 1.25 MHz channel is composed of 128 subcarriers.

A wireless interface consumes more power when accessing a wider spectrum bandwidth, because the interface needs to sample with higher sampling rate. It is unrealistic to expect that the control interface can access the whole licensed spectrum simultaneously. Therefore, to make the discussion more realistic, we include a limitation that the control interface can access spectrum that spans over n channels that is equal to $n \cdot \frac{N}{M}$ MHz ($n \cdot K$ RUs). The RU allocation of SU u's control interface's current accessing can be denoted as a bit stream $I(u) = r(u)_0 r(u)_1 ... r(u)_{M \cdot K-1}$. If SU u's control interface is currently accessing RU i, $r(u)_i$ is 1, otherwise $r(u)_i$ is 0. Note that in $I(u)$, the index of the last 1 minus the index of the first 1 must be less than or equal to $(n \cdot K)$, because the control interface is assumed to be restricted to access spectrum that spans over $n \cdot K$ RUs only. The intersection operation of $I(u) \cap I(v)$ is defined as $r(u \cap v)_0 r(u \cap v)_1 r(u \cap v)_2 ... r(u \cap v)_{M \cdot K-1}$, where $r(u \cap v)_i = (r(u)_i \ \& \ r(v)_i)$ for all $0 \leq i \leq (M \cdot K - 1)$. The union operation of $I(u) \cup I(v)$ is defined as $r(u \cup v)_0 r(u \cup v)_1 r(u \cup v)_2 ... r(u \cup v)_{M \cdot K-1}$, where $r(u \cup v)_i = (r(u)_i \mid r(v)_i)$ for all $0 \leq i \leq (M \cdot K - 1)$. Two SUs u and v can establish control channel if $I(u) \cap I(v) \neq 0_{M \cdot K-1}$, where $0_{M \cdot K}$ means all $M \cdot K$ bits are 0. The intersection and union operations $R(u) \cap R(v)$ and $R(u) \cup R(v)$ are defined similarly.

The NCCC establishment problem is defined as following: Given an OFDM-based CRN consisting of $S, S \geq 2$, SUs and M channels in a licensed spectrum. Each SU is equipped with an NC-OFDM-based wireless interface for control purposes. For each SU $u \in S$, we need to find *at least one* available RU to serve as the control channel for SU u and all u's one-hop neighbors. That is, for each SU u and each of u's one-hop neighbors v, we need to find the RU allocation $I(u)$ and $I(v)$, such that $I(u) \cap I(v) \neq 0_{M \cdot K}$. For SU u and v, they can perform handshaking and exchange control messages using the resources in $I(u) \cap I(v)$.

2.2.2 Probe and Rendezvous

The goal of our study is for SUs to obtain a set of NCCC allocations in licensed spectrum efficiently without using any preselected dedicated control channel. By exploiting the OFDM subcarrier pulses, we propose a *probe and rendezvous* method that probes the neighboring SUs of local

available channels with minimized interference to PU activities.

When the CRN is still negotiating control channels, all SUs' control interfaces are tuned to a predefined *probing channel* denoted as c_p. In contrast to blind rendezvous approach, we propose a *probe and rendezvous* approach. The basic idea is to enable one SU to *probe* its neighbors about *one* of its available channels using *one* pulse on a specific subcarrier. Each SU transmits a pulse on c_p every random amount of time. Suppose channel i is available to SU u, u transmits a pulse on $sc_{(p \cdot m \cdot K+i)}$ (ith subcarrier in channel p), indicating c_i is available to SU u. Because u is checking the availability of channel i to u's neighbors, channel i is the current *checking channel* of u.

After the pulse is sent, SU u switches its control interface to checking channel i and waits for any neighbor to respond. For an SU v, on detecting the pulse, if $c(v)_i$ is available, v also tunes its control interface to channel i and channel i now serves as the *rendezvous channel* between u and v. SU u and v perform handshaking. We call this a *successful probing* between u and v. If c_i is unavailable to v, v simply ignores this probe and waits for future pulses. When SU u times out without receiving any response from v, u knows that $c(v)_i$ is unavailable and this is a *failed probing*.

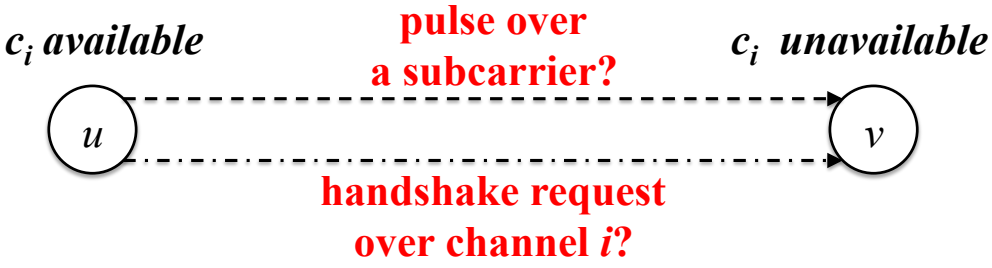


Figure 2.8: The scenario of a secondary network activity interfering with primary networks on channel c_i.

2.2.3 Probing Causes Less Interference

Consider two SUs u and v in Figure 2.8, c_i is unavailable to v and c_i is available to u. Some primary network activities on c_i must be taking place near v, but the activities might be too far away from

u and u is unable to sense the primary activity. u needs to know if c_i is available to v and if c_i can be used as a control channel.

In our approach, SU u probes the availability of channel c_i by transmitting a pulse on sc_i to SU v. One may argue that the pulse from u is causing interference to PU activities. However, as we discussed earlier, the pulse carried by one subcarrier only interferes with one subcarrier in the frequency domain and a small number of symbols in the time domain. ($k + 1$ symbols are affected if the duration of the pulses is k symbols.) Moreover, there is some *link margin* in each successful OFDM transmission that can sustain the pulse.

Nevertheless, this is also a concern in CH-based schemes, but it is seldom addressed. Whenever an SU hops to a channel, some handshaking procedures through packet exchange are required. Even for the most straightforward two-step rendezvous procedure, it requires at least two packets: *Handshake Request* and *Handshake Grant*. Each time an SU hops to the next channel, it must broadcast a handshake request, and it must wait for its neighbor to respond.

Consider the same scenario in Figure 2.8, u hops to c_i, and it transmits a handshake request. c_i is available to u, but *is not available to* v. When u hops to c_i, at least u needs to broadcast a handshake request. However, if a handshake request sent over the entire channel c_i, is causing interference to PU activities, the request must cause interference that is more severe than *a single pulse*. In contrast, the interference to primary activities caused by a failed probing to v is *minimized* to a subcarrier and is *less than CH-based approaches*.

2.2.4 NC-OFDM-based Control Channel Establishment

In this section, we describe the detailed NC-OFDM-based Control Channel (NCCC) establishment algorithm. We adopt the same assumption in most related studies that each SU u knows its one-hop neighbors, denoted as $N(u)$. This assumption is a common assumption of control channel establishment studies. The NCCC establishment algorithm requires each SU to perform *two* rounds of successful probing to each of its neighbors. Initially, the control interfaces of all SUs are tuned to the probing channel. Each SU u only knows about its local RU availability ($R_l(u)$). In the first round

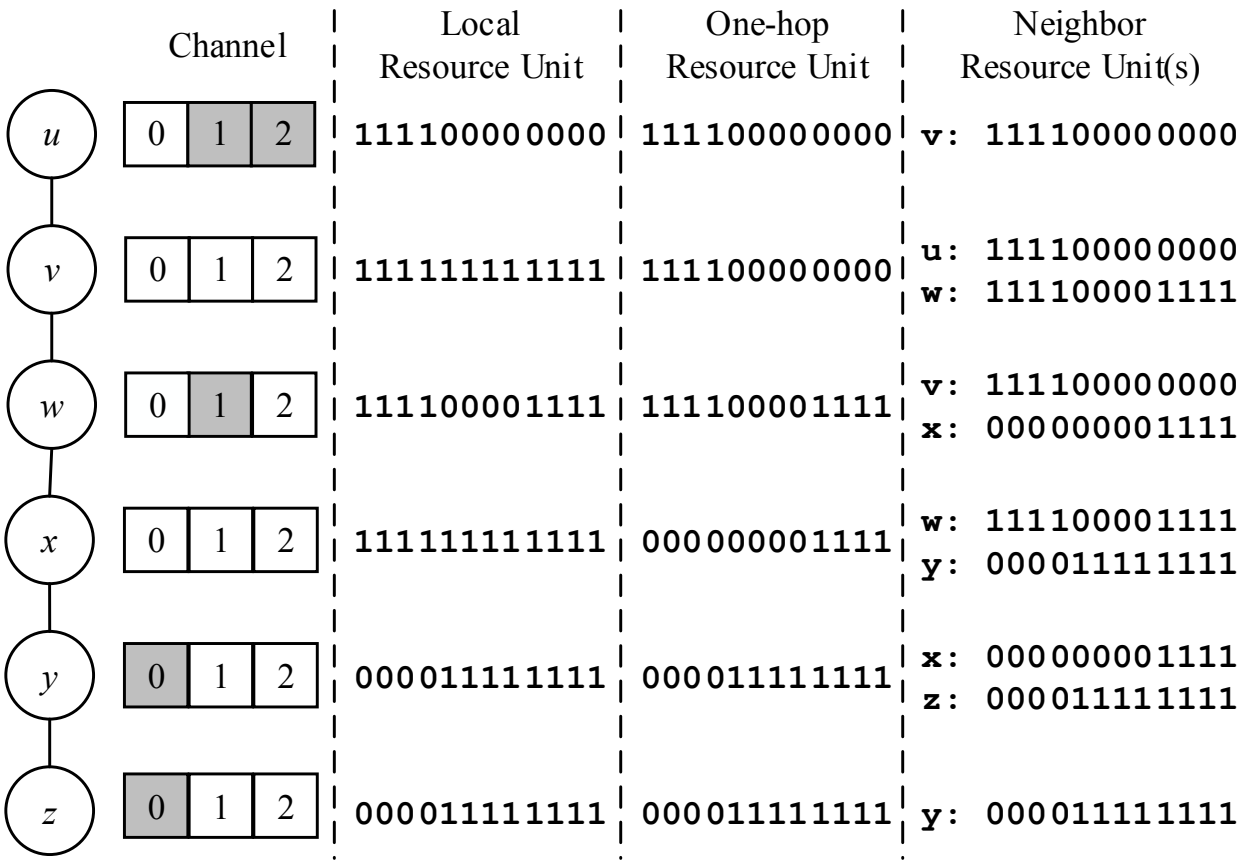


Figure 2.9: An NCCC establishment example. There are three channels ($M = 3$) and each channel is composed of 4 RUs ($K = 4$). $|R| = M \cdot K = 12\ bits$.

of probing, the SUs exchange their R_l information using the negotiation packets. After an SU u collected R_ls of all its neighbors, u knows its *one-hop* RU availability ($R_o(u)$). In the second round, the SUs exchange their R_os. After an SU collected all R_os information of its entire neighborhood, it can determine its control RU allocation, the NCCC, locally. The negotiation packet is a three-tuple ($SU\ ID$, R_{ngt}, $flag$). The $flag$ is *false* if this R_{ngt} is R_l and is *true* if R_{ngt} is R_o.

2.2.4.1 The Probe and Rendezvous Algorithm

Consider a simple CRN shown in Figure 2.9. Each channel is composed of 64 subcarriers. We define each RU to be 16 subcarriers, and thus each channel is composed of $K = 4$ RUs. Suppose

that the NC-OFDM control interface can access spectrum that spans over the bandwidth of two channels, and we use channel 0 as the probing channel, i.e., $n = 2$ and $p = 0$. As a result, the RU availability can be represented by $M \cdot K = 3 \cdot 4 = 12$ bits and the subcarriers that carry pulses are sc_0 to sc_2. Initially, the control interfaces of all SUs are tuned to $c_p = c_0$.

Input: $N(u) = \{\text{one-hop neighbors}\}$, $R(u)_{local}$;
Output: $I(u)$: u's control resource allocation;
// Initialize
for $v \in N(u)$ **do**
    $R(v)_{local} = NULL$;
    $R(v)_{one-hop} = NULL$;
end
while *any* $v \in N(u)$, $R(v)_{local}$ *or* $R(v)_{one-hop}$ *is NULL* **do**
    next_time = random(20 ms, 25 ms);
    sleep(next_time);
    checking_channel = get_next_checking_channel();
    transmit_pulse(checking_channel);
    change_channel(checking_channel);
    while *wait_for_negotiation(20 ms)* **do**
        $(v, R(v), flag)$ = receive_negotiation();
        update(v, $R(v)$, $flag$);
    end
    change_channel(probing_channel);
end
$I(u)$ = compute_control_resource_allocation();

Algorithm 1: Probing Algorithm of an SU u

Suppose that SU v probes channel 1 (c_1) by transmitting a pulse on sc_1. Then SU v tunes its control interface to c_1 and waits for rendezvous with its neighbors. Channel c_1 is unavailable to both SU u and w, so they do not respond to this probing. After SU v times out, it knows that c_1 is unavailable on both SU u, and SU w and SU v tune their control interfaces back to c_0 for detecting future pulses. Suppose SU w is the next SU to probe and SU w probes channel 2 (c_2) by transmitting a pulse on sc_2. c_2 is available to both SU v and x, so both v and x tune their control interfaces to rendezvous with v and exchange negotiation packets. Since both v and x only know about their R_ls, their negotiation packets are $(x, R_l(x), false)$ and $(v, R_l(v), false)$. On receiving the negotiation packets from v and x, w knows all its one-hop neighbors' R_l, and w knows its

one-hop RU availability $R_o(w)$ = 111100001111.

An SU cannot determine its NCCC allocation relying merely on R_o without considering its neighbors' R_os. In Figure 2.9, without considering $R(v)_o$ and $R(y)_o$, SU w might think either c_0 or c_2 can act as the control channel according to $R(w)_o$. However, c_0 is not a feasible choice for SU x, because when SU x transmits using $c(x)_0$, it might interfere with primary activities that are currently using c_0 near SU y. c_2 has similar a concern to SU v and u. Therefore, a second round probing and negotiation are needed for each SU to $collect$ its neighbors' R_o. In the second round probing, the negotiation packets from an SU u are $(u, R(u)_o, true)$. The probing algorithm is summarized in Algorithm 1.

Input: $N(u)$, $R(v)_{one-hop} \forall v \in N(u)$ and $R(u)_{one-hop}$;
Output: $I(u)$: the control resource allocation;
$U(u) = 0_{M \cdot K}$;
for $v \in N(u)$ **do**
    $R(uv) = R(u)_{one-hop} \cap R(v)_{one-hop}$;
    if $R(uv) == 0_{M \cdot K}$ **then**
        return $0_{M \cdot K}$;
    end
    $U(u) = U(u) \cup R(uv)$;
end
Let the index of first 1 in $U(u)$ be $first$;
Let the index of last 1 in $U(u)$ be $last$;
while $last - first > n \cdot K$ **do**
    $U(u)[first] = 0$, $U(u)[last] = 0$;
    $first+ = 1$, $last+ = 1$;
end
for $v \in N(u)$ **do**
    // check if $U(u)$ can still cover all $R(uv)$;
    if $U(u) \cap R(uv) == 0_{M \cdot K}$ **then**
        return $0_{M \cdot K}$;
    end
end
return $U(u)$ as $I(u)$;

Algorithm 2: NCCC Allocation Algorithm of an SU u

2.2.4.2 The Control Resource Allocation Algorithm

When an SU u collected all its neighbors' R_o, u can determine the NCCC allocation for its control interface. For each $v \in N(u)$, u computes $R(uv) = R_o(u) \cap R_o(v)$. $R(uv)$ is the candidate RUs that can be used as the control resources between u and v. However, u needs to make sure if its control interface can cover all RUs in $R(uv) \forall v \in N(u)$. u computes a union $U(u) = U(u) \cup R(uv) \forall v \in N(u)$. Let the index of first 1 in $U(u)$ be *first*, and the index of last 1 in $U(u)$ be *last*. $U(u)$ must satisfy that $last - first \leq n \cdot K$, because the control interface can access spectrum that spans over $n \cdot K$ RUs as defined in Section 2.2.1. If $U(u)$ requires RUs that span spectrum that is wider than the capability of the control interface, we need to check if we can shrink $U(u)$ by giving up RUs at the edges. The detailed algorithm is described in Algorithm 2.

The SUs compute NCCC using the common available RUs based on their R_os and their neighbors' R_os. If there are more than one available RUs, the SUs choose the RUs with smaller indices to be the NCCC. For SU u in Figure 2.9, $R_o(u) = R_o(v)$, u simply use c_0 as the control interface and $I(u) = 111100000000$. Note that u does not utilize RU 4 to 7 because c_1 is not available to u. Similarly, SU v, x, and y can also easily determine their control RU allocation to 11110000000, 000000001111 and 000000001111, respectively. For SU z, the initial NCCC allocation would be 000011111111, and it can communicate with y through RU 8 to 11.

However, the NCCC allocation of SU w needs some adjustment. By comparing $R_o(w)$ and w neighbors' R_o, $U(w) = 111100001111$ and w knows that it can communicate with v using RU 0 to 3 and with x using RU 8 to 11. Because of the restriction that the control interface can only span over $n = 2$ channel wide spectrum, the control RU allocation can only span over 8 RUs. The control interface needs to give up access to the first two and last two RUs. Therefore, x established an NCCC with its neighbors and the final $I(x) = 001100001100$. RUs 2 and 3 are used for communication with v and RU 8 and 9 are used for communication with y. On the other hand, if $n = 3$, the restriction to $I(x)$ is relaxed and the $I(x)$ becomes 111100001111. In contrast, if $n = 1$, then there is no possible NCCC allocation that can establish control channels for this CRN. For SU z in Figure 2.9, if n is 1 instead of 2, z's NCCC allocation would be 000000111100. However, an

wireless interface consumes more power for larger n because it needs to sample wider spectrum.

2.2.4.3 The Probe Detection Algorithm

There are several scenarios to be addressed when an SU is detecting pulses, including deferring one-hop neighbors probe and handling simultaneous pulses.

Notification Pulse: When an SU v detects a pulse from u on subcarrier i ($sc_{p\cdot m\cdot K+i}$), v checks if $c_{p\cdot m\cdot K+i}$ is available. If the checking channel $c_{p\cdot m\cdot K+i}$ is available to SU v, SU v tunes its control interface to channel i to perform handshaking. When v is transmitting using channel i, its control interface is no longer accessing the probing channel c_p, and SU v can no longer detect pulses from its neighbors. However, if SU x transmits a pulse on $sc_{p\cdot m\cdot K+j}$ while v's control interface is not at the probing channel, v is not able to respond to this probe from x and x might incorrectly assume channel j is not available to v. This is a false negative failed probing.

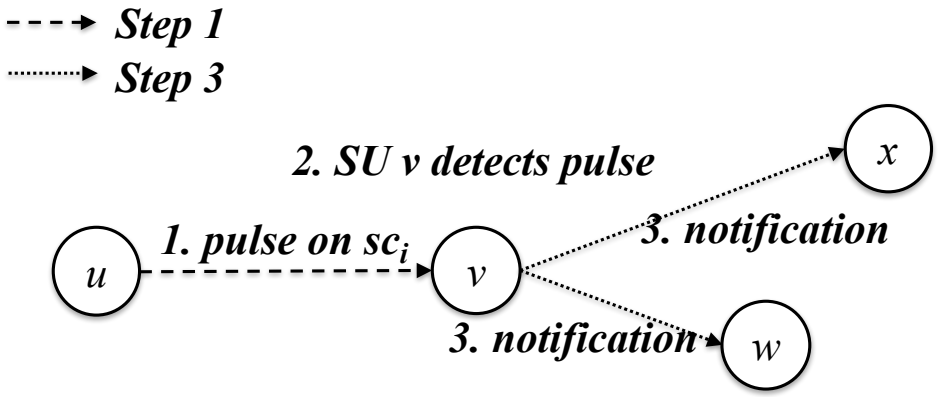


Figure 2.10: SU v notifies its neighbors to reschedule their probing.

Therefore, in order to avoid false negative failed probing, before v tunes its interface to the checking channel, it needs to notify all its other neighbors $N(v)$ about it. Each SU $w \in N(v)$ needs to *defer* its next probing because this v is already responding to a probing. As shown in Figure 2.10, SU u transmits a pulse on $sc_{p\cdot m\cdot K+i}$ to check if c_i is available to SU v. On detecting this pulse, SU v transmits a notification pulse to SU v's neighbors so that SU v's neighbors can adjust their probing schedule (by changing $next\_time$ in probing algorithm) to correctly probe v.

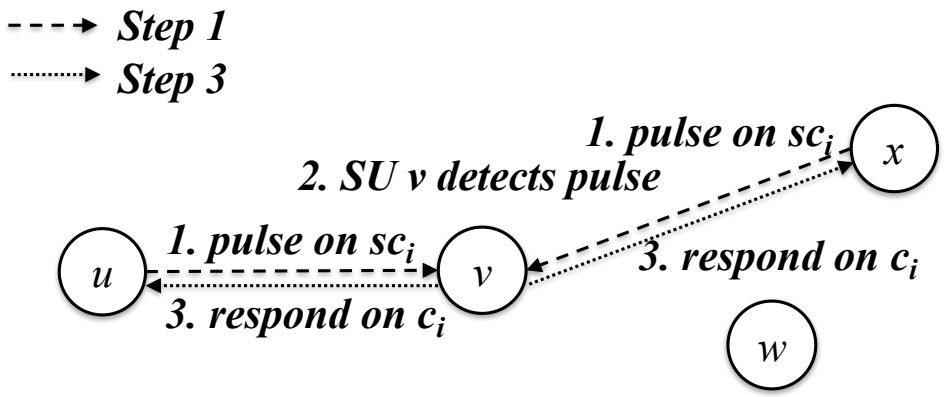


Figure 2.11: SU u and SU x probe c_i at v simultaneously. SU v responds to both u and x as if only one probing only.

Multiple Pulses Simultaneously: Although SUs transmit pulses in random period of time, it is still possible that more than one SU is probing and multiple pulses are detected by an SU simultaneously. If multiple SUs transmit pulses on the same subcarrier, it is a trivial case because they are all probing the same channel. The receiver does not care if there are multiple pulses. The receiver still tunes its interface to the probing channel and responds a negotiation packet. The negotiation packet will be received by all transmitting SUs as if only one pulse is detected. As shown in Figure 2.11, SU u and x probes the availability of c_i to SU v. SU v does not care if there are multiple pulse on subcarrier sc_i or not. SU v simply responds a negotiation packet to both u and x using channel i as if only one pulse is detected.

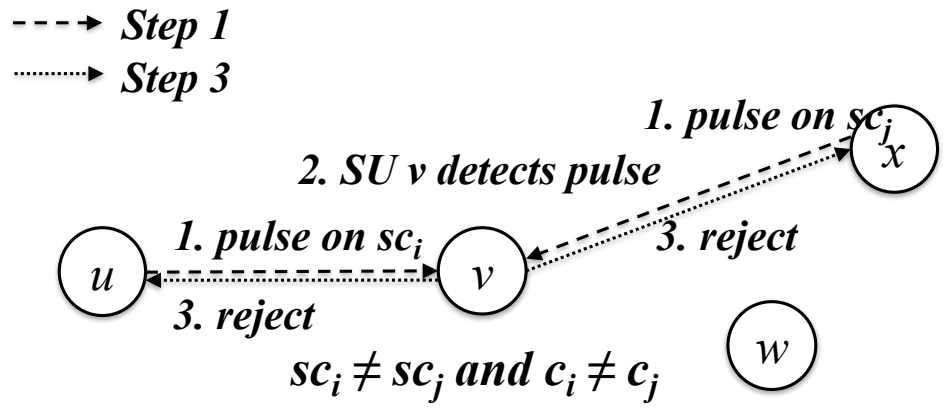


Figure 2.12: SU u probes c_i at v and SU x probes c_j at v simultaneously. SU v rejects both probes.

However, if the SUs transmit pulses for probing different channels and an SU detects multiple pulses on *different* subcarriers at the same time, the detecting SU cannot respond on all these channels simultaneously. The SU has no choice but rejecting all the probing pulses as shown in Figure 2.12. On detecting the two pulses, SU v cannot respond both of them and has to reject both probings. When the transmitting SUs detect rejection pulse, the transmitting SUs x and u give up this probing and reschedule another probing to the same channel later.

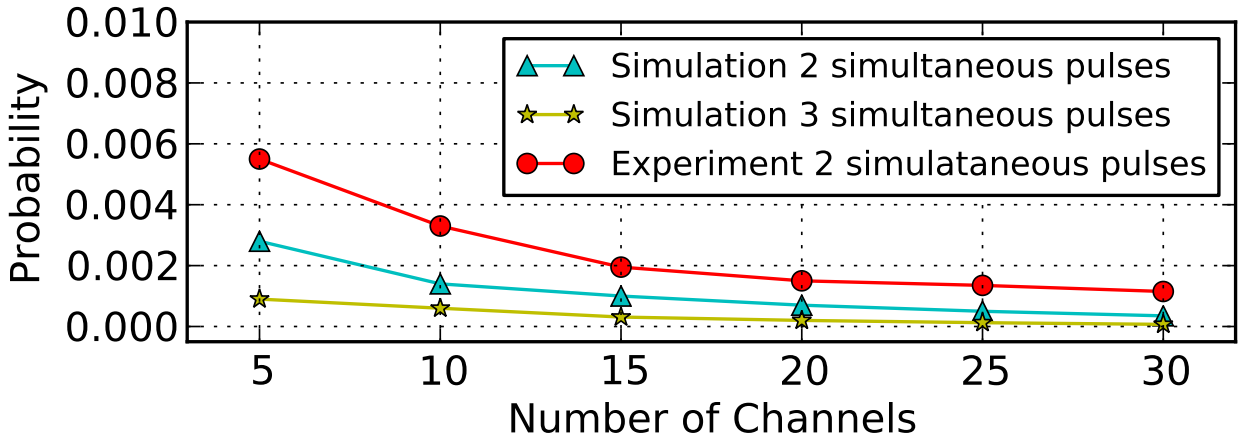


Figure 2.13: The probability of pulses detected at the same time.

The Probability of Simultaneous Pulses: We conduct both simulations and experiments to gain some insight into the probability of simultaneous pulses. The results are plotted in Figure 2.13. In our simulation settings, a pulse has a duration of 4 μs, and each SU transmits a pulse every 20 to 25 ms. The topologies are the same set as in Section 2.3. The average degree of the nodes is 3.4. Thanks to the notification pulses, the chance of two simultaneous pulses is around 0.25%, and the chance of three simultaneous pulses is lower than 0.1% when there are five channels. The probability of simultaneous pulses is even lower as the number of channels increases.

We also conducted experiments on GNU Radio/USRP. Two USRPs act as pulse transmitting SUs and another USRP is used to receive the signal. The USRPs are placed in a line in the order of transmitter, receiver, and transmitter. Each pulse transmitter transmits a pulse on a random subcarrier in a channel consists of 64 subcarriers every 20 to 25 ms. The pulses transmitted by the USRPs are of length 10.67 μs, and the receiver does not return notification pulses. The receiver

```
Input: N(v) = {one-hop neighbors};
Output: None;
while still negotiating I(v) do
    checking_channel = detecting_pulses();
    if more than one pulse then
        transmit_pulse(p · m · K + M + 1);
    end
    else if checkin_channel is M) then
        reset_next_time_in_probing();
    else if checkin_channel is M + 1) then
        transmit_same_pulse_again_in_probing();
    else
        transmit_pulse(M);
        change_channel(checking_channel);
        if R(v)_{one-hop} is known then
            send_negotiation(v, R(v)_{one_hop}, true);
        end
        else
            send_negotiation(v, R(v)_{local}, false);
        end
        change_channel(probing_channel);
    end
end
```

Algorithm 3: Probe Detection Algorithm of an SU v

records the signals as a file, and we decode it using GNU Radio. The power spike of multiple pulses on the same subcarrier is higher than the power spike when pulses are on different subcarriers. We ran the experiment five times for a different number of channels. For each experiment, 60 seconds of the log from receiver were analyzed. The probability of multiple pulses is still as low as around 0.6% when there are five channels.

Special Purpose Pulses: As a result, the NCCC establishment algorithm needs at least two types of special purpose pulses, including notification and rejection. Because there are M available channels, we only need subcarrier 0 to $M-1$ in channel c_p for probing all available channels. We can take advantage of the free subcarriers $sc_{p\cdot m\cdot K+M}$ to $sc_{(p+1)\cdot m\cdot K-1}$ for these special purposes. In real-world wireless communication systems, there are guard band subcarriers at the two edges (the first and last few subcarriers in a channel) of a channel in avoid cross-band interference. We utilize these guard band subcarriers for special purpose pulses to reduce interference to PU

activities. We use $sc_{(p+1)\cdot m\cdot K-1}$ as pulses to notify neighbors to reschedule their probing pulses and $sc_{(p+1)\cdot m\cdot K-2}$ to reject simultaneous probing pulses. The probe detection algorithm is summarized in Algorithm 3.

2.2.4.4 Guaranteed NCCC Establishment

Each SU probes its neighbors about each of its available channels. There are four possible situations when an SU u is performing a probe and rendezvous on channel c. Firstly, before the pulse is actually sent, SU u receives a notification pulse from its neighbor and u has to reschedule the probing pulse. u is still waiting to perform a probe and rendezvous on channel c. Secondly, a pulse is sent, and neighbors respond with a rejection pulse. SU u has to give up this probe and probes the same channel again later. Thirdly, the probing pulse is responded by its neighbor and is a successful probing. Alternatively, the probing pulse is sent without any response, and it is a failed probing.

In summary, if an SU cannot successfully probe any of its neighbors after probing all channels, the SU knows that the none-responding neighbors and itself do not have any available resource in common. There is no feasible NCCC allocation. On the other hand, an SU is guaranteed only to continue to probe another channel if and only if the probing of the current channel is done. An SU knows its own R_l and all its one-hop neighbors' R_os if it has successfully probed all its neighbors. Naturally, the NCCC establishment is bounded by the number of channels (M), and its complexity is $O(M)$. As described in Algorithm 2, The SUs are guaranteed to compute an NCCC assignment based on their R_ls and R_os of their neighbors.

2.2.4.5 Overhead Analysis

Each SU expects two negotiation packets from each of its neighbors. One negotiation packet for R_l and another for R_o. Suppose the SU ID can be represented by d bits. One negotiation packets is $d + M \cdot K + 1$ bits. An SU only needs to exchange information with its neighbors. Let $deg(u)$ denote the degree of node u. The total number negotiation packet required is $|S| \cdot \frac{\Sigma_{u\in S} deg(u)}{|S|} \cdot 2 =$

$\Sigma_{u \in S} deg(u) \cdot 2$. The number of data transmitted is $\Sigma_{u \in S} deg(u) \cdot 2 \cdot (d + M \cdot K + 1)$ bits. The overhead only increases linearly along with the increase in the number of channels.

2.3 Performance Evaluation

We simulate and evaluate our algorithms with the different number of SUs and PUs randomly deployed in an area of $1500 \cdot 1500$ square meters. The SUs have the same transmission range of 250 m. We assume that each SU is equipped with one NC-OFDM-based CR interface for control purpose and another interface for data communication. We evaluate the different max span (different n) of the spectrum that each control interface can access. The control interfaces exchange negotiation packets in CSMA/CS fashion. Each channel is composed of 8 RUs as the basic allocation unit, and each RU consists of 16 subcarriers. Therefore, there are 128 subcarriers in a channel. The pulse length is 4 μs. Each SU performs a channel probing every random amount of time between 20 and 25 ms.

We compare NCCC with EJS algorithm [71], which is one of the most representative CH-based algorithms. The length of each hopping slot in EJS algorithm is set to 20 ms. EJS algorithm is a CH-based rendezvous algorithm that guarantees rendezvous in a finite time if a CCC exist. However, EJS algorithm does not consider the use of NC-OFDM-based control interface. Our work focuses on exchanging RUs R, and finds a set of feasible NCCC allocation for the network using *probe and rendezvous*. Therefore, instead of using the term TTR, we use the term *time consumption* to establish NCCC. Moreover, we show that the NCCCs are established even if the control interface can only access spectrum that spans across one channel ($n = 1$).

2.3.1 Symmetric Model Time Consumption

In simulations for symmetric models, we randomly select some channels to be unavailable across the whole network. The ratio of unavailable channels over the number of channels (M) is defined as θ. Higher θ means less available channels for the SUs. Every SU has exactly the same channel availability across the network. We perform six sets of simulations for the symmetric model by

varying θ and M. Each set of simulation consists of fifty randomly generated topologies and the SUs deployed in the topologies has unified channel availability.

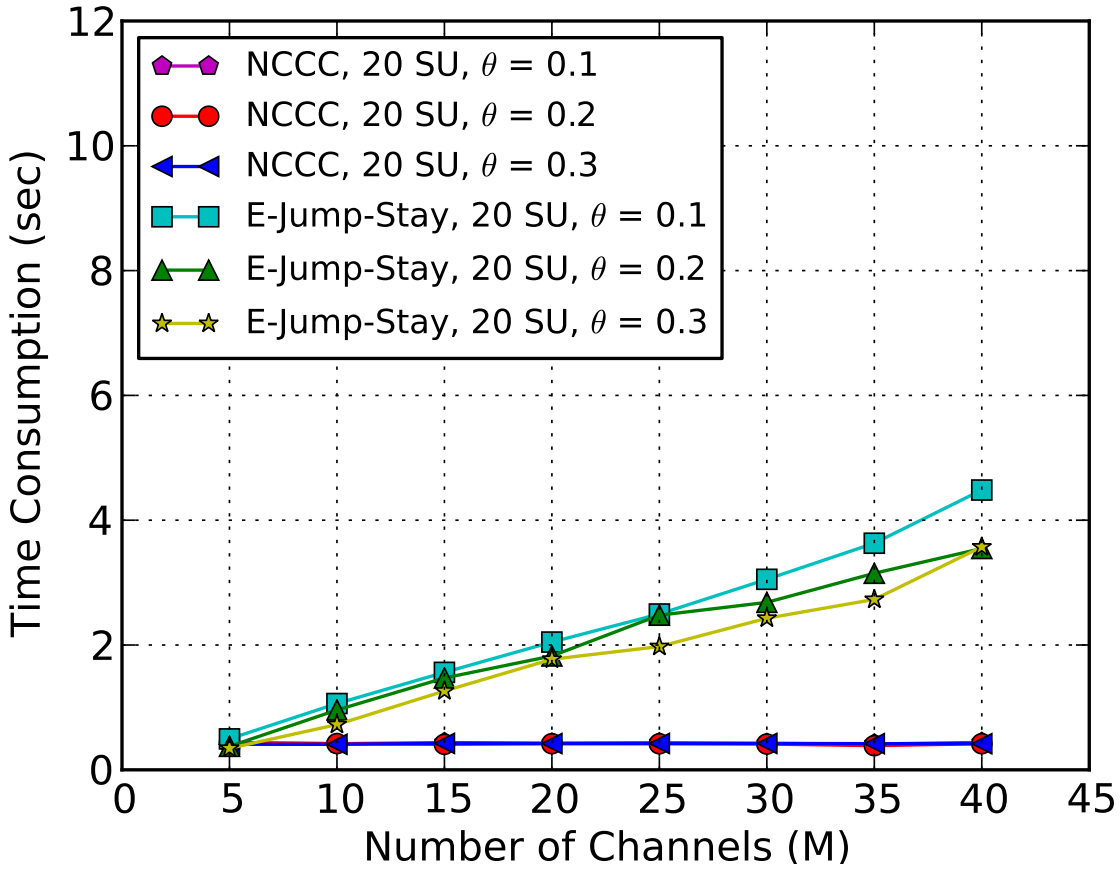


Figure 2.14: The time consumption of NCCC and the TTR of EJS algorithm under the symmetric models varying θ and M.

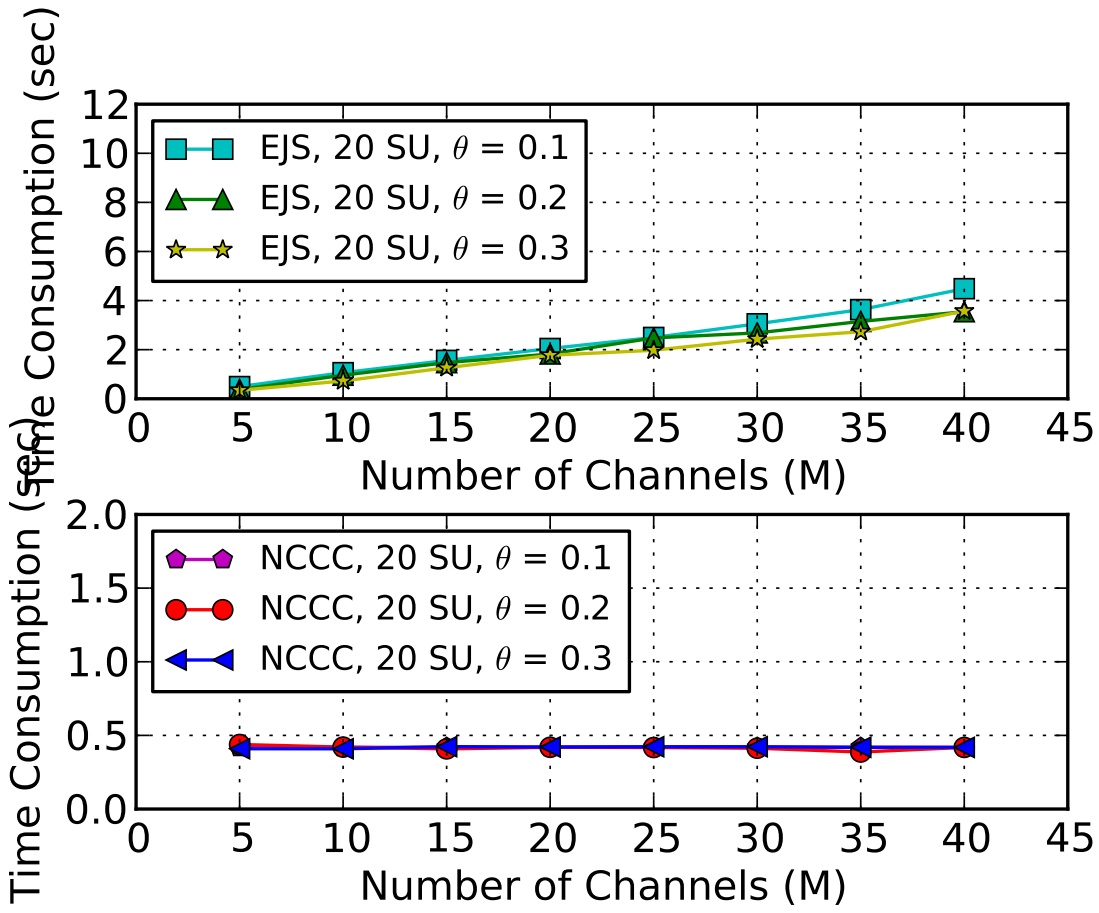


Figure 2.15: The time consumption of NCCC and the TTR of EJS algorithm under the symmetric models varying θ and M.

The results for symmetric models are shown in Figure 2.14 and 2.15. In symmetric models, the worst-case complexity of EJS algorithm is $O(P)$, where P is the smallest prime that is greater than M. More unavailable channels (higher θ) means there are less possible options in the EJS algorithm CH-sequences and higher possibility for SUs to rendezvous. When M is fixed, the network rendezvous in less time with higher θ. When there are only five total channels, the time consumption of EJS algorithm is slightly lower than that of NCCC. It is because NCCC needs to perform two rounds of negotiations to exchange R_l and R_o and the two rounds negotiation in our algorithm are inevitable. However, for NCCC in symmetric models, every probing is successful probing, because a channel is available to all SUs. M value becomes irrelevant to NCCC establishment and the time consumption becomes almost constant. In contrast, the TTR of the

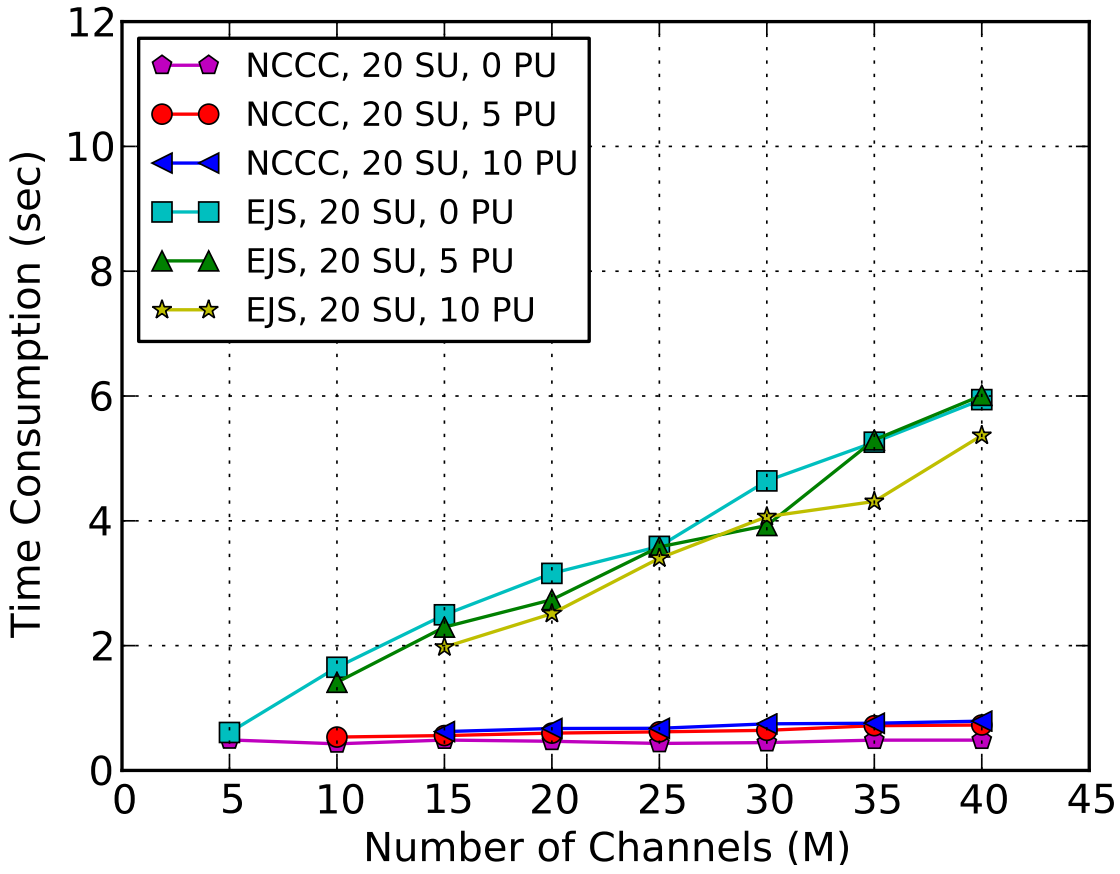


Figure 2.16: The time consumption of NCCC and the TTR of EJS algorithm under asymmetric models varying the total number of PUs and M. (Missing dots means the time/TTR is inapplicable.)

CH-based approaches increases along with the number of channels.

2.3.2 Asymmetric Model Time Consumption

In asymmetric models, for each randomly generated topology, we simulate by deploying PUs in the area and each PU occupies one randomly selected channel. The PUs also have transmission ranges of 250 meters. Then we compute the channel availability of SUs based on these PUs. If an SU u is located within a PU's transmission range, the corresponding channel is marked as unavailable to SU u. It is possible that a CCC does not exist in such kind of randomly generated asymmetric topologies. We perform six sets of simulations by varying the locations of PU, the amount of PUs and M. Each set of simulation consists of 50 randomly generated topologies.

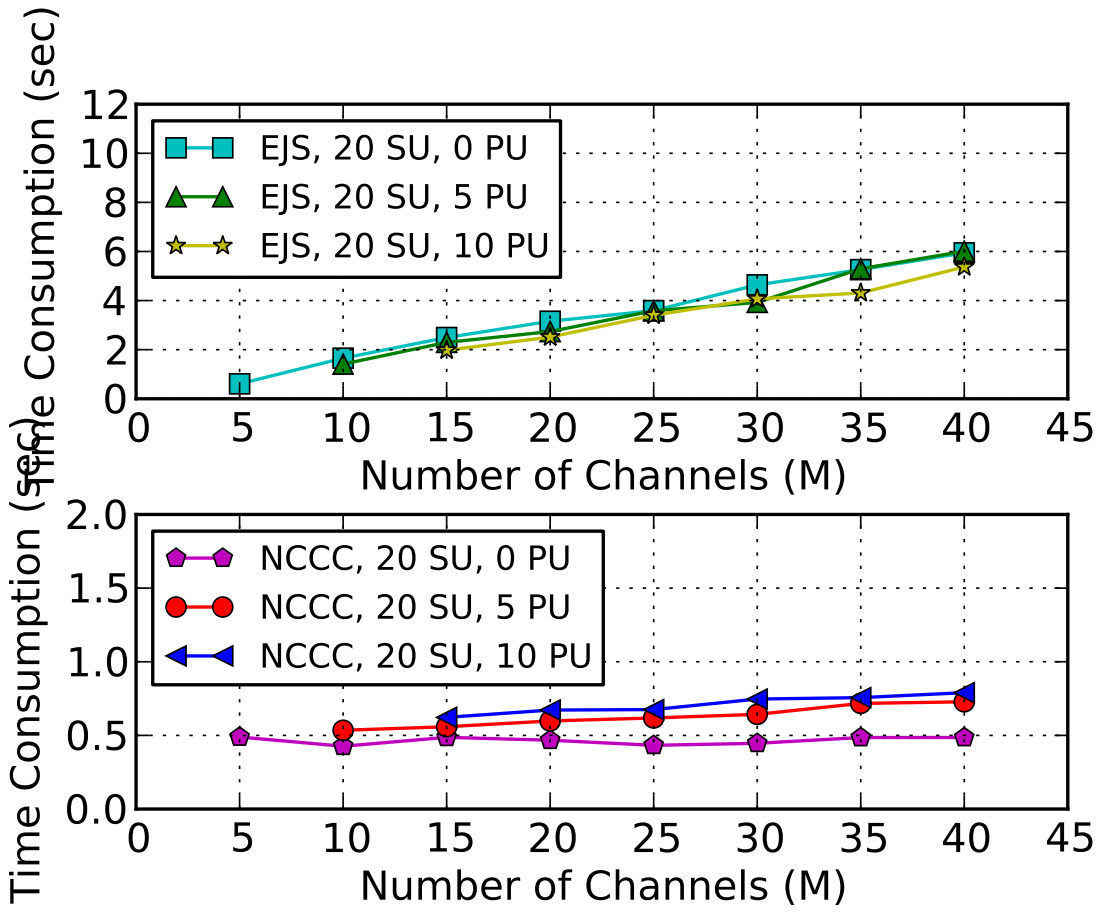


Figure 2.17: The time consumption of NCCC and the TTR of EJS algorithm under asymmetric models varying the total number of PUs and M. (Missing dots means the time/TTR is inapplicable.)

The results for asymmetric models are shown in Figure 2.16 and 2.17. If M is low, the number of possible available channels for an SU to hop on is higher and the chance of rendezvous increases. However, even in the scenario with 20 SUs, 0 PU, and five channels, the time consumption of our approach is 19.60% lower than EJS algorithm's TTR. The time consumption of NCCC is much lower than EJS algorithm's TTR as the number of channel increases. More PUs occupy more channels means less available channels for the SUs. NCCC start to have some failed probing. The time consumption starts to increase compared with that of scenarios of less PU due to failed probing. When there are 5 or 10 PUs in networks with $M = 5$, both NCCC and EJS algorithms cannot establish control channel, because some SUs do not have any available channel at all. Similar situations happen in some simulations where there are 10 PUs in networks with $M = 10$.

2.3.3 No Common Channel Available

One of the CH-based approaches' drawbacks is that it takes a long time for SUs to determine if there is no CCC available. Some of the CH-based approaches keep hopping on the CH-sequence indefinitely. If no CCC is available, for the upper bound of the CH-sequences can be determined, such as EJS algorithm [71], the SUs need to go through the whole CH-sequences. For CH-sequences whose upper bound is infinite, such as AMRCC [34], the SUs keep hopping on the CH-sequences indefinitely. In contrast, even if there is no feasible control channel allocation existing in a CRN, NCCC establishment approach can determine that there is no feasible allocation.

There are two possible ways for an SU u to discover that there is no feasible NCCC allocation. Firstly, suppose u is able to collect the R_o from all its neighbors. When computing the NCCC, u knows it is unable to establish control channel with v if $R(uv) = 0_{M \cdot K}$ for any $v \in N(u)$. Secondly, if an SU u has performed probing on all its available channels and it is still unable to negotiate with *any* of its neighbors, it can infer that the non-responding neighbors do not have any channel available for control purpose with u. Therefore, worst-case time consumption for our NCCC establishment approach to finding out that there is no feasible NCCC allocation is bounded by M. This information can be propagated to other SUs using their R_o.

The time consumption of our approach only slightly increases as the total number of channels M increases. It is because an SU gets all RU information R_l or R_o from at least one of its neighbor when there is a *successful probing*. In our simulation, every SU starts probing from channels with smaller indices. If an SU u probes on a channel that is available to all its neighbors, all its neighbors send negotiate packets to u and u can determine $R_o(u)$ right away. It is very likely that an SU successfully probes all its neighbors using the first few channels. As shown in the results, with our approach SUs are able to establish control channels in less time if there are more available channels. It is because there are more possible channels in the CH-sequences to hop on and the chance that two SUs hop to the same channel decreases. More possible channels lead to longer CH-sequences and the chance that two SUs hop to the same channel decreases.

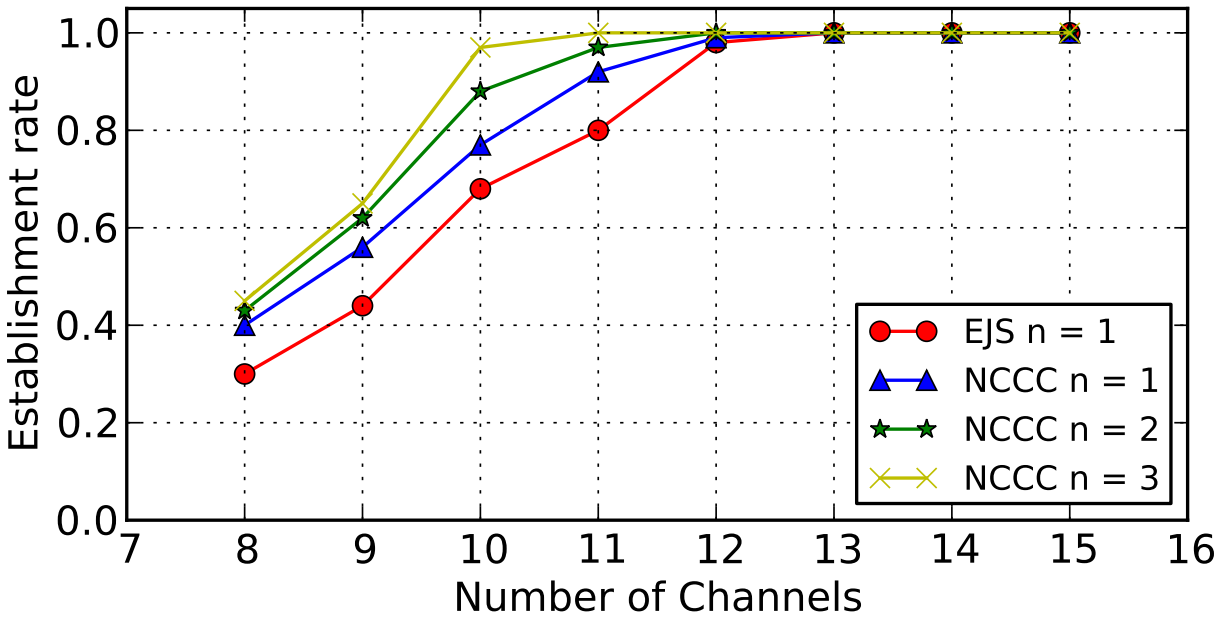


Figure 2.18: Control channel establishment rate for 20 SUs and 5 PUs varying the value of n.

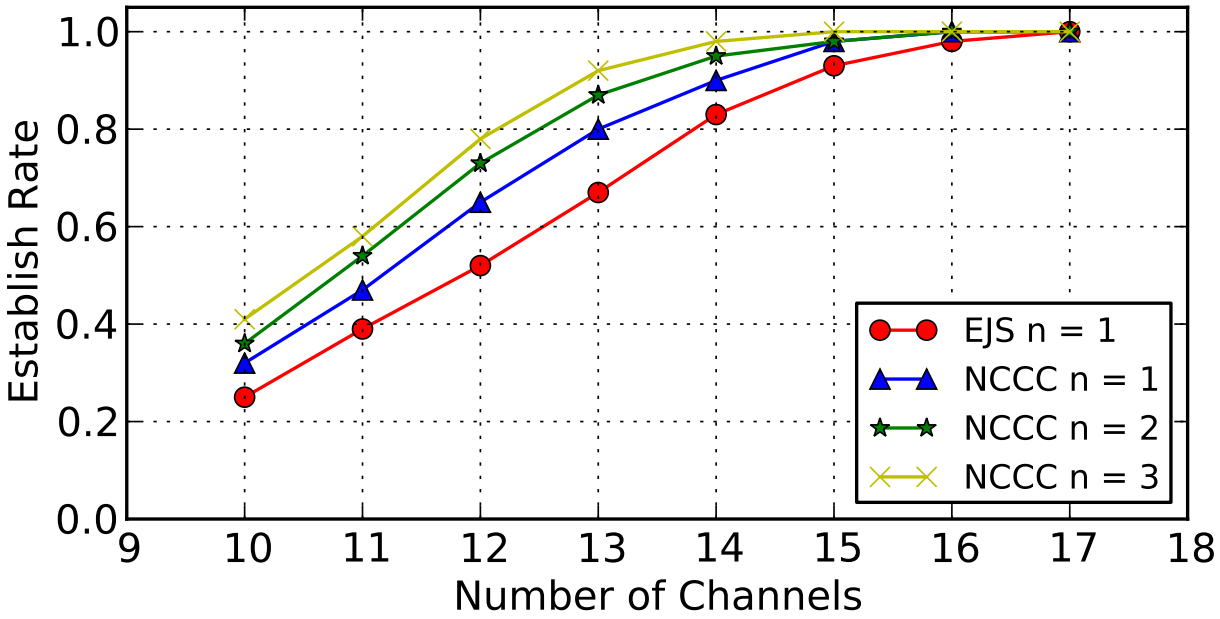


Figure 2.19: Control channel establishment rate for 20 SUs and 8 PUs varying the value of n.

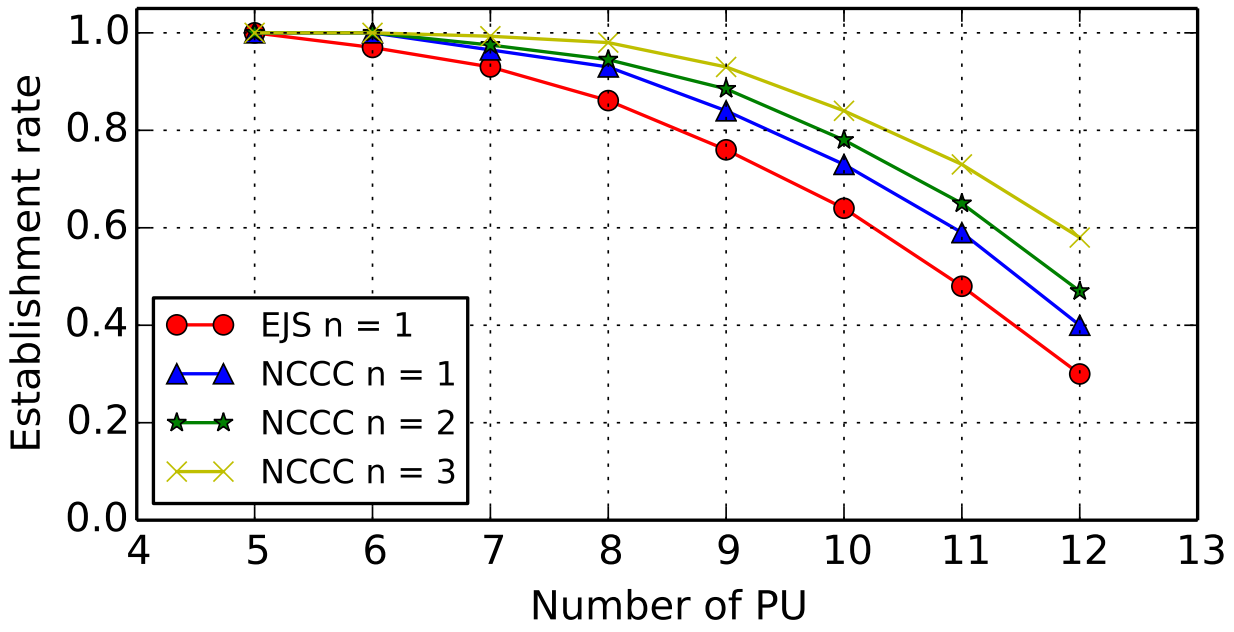


Figure 2.20: Control channel establishment rate for 15 channels, 20 SUs and varying the number of PUs.

2.3.4 Control Channel Establishment Rate

One of the main advantages of adopting NC-OFDM-based control interfaces is to access and allocate partial channels for control purpose. NC-OFDM-based control interfaces is able to form NCCCs while a CCC does not exist in the CRN. We conducted simulations using different topologies to gain some insight into the advantage of NCCC. We randomly generate three sets of 100 asymmetric topologies and deploy SUs and PUs randomly. Each of the PUs owns a channel and when the PU is active, the channel is considered busy and the SUs do not use the channel for control purpose. We define *establishment rate* to be the number of topologies that is able to establish control channels divides by the number of total topologies simulated.

The results are shown in Figure 2.18 and 2.19. We can see that the utilization of NC-OFDM-based control interfaces significantly improved the establishment rates even if the NC-OFDM-based control interfaces can only access spectrum bandwidth that is equal to one channel ($n = 1$). In Figure 2.18, the NCCC establishment rates are 13.8%, 21.8% and 27.2% higher than that of EJS algorithm for $n = 1, 2, 3$, respectively. In Figure 2.19, the NCCC establishment rates are 9.88%,

15.4% and 19.7% higher than that of EJS algorithm for $n = 1, 2, 3$, respectively. We also simulate the establishment rate of NCCC and EJS with fixed number of channels, 20 SUs and varying the number of PUs. The results are shown in Figure 2.20. More PUs means there are less available channel for SUs to find a CCC or form an NCCC. The NCCC establishment rates are 8.65%, 12.86% and 18.71% higher than that of EJS algorithm for $n = 1, 2, 3$, respectively. Note that not only the establishment rates of NCCC are higher than EJS algorithm, but the time consumption of NCCC establishment is also *at least 19.60% lower* than that of EJS algorithm.

2.4 Summary

In this chapter, we propose an efficient NC-OFDM control channel (NCCC) establishment method for distributed OFDM-based CRN. We propose a *probe and rendezvous* approach by taking advantage of OFDM subcarrier pulses. Our simulation results show that the time consumption to obtain a set of NCCC allocations is *at least 19.60% lower* than that in channel hopping-based schemes. The time consumption of our approach increases slightly along with the increase in the total number channels M for asymmetric models and are irrelevant to M for symmetric models. A common requirement of previous studies is that a channel across the network must be available to all SUs to serve as the control channel. However, such a channel may not exist in some CRNs. We propose the use of NC-OFDM-based control interfaces for control purpose so that the control channels can be established using non-contiguous spectrum fragments. In our simulations, the use of NC-OFDM control interfaces improves the control channel establishment rate by *at least 10%* even when the control interface can only access the spectrum bandwidth that is equal to one channel.

In CRNs, the SUs can establish communication sessions and perform communication tasks based on their needs. However, some of the characteristics of CRNs might break the routing path between two SUs, which is a critical issue that needs to be handled with care. To address this issue, we present the k-protected routing protocol in the next chapter.

CHAPTER 3

BUILDING K-PROTECTED ROUTES IN MULTI-HOP COGNITIVE RADIO NETWORKS

The second problem deals with the interruption to a routing path in CRNs when PUs become active again. In CRNs, the SUs reutilize the unused licensed spectrum as long as the SUs do not interfere with the owner of the licensed spectrum (i.e., the PUs). When a PU who owns the spectrum band starts to be active again, the SUs that are currently using the spectrum band must stop using it, which means the routing path might be broken and the ongoing communications might be interrupted. The possible interruption due to PU activity significantly limits the applications of CRNs, especially for quality-of-service (QoS) sensitive applications like audio and video conferencing. To address such potential interruptions, we present the k-protected routing protocol which builds routing paths that will not be interrupted even after k PU returns in CRNs.

In this chapter, *PU returns* means a PU becomes active again and is reclaiming the spectrum the PU owns. The SUs must relinquish the spectrum resource when the owner PU returns. A *link* between two SUs exits if the SUs are within each other's communication range. A *path* from SU s to d is a set of links that connect s to d with distinct intermediate SUs. The term *spectrum resources*, *spectrum holes* or *resources* denote idle spectrum fragments that can be utilized by the SUs for data communication.

An effective secondary network is expected to continue its operation even when PUs return. There are two major approaches for the SUs to handle returning PUs and to maintain the ongoing session. Firstly, the SUs perform *spectrum handoff* to another spectrum resource on the same affected links [67]. Secondly, the SUs replace the interrupted link with an alternative link or path, known as *backup path*[1] to maintain the ongoing session between the SUs [39, 83].

In either approach, the backup resource and path can be preassigned or determined reactively

[1] In the networks described in this chapter, there is only one link between any two SUs. If a link does not have any available spectrum resource for backup purpose, the protection must be done via a multi-hop *backup path*.

upon detecting PU activities. In the latter case, even if the SU attempts to find an alternative idle spectrum using the same link, the SUs need to perform spectrum sensing (tens to hundreds of milliseconds depending on the sensing technique [127]), neighbor discovery and channel switching (several μs). It takes even longer if the SUs attempt to find an alternative multiple-hop path to replace the broken link. The SUs reactively seeking for backup spectrum resources or even backup paths may cause delay to the ongoing session. In the worst case, it is possible that neither an alternative resource nor a path exists and the ongoing session is interrupted. The potential delay and possible interruption are obstacles towards realizing DSA and make CRN infeasible for quality of service-sensitive (QoS-sensitive) applications, such as audio/video conferencing, multimedia streaming, or DSA for future cellular networks. In this chapter, we study the problem of building k-protected routes in CRNs with one main resource and k preassigned backup resources or paths that are guaranteed to sustain from k PU returns.

In CRNs, the links that are interrupted due to returning PUs may perform spectrum handoff to another spectrum resource, and the same link is still perfectly functional. Moreover, the coverage of PUs is usually significantly larger than that of SUs. For example, the Federal Communications Commission (FCC) has approved the unlicensed access to TV band [29, 30], and Wi-Fi is one of the possible applications. The transmission range of TV stations can be as broad as tens of miles, while the unlicensed Wi-Fi devices have power limitation (several Watts depending on the types of the devices [30]) and limited range. Multiple SU links far apart from each other might be affected by one returning PU, while in wired networks, links that are physically far apart from each other are unlikely to fail simultaneously. Existing work on handling returning PUs in CRNs mainly focuses on minimizing the delay by spectrum handoff or backup path. Some of the previous studies [39, 131] also consider maintaining the connectivity of the CRNs on PU returns. However, *providing a guaranteed level of protection* from returning PUs is seldom discussed.

In this chapter, we study the problem of providing a guaranteed level of protection by building k-protected routes in CRNs with preassigned resources and non-contiguous orthogonal frequency division multiplexing (NC-OFDM) interfaces. NC-OFDM enables the SUs to access the spectrum

in a more flexible fashion, including aggregating multiple spectrum fragments and access partial channel. A k-protected route is a set of *main links* with *main spectrum resource* plus k sets of preassigned *backup spectrum resource* and *backup paths*. A k-protected route is guaranteed to sustain from k returning PUs without being interrupted. For every request made to a CRN, we want to compute a k-protected route considering the CRN's current spectrum resource availability for the request. Meanwhile, we want to maximize the number of k-protected sessions that can be supported in this CRN.

The preassigned backup paths and spectrum are activated upon PU returning with minimal delay. Moreover, the backup paths are interleaved with the main path, instead of edge-disjoint backup paths. When a PU appears, only *part of the route* is affected, and thus the activation delay is lower than edge-disjoint backup paths. k-protected routing protocol in CRNs is suitable for all kinds of applications. Applications that are intolerable to interruptions and are QoS-sensitive can issue requests with higher protection level, while applications that are insensitive to delay can issue requests with low protection level, even 0-protected route requests. For example, in cellular networks with DSA, making calls should request k-protected route with high k value, while a 0-protected route is sufficient for file transfer applications. We propose both centralized and distributed algorithms for building k-protected routes. Some preliminary results have been presented in [75].

We organize the rest of the chapter as follows. Section 3.1 provides some discussions on related work and the reasons why they cannot be applied to CRNs directly. We give a problem formulation and introduce our k-protected routing problem in Section 3.2. Our centralized and distributed algorithms are discussed in Section 3.3 and 3.4, respectively. Section 3.5 is devoted to evaluating our algorithms. Finally, we give a summary of this chapter in Section 3.6.

3.1 Related Work

Network protection for wired networks, especially for optical networks are widely studied [19, 58, 98, 103, 133]. Edge-disjoint backup paths, extra physical backup wires or preallocated

additional link capacity for restoration are usually required in these previous studies. It is because unlike link failures in wireless networks, link failures in wired networks are often caused by the damage of the wire medium such as the optical fibers. The link failure due to physical link cannot be recovered without human labor. In contrast, such kind of damage is less common in wireless networks.

The paper [98] discusses the survivability and restoration time of wavelength-division-multiplexing (WDM) optical network. Authors in [103] propose 1 + 1 protection by using edge-disjoint backup paths for single link failures in optical networks. The paper [58] proposes network protection for optical networks through network coding and *p*-cycle edge-disjoint paths. Each connection needs to carry coded information for other connection. In [19], edge-disjoint backup paths are needed for dual-link failure. Although [133] discusses multi-link failure and network survivability for optical networks, the scenario is data center networks, where the failed links are located in proximity. Instead of providing network protection in the lower layer, [60, 121] deliver to provide protection in the network layer.

The paper [14] discusses graph search-based approaches for failure recovery in software defined networks. Based on the OpenFlow fast-failover scheme, three algorithms are proposed to reach the flows' destination. However, the proposed algorithms are built on top of the OpenFlow fast-failover. DDC [78] guarantees connectivity, but might suffer from forwarding loops and might result in stretched routes. The authors in [25] achieved robust failover approach using static forwarding tables. Although these approaches deal with multi-link failure, they seek backup paths after failure happens and are designed for wired networks. These approaches cannot be applied to CRNs due to the intrinsic differences between wired and wireless networks.

In most wireless networks, common approaches for handling link failures include switching to an alternate channel and re-route current traffic. Protection of wireless networks is briefly discussed in [61] by scheduling traffic. However, such reactive approach does not offer guaranteed protection. The work [63] discusses guaranteed protection against single link failures for wireless networks with a single channel. Extra protection paths between source and destination are established with

scheduled time slots for accessing the wireless channel. The time slots are carefully allocated to links to ensure that when a wireless link fails, the source is still able to deliver data to the destination. However, this work focus on single channel and the slotted channel cannot provide stable performance regarding throughput. Many existing studies on network survivability [109] focus on improving the reliability of networks. In [85], the authors prepare predetermined backup paths for traffic flows by using a redundancy tree. However, in CRNs, links that are affected by one PU can switch to another channel and still be functional.

However, these previous studies mainly focus on keeping the network from *single link failure*. In contrast, the transmission range of the PUs is usually significantly larger than that of the SUs. Multiple links might be affected by one returning PU (e.g., Wi-Fi using TV whitespace), which makes the edge-disjoint path approaches insufficient. If two links in two edge-disjoint paths use the same spectrum resource of PU i, both paths are interrupted when PU i returns to active. Therefore, spectrum disjointness is also required to provide protection to CRNs. Although multi-link failure is discussed in [19, 133], the links are in proximity, and the links fail simultaneously due to physical damage.

Several previous studies attempt to maintain ongoing traffic flows after detecting link failures due to returning PUs. A self-healing algorithm for returning PUs is studied to improve the survivability of CRNs [68]. A spectrum aware routing algorithm is proposed in [94]. The algorithm attempts to re-route affected traffic after link failure due to returning PU. These reactive approaches not only suffer from significant delay time but also substantial traffic loss for recovering from link failures. Moreover, it is possible that the interruption cannot be recovered reactively and the session is interrupted.

Instead of reactively re-route after PUs return to active, proactively allocating backup spectrum and paths provides protection with lower delay. The authors in [67] propose to prepare backup spectrum for each link in the CRN. The SU switches to the backup spectrum when the PU reclaims the main spectrum. The authors in [83] formulate the problem of preassigning backup paths as an integer programming problem. In [36], two edge-disjoint and spectrum-disjoint paths are

required for each session for protecting from returning PUs. On the other hand, several previous studies [39, 131] discuss maintaining the connectivity of CRN when PUs return to active. The paper [39] studies multiple links handoffs while maintaining connectivity and minimize total completion time. In [131], one or more channels are assigned to each link with a minimal level of interference to maintain connectivity when PUs appear. Existing work on CRNs mainly focuses on minimizing the delay time for handling returning PUs by spectrum handoff and/or backup path. However, guaranteed network protection from returning PUs for QoS-sensitive applications is seldom discussed.

3.2 System Model and Problem Formulation

In this section, we present the problem formulation and introduce the k-protected routing.

3.2.1 System Model

We consider a multi-hop mesh cognitive radio network that consists of multiple SUs, and there is a primary network with $\mathcal{P}$ PUs located in the same geographical area. As mentioned above, the SUs might initiate requests with different bandwidth requirement B and protection level k. We want to maximize the number of sessions that can be supported by the CRN.

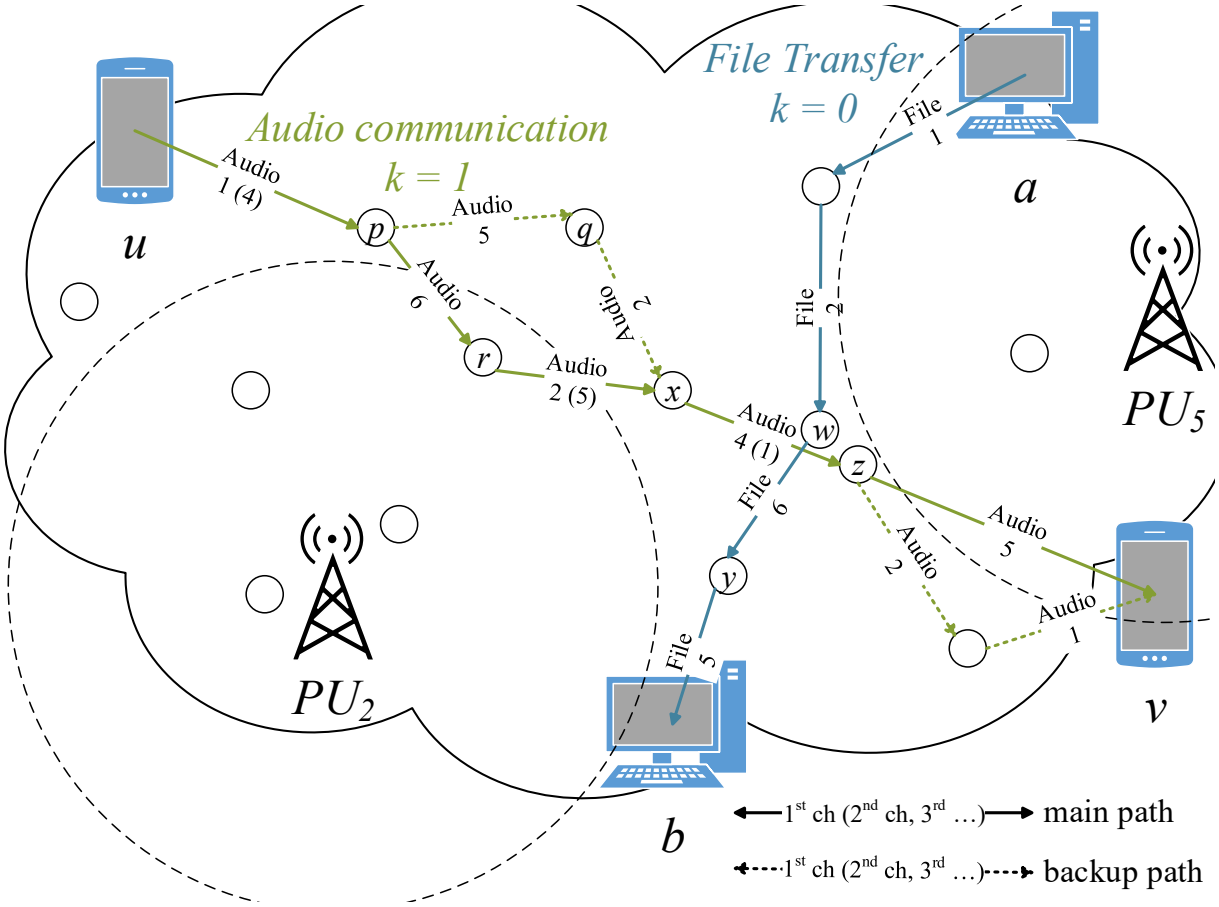


Figure 3.1: An example of a 1-protected route (green, labeled as “Audio" 300 kbps [107]) and a 0-protected route (blue, labeled as “File", bandwidth varies). There are eight channels, labeled as 0 - 7, in the network. For easy presentation, PUs that are not shown cover the whole area. The interference model is two-hop interference model [57][2].The solid/dashed lines denote the primary/backup link. If multiple channels are on a link, the link is assigned with a primary frequency and some backup frequencies.

A network with one 0-protected session and one 1-protected session is shown in Figure 3.1. PU i denotes the PU who owns the access right to channel i. PU 0, PU 3, and PU 7 are currently active and accessing the spectrum they own (i.e., channel 0, 3 and 7 are not available). A 1-protected route for an audio communication session from SU u to v and a 0-protected route for a file transfer session between SU a and b are established. The 1-protected session will not be interrupted no matter any

[2]Two-hop interference model means node N's transmission using channel C interferes with N's two-hop neighbor. Therefore, N's two-hop neighbors need to avoid transmission using channel C. For example, in Figure 3.1, cellphone u transmits using channel 1 and thus u's two-hop neighbors (i.e., p, q and r) must not use channel 1 again.

PU appears. Audio communication applications might require lower bandwidth (smaller b) than file download, but audio communication applications require relatively stable bandwidth, and they are less delay-tolerant (higher k).

According to Skype[107], the recommended video calling bandwidth is 1.5 Mbps. Take TV white space [30, 40] for instance, if a 6 MHz channel used to carry Wi-Fi signals for a hundred meters, the capacity of a 6 MHz channel is significantly larger than 1.5 Mbps, allocating the whole channel to a video communication session is a waste of resource. Therefore, flexible spectrum assignment is desired and can be achieved using non-contiguous orthogonal frequency division multiplexing (NC-OFDM) interfaces.

With NC-OFDM, a routing protocol can assign the exact amount of radio resource to each link that satisfies the bandwidth requirements of the different requests. In this chapter, we discuss the SUs equipped with NC-OFDM based radio interface, and we allocate spectrum fragments to the interfaces exclusively. None of the SUs in the interference range of the spectrum fragment can reuse the same fragment. In other words, a link is guaranteed not to experience any interference from any other nodes, because none of the nodes in this link's communication range will have access to the spectrum fragment. For the example shown in Figure 2.2, the green spectrum fragment in channel i is assigned to SU x and is not interfered by SU w accessing the blue spectrum fragment in channel i.

3.2.2 Problem formulation

Consider an OFDM-based multi-hop CRN consists of S SUs, $S \geq 2$, and each SU is denoted as SU i. There is one or more PUs located in the area. The SUs can sense the spectrum (channels) and detect PU activities. That the SUs do not know when the PUs will return, but the SUs remembers which PU might affect itself after a PU has ever returned. We can let each SU keep a history record of all PUs that have affected the SU.

The term *spectrum resources*, *spectrum holes* or *resources* denote idle spectrum fragments that can be reutilized by the SUs. Let M be the set of spectrum resources that the NC-OFDM interface

can sense and operate on. Let us denote the set of spectrum holes at SU i as M_i, for each $i \in S$. For SU u and v at different locations, different PUs may affect them and it is very likely that $M_u \neq M_v$, for $u \neq v$.

We equip each SU with three interfaces comprising an outgoing interface (Tx), an incoming interface (Rx), and a control interface, similar to the NS-3 cognitive radio extension [6]. The Tx and Rx are NC-OFDM interfaces for data transmission, and the control interface is a legacy interface that accesses unlicensed spectrum (i.e., 2.4 GHz). The control interface is only responsible for exchanging control messages as in [24, 104]. The SUs attempt to allocate spectrum resources for Tx and Rx dynamically.

The Tx and Rx are the data interfaces that can access spectrum that spans over n channels. It is because the wider the interface accesses, the higher sampling rate it requires, and the more power it consumes. For battery-powered SU, a smaller n makes more sense to them and for SUs that power supply is not an issue, higher n can be employed. The spectrum resource allocation of a radio interface Tx and Rx on SU i is denoted as Tx_i and Rx_i, respectively. Moreover, the NC-OFDM interfaces can send and receive data from multiple neighbors as long as two data interfaces share some common spectrum resources.

We say a link l_{uv} exists between the Tx on SU u and the Rx on SU v if the two SUs are within each other's transmission range. Let $Allocate(l_{uv}) = \{f_0, f_1, ..., f_k\}$ denote the spectrum resources allocated to link l_{uv}. A spectrum resource f_x can be allocated to l_{uv} (i.e., $Allocate(l_{uv}) = f_x \cap Allocate(l_{uv})$) if and only if l_{uv} exists, $f_x \in M_u$ and $f_x \in M_v$. SU u can send data to v using l_{uv} if and only if $Allocate(l_{uv}) \cap Tx_u \neq \emptyset$, and $Allocate(l_{uv}) \cap Rx_v \neq \emptyset$. The amount of spectrum resources (in MHz) allocated to the link l_{uv} is defined as

$$|Allocate(l_{uv})| = \sum_{f_x \in Allocate(l_{uv})} |f_x| \tag{3.1}$$

where $|f_x|$ denotes the bandwidth of the spectrum resource f_x in MHz. Moreover, the spectrum resources allocated to the link l_{uv} needs to satisfy the bandwidth requirement, which means

$$|Allocate(l_{uv})| \geq B \tag{3.2}$$

where B is the bandwidth requirement in MHz, which will be discussed in detailed later. We now define the protection level $Protect(l_{uv})$ as follows.

$$Protect(l_{uv}) = \left\lfloor \frac{|Allocate(l_{uv})|}{B} \right\rfloor \tag{3.3}$$

If Eq. 3.2 is satisfied, $Protect(l_{uv})$ is greater or equal to 1.

A path from SU s to d is denoted as $P_{sd} = \{l_{si}, l_{ij}, ..., l_{kd}\}$. The paths in k-protected route must be loopless, which means the path must not visit any intermediate node more than once. The amount of spectrum resources allocated on a path P is defined as $|Allocate(P)| = \sum_{l \in P} Allocate(l)$. The protection level on a path P can be expressed as

$$Protect(P) = min(Protect(l), \forall l \in P) \tag{3.4}$$

Take Figure 3.1 for example. Assume the bandwidth requirement is B and the bandwidth of a channel in the figure is also B. Although $|Allocate(l_{rx})| = 2B$ and $Protect(l_{rx}) = 2$, the path $P_{prx} = \{l_{pr}, l_{rx}\}$, $Protect(P_{prx}) = min(Protect(l_{pr}), Protect(l_{rx})) = 1$ and $Protect(P_{pqx}) = 1$.

For each communication request between SU s and d with a specified protection level k, we compute the k-protected route from s to d. A k-protected route from SU u to v (denoted as L_{uv}) is one set of *main path* with *main spectrum resource* plus k sets of preassigned *backup spectrum resources* and *backup path*s that is guaranteed to sustain from k returning PUs to active without being interrupted.

The *main links* with *main spectrum resources* are the links in the path that are activated right after the k-protected route is established, while the *backup spectrum resources* and *backup path*s are activated when PU returns and the k-protected route has to relinquish the main spectrum resource. Take Figure 3.1 for example. P_{prx} is the main path. Channel 6 on l_{pr} is a main spectrum resource and channel 5 on l_{rx} is a backup resource. P_{pqx} is a backup path and channel 5 on l_{pq} is a backup spectrum resource.

The protection level of a k-protected route $Protect(L)$ is defined as $\sum_{P \in L} Protect(P) - 1$. (We minus one because one path is used as the main path and main resource.) The 1-protected route L_{px} in Figure 3.1 is $\{P_{prx}, P_{pqx}\}$ and it has *one* main path (P_{prx}) and *one* backup path

(P_{pqx}), $Protect(P_{prx}) + Protect(P_{pqx}) = 2$. Thus, $Protect(L_{px}) = \sum_{P \in L} Protect(P) - 1 = 1$. If a k-protected route cannot be found, our protocol rejects this request. The k-protected route has to satisfy the following constraints.

Exclusive Spectrum Fragment Allocation: In wireless communications, two wireless nodes sharing the same spectrum might interfere with each other. In this chapter, we consider two-hop interference model [57]. In two-hop interference model, if an SU u is transmitting using channel c, all two-hop neighbors of node SU u (denoted as $2hop(u)$) who also transmit using channel c may be interfered, which is undesirable. Therefore, in this chapter, we eliminate such kind of interference by allocating spectrum fragments exclusively to a link. Any other SUs in two-hop neighborhood to SU u cannot reuse the spectrum fragment. The constraint is defined as follows.

$$\begin{gathered} Allocate(l_{uv}) \cap Allocate(l_{xy}) = \emptyset, \forall l_{uv}, l_{xy} \in L, \\ u, v \notin 2hop(x) \cup 2hop(y) \wedge x, y \notin 2hop(u) \cup 2hop(v). \end{gathered} \tag{3.5}$$

Bandwidth Requirement: Suppose a request needs a data rate of b, all links in L must allocate spectrum fragment that is able to satisfy the bandwidth requirement presented as follows. Based on Shannon-Hartley theorem [106], the capacity of a spectrum fragment of bandwidth is The capacity allocated must satisfy the requested data rate b.

$$Capacity = B \cdot log_2(1 + SNR) \geq b \tag{3.6}$$

where B is the bandwidth in MHz, and SNR is the signal-to-noise ratio. The spectrum resources allocated to the links in L must satisfy bandwidth requirement.

$$|Allocate(l)| \geq B \geq \frac{b}{log_2(1 + SNR)}, \forall l \in P, \forall P \in L \tag{3.7}$$

As mentioned in previous constraint, we allocate spectrum fragments to links exclusively. The SNR in the same channel is high and we consider it to be a fixed value.

Note that in this chapter, although we consider the NC-OFDM interface allocating multiple spectrum fragments on a link, we *do not* consider using multiple paths to satisfy one bandwidth requirement. If a bandwidth requirement cannot be satisfied at a link (Eq. 3.2) with its available

spectrum resource, the k-protected routing protocol simply does not consider this link. For example, consider a 1-protected route requests with bandwidth requirement corresponding to 5 Mbps, we want to find two paths, in which all links can provide a capacity of 5 Mbps. We will not seek five 1 Mbps path as the main links and another set of five 1 Mbps paths to form the backup path. It is because, in this case, each link on the paths is only carrying part of the traffic. When a link is down due to PU activity, all links in all five paths needs to be informed and stop working. Similarly, all links in all five paths in the backup path needs to be activated. It is inefficient in both building the k-protected route and recovering from PU returns.

Cost: Due to the exclusive spectrum fragment allocation constraint, when allocating a spectrum fragment f_x to a link l_{uv}, all links involved in any SU in $2hop(u) \cup 2hop(v)$ are no longer available for allocation. Let us denote the set of links involved in any SU in $2hop(u) \cup 2hop(v)$ as $E(l_{uv})$. In other words, allocating a spectrum resource f_x to l_{uv} costs all f_x at all links in $E(l_{uv})$. Let us define the cost of assigning a set of spectrum resource (F) to the link l_{uv} as the following equation.

$$cost(l_{uv}, F) = \sum_{f_x \in F} |E(l_{uv})| \cdot |f_x| \tag{3.8}$$

Therefore, the cost of allocating a spectrum resource f_x to l_{uv} can be expressed as

$$cost(l_{uv}, \{f_x\}) = |E(l_{uv})| \cdot |f_x| \tag{3.9}$$

The cost of allocating the set of spectrum resources $Allocate(l_{uv})$ to l_{uv} can be expressed as

$$cost(l_{uv}, Allocate(l_{uv})) = \sum_{f_x \in Allocate(l_{uv})} cost(l_{uv}, f_x) \tag{3.10}$$

The cost of a path route P can be defined as

$$cost(P) = \sum_{l \in P} cost(l, Allocate(l)) \tag{3.11}$$

Similarly, the cost of a k-protected route L is therefore defined as

$$cost(L) = \sum_{P \in L} cost(P) = \sum_{P \in L} \sum_{l \in P} cost(l, Allocate(l)) \tag{3.12}$$

In Figure 3.1, assume the bandwidth of a channel is B. $|l_{pr}| = B$ (one channel, channel 6) and both p's and r's two-hop neighbors cannot reuse channel 6. Within l_{pr}'s two-hop neighborhood, there are 5 links (i.e., l_{up}, l_{pq}, l_{qx}, l_{rx} and l_{xz}) and therefore $cost(l_{pr}, \{channel\ 6\}) = 5B$. As a result, when building the latest k-protected route for a request, we need to minimize the cost of resources for the request. With minimized cost for serving L, the available spectrum resources left over after L is granted is maximized and the network can serve more future requests. Therefore, the we propose algorithms that attempt to optimize the following function with different bandwidth requirement B and level of protection k.

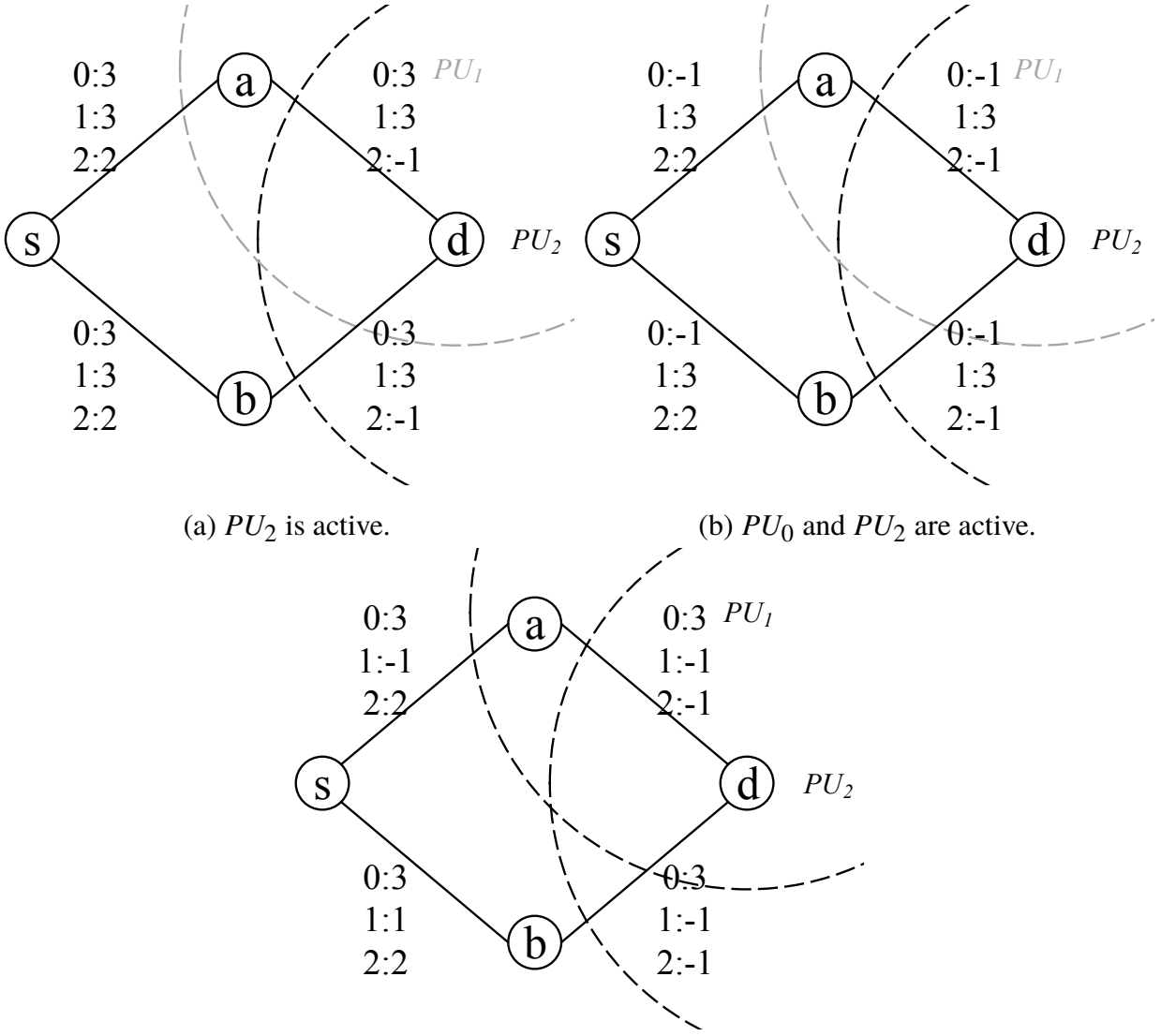
$$Minimize\ cost(L) = Minimize \sum_{P \in L} \sum_{l \in P} (cost(l, Allocate(l)) \tag{3.13}$$

with the following constraints.

$$\begin{aligned} &|Allocate(l)| \geq B, \forall l \in P, \forall P \in L \\ &Protect(L) \geq k \end{aligned} \tag{3.14}$$

Spectrum Handoff and Re-route Constraint: Suppose we are building a k-protected route for a given communication request from SU s to d. Intuitively, we can enumerate all possible paths and allocations from s to d. For each allocation, we compute their costs for current spectrum resource availability and for handling different combinations of returning PUs.

There can be different resource allocation on a path P ($Allocate(P)$). Obviously, their cost is also different when different PU is active. Let $cost(P)_{sd_{i,j}}$ denote the jth lowest cost of allocation to path P_{sd} when PU_i is active. For example, $cost(P)_{sd_{2,1}}$ denotes the second cost of the path P_{sd} when PU 2 is active.



(a) PU_2 is active.

(b) PU_0 and PU_2 are active.

(c) PU_1 and PU_2 are active.

Figure 3.2: An extreme case. The number pairs at the links (l) denote *resource:cost(l, resource)*. -1 denotes the channel is unavailable at this link and therefore the cost is -1. For link l_{ad}, $cost(l_{ad}, channel\ 0)$ is 3 and because channel 2 is unavailable, $cost(l_{ad}, channel\ 2)$ is -1. Suppose PU_1 was active before and SUs (i.e., a, d) that are in PU_1's coverage know it. PU_2 is active, and PU_0 are unknown to the SUs. For easy presentation, the interference model in this figure is one-hop interference model. Suppose we are building a 1-protected path with current channel availability in subfigure (a). PU_0 appears in the middle figure after the 1-protected route is established in subfigure (a). PU_0's location is currently unknown, the cost of PU_0 returning is shown in subfigure (b). Subfigure (c) shows the cost after PU_1 appears after subfigure (a).

Let $C(i)$ denote the set of costs when PU i appears. Suppose we are constructing a 1-protected route L_{sd} using this approach, we will have a set of costs for a 1-protected route. We then sort each

element in every $C(i)$ in ascending order and we will have the following cost vectors.

$$\begin{aligned} C(0) &= \{cost(P)_{sd_{0,0}}, cost(P)_{sd_{0,1}}, ..., cost(P)_{sd_{0,n}}\}, \\ C(1) &= \{cost(P)_{sd_{1,0}}, cost(P)_{sd_{1,0}}, ..., cost(P)_{sd_{1,n}}\}, \\ &... \\ C(\mathcal{P}-1) &= \{cost(P)_{sd_{\mathcal{P}-1,0}}, cost(P)_{sd_{\mathcal{P}-1,1}}, ..., cost(P)_{sd_{\mathcal{P}-1,n}}\} \end{aligned} \tag{3.15}$$

We can simply use the lowest cost paths in each group to form a k-protected path $L = \{P_{sd_{0,0}} \cap P_{sd_{1,0}} \cap ... P_{sd_{\mathcal{P}-1,0}}\}$.

In Figure 3.2(a), allocating channel 2 to l_{sb} can be denoted as $Allocate(l_{sb}) = \{channel\ 2\}$ and its cost $cost(l_{sb}, \{channel\ 2\}) = 2B$, where B is the channel bandwidth. For easy presentation, let us denote an allocation to path $P_{uxv} = \{l_{ux}, l_{xv}\}$ as $[u-(f_i) \rightarrow x-(f_j) \rightarrow v, cost]$ In Figure 3.2(a), an allocation to path P_{sbd} can be denoted as $Allocate(P_{sbd}) = [s-2 \rightarrow b-0 \rightarrow d, 2+3]$. Suppose a 1-protected route is being built in Figure 3.2(a) when PU_2 is already active. The cost vector $C(PU_0)$ of L_{sd} for the extreme case shown in Figure 3.2(a).

$$\begin{aligned} C(PU_0) = \{&[s-(2) \rightarrow a-(1) \rightarrow d, 5], \\ &[s-(2) \rightarrow b-(1) \rightarrow d, 5]\}, \\ C(PU_1) = \{&[s-(1) \rightarrow b-(0) \rightarrow d, 4], \\ &[s-(2) \rightarrow b-(0) \rightarrow d, 5], \\ &[s-2 \rightarrow a-0 \rightarrow d, 5]\}, \\ C(PU_2) = \{&[s-(2) \rightarrow b-(1) \rightarrow d, 5], \\ &[s-(2) \rightarrow a-(1) \rightarrow d, 5], \\ &[s-2 \rightarrow b-0 \rightarrow d, 5], \\ &[s-2 \rightarrow a-0 \rightarrow d, 5], \\ &...\}, \end{aligned} \tag{3.16}$$

Intuitively, to form a 1-protected route, for handling PU_0 returning, we pick the lowest cost path $[s-(2) \rightarrow a-(1) \rightarrow d, 5]$, for handling PU_1 returning, we pick the lowest cost path

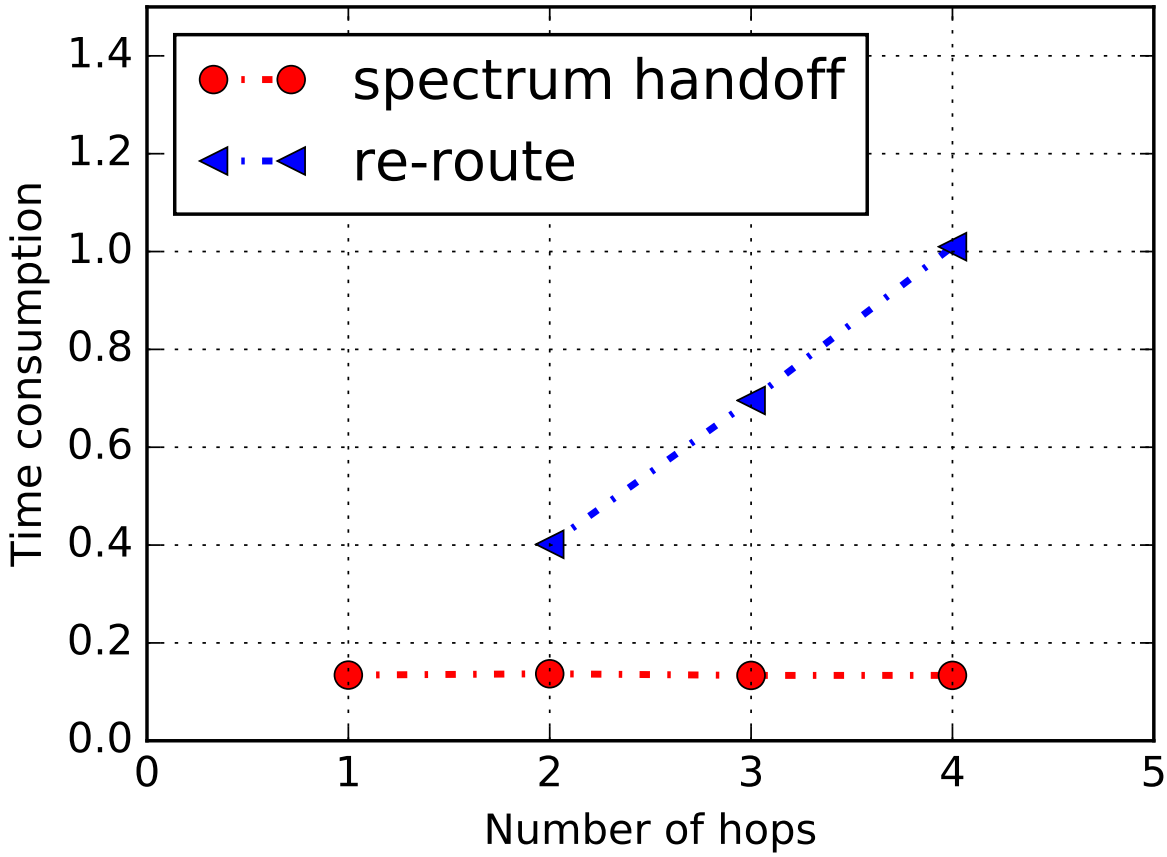


Figure 3.3: Switch delay for backup channel and backup path. The x-axis denotes the number of hops. A one-hop handoff is definitely a spectrum handoff, because there is only one link between a pair of wireless nodes.

$[s-(1) \rightarrow b-(0) \rightarrow d, 4]$, and for handling PU_2 returning, we pick the lowest cost path $[s-(2) \rightarrow b-(1) \rightarrow d, 5]$. The total cost is $5+4+5=14$, which is the optimal cost.

However, lowest cost allocation may not be the optimal solution in terms of delay, because they might require longer spectrum handoff and re-route time than allocations with higher cost with fewer re-routes. As shown in Figure 3.3, in our simulation, we observed that the time required for re-routing is significantly higher than that of spectrum handoff. In CRE-NS3 [6], the SUs perform spectrum sensing every 100 ms. It is because every SU itself detects PU activity and performs spectrum handoff right away. On the other hand, the SUs on the backup path need their neighbors to inform the activation of the backup path hop by hop, which is more time consuming than spectrum

handoff.

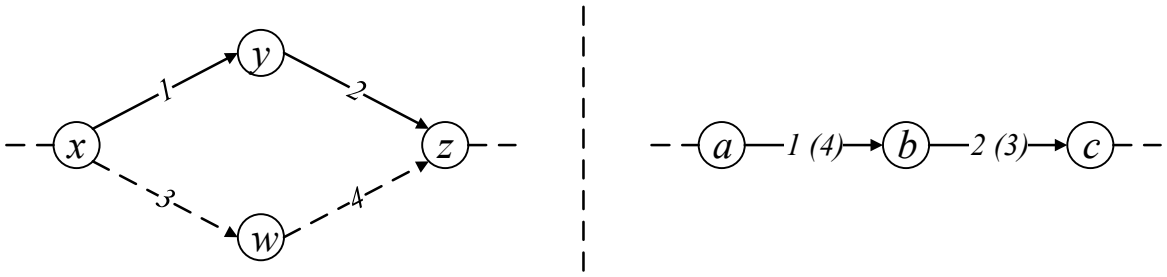


Figure 3.4: Difference between backup path and backup spectrum.

Considering the example shown in Figure 3.4, the path P_{xy} in left hand side has a backup path P_{xwz} and the path P_{ac} in the right hand side has backup channels. Suppose PU_2 returns, l_{bc} simply switches to backup spectrum in channel 3 while l_{yz} needs to inform x and y about the failure of l_{yz}. x needs to activate the two-hop backup path from x to z. Therefore, the final constraint is to minimize the number of backup paths and only to employ backup path if and only if a feasible backup spectrum does not exist.

Take the aforementioned lowest cost solution for example, it is far from a good solution in terms of spectrum handoff and re-route. It requires two spectrum handoffs for handling PU_1 returns to active, including switching l_{sb} from channel 2 to channel 1 and switching l_{bd} from channel 0 to channel 1.

A better solution for the example shown in Figure 3.2 is $[s-(2) \rightarrow b-(1) \rightarrow d, 5]$ (for PU_0 returns to active), $[s-(2) \rightarrow b-(0) \rightarrow d, 5]$ (for PU_1 returns to active), and $[s-(2) \rightarrow b-(1) \rightarrow d, 5]$ (PU_2 is already active). Although the total cost is 15 in this solution, it requires only 1 spectrum handoff for handling PU_1 returns to active. Therefore, instead of choosing minimum cost allocation from the cost vector, the *allocation difference* needs to be takin into account. For example, the allocation difference of $[s-(2) \rightarrow b-(1) \rightarrow d, 5]$ and $[s-(1) \rightarrow b-(0) \rightarrow d, 4]$ is 2 and the allocation difference of $[s-(2) \rightarrow b-(1) \rightarrow d, 5]$ and $[s-(1) \rightarrow b-(0) \rightarrow d, 5]$ is only 1.

However, this approach is unsuitable for solving the k-protected routing problem for the follow-

ing reasons. Firstly, enumerating all paths between a pair of SUs is a computationally expensive task. Secondly, it has $\binom{|PU_{off}|}{k}$ different possible combinations of returning PUs where $|PU_{off}|$ is the number of inactive PUs and k is the desired level of protection. The cost evaluation of all possible paths on all possible combinations is a huge task. Therefore, instead of enumerating all possible allocation and choose the minimum cost allocations, we propose greedy algorithms that attempts to allocate as much backup resources as possible before seeking backup paths.

3.2.3 Hardness of the Problem

We prove the hardness of k-protected routing problem by reducing the k-protected routing problem to the optimum cost chromatic partition problem (OCCP) [54].

OCCP: OCCP is described as follows. For a graph $G = (V, E)$ with n vertices and a coloring cost vector $K = \{k_0, k_1, ...k_{n-1}\}$. The OCCP problem finds a feasible coloring $f(v)$ for each vertex for all $v \in V$, such that the total costs $\sum_{v \in V} (k_{f(v)})$ are minimum. A coloring $f(v)$ is said to be feasible, if adjacent vertices are assigned with different colors. The input to OCCP is (G, K), where G is a graph (V, E) with $|V| = n$ vertices, and K is a coloring cost vector $\{k_0, k_1, ...k_{n-1}\}$. The output is $f(v)$, which is the minimum cost coloring that the adjacent vertices in G are assigned with different colors.

k-protected routing: The input to k-protected routing is (G, M, k, B), where G is a graph (V, E), V is the set of SUs, E is the set of links, M is the spectrum resources, k is the level of protection, and B is the bandwidth requirement. The output is the L, which is the k-protected routing.

We verify k-protected routing is NP. For a solution $L \in E$ and $Allocate(L) \in M$. Accept the solution if all the following statements are all true. Otherwise, reject the solution.

- $Protect(L) \geq k$
- $Allocate(l) \geq B, \forall l \in P, \forall P \in L.$

We now show k-protected routing is at least as hard as OCCP. Firstly, we reduce k-protected routing to 0-protected routing by considering the simplest version of our k-protected routing problem, the 0-protected routing. This step can be done in polynomial time (actually constant). Given a pair of source and destination SUs in the CRN, we can find a shortest path (P) from the source node s to the destination node d by using any algorithm that solves *single-pair shortest path problem*. We have a subgraph $G' = (V', P)$, where V' is the set of vertices visited when traverse from node s to d via P, $P \subset E$ and $V' \subset V$.

At this stage the links in P do not have any spectrum resources. For each link $l_{uv} \in P$, there is an array of candidate spectrum resources that can be expressed as $M_u \cap M_v$. The cost of a spectrum resources $f_x \in M_u \cap M_v$ is $cost(l_{uv}, \{f_x\})$. Therefore, for each link $l_{uv} \in P$, there exists a cost vector K_{uv}.

We now convert $G' = (V', P)$ to another graph H, where a vertex in H is an edge in G'. A vertex t_{uv} is added to H, if l_{uv} is in P. An edge l_v is added to H if two edges l_{uv} and l_{vx} are in P. For example, $P = \{l_{su}, l_{uv}, l_{vd}\}$ can be converted to another graph $H = (V_H, E_H)$, where $V_H = \{t_{su}, t_{uv}, t_{vd}\}$ and $E_H = \{l_u, l_v\}$. This also can be done in polynomial time by performing a graph traversal in any order.

Since an edge $l_{uv} \in P$ is converted to the vertex $t_{uv} \in V_H$, the color vector K_{uv} is also assigned to t_{uv} At this stage, the cost vector on each vertex in H (each edge in P) might be different as the cost of a link is defined in Equation 3.13. However, we further reduce the 0-protected routing to let the cost vector on each vertex in V_H be the same, say K'. The final 0-protected routing problem of (V_H, K', k, B) (k and B is irrelevant), is now equivalent to OCCP. All the reduction steps can be done in polynomial time.

In summary, we can reduce our k-protected routing problem 0-protected routing problem and reduce 0-protected routing problem again by using the same cost array for all $v \in V_H$ to make the problem equivalent to OCCP. Therefore, k-protected routing problem is NP-hard. To solve this NP-hard problem, we propose both centralized and distributed algorithms for finding k-protected routes.

3.3 Centralize Algorithm

In this section, we propose a *centralized recursive k-protected route establishment (CRKRE)* algorithm for finding a k-protected route between s and d with bandwidth requirement B. From the observation that a spectrum handoff incurs lower delay than re-route, our CRKRE finds a shortest path first, attempts to satisfy the level of protection by allocating resources on the path and seeks backup routes only if no more backup spectrum can be found to satisfy the level of protection.

Given a network G with current spectrum availability and a request from s to d with bandwidth requirement B and protection level of k, the CRKRE finds a set of k-protected route for the request. The basic idea of CRKRE is as follows. The CRKRE algorithm performs a path discovery based on a modified weighted breadth-first search (BFS) to find a 0-protected route L_{sd} from s to d with minimum cost as described in Equation 3.12. The resources allocated in the 0-protected route are the *main resources*. After the 0-protected route is found, CRKRE attempts to add another k sets of *backup resources* to provide the k-protection by *only allocating more backup spectrum on* l to each link $l \in L_{sd}$.

If the desired protection level cannot be found for any sub-route from u to v in L, the algorithm performs another recursion of CRKRE on u and v to find a backup path for them. It is possible that the desired protection level cannot be satisfied between two SUs u and v on the 0-protected path. Suppose the protection level of the subroute P_{uv} is $min(p(l), \forall l \in P_{uv}) = m$ and is lower than k. In this case, a backup path between u and v is the only way to achieve the desired protection level. If $m, m < (k-1)$ backup spectrum cannot be satisfied between intermediate node u and v, the algorithm recursively finds another m-protected route L' that shares no common links in L. That is $L' \cap L = \emptyset$ and L' is a backup path that does not overlap with L. After L' is found, the algorithm allocates $(m-1)$-protection resources. It is because the original subroute from u to v in L already contain a *main path* and L' does not need another one.

When a backup route L' is completely constructed, the algorithm adds L' to L. Similarly, another recursion will be invoked if the desired protection level cannot be found for any sub-route from u' to v' in L'. If an L' cannot be found, the algorithm has no choice but reject the request. The

procedure stops when a k-protected route L satisfying the bandwidth requirement B and protection level k is found. If a complete k-protected route L is successfully constructed, the algorithm updates G by removing all spectrum resources that are assigned to L.

Input: G, s, d, k, b;
Output: L;
$L = path\_discovery(G, s, d, k, b)$;
if $L = NULL$ **then**
    return $NULL$;
end
$tmpG = G$, $unsatisfied = \{\}$;
for $l \in L$ **do**
    $i = 0$;
    while $i < k$ **do**
        find the fragment min_f with minimum cost;
        if $min\_f \neq NULL$ **then**
            $Allocate(l) = Allocate(l) \cup min\_f$;
            $tmpG = tmpG - min\_f$;
        else
            break;
        end
        i += 1;
    end
    $unsatisfied.append((l, k - i))$;
end
combine ls in $unsatisfied$ to $P' = \{(s_1, d_1, m_1), (s_2, d_2, m_2)...\}$;
for $(u, v, m) \in P'$ **do**
    $L' = CRKRE(tmpG, u, v, m, b)$;
    if $L = NULL$ **then**
        return $NULL$; // sub-route unsatisfied
    end
    $L = L \cup L'$;
end
$G = tmpG$; // Update G with $tmpG$
return L;

Algorithm 4: Summary of CRKRE Algorithm.

3.3.1 CRKRE Path Discovery

The path discovery procedure starts with the source node s and a queue Q consisting of 3-tuples. The tuples in the queue are composed of (id, C, L), where id is the id of the vertex, C is the current

Input: G, s, d, k, b;
Output: L;
$QueueQ = \{(s, 0, \{\})\}$;
while *Q is not empty()* **do**
    sort Q by cost;
    $(v, cost, L) = Q.dequeue()$;
    if $v == d$ **then**
        return L;
    end
    for $n \in neighbor(v)$ *and* $n \notin L$ **do**
        find the link l_{vn} and fragment min_f with minimum cost;
    end
    if $min\_f \neq NULL$ **then**
        $Q.enqueue((n, cost + cost(l_{vn}, min\_f), L \cup (l_{vn}, min\_f))$;
    end
end
return $NULL$; // s is unable to reach d

Algorithm 5: Summary of CRKRE's path discovery procedure.

cost, L is the current path and spectrum allocation for s to reach id. The algorithm starts with enqueuing a 3-tuple $(s, 0, \{\})$ into the queue Q. While Q is not empty, the algorithm removes the first tuple (v, C, L) from Q. If v is the destination, the procedure has found the lowest cost 0-protected route L from s to d. The procedure terminates and returns L. If v is still not the destination, the procedure continues and computes the costs from v to its neighbors. For each neighbor n of v, the procedure finds the spectrum fragments f that satisfies the bandwidth requirement B with minimum $cost(l_{vn}, f)$, denoted as min_f. The tuple $(n, C + min\_f, L \cup (l_{vn}, f))$ is enqueued to Q for future search. After all neighbors of v is visited, the procedure sorts the tuples in Q to ascending order based on their costs C.

On the other hand, if a spectrum fragment satisfying the bandwidth requirement B cannot be found for l_{vn}, it means there is not enough spectrum fragment for v and its neighbor n to satisfy the bandwidth requirement. This path from v to n is useless and the procedure does not enqueue this path. If Q is empty before s reaches d, it means there is not even 0-protected route for s to communicate with d. This request is rejected by returning an empty route. After a 0-protected route L is found, for each link l in L, the algorithm allocates k-protection resources by finding the

k lowest cost spectrum fragments satisfying the bandwidth requirement B and add them to L. The CRKRE algorithm is summarized in Algorithm 4 and the path discovery algorithm is summarized in Algorithm 5.

The time complexity of the CRKRE is composed by the weighted BFS, backup resource allocation and recursions of CRKRE. The weighted BFS's complexity is $O(|V| + |E|)$, where $|V|$ is the number of SUs and $|E|$ is the number of links in the network. The initial 0-protected route L can only have the number of links $|E'|$ and $|E'| \leq |E|$, since the links in $L \subset E$. The allocation of k backup resource for each links in L is therefore $O(k|E'|)$. Finally, for sub-routes that cannot satisfy the level of protection, another recursion of CRKRE with protection level m, $m \leq k$, is needed and the complexity is $O(|V| + |E|) + O(m|E'|)$. When the next recursion happens, it means the current recursion cannot satisfy the protection level requirement k. For example, current recursion can only satisfy 2 protections, then $m = k - 2$. If k is satisfied in current recursion, there will be no next recursion and the complexity of current recursion is $O(|V| + |E|) + kO(|E'|)$ However, in worst case, each recursion can only find 1 protection, k recursions happens and thus the complexity is $k(O(|V| + |E|) + 1 \times O(|E'|)) = O(k|V| + k|E|)$.

3.4 Distributed Algorithm

We propose a distributed algorithm for building a k-protected route. Unlike centralized approach (CRKRE), an SU does not have global view to the network (G) and backup spectrum allocations of other SUs. Therefore, an SU needs to satisfy the bandwidth requirement and the desired level of protection locally before it attempts to request routes to its neighbors. On SU v, the backup spectrum decided locally is sent to SU v's neighbors along with its current route from s to v in route request (REQ) message. Moreover, the SUs should only determine *one set* of local lowest cost neighbor(s) that satisfy the bandwidth and protection level and expand the path discovery to them. It is because when one neighbor cannot satisfy the requirement it attempts to separate the k-protected route into multiple protected route with a combination of lower protection levels. If an SU attempts to expand the route requests with multiple possible combinations of neighbors that

are able to achieve k-protection simultaneously, it results storm of path discovery messages from different combinations.

We propose a *distributed k-protected route establishment (DKRE)* algorithm for finding k-protected route between s and d with bandwidth requirement B. The DKRE algorithm starts with the source s. We assume that the SU periodically scans PU activity and broadcasts neighbor discovery messages. The neighbor discovery message from an SU u includes its current spectrum availability, its one-hop neighbors and all their spectrum availability. Therefore, the SUs know their one-hop and two-hop neighbors and the spectrum availabilities at these SUs.

Input: s, d, b, k, C, L;
if $v == d$ **then**
    // reached the destination d
    send GRANT to s;
    return L;
end
$next\_hop = \{\}$; $i = 0$;
while $i < k$ **do**
    for $n \in neighbor(v)$ *and* $n \notin L$ **do**
        find the link l_{vn} with minimum cost;
        attempt to find as many fragments to satisfy k as $p(l_{vn})$;
    end
    $i = i + p(l_{vn})$; $L = L \cup l_{vn}$;
    $next\_hop = next\_hop \cup n$;
end
send $REQ(v, d, b, k - p(l_{vn}), C, L) \forall n \in next\_hop$;
$message = receive(n) \forall n \in next\_hop$;
if *message is REJECT* **then**
    send REJECT to s;
end
if *all* $n \in next\_hop$ *are GRANTed* **then**
    send GRANT to s;
end

Algorithm 6: Summary of DRKE algorithm on an SU v. The algorithm is invoked when v receives a REQ message or v is instructed to find a k-protected route.

3.4.1 DKRE Path Discovery

The source starts the establishment of the k-protected route. When an SU v receives a route request (REQ) message or a source SU v is instructed to find a k-protected route, for each neighbor n of v, v computes the cost of providing k protections with bandwidth requirement b on link l_{nv} to each of its neighbors. SU v chooses the neighbor that can provide the desired level of protection with lowest cost. If a neighbor u is already in L, it means u has been visited and v do not consider expanding the route t u. v sends a REQ message along with its current backup spectrum to that neighbor.

The REQ message contains a following information (s, d, b, k, C, L), where b is the bandwidth requirement, k is the desired level of protection, C is the total cost and L is the current route from s to the sender of this REQ. However, it is possible that none of the neighbors can provide the desired protection on its own. Instead of finding one neighbor to satisfy the required protection, a subset of neighbor is needed. Suppose a subset of the neighbors N with lowest costs can only provide m-protected routes. We have no choice but to find another backup route. v keeps finding $(k-m-1)$-protection from other neighbors until the desired level of protection is satisfied by N. v sends a REQ $(s, d, b, k_n, C, L + l_{vn})$ to each $n \in N$ and $\sum_{n \in N} (k_n) = k$. DKRE should avoid increasing the size of N, because it means branching the k-protected routes.

If a REQ (s, d, b, m, C, L) reaches the destination, the destination grants the request and responds with a GRANT message consisting of (m, L), where m is the protection level of this route and L is constructed m-protected route. The GRANT message is sent along with the path in L to the source node. Each SU on receiving the GRANT message, examines if m is the same as the protection level it was requested. If it has more than one branch, it waits for all branches to GRANT. If $m \leq (k-1)$, it means that there are some other backup routes with protection level of $k-m-1$ that are still under construction. On the other hand, if an SU u receives a REQ and it is unable to find a neighbor to satisfy the bandwidth and protection requirement. It sends a REJECT message to the REQ sender. The REQ sender, upon receiving the REJECT message, attempts to find another path to satisfy the bandwidth and protection requirement by replacing neighbor that issues the REJECT message in

N with other neighbors. If the REQ sender has tried all its neighbors and cannot find an alternative path, it also send the REJECT to its upstream.

3.5 Evaluation

We evaluate our algorithms by simulation using NS-3.17 [50] and an extension for CRN called CRE-NS3 [6]. We randomly deploy different number of SUs and PUs in a $2000 \cdot 2000\ m^2$ area. The SUs have the same transmission range of 250 meters and the PUs have the transmission range of 1000 meters. We remove topologies that SUs do not form a connected graph. For each number of SUs, we randomly generate ten topologies and compute the results from the average of them. Some of the PUs are active and some are not, but the inactive PUs might be active later. The SUs might have different data rate requirement, but their data rates are stable after they obtained a k-protected routes, because the spectrum is assigned to the SUs exclusively and the interference is minimized.

The primary channels are 802.11g channels with physical layer bandwidth of 54 Mbps [116] and we use the NS-3 and CRE-NS3 built-in `AdhocWifiMAC`. The width of each 802.11g channel is 22 MHz and the max span of the NC-OFDM data interfaces is set of $n = 3$ channels. In our simulation, there are 15 PU channels. Two PUs are active when computing the k-protected route, which means 13 channels are available for SUs. For distributed algorithm, a dedicated common control channel is used for control purposes.

In this chapter, an NC-OFDM interface allocates a spectrum fragment exclusively to a link in k-protected route. In other words, it is guaranteed not to experience any interference from any other SUs that are assigned with *different* spectrum fragment in the channel. We simulate the behavior and performance of NC-OFDM using the following approach. We disable the interference in the physical layer in NS-3. If an interface allocates $n\%$ of a channel, it transmits using the channel $n\%$ of the time to obtain a throughput for a partial channel. However, when PUs return to active, SUs still relinquish the channel and resort to backup spectrum or backup path.

We compare our k-protected routing with eCRTCA and dCRTCA proposed in [131]. The goal of eCRTCA and dCRTCA is to assign channels to links in CRN to guarantee the CRN to

remain connected when PUs return to active. eCRTCA and dCRTCA do not consider NC-OFDM interfaces, but their SUs are equipped with multiple interfaces. Note that the NC-OFDM interfaces in our approach can access a spectrum that spans over $n = 3$ channels. To evaluate more reasonably, we equip each SU in eCRTCA and dCRTCA with three interfaces, so that they can also access three channels.

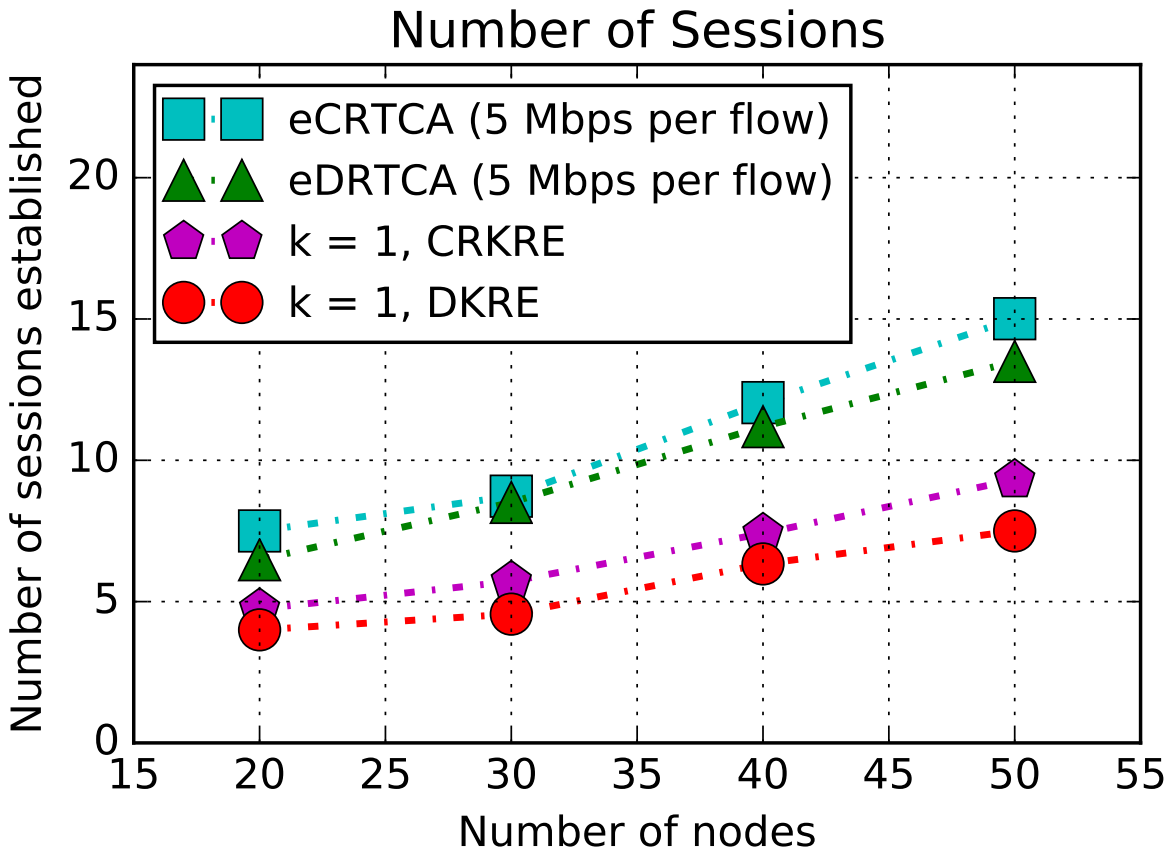


Figure 3.5: The number of sessions for 1-protected session requests with bandwidth requirement 5 Mbps.

3.5.1 Number of Sessions

We evaluate the number of k-protected routes that a network can accommodate by letting the simulator generate requests from random source SU to random destination SU. Note that we discarded special cases that source and destination are less than two hops away, so that short

distance results do not make the overall results too optimistic. The algorithms attempt to establish k-protected route for this pair of source and destination. If *twenty* consecutive requests are rejected, we say the network has reached its limitation and the simulator stops generating source and destination. The NC-OFDM interfaces can access spectrum spans over 3 consecutive channels. For easy understanding, we use the term data rate in the evaluation, instead of bandwidth requirement. The evaluated data rate requirements are 5 Mbps (5 people video conferencing) and 2 Mbps (one-to-one video calling) [107], respectively. We use Eq. 3.6 to convert data rate (b) to bandwidth requirement (B) and vice versa.

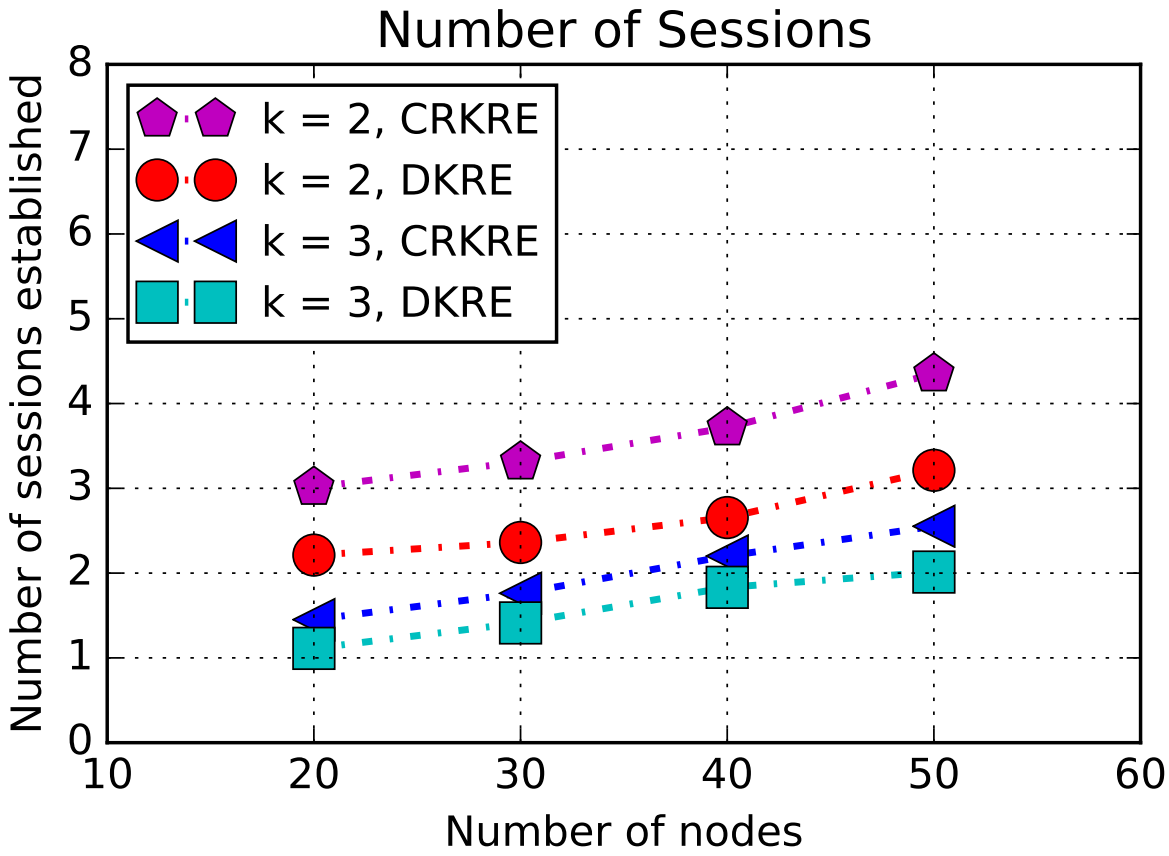


Figure 3.6: The number of sessions for 2 and 3-protected routes with bandwidth requirement 5 Mbps

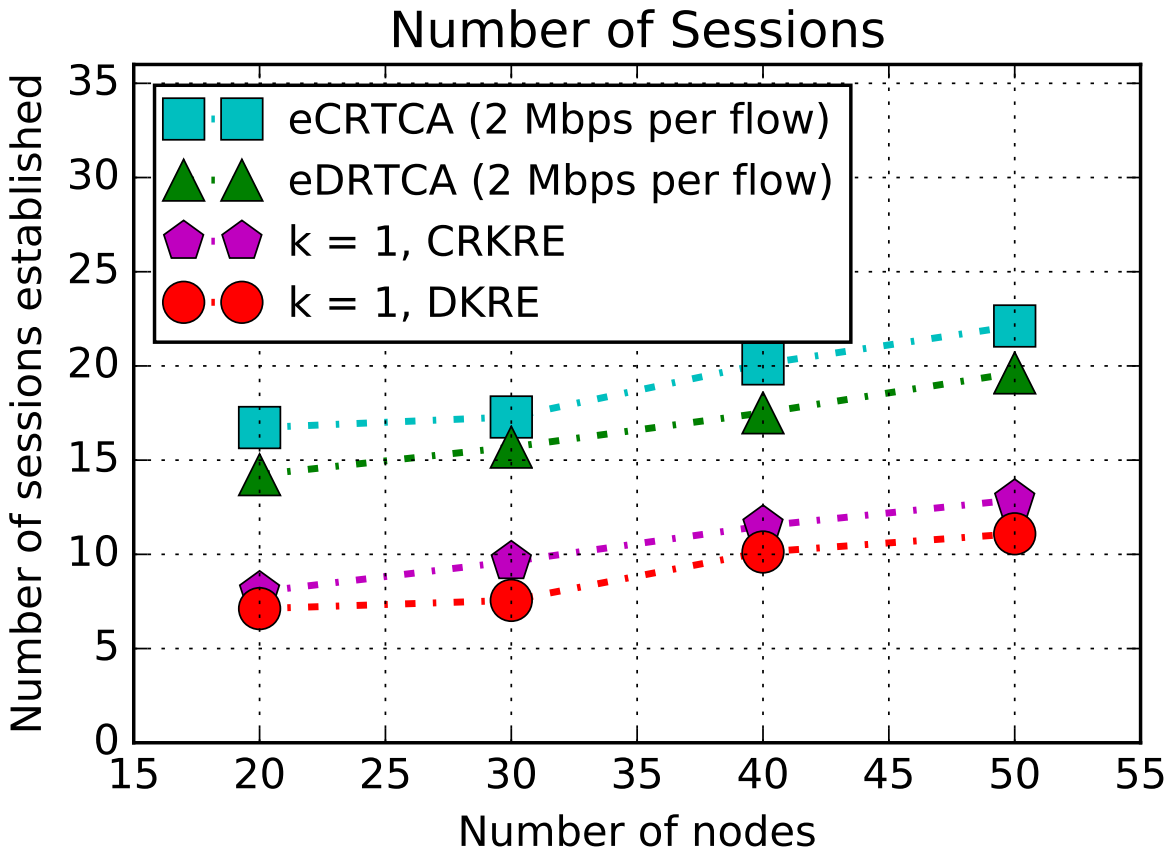


Figure 3.7: The number of sessions for 1-protected routes with bandwidth requirement 2 Mbps.

Similarly, we attempt to establish traffic flows in eCRTCA and dCRTCA for each source and destination request. We use NS-3 built-in AODV protocol for establishing the traffic flow in eCRTCA and dCRTCA. Note that eCRTCA and dCRTCA do not consider NC-OFDM interfaces and the channel bandwidth of each link might be the bottleneck when the sessions have saturated the bandwidth of a channel. However, eCRTCA and dCRTCA assumes each node is equipped with multiple interfaces. In our simulation, we assign three interfaces for each SU in eCRTCA and dCRTCA. Therefore, the limitation for the number of sessions of eCRTCA and dCRTCA is observed when specific node's packet queue is full and significant packet loss happens.

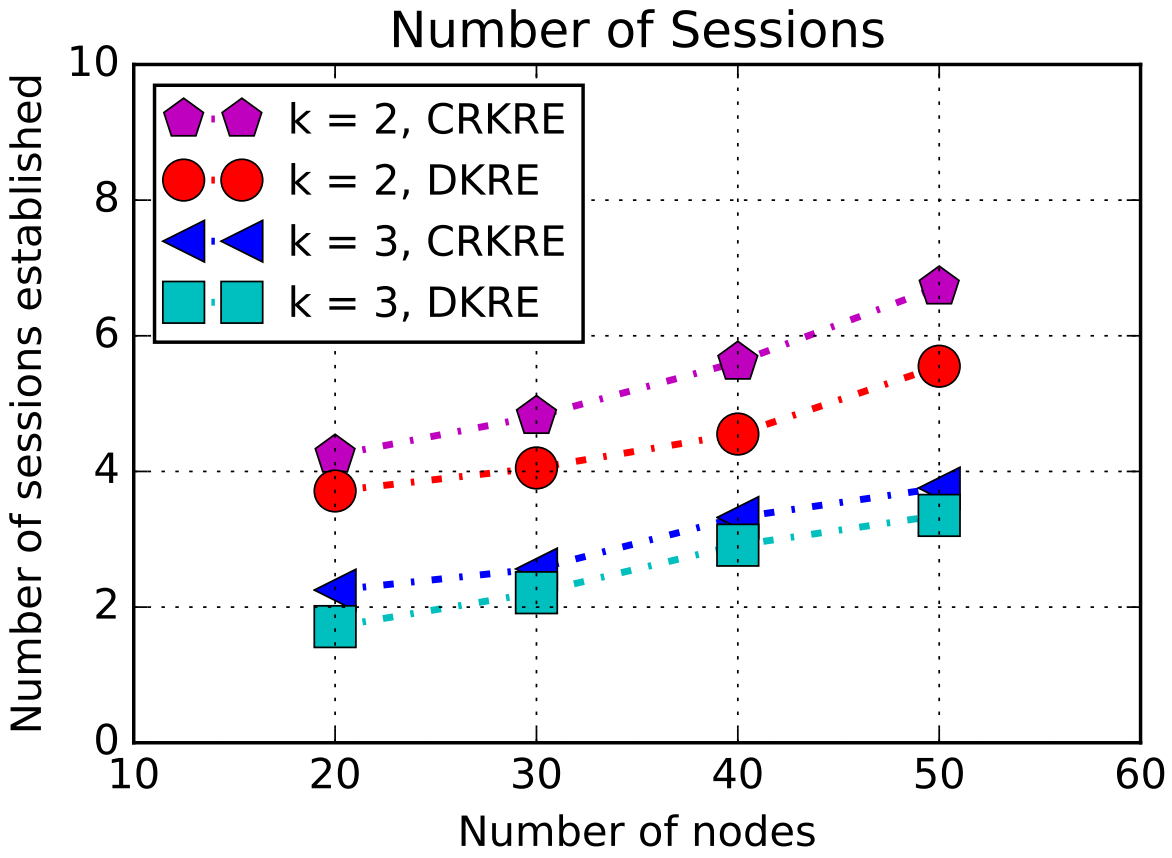


Figure 3.8: The number of sessions for 2 and 3-protected routes with bandwidth requirement 2 Mbps.

The simulation results for the number of 1-protected routes with $b = 5$ Mbps is shown in Figure 3.5. The number of 2 and 3-protected routes with $b = 5$ Mbps is shown in Figure 3.6. The simulation results for number of 1-protected routes with $b = 2$ Mbps is shown in Figure 3.7. The number of 2 and 3-protected routes with $b = 2$ Mbps is shown in Figure 3.8. We can see that the number of k-protected routing established is lower than either eCRTCA or dCRTCA. It is not surprising because k-protected routing trade considerable amount of resources for protection.

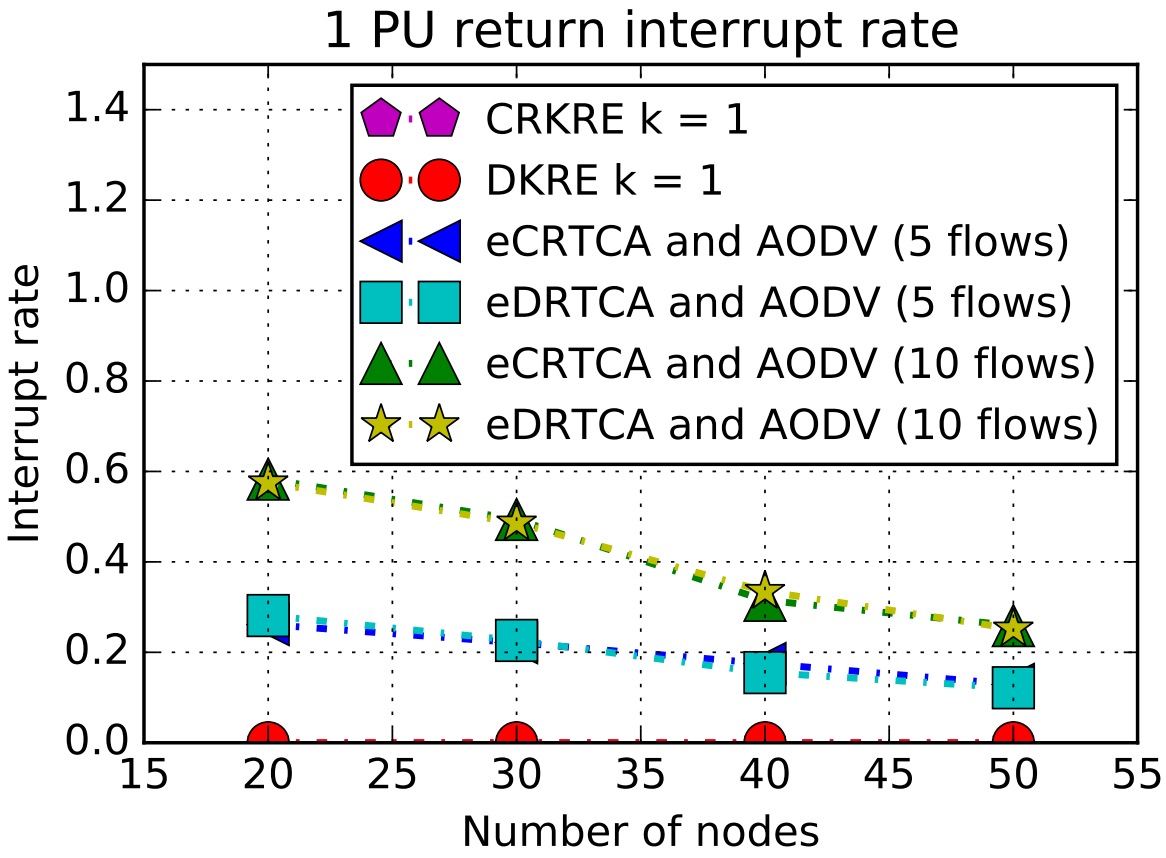


Figure 3.9: The interruption rates for one PU returns to active.

On the other hand, the number of routes established by CRKRE and DKRE for 2 Mbps requests is not significantly higher than that for 5Mbps. It is because of the NC-OFDM interface span limitation ($n = 3$). Suppose an NC-OFDM interface has allocated some spectrum fragment in channel 11 and its span limitation is three channels. It can no longer allocate spectrum fragment that is more than three channels away, such as spectrum fragment in channel 1. However, eCRTCA and dCRTCA assumes that an SU is equipped with multiple independent interfaces that are able to handle such requests. The advantage of k-protected routing is for handling returning PU using preassigned resources with minimized delay.

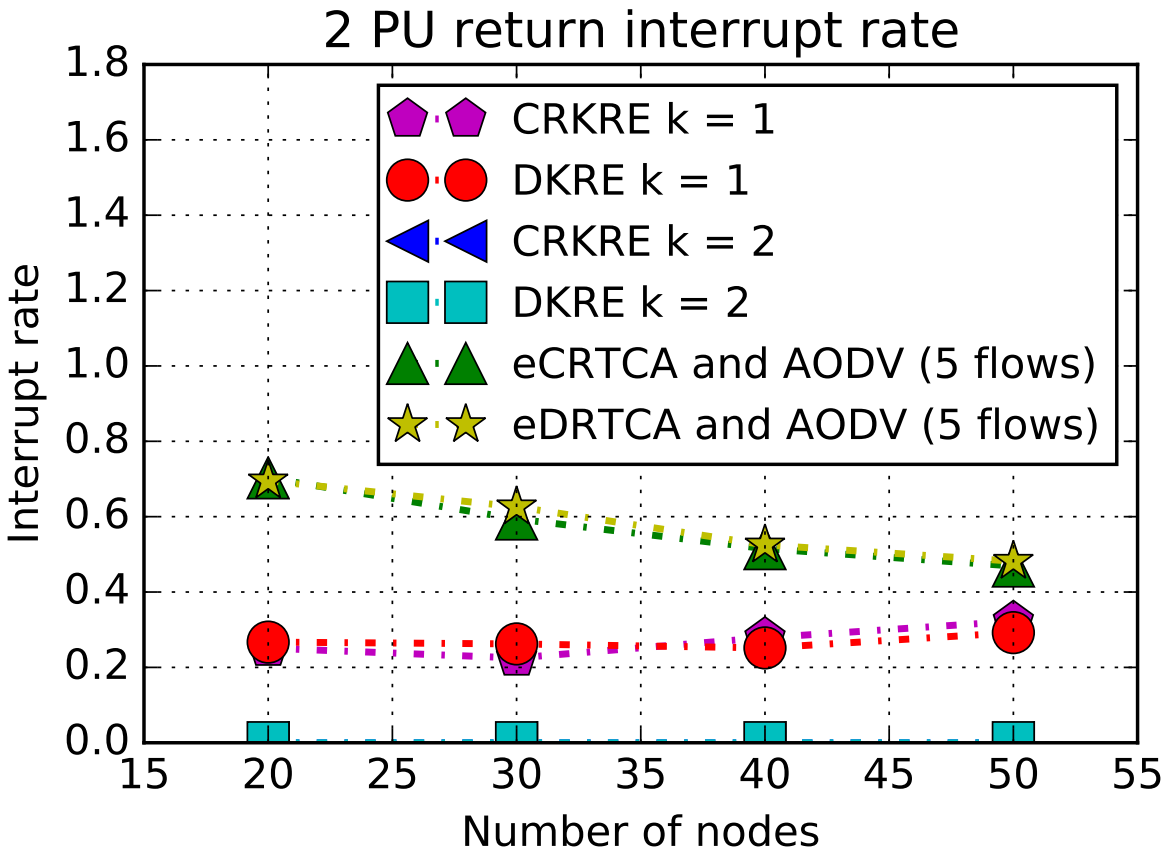


Figure 3.10: The interruption rates for two PUs return to active.

3.5.2 Interruption Rate

We simulate the interruption rate after the k-protected routes are constructed. For one returning PU simulations, for each PU, we deploy the routes as described in last section and let the PU be active. We wait for 10 seconds and terminate the simulation. We examine the packet queue overflow message. If a source cannot reach the destination at the end of simulation, the route is interrupted. For each set of PU combination, we let all PUs in the same combination be active and check the interruption rate. The interruption rate is defined as the number of interrupted routes over the total number of routes.

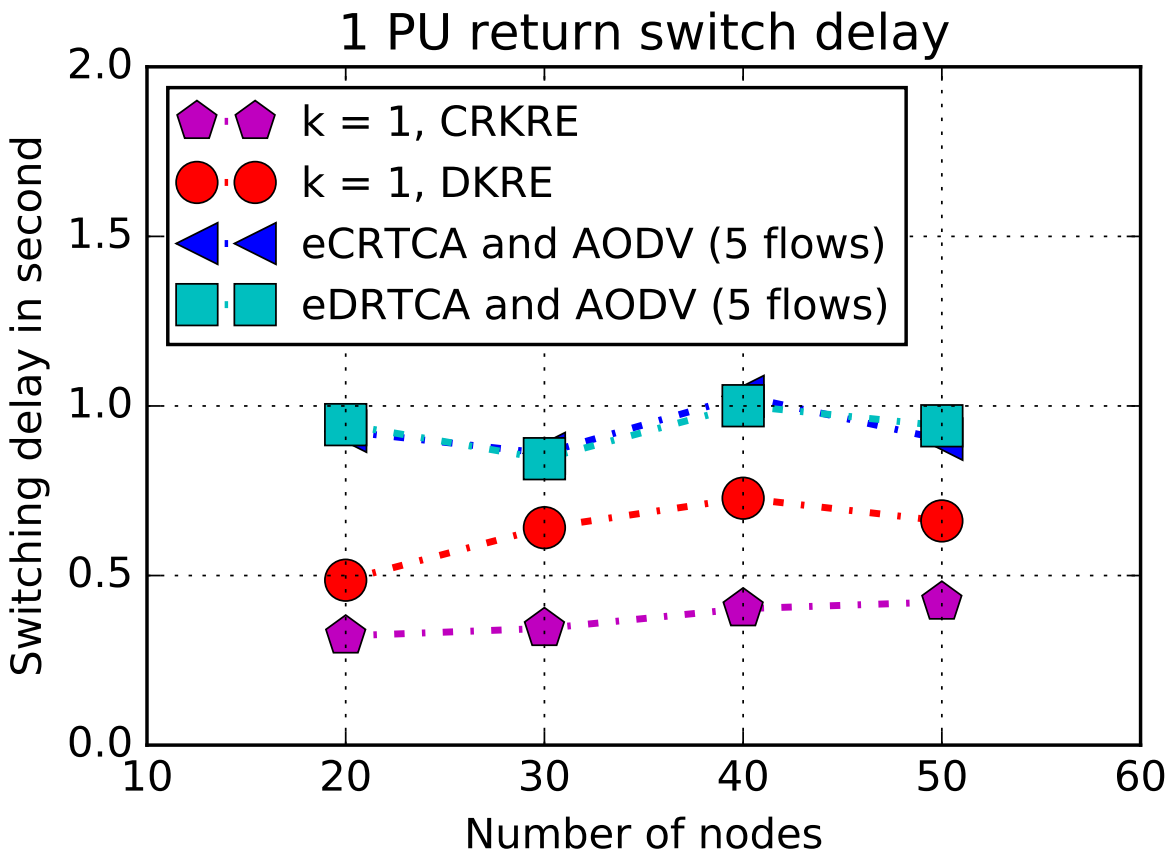


Figure 3.11: Average delay for recovering from one returning PU.

The interruption rate of one returning PU is shown in Figure 3.9. For eCRTCA and eDRTCA, we attempt to establish traffic flows for up to 5 or 10 flows. When PU appears, the ongoing traffic flow in eCRTCA and eDRTCA is interrupted and the source attempts to recover the interrupted flow by establishing another flow using AODV. Therefore, when there are more flows already in the network, there are fewer resources for recovering from interruption due to PUs return to active. For networks with more SUs, eCRTCA and eDRTCA have higher chance to recover from returning PUs, because there are more available SUs to establish the interrupted routes. In contrast, the 1-protected routes are never interrupted due to one returning PU and the interruption rate is 0%, which means 1-protected routes are robust against one returning PU.

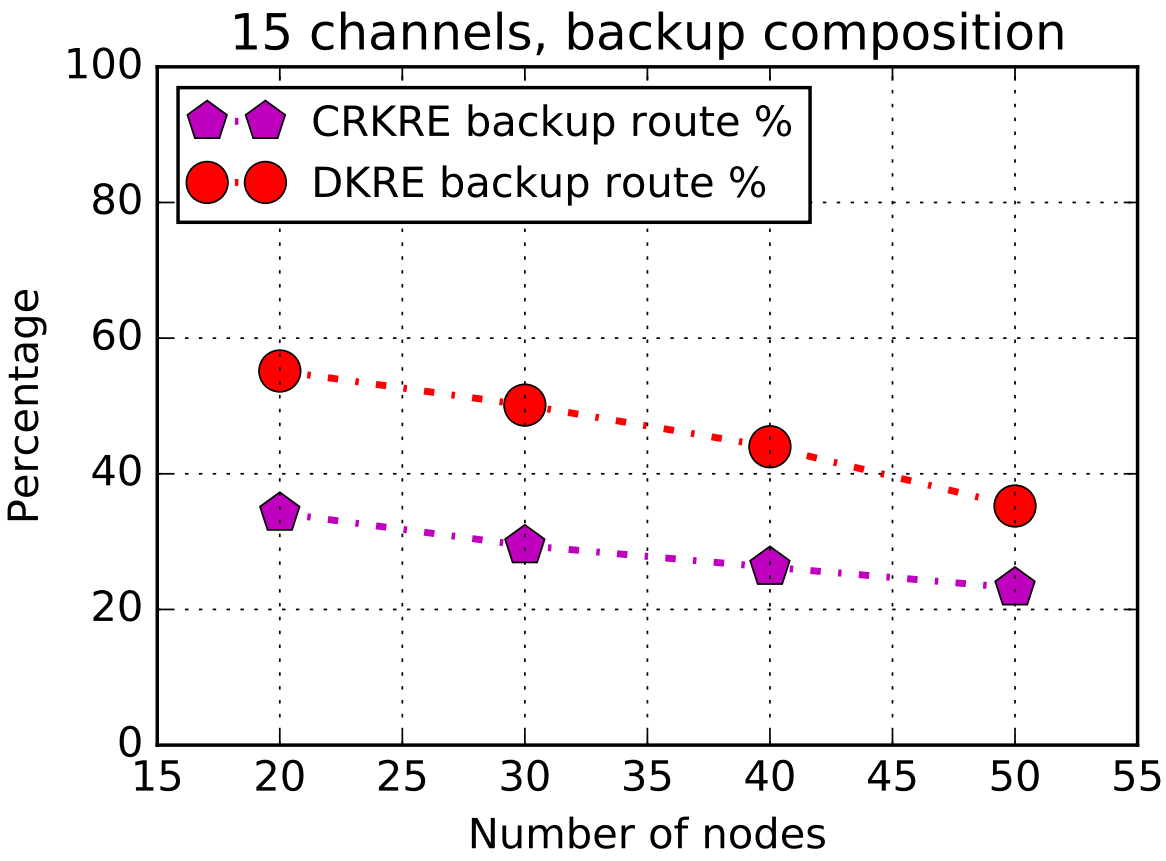


Figure 3.12: The ratio of backup routes in 1-protected routes.

The interruption rate of two PUs return to active is shown in Figure 3.10. When two PUs appear, more than 10% of 1-protected routes are interrupted. However, more than 80% of the 1-protected routes are still connected. Because the interruption only happens to links that use *exactly the two spectrums owned by the returning PUs*. For links that only use one of the returning PU as their main or backup resource, they have another backup resource or path to maintain connection. Similarly, all 2-protected routes are not interrupted.

3.5.3 Handoff Delay and Throughput

The delay for recovering from one returning PU is shown in Figure 3.11. For CRKRE and DKRE, the delay is composed of spectrum sensing time (100 ms in our simulation) plus activating the backup path or backup spectrum. As shown in Figure 3.3, activating a backup spectrum is much

faster than activating a backup route. For eCRTCA and eDRTCA, the delay time is composed of spectrum sensing time plus the time ADOV takes to re-route the traffic. Both CRKRE and DKRE have low handoff delay. It is because both backup spectrum and backup paths are preassigned resources to the route. The backup resources are just not being used. Both CRKRE's and DKRE's delay times are mainly dominated by the time for activating backup paths.

On the other hand, as shown in Figure 3.12, the CRKRE has much fewer backup routes compare with DKRE. It is because the CRKRE is a centralized algorithm. The CRKRE is able to find a 0-protected route as a base route and keeps adding protection to each links. The CRKRE only resorts to backup routes when backup spectrum is no longer available between any pair of SU u and v. Even if there are backup paths, the backup path starts from SU v converge to the main route when it reaches SU u. In contrast, whenever an SU in DKRE needs more than one neighbor to provide the desired level of protection, DKRE often resorts to backup paths. The backup paths are often harder converge to a single path due to the lack of knowledge of the network.

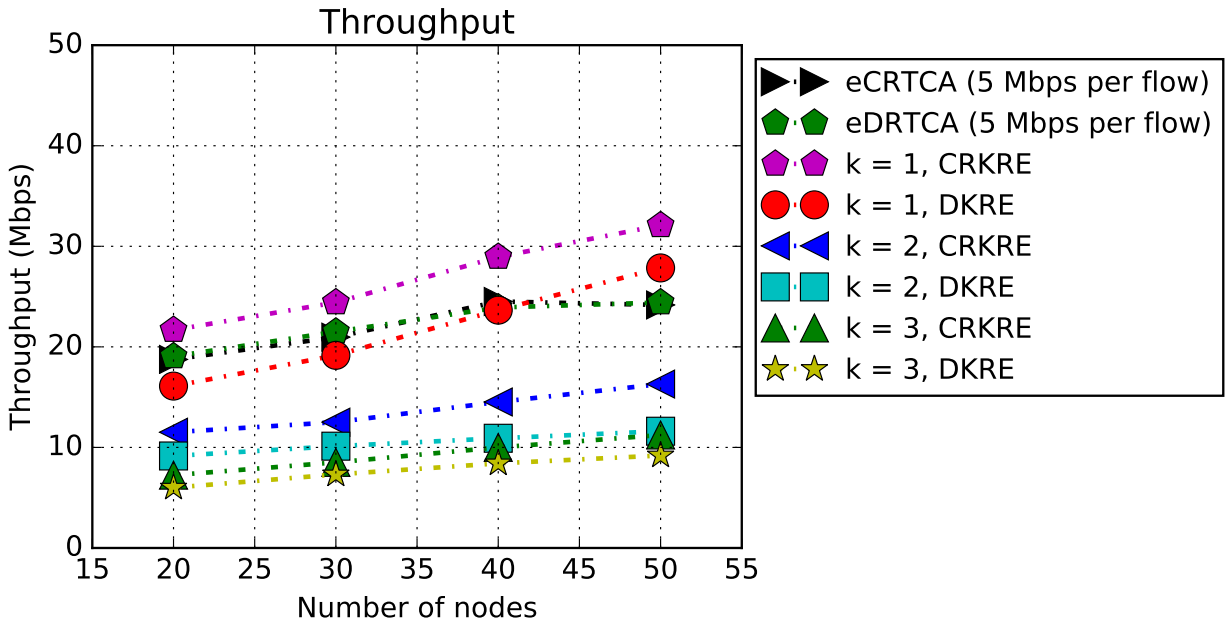


Figure 3.13: The throughput measured in different set of simulations.

We use NS-3 built-in `BulkSendApplication` to generate traffic for all simulations. The throughput comparison is shown in Figure 3.13. As mentioned above, the NC-OFDM data interfaces

allocate spectrum fragment exclusively. Almost every k-protected route achieves the bandwidth it requested, no matter it is built using CRKRE or DKRE. Therefore, the throughput has similar trend as the number of routes. On the other hand, the traffic flows in eCRTCA and eDRTCA might saturate some of the links and the throughput is bounded by those links.

3.6 Summary

In this chapter, we argue that guaranteed network protection is essential for an effective CRN because returning PUs might interrupt the ongoing traffic between the SUs. Such characteristic makes CRNs unsuitable for QoS-sensitive applications such as audio and video conferencing and multimedia streaming. Moreover, DSA is envisioned in future generation cellular network, the potential delay and possible interruptions when PUs become active need to be handled with care. We study the problem of establishing k-protected routes for CRNs. A k-protected route is a set of *main link*s with k sets of preassigned *backup spectrum* and *backup path*s that is guaranteed to sustain from k returning PUs without being interrupted. Moreover, we propose the use of NC-OFDM interfaces to allocate spectrum fragments flexibly. We proposed both centralized and distributed for constructing k-protected routes. Simulation results show that the 1-protected routes are never interrupted when one PU appear, and the average interruption rate is lower than 18% when two PUs return. Moreover, the delay time for the k-protected routes built by CRKRE is about 20% to that of previous studies, and DKRE is at most 80% to that of previous studies. We have presented how the CRNs find control channel and establish protected communication routes, which significantly improves the feasibility of CRNs. In the following two chapters, we will present our work to address the challenges for interference identification and mitigation in HetNets.

CHAPTER 4

INTER-FEMTOCELL INTERFERENCE IDENTIFICATION AND RESOURCE MANAGEMENT

For HetNets, the first research challenge is deciphering the interference relations in the network. HetNets formed by smallcells with different sizes of coverage (i.e., femtocells, picocell, and microcells) and macrocells have been proposed to satisfy increased bandwidth demand with the limited and crowded wireless spectrum. Although cellular networks provide almost ubiquitous coverage in most areas, complaints of low-quality service regarding signal reception are often reported by indoor users. The indoor signal attenuations are mainly attributed to the penetration losses through building structures and multipath propagation. To improve the indoor service quality and coverage, low-power and low-cost indoor cellular base stations called femtocells [20, 129] had been developed. The femtocells operate on the same licensed spectrum band and use the same technology as the macrocellular infrastructures. Hence, no modification to existing user devices is required. Several orthogonal frequency division multiple access (OFDMA [90]) based 4G cellular network standards such as Mobile WiMAX, and LTE-A have endorsed femtocell as a mandatory technology. The femtocells attach to the core cellular network via IP-based broadband backhauls such as cable or DSL. The femtocells are promising in terms of providing superior voice quality and higher data rates for indoor users. Large-scale deployment of femtocells in the urban area is expected to be realized in the future.

Because all base stations operate in the same licensed spectrum band, inter-cell interference (including inter-femtocell and cross-tier macrocell/femtocell interference) significantly limits the achievable throughput of a femtocell cellular network. A typical approach to mitigate inter-cell interference is known as resource isolation, which assigns non-overlapping resources to the interfering stations. The most critical step to mitigate interference is to identify the interference relations in the network precisely.

The service providers install the macrocells with careful planning. Thus, the inter-macrocell

interference is minimized at coverage edges. The interference management between macrocells and femtocells is also widely studied [21, 28, 120] where the information (e.g., configuration and location) of macrocell base stations are known. In this chapter, we focus on a more challenging problem, which is the inter-femtocell resource management. Several characteristics of femtocells make the interference identification challenging. The femtocells are often installed by end-consumers without any pre-planning. An individual femtocell possesses no information as to either its location or the existence of nearby femtocells. Since very limited information about the femtocells is available, it is hard to decipher the inter-femtocell interference. In addition, it is unrealistic to expect the coordination and assistance of the user devices because the goal of femtocell cellular network is to augment the coverage without modifying any existing user devices.

Several efforts have been made to improve inter-femtocell resource management. Authors in [112] propose to let femtocells assign resources to mobile stations (MSs, e.g., cellphones and tablets) based on a hashing method. If an MS is experiencing interference in certain resources, the femtocell invokes a collision resolution procedure and tries to assign the MS other resources. The collision resolution procedure does not require the identification of interferer, but it may require a certain amount of subsequent frames to find available resources. FERMI [8] introduces two dedicated measurement zones in a frame. If an MS is experiencing interference, the delivery rates of the two zones would be significantly different. A central controller determines whether an MS experiences interference by observing the difference between the delivery rates in the two zones. If the MS were experiencing interference, the femtocell considers all nearby femtocells whose *received signal strength (RSS)* is over a threshold as interferers. This approach is highly sensitive to the threshold of the RSS [97], and it is extremely difficult to determine a uniform threshold across the whole network.

Instead of using the RSS for interference identification, we propose an algorithm to improve the interference identification by taking advantage of the availability of multiple subchannels in an OFDMA system. We let each femtocell transmit data using a subset of subchannels. The different combinations of subchannels are called *patterns* in this chapter. Femtocells transmit with different

transmission patterns, which impact the subchannels the MSs can receive data. The subchannels from which an MS receives data constitute a received pattern. Experiments on GNU Radio/USRP are conducted to demonstrate the MSs observe the received patterns, and the transmission patterns do affect their received patterns. The proposed interference identification algorithm identifies the interference relations in the network by examining these patterns intelligently. The evaluation results show that our method successfully identifies all real interferers and most non-interfering femtocells compared with approaches that use RSS. Because the method identifies real interferers while excluding non-interfering femtocells, the number of edges in the conflict graph is reduced, and the achievable throughput is improved. Further, we propose an efficient weighted vertex-coloring based resource assignment algorithm that allocates resources with better fairness and achieves higher throughput than previous studies. Some preliminary results have been presented in [74].

The rest of this chapter is organized as follows. Section 4.1 provides a brief introduction to the background of OFDMA systems and discussions on related work. We give an overview of our interference identification algorithm and describe the experiment to prove our concept in Section 4.2. Details about interference identification are presented in Section 4.3. In Section 4.4, we explain our algorithm for resource management. Section 4.5 is devoted to evaluating our interference identification and the resource management algorithm. Finally, a summary is presented in Section 4.6.

4.1 Background and Related Work

In this section, we briefly introduce the background of OFDMA systems and related work.

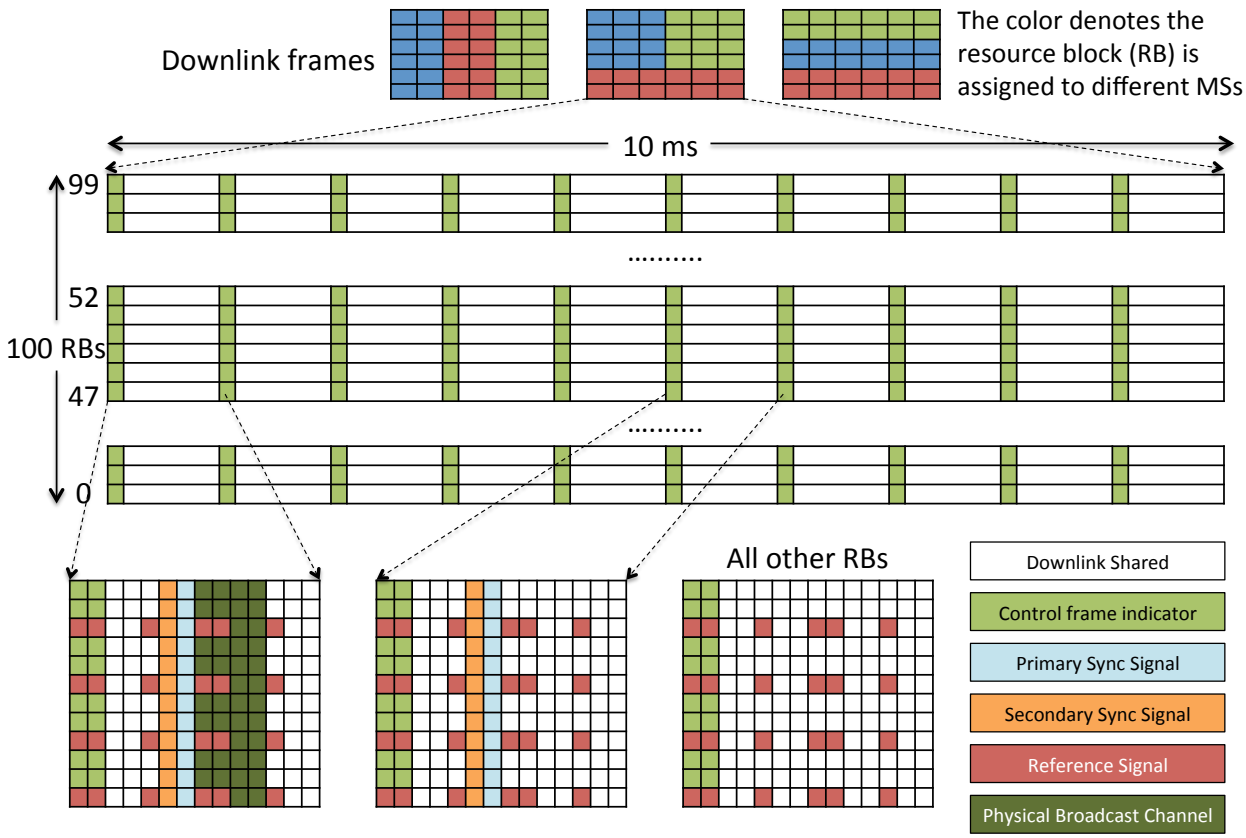


Figure 4.1: The LTE downlink frame structure of a 20 MHz channel.

4.1.1 OFDMA Preliminaries

The OFDMA is the underlying technology for 4G cellular networks. In an OFDMA system, the radio resources are two-dimensional frames of frequency (subchannel) and time (symbol). The continuous spectrum is divided into multiple equally spaced tones (subcarriers), and several subcarriers are grouped to form a subchannel. An OFDMA device operates in all or part of orthogonal subchannels in each frame. Several time symbols in a subchannel form a *resource block (RB)*, also known as a *tile*. Two RBs form a slot, which is the basic resource unit to assign to MSs. Each RB in an OFDMA subframe has a unique symbol and subcarrier offset.

The frame structure of a 20 MHz channel is shown in Figure 4.1 and the band is divided into 2048 subcarriers with 1200 occupied subcarriers. A frame in LTE Advanced is of length 10 ms, and each frame is further divided into ten subframes. Multiple devices can transmit using the

same channel simultaneously as long as they occupy different RBs (the downlink shared blocks in Figure 4.1). The control frame indicator (CFI) informs the MSs about the RBs allocation and modulation. Through CFI, each MS knows which RBs belongs to it and what modulation is being used in the RBs.

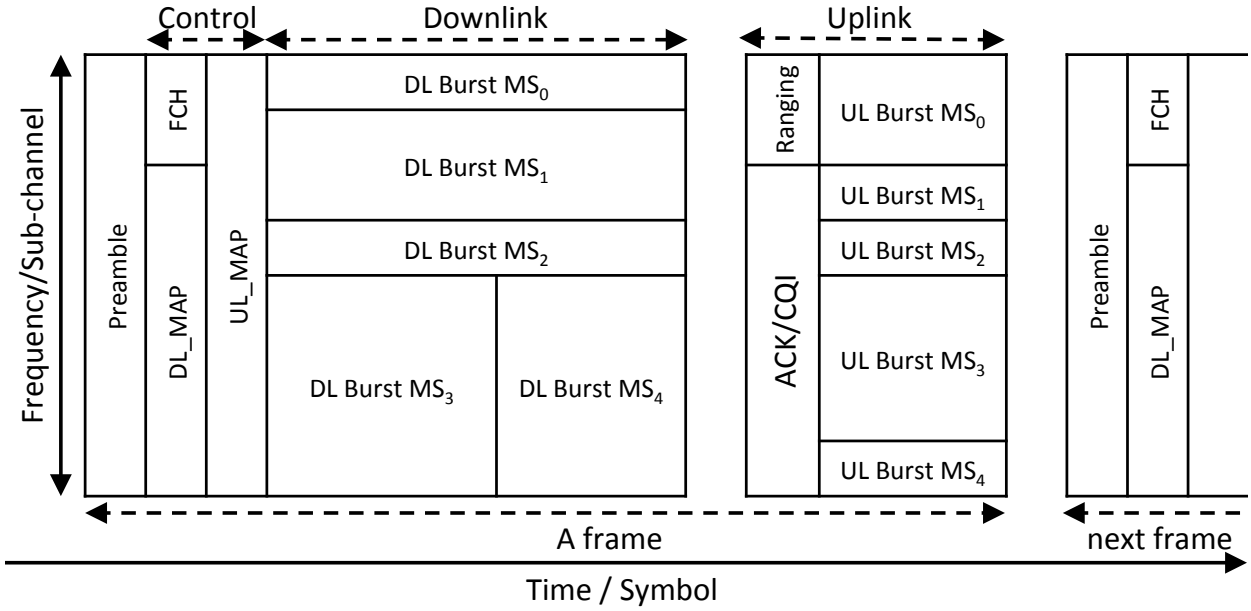


Figure 4.2: WiMAX frame structure.

Figure 4.2 shows a Mobile WiMAX Time Division Duplex (TDD) frame, where the horizontal axis denotes the time, and the vertical axis denotes the frequency. A TDD frame has a fixed duration and is composed of a downlink (DL) and an uplink (UL) sub-frame. DL also contains a control part, which includes the preamble, FCH (Frame Control Header) and downlink map (DL_MAP). The control part is transmitted using the most robust modulation and coding scheme (MCS). When a base station transmits a frame, all MSs that are associated with this base station receive this frame. The DL_MAP in the control part informs MSs of the RB assignments and MCSs of these RBs. An MS knows which RB belongs to it from the DL_MAP. MSs acknowledge (ACKs) the reception of each RB through the dedicated subchannels in UL.

4.1.2 OFDMA Systems Related Work

The service providers deploy macrocells with careful planning, including location, height, frequency, and even antenna configuration. Therefore, inter-macrocell interference is minimized at coverage edges. Multi-cell OFDMA frequency planning is also discussed in [22, 96], but these approaches require certain amount of knowledge of the base stations, such as location and antenna configuration. Because femtocells possess little knowledge of their information, frequency planning is not feasible for femtocell networks. Therefore, interference management is one of the most critical components for dense deployment of femtocells [128].

4.1.2.1 Cross-tier Interference Management

Cross-tier interference management has been widely studied as well. In earlier generation cellular networks (e.g., CDMA and GSM), there is only one carrier, and frequency reuse is not possible. Interference management mainly relies on pure frequency isolation [120] or power auto-configuration [21, 28]. Authors in [120] propose to assign different frequency spectrum band to macrocells and femtocells arbitrarily. For OFDMA-based cellular networks, a centralized approach for cross-tier interference management is proposed in [112], which prevents femtocells from reusing the resources that are occupied by macrocells. In these work, the knowledge of macrocells is relatively easier to obtain.

4.1.2.2 Inter-femtocell Interference Management

The very first step to inter-femtocell interference management is to *identify interference*. Previous papers also proposed several approaches to identify inter-femtocell interference, but they either require additional functionality such as channel sensing [9, 51, 70, 80] or additional information from the femtocells [66].

Location Information: The most straightforward approach to determine inter-femtocell interference is based on the locations of the femtocells. A centralized inter-femtocell interference

mitigation approach is proposed in [66]. The inter-femtocell interference is determined by a femtocell gateway using the location information provided by the femtocells. Based on the interference relations, the authors in [66] proposed an adaptive fractional frequency reuse (FFR) algorithm to adjusts the coverage area by controlling femtocells' transmit power. However, the location information is difficult to obtain in practice because the femtocells are installed by non-expert end-consumers. It is impractical to include global positioning systems (GPS) in femtocells either because femtocells are designed for improving indoor cellular coverage and GPS is also designed for outdoor environments. Moreover, femtocells that are located close to each other does not necessarily mean they are interfering.

Sensing: In [9, 51, 70, 80], the authors employ channel sensing to detect potential interference. However, it is not realistic to anticipate that off-the-shelf femtocells and MSs to have carrier sensing ability. The authors in [70] suggest performing channel sensing periodically to all radio resources as in cognitive radios. During the sensing period, the femtocells and the MSs cannot communicate. Channel sensing for inter-femtocell interference detection needs to be performed very frequently and is likely to result in high overhead. The authors in [80] propose that the femtocells need to be self-organizing to sense interference and to tune its transmission power to avoid interference. However, the reduction of transmission power not only shrinks the coverage but also decreases the received signal strengths at the MSs and affects the throughput. The work [51] also relies on cognitive femtocell to sense interference and adopts the game-theory approach to allocate wireless resources to avoid interference.

Message Exchange: The paper [69] discuss interference avoidance and quality of service (QoS) in OFDMA femtocell networks. The authors also recommend that the femtocells be self-organizing and self-optimizing. At configuration stage, the femtocells transmit neighboring information message with a transmission power two times as regular transmission power, to inform femtocells that are two hops away and to avoid the hidden terminal problem. The neighboring femtocells are grouped using this neighboring information message. Only the femtocells in the same group are possible to interfere with each other.

Signal-to-Interference-plus-Noise Ratio (SINR): The MSs in [115] reports nearby femtocells with SINR higher than a threshold to its femtocell. The femtocells obtain a list of neighboring femtocells and report it to a central server, where the interference relation graph is constructed. A resource partitioning method is proposed based on graph coloring that assigns wireless resources. However, neighboring femtocells with high SINR are not necessarily interferers.

Measurement Zone: FERMI [8] introduces two dedicated measurement zones (the *free* zone and the *occupied* zones) in the downlink of a frame. In **free** zone, only γ out of n femtocells transmit using all subchannels. In **occupied** zone, *all* femtocells transmit using all subchannels. [8] and [125] introduce *Burst Delivery Rate (BDR)*, which is defined as the number of bursts received at the client divide by the total number of bursts transmitted. If an MS were experiencing interference, the BDR in the *occupied* zone would be significantly lower than the BDR in the *free* zone. Through comparisons between the BDR in the free and the occupied zone, the central controller infers the interference in the femtocell networks without knowing femtocells' information. An MS maintains a list of nearby cells for potential handovers and measures the RSS from these cells. If the central controller finds that an MS is experiencing interference, the corresponding femtocell considers all nearby femtocells with RSS over a threshold as interferers. However, relying on the RSSs makes such interference identification highly sensitive to the threshold. It is also difficult to determine the threshold [97]. Especially for femtocell networks, the MSs are indoor, and many factors affect wireless links. Meanwhile, the indoor environment makes it impractical to use a universal threshold across the whole network.

Collision Resolution: A distributed hashing based inter-femtocell resource management is proposed in [112]. Femtocells assign resources to MSs by a random hashing scheme without any coordination with other femtocells. Collisions might occur if two interfering femtocells assign the same RB to their MSs. Due to the collision, the MS is unable to receive the RB correctly, and the femtocell notices the collision through the ACK. A collision resolution procedure is thus invoked to resolve the collision, rehashing another available RB to the MS. The hashing scheme and the collision resolution procedure avoid the need of identifying the interference relations. Instead,

this work [112] resolves collisions after they happen. However, the potential collisions degrade the quality of service, and it may take multiple subsequent frames to resolve one collided RB. In addition, it is possible that some RBs are mistakenly considered as collisions and trigger the rehashing unnecessarily.

In RADION [125], instead of assigning resources randomly, a probing method is proposed for the femtocells to search for available subchannels opportunistically. Each femtocell periodically transmits data bursts on certain resources and observes the difference of the BDR. The resources with higher BDR are considered available, and the femtocell assigns these resources to its client. However, the resource allocated to each client may result in fragmented spectrum and unfair resources allocation.

4.2 Interference Experiment

In this section, we explain the harmful effects of inaccurate interference identification. We also conduct experiments to demonstrate the impact of interference on received patterns.

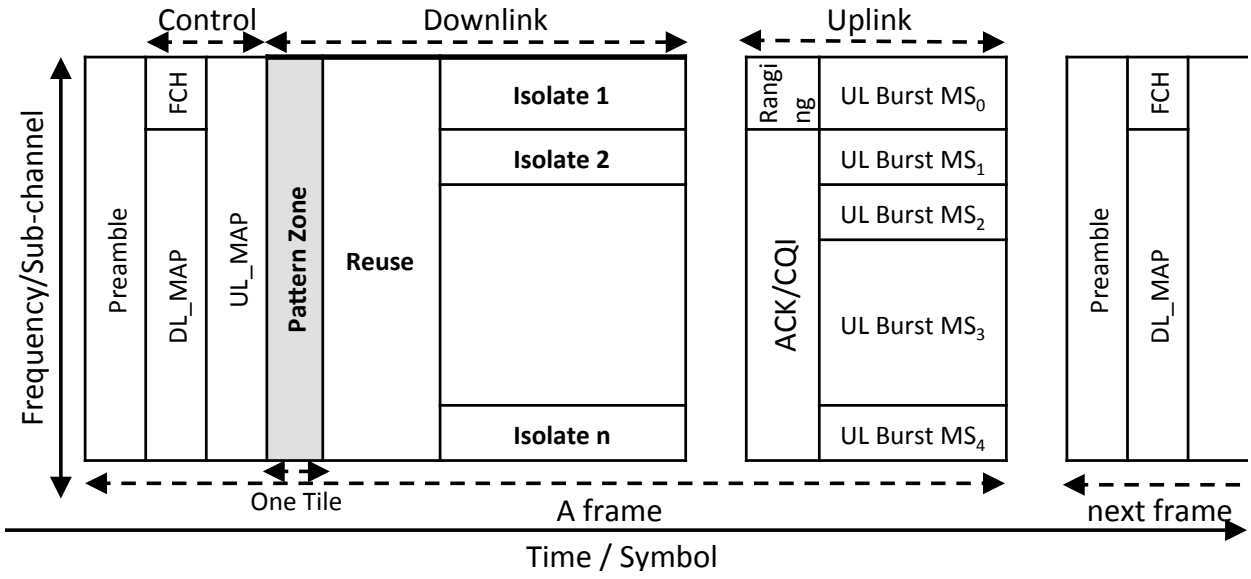


Figure 4.3: Zoning in our design.

4.2.1 Harmful Impact of Inaccurate Identification

The goal of resource management is to assign downlink RBs to MSs with minimized interference, maximized throughput, and maximized fairness. The most critical step to mitigate interference is to identify the interference relations in the network. Interference identification directly affects the performance of resource management. We propose an efficient method that takes advantage of the availability of multiple subchannels in an OFDMA system. As shown in Figure 4.3, we introduce a *pattern zone* in the downlink part. The pattern zone is used to distinguish real interferers from non-interfering femtocells so that the throughput can be improved with a fewer number of edges in the constructed conflict graph. Other OFDMA-based systems such as LTE-A also has a similar frame structure, and we can introduce pattern zone in their frame. The fundamental idea is to identify the inter-femtocell interference by generating different received patterns on the MSs.

In the pattern zone, each femtocell transmits data using a subset of n subchannels to its MSs. The interfering femtocells to an MS m will impact the received patterns on the MSs. Each femtocell extracts the received pattern on its MS from their ACKs. The received patterns of the MSs and the transmission patterns of the femtocells are collected at the central controller through the backhaul. The central controller identifies the interference relations in the network and builds a conflict graph. Based on the conflict graph, the central controller assigns RBs in isolation zones to interference-free MSs to avoid interference.

4.2.2 Interference Experiment Setup

We conducted experiments to demonstrate that different transmission patterns from interferers do affect the received patterns on the MSs. Although commercial OFDMA-based cellular services have been deployed in many countries, we have no access to commercial femtocells and their proprietary software. Therefore, we implemented a subset of OFDMA system by extending the implementation in [122] on GNU Radio [13]/USRP [37] to prove our concept. Since we cannot access licensed spectrum band, the central frequency is set to 5.3 GHz. We conducted the experiments with two system bandwidths 1 MHz and 10 MHz. In both experiments, the modulation is BPSK, and we

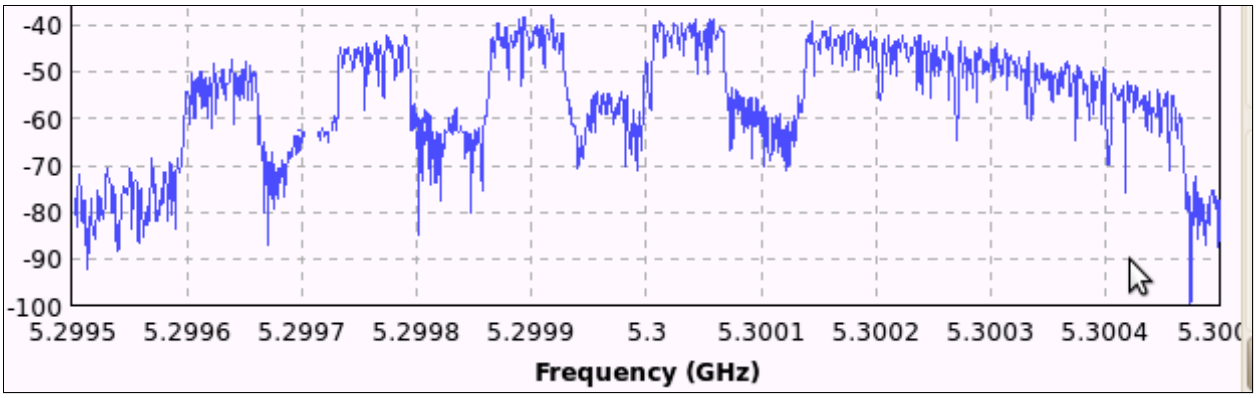


Figure 4.4: The received RSS of a transmission pattern 01010101011111.

split the frequency into 256 tones, and thus each subcarrier has a bandwidth of 3.90625 kHz and 39.0625 kHz, respectively. We do not use the central two subcarriers due to the DC-offset, and we skip eight subcarriers at two ends because of the roll down of filters.

For downlinks, we group 16 subcarriers into one subchannel, and this partition forms 14 subchannels in total. We also spare one subcarrier between two subchannels as a small guard band to avoid cross-subchannel interference. We define one RB as a subchannel over six symbols, and an RB carries 96 bits. (BPSK carries 1 bit per symbol.) To examine if an RB is received correctly, we reserve 4 bytes for CRC, and 8 bytes are left for data. When receiving an RB, the receiver determines whether the RB is received correctly by checking whether the received data matches CRC. If the CRC matches, it is said that this RB is transmitted, received and decoded correctly.

As shown in Figure 4.3, the pattern zone is defined as the *first RB over all subchannels* in each frame. The central controller generates a transmission pattern for each femtocell to transmit in the pattern zone. We will discuss more about the pattern generation strategy in central controller in the later context. Let us denote the transmission pattern to femtocell f as a string $p_{tx}(f) = c_1 c_2 ... c_n$, where $c_i = 1$ if subchannel i is used by f, otherwise $c_i = 0$. A *bit* in the pattern c_i is denoted as $p_{tx}(f)[i]$ and the length of a pattern p is denoted as $|p|$. If $p_{tx}(f)[i] = 1$, the subchannel i is used for transmission, otherwise, the subchannel i is left idle. In our implementation, there are 14 subchannels ($n = 14$). Figure 4.4 illustrates the RSS of a transmission pattern $p_{tx} = 01010101011111$. The RSSs of 1s are clearly higher than 0s.

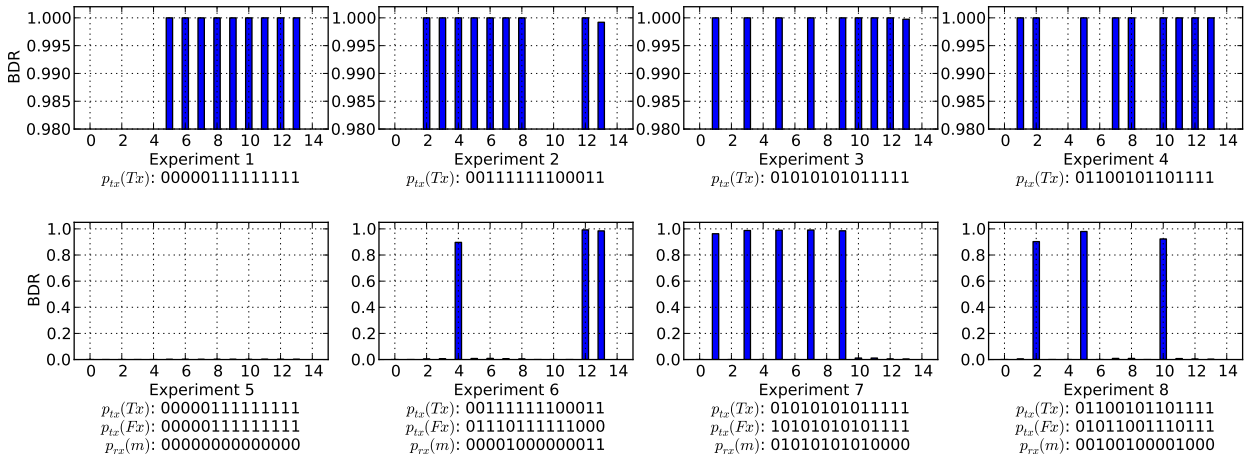


Figure 4.5: The burst delivery rate (BDR) on each subchannel for four transmission patterns (bandwidth 1 MHz).

Let us also denote the received pattern on an MS m as a string $p_{rx}(m) = c_1 c_2 ... c_n$, where c_i is marked as 1 if the RB on subchannel i is received correctly, otherwise c_i is marked as 0. If the RB carried by subchannel i is not destined for this m, c_i is also marked as 0. If a pattern is composed of all 0 or all 1, it is denoted as 0_n or 1_n.

Our testbed is composed of three USRPs, one USRP1 as transmitting femtocell, Tx, one USRP E110 as receiving MS, m, and one USRP1 as interferer femtocell, Fx. Tx and m are 1.5 meters away, and their antennae are 1 meter above the floor. The distance between Tx and Fx is 3 meters. The transmission gains of both Tx and Fx are 10 dBi, and the receiver gain of m is 10 dBi.

4.2.3 Interference Experiment

We first conducted a set of experiments with no interferer and the transmitter Tx transmits four representative transmission patterns. Figure 4.5 and 4.6 show the experiment results with system bandwidth 1 MHz and 10 MHz, respectively. The received patterns on m are shown in experiments 1 to 4 in Figure 4.5 and Figure 4.6. Taking experiment 2 as an example, $p_{tx}(Tx) = 00111111100011$ indicates that Tx does not transmit on subchannel 0, 1, 9, 10, and 11. These experiments demonstrate that when m is interference-free, almost 100% of the RBs are received correctly. If there is no interference at all, $p_{rx}(m)$ should be identical to $p_{tx}(Tx)$.

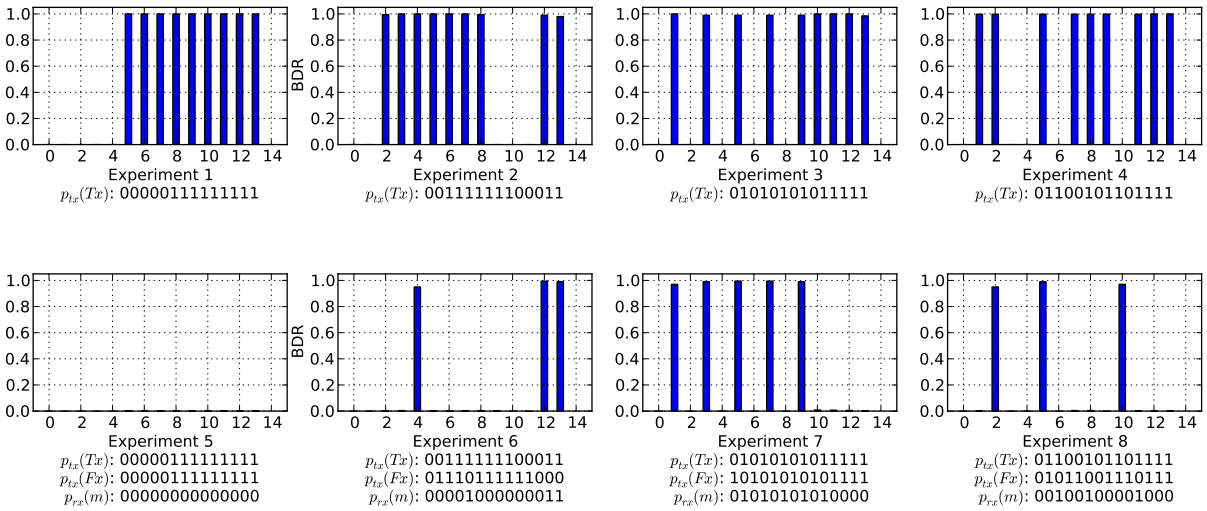


Figure 4.6: The burst delivery rate (BDR) on each subchannel for four transmission patterns (bandwidth 10 MHz).

We then conducted another set of experiments that includes an interferer Fx, which also transmits four patterns. Tx still transmits using the original four patterns. Tx, m, and Fx are placed in a straight line in the same order. The transmission gain of Fx is set to 5 dBi. The results are shown as experiments 5 to 8 in Figure 4.5 and 4.6. In experiment 5, the $p_{tx}(Tx)$ is precisely the same as $p_{tx}(Fx)$, and thus $p_{rx}(m)$ is expected to be 0_n, which is confirmed by the result. Collisions happen in all subchannels and those BDRs are very close to 0%. From experiments 5 to 8, we observe that for subchannels occupied by both Tx and Fx, the BDRs are very close to 0%. For subchannel that is occupied by Tx but not by Fx, m still receives most of the RBs correctly. However, in Figure 4.5 (system bandwidth 1 MHz), the BDR of subchannel 4 in experiment 6 and subchannel 2 in experiment 8 drops to around 90% because several adjacent subchannels are subject to interference. Those subchannels BDR in Figure 4.6 (system bandwidth 10 MHz) also dropped, but the decrement is less and the BDR is still above 95%. The experiment result shows that the receive patterns on the MS do reflect the existence of interferers. By examining these p_{rx}s and p_{tx}s as introduced below, we can decipher the interference relations in a femtocell network.

4.3 Interference Identification

The experiments above show that the interference can be identified from the subchannel BDR differences. This section introduces the interference identification algorithm that finds out which interferer causes the interference. Consider a femtocell network that consists of a set of femtocells F and a set of MSs M. The femtocell to which an MS m is associated is denoted as $f(m)$. m can only receive RB from $f(m)$, and $f(m)$ is the only Tx to m. All other transmissions from $f \in (F - f(m))$ are considered potential interference signals.

In the pattern zone of each frame, every femtocell f transmits a transmission pattern $p_{tx}(f)$ coordinated by the central server. m replies the ACKs about the reception of the RBs back to $f(m)$. According to these ACKs, $f(m)$ can extract the received pattern $p_{rx}(m)$. Each pattern can be represented as several integers depending on the length of the pattern. The system bandwidth in Mobile WiMAX can be 5, 10 or 20 MHz, which gives us $|p| = 15,\ 30$ or 60. Thus, one or two four-byte integers can represent a pattern in Mobile WiMAX. All received patterns p_{rx}s are gathered at a central controller through the backhaul. The central controller is responsible for identifying the inter-femtocell interference based on these patterns. We propose the following algorithm by examining these patterns for interference identification.

$p_{tx}(f(m)) = p_{rx}(m)$: The received pattern on m is exactly the same as the transmission pattern from $f(m)$. The data carried by all subchannels are received correctly. Obviously, this m is not experiencing any interference from nearby femtocells. It is classified as an interference-free MS. No further interference processing is needed.

$p_{tx}(f(m)) \neq p_{rx}(m)$: The femtocell $f(m)$ observes that m receives a different pattern. For a subchannel i occupied by $f(m)$ (i.e., $p_{tx}(f(m))[i] = 1$), it expects that m receives the RB correctly and $p_{rx}(m)[i]$ equals 1. If $p_{rx}(m)[i]$ is not 1, it means that *there is an interferer whose* $p_{tx}(Fx)[i]$ *is 1* or *subchannel i is suffering from frequency selective fading*. The m is classified as a victim MS.

If $p_{rx}(m)[i] = 1$ and a suspicious interfering femtocell has $p_{tx}(f)[i] = 1$, there are two possible reasons. First, the suspicious interfering femtocell f is *not* a real interferer to m. Second, the suspicious interfering femtocell is a weak interferer to m because the transmission of the suspicious

interferer does not cause a loss on the subchannel i.

Suppose there are two suspicious interfering femtocells fx_1 and fx_2 a $Tx = f(m)$, whose transmission patterns are as follows:

$$p_{tx}(fx_1) = 01011001110111$$

$$p_{tx}(fx_2) = 11100011111010$$

$$p_{tx}(f_m) = 10100101101111$$

However, only fx_1 is a real interferer, $Fx = \{fx_1\}$ and fx_2 is a non-interfering femtocell. We propose an efficient interference identification algorithm by computing several patterns based on p_{tx} and p_{rx} described as follows. We will demonstrate how real interferer is detected and how to rule out non-interfering femtocell using this example.

4.3.1 The Interference Free Pattern

We may assume that all femtocells are suspicious interfering femtocells, but it is unnecessary to examine all femtocells in the network. In cellular networks, the MSs maintain a list of nearby cells for potential handovers. Therefore, we can take advantage of this information and only examine femtocells whose RSS at MS m is above a certain *suspect threshold* θ. Note that we do not rely on this suspect threshold θ solely. Instead, we just list those femtocells as suspicious interfering femtocells, denoted as *suspect(m)*. We exclude *non-interfering femtocells* from suspicious interfering femtocells to an MS m based on *interference-free pattern* p_{free} defined below.

Definition 1. *The interference-free pattern between an MS m and a femtocell* $f \neq f(m)$ *is defined as* $p_{free}(f, m) = p_{tx}(f) \cap p_{rx}(m)$. *If* $p_{free}(f, m) \neq 0_n$, *m is said to be interference-free from f because even with the concurrent transmission of f, m is able to receive data correctly from its associated femtocell* $f(m)$.

Suppose the received pattern $p_{rx}(m)$ at m is 10100100001000. The interferer fx_1's transmission

pattern perfectly collides with $p_{tx}(f(m))$. The p_{free}s as follows:

$$p_{free}(fx_1, m) = 00000000000000$$

$$p_{free}(fx_2, m) = 10100000001000$$

Clearly, fx_1 is a real interferer and fx_2 is an non-interfering femtocell.

However, the interference-free pattern faces two problems. First, if a suspicious interfering femtocell f generates a pattern that is exactly the same as $f(m)$, $p_{rx}(m)$ becomes 0_n because the interferer collides with $f(m)$ in all subchannels. If $p_{rx}(m) = 0_n$, for all other $f \in$ *suspect(m)*, $p_{free}(f, m) = 0_n$. All suspicious interfering femtocells are mistakenly considered as real interferers. Therefore, we require that the transmission patterns to each femtocell must be unique. This can be coordinated by the central controller. Second, it is possible that a *weak interferer* does not introduce a clear interference pattern. Suppose interferer fx_1 is a weak interferer and fails to cause significant RB losses on three subchannels (underlined bits). The received pattern $p_{rx}(m)$ on m may be 1$\underline{1}$10$\underline{1}$100001010. The interference-free pattern

$$p_{free}(fx_1, m) = 01001000000010 \neq 0_n$$

which mistakenly identifies m as interference-free from fx_1. Therefore, depending on interference-free pattern solely is insufficient. To improve the accuracy, we add more strict constraints that can better identify real interferers.

4.3.2 The Process of Inclusion

We define an *interference pattern* p_{ix} to identify subchannels that are experiencing interference.

Definition 2. *The interference pattern on a mobile station m is defined as* $p_{ix}(m) = \overline{p_{rx}(m)} \cap p_{tx}(f(m))$.

The $p_{rx}(m)$ in our example is:

$$\overline{p_{rx}(m)} = 0\underline{0}01\underline{0}011110\underline{1}01$$

$$p_{tx}(f_m) = 10100101101111$$

$$p_{ix}(m) = 00000001100101$$

In Definition 2, $p_{ix}(m)[i] = 1$ indicates that the RB carried by subchannel i is lost. In the case that $p_{rx}(m) = p_{tx}(f(m))$, all data are received correctly on all subchannels. The interference pattern $p_{ix}(m)$ should be 0_n, which means that m is interference-free to all femtocells and m is a interference-free MS and *suspect(m)* is empty. This is the condition 1 $p_{tx}(f(m)) = p_{rx}(m)$.

However, if $p_{ix}(m) \neq 0_n$, some subchannels are subject to interference. The m is a victim MS. The next step is to figure out which interferer contributes to the interference. We define a *unique pattern* p_{ux} for this purpose.

Definition 3. *The unique pattern of a suspicious interferer f is defined as* $p_{ux}(f) = p_{tx}(f) \cap \overline{p_{tx}(g)}, \forall g \in \{suspect(m) - f\}$. *That is* $p_{ux}(f) = c_1c_2...c_n$, *where* $c_i = 1$ *if* $p_{tx}(f)[i] = 1$ *and for all* $g \in \{suspect(m) - f\}$, $p_{tx}(g)[i] = 0$; *otherwise,* $c_i = 0$.

We can compute $p_{ux}(fx_1) = 00011000000101$ and $p_{ux}(fx_2) = 10100010001000$. The unique pattern $p_{ux}(f)$ denotes the subchannels that are *only* occupied by f. If c_i in $p_{ux}(f)$ is 1, it means that f is the only possible interferer that affects subchannel i.

We propose a **process of Inclusion** that identifies which suspicious interfering femtocell is responsible for causing the RB lost on the subcarriers by using *conflict pattern* p_{cx}, which is defined as follows.

Definition 4. $p_{cx}(f, m) = p_{ux}(f) \cap p_{ix}(m)$ *is the conflict pattern on m caused by f, where* $c_i = 1$ *means the RB loss on subchannel i is definitely caused by f.*

The p_{cx}s in our examples are $p_{cx}(fx_1, m) = 00000000000101$ and $p_{cx}(fx_2, m) = 00000000000000$. From p_{cx}s we can infer which interferer is responsible for the RB loss in a certain subchannel. If $p_{cx}(f, m) \neq 0_n$, f is a real interferer to m. In our example, we can see that fx_1 is responsible

for two RB losts at subcarrier 11 and 13. Moreover, we successfully rule fx_2 out as a suspicious interfering femtocell.

4.3.3 The Process of Exclusion

We can also exclude non-interfering femtocells from suspicious interfering femtocells according to Definition 5.

Definition 5. *$p_{irr}(f, m) = p_{ux}(f) \cap p_{rx}(m)$ is the irrelevant pattern of m caused by f, where $c_i = 1$ means f is the only possible interferer, but m is not experiencing interference from f as m still receives the RB correctly.*

In our example, the $p_{irr}(f, m)$s are $p_{irr}(fx_1, m) = 00001000000000$ and $p_{irr}(fx_2, m) = 10100000001000$. Obviously, if $p_{irr}(Fx, m) \neq 0_n$, it means although some subchannels are occupied by both the Fx and the $f(m)$, no collision is observed on m, confirming that Fx is an non-interfering femtocell to m. Note that although $p_{irr}(fx_1, m) \neq 0_n$, it does not mean fx_1 is not interfering with m. p_{cx} has already determined that fx_1 is a real interferer.

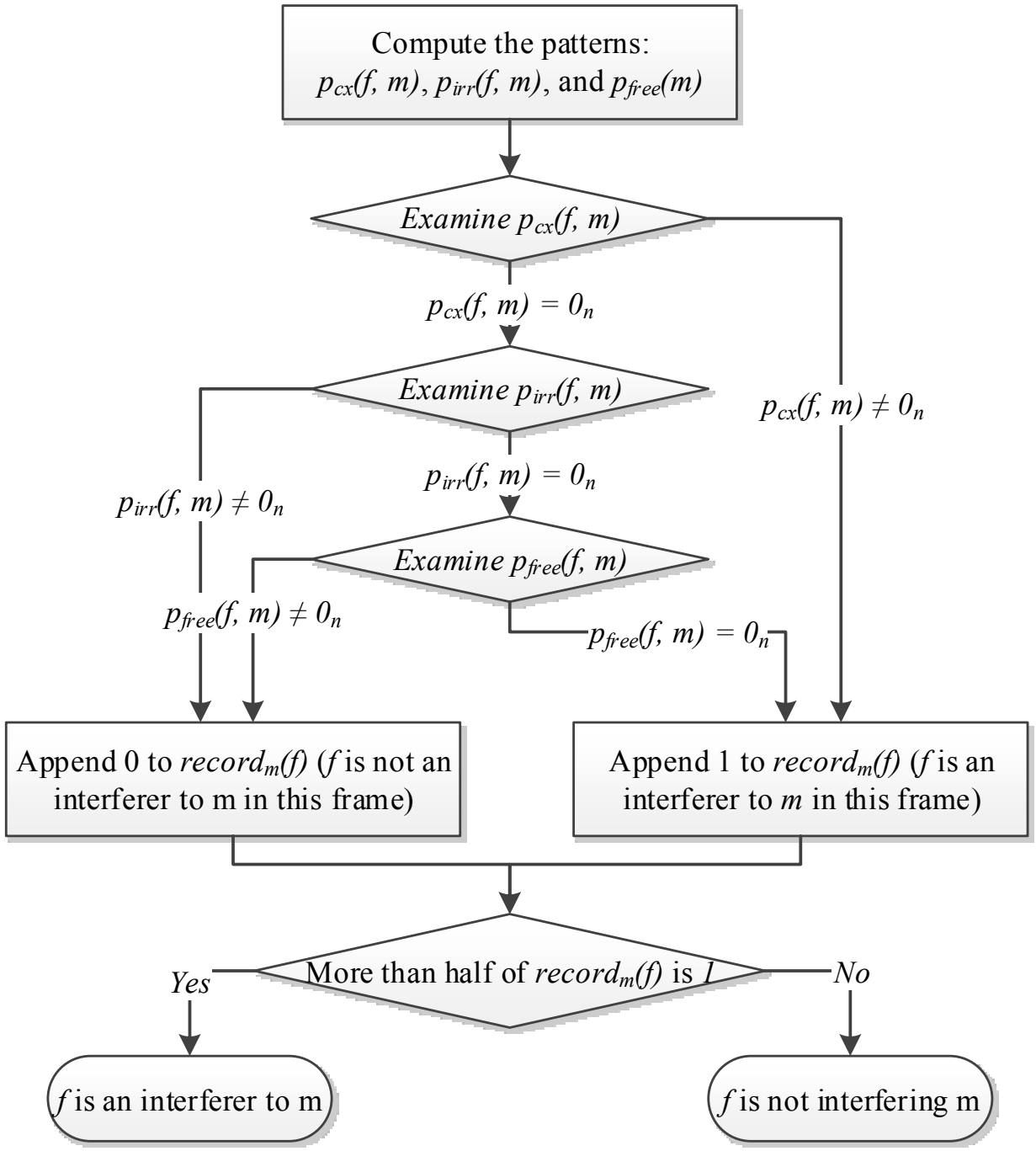


Figure 4.7: Decision flow diagram of each $f \in suspect(m)$ for each frame received by m.

4.3.4 Summary of Interference Identification

We determine whether or not m is experiencing interference from f based on three criteria in the following order.

- $p_{cx}(f, m) \neq 0_n$, f is a real interferer to m.
- $p_{irr}(f, m) \neq 0_n$, f is not interfering m.
- $p_{free}(f, m) \neq 0_n$: m is interference-free from f.

In other words, in addition to the loose constraint in Definition 1, we first check whether there exist subchannels in which f is the only suspicious interfering femtocell and RB loss is observed. If the condition is true, f is definitely a real interferer to m; otherwise, we check whether f can be excluded from suspicious interferers according to Definition 5. Finally, we use the interference-free pattern in Definition 1 to check whether m is interference-free from f if there is no such a subchannel where f is the only suspicious interfering femtocell.

If all of the three criteria do not hold, the f is conservatively considered as a real interferer. As mentioned earlier, we only examine *suspect(m)* whose RSS at m is above a threshold θ. The worst case performance is bounded by identifying all $f \in$ *suspect(m)* as real interferers to m. However, we cannot blindly trust the identification from merely one frame. An RB on subchannel might be lost due to frequency selective fading, instead of due to interfering femtocells. The decision needs to be made based on several consecutive frames, so that the side effect of frequency selective fading and weak interferers and weak receivers is alleviated. The interference identification algorithm is summarized in a flow diagram in Figure 4.7.

Overhead. As mentioned in Section 4.3, the transmission patterns and received patterns are gathered in the central controllers through the backhaul and stored as several integers. The operations to compute the patterns in our algorithm are bitwise operations that take constant time. Therefore, the time consumption of processing the patterns in the central controller is $O(|F| \cdot |M|)$, where $|M|$ and $|F|$ are the number of MSs and femtocells in the network.

Building the conflict graph. The central controller performs the interferer identification for each frame. The input to the interference identification algorithm is a network N and the output is a conflict graph G. Considering a femtocell network $N = \{F, M\}$ where F is a set of femtocells and M is a set of MSs, $|F|$ and $|M|$ denote the number of femtocells and MSs respectively. Each MS

m is associated with exactly one femtocell, denoted as $f(m)$. m can only receive data from $f(m)$. Each m puts the femtocells whose RSS at m are greater than θ in the list of suspicious interferers, denoted by *suspect(m)*. For each f in *suspect(m)*, we keep a history record $record_m(f)$, 1 denotes f is a real interferer to m, 0 otherwise. If more than half of the records are 1, f is regarded as a real interferer to m. The femtocells are modeled as vertices in $G = \{V, E\}$, where $V = F$. For f, $g \in F$, if f is interfering with g, an edge (f, g) is added to E. The weight of the vertex $w(f)$ is the number of victim MSs associated to f.

In each frame, the central controller generates a unique disjoint pattern to every femtocell.Each femtocell f transmits this $p_{tx}(f)$ From the ACKs of each m, $f(m)$ extracts the received pattern $p_{rx}(m)$. These p_{rx}s are forwarded to the central controller. The central controller infers the interference relationships by applying the three rules and build the conflict graph. Based on the conflict graph, the central controller assigns isolated resource to interferer femtocells, which will be discussed in Section 4.4. We introduce the transmission pattern generation that improves the interference identification performance.

4.3.5 Transmission Pattern Generation

The main idea of our interference identification algorithm is to generate and examine collisions in the pattern zone. In order to identify which suspicious femtocell is responsible for the collision in the specific subchannel, we proposed the *unique pattern* p_{ux} in Definition 3. We want to maximize the useful bits in p_{ux}s to maximize the power of p_{ux}s. An intuitive assignment to the previous example is as follows.

$$p_{tx}(f) = 11111111111111$$

$$p_{tx}(fx_1) = 11111111100000$$

$$p_{tx}(fx_2) = 00000011111111$$

In this case, $p_{ux}(fx_1) = 11111100000000$ and $p_{ux}(fx_2) = 00000000011111$ have five useful bits. However, to m and f, fx_1 and fx_2 are suspicious interfering femtocells, but to other

MSs that are associated with fx_1 or fx_2, f might also be an interferer. Let us define $F(m) = \{f(m) \cup suspect(m)\}$ and $|F(m)|$ denotes the size of $F(m)$. Therefore, for all femtocells in $F(m)$, their transmission pattern assignments must satisfy the following constraints. For each pair of $f_1, f_2 \in F(m)$, $p_{tx}(f_1) \cap p_{tx}(f_2) \neq 0_n$, and for each triplet $f_1, f_2, f_3 \in F(m)$, $p_{tx}(f_1) \cap p_{tx}(f_2) \cap p_{tx}(f_3) = 0_n$.

In other words, for any pair of femtocells in $F(m)$, they must share some common bits that are not shared with any other femtocell. These bits will become the p_{ux} between this pair of femtocells. This can be achieved by using combinations of 2 out of n femtocells. For example, suppose $F(m) = \{f_1, f_2, f_3\}$, the number of combinations is $\binom{3}{2} = 3$ and the combinations are $C(3) = \{(f_1, f_2), (f_1, f_3), (f_2, f_3)\}$. The central controller assigns 1 to $p_{tx}[i]$ to fs in the tuple at index i. For example, because $C(3)[0] = (f_1, f_2)$, $p_{tx}(f_1)[0] = 1$ and $p_{tx}(f_2)[0] = 1$. The transmission patterns are shown as follows.

$$p_{tx}(fx_1) = 110**********$$

$$p_{tx}(fx_2) = 101**********$$

$$p_{tx}(fx_3) = 011**********$$

We can generate the combinations for an $F(m)$, whose $|F(m)| = n$ by the equation defined as follows.

Definition 6. *The combinations C(n) for an $F(m)$, whose $|F(m)| = n$ is defined as $C(n) = C(n-1) \cup \{(f_1, f_n), (f_2, f_n), ..., (f_{n-1}, f_n)\}$, where $C(2) = \{(f_1, f_2)\}$ and $n \geq 3$.*

The patterns generated from $C(5)$ is shown below.

$$p_{tx}(fx_1) = 1101001000****$$

$$p_{tx}(fx_2) = 1010100100****$$

$$p_{tx}(fx_3) = 0110010010****$$

$$p_{tx}(fx_4) = 0001110001****$$

$$p_{tx}(fx_5) = 0000001111****$$

We can repeat the patterns until it reaches the desired pattern length.

$$p_{tx}(fx_1) = 11010010001101$$

$$p_{tx}(fx_2) = 10101001001010$$

$$p_{tx}(fx_3) = 01100100100110$$

$$p_{tx}(fx_4) = 00011100010001$$

$$p_{tx}(fx_5) = 00000011110000$$

However, a femtocell f might be suspicious interfering femtocell to more than one MS, but the femtocell can have only one p_{tx} in a frame. The central controller needs to assign patterns to this f that maximizes the number of useful bits in the unique patterns. For a pattern generated in $C(n)$, the number of 1s is n out of $\binom{n}{2}$ bits. The larger $F(m)$s have less useful bits available for unique patterns. Therefore, we need to generate patterns for m with larger $|F(m)|$ first.

For each $m \in M$, the central controller assigns patterns to $F(m)$ in descending order based on $|F(m)|$. The central controller generates patterns as described above and randomly assigns the patterns to each $f \in F(m)$. If there is an $f \in F(m)$ that has already been assigned a pattern in $F(m')$ (because $|F(m')| > |F(m)|$), the central controller skips it and remain using the pattern assigned in the set $F(m')$. The patterns that have skipped in might affect this $F(m)$. However, there are still $\binom{n}{2} - n$ useful bits for this $F(m)$, where $n = F(m')$. When the pattern assignment is done, the central controller shuffles the columns of the patterns in order to offset the effects of frequency selective fading. Finally, these patterns are sent to the femtocells through the backhaul as their p_{tx}s.

4.4 Resource Allocation and Assignment

Because there are interference-free MSs that do not experience interference from any femtocell, their associated femtocells can transmit data to them at the same time. The transmission period is thus divided into *reuse zone* and *isolation zone* similar to FERMI [8] as shown in Figure 4.3. In the *reuse zone*, all interference-free femtocells transmit at the same time, and the *isolation zone* is used by femtocells that are subject to interference. Compared with prior schemes that isolate resources

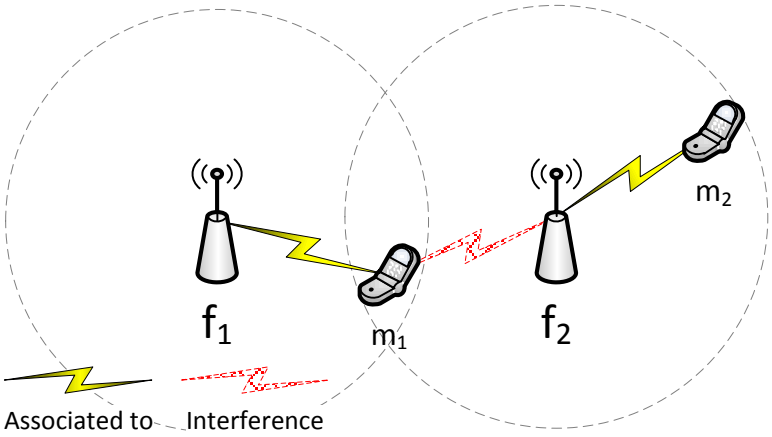


Figure 4.8: Interference scenario.

for each femtocell, the structure is more efficient as interference-free femtocells are allowed to utilize the whole band. To maximize the benefit of *reuse zone*, accurate interference identification is critical.

The example in Figure 4.8 demonstrates the importance of interference identification. Assuming that m_1 is not experiencing interference from f_2, both m_1 and m_2 are interference-free MSs. f_1 can assign all resources to m_1 and f_2 can assign all resources to m_2. There is no need to isolate resources for them. They can reuse the same resource without interfering with each other.

However, if f_2 is *mistakenly* identified as a real interferer of m_1, m_1 and m_2 have to give up the *reuse zone* and compete with others in the *isolation zone*. The unnecessary resource isolation results in resource underutilization. Our interference identification algorithm introduced above aims at identifying real interferers and excluding non-interfering femtocells. Interference-free femtocells can utilize the whole band at the same time in the *reuse zone*. For the *isolation zone*, we propose a weighted vertex-coloring (WVC) algorithm to allocate resources based on the conflict graph. The goal is to assign some isolated resource blocks to the victim MSs in each femtocell with minimized interference and maximized resource utilization while maintaining weighted max-min fairness [53].

FERMI [8] proposes a resource assignment algorithm that assigns resource by identifying maximal cliques in the conflict graph and assigning the resources to the vertices in those maximal

cliques. For general graphs, listing all maximal cliques takes exponential time, but for *chordal graphs*, listing all maximal cliques can be done in linear time. Therefore, they adopt an $O(|V| \cdot |E|)$ complexity triangulation algorithm [10] that converts general graphs into chordal graphs by adding *fill-in* edges. As long as the graph is chordal, FERMI guarantees to produce optimal assignment. However, adding an extra edge between two vertices u and v means u and v are also interfering with each other, and they need resource isolation. The system needs to assign some isolated resource blocks to u and v even though they do not interfere with each other and thus the resource utilization is decreased. As the density of femtocell increases, the number of vertices increases and the number of *fill-in* edge increases drastically. In addition, the triangulation algorithm [10] involves paths with special property and is very time-consuming in large graphs.

Since the conflict graphs are weighted graphs, we cannot merely formulate the problem as a multi-coloring problem. We formulate the problem as a weighted vertex coloring problem (WVCP). Unlike graph coloring for Wi-Fi [88], we cannot model one subchannel as one color and minimize the number of colors used. This is because a subchannel may be assigned to different mobile stations in different resource blocks in a frame. Moreover, if we model one subchannel as one color, the subchannel assignment to the vertices is bundled with the color class and results in *discontinuous* subchannel assignment. In our work, we model a color class as a set of nonadjacent vertices that can share the same *set of subchannels* without causing interference. Our WVC assignment algorithm is a two-pass procedure: (1) assign colors to all vertices in G and (2) allocate and assign subchannel subsets to the vertices based on the color assigned.

A proper coloring C of a weighted graph $G = (V, E)$ is $\{S_1, S_2, ..., S_k\}$, where S_i is a disjoint independent subset of V and $S_1 \cup S_2 \cup ... \cup S_k = V$. An independent set S_i in G is a set of pairwise nonadjacent vertices. The weight of a vertex set S_i is denoted as $\alpha(S_i)$ and is defined as the maximum weight vertex in that S_i. The weight of the proper coloring C is denoted as $W(C) = \sum_{i=0}^{k} \alpha(S_i)$. The goal is to minimize $W(C)$.

It is known that vertex coloring is NP-hard. We can reduce WVCP to vertex coloring by letting the weights on all vertices to 1. Therefore, we propose a heuristic algorithm that colors the vertices

in the order of a breadth-first-search tree rooted at the highest degree vertex. When assigning a vertex v, we assign v to the independent set S_i such that the *increment of W(C) is minimized*. We check if S_i can accommodate v in the order of descending $\alpha(S_i)$. If none of these independent set can accommodate v, a new independent set $\{v\}$ is added to C.

After computing the independent sets $\{S_1, S_2, ..., S_k\}$, we assign actual subchannels in the descending order of $w(v)$ in each independent set S_i. Let us denote the vertices that are adjacent to v as $adj(v)$. For each S_i in C, $S_i' = adj(v) \cap S_i$ is the set of adjacent vertices that are in the same independent set, and they can share the same subset of subchannels. $\alpha(S_i')$ is the maximum weight of v's adjacent vertices that *have not been* assigned subchannels yet. $\alpha(S_i')$ is the *contending load* of independent set S_i to v. Therefore, the sum of contending load to v is $\sum_{i=0}^{k}(\alpha(S_i'))$. The number of available subchannels that are assigned to v is $avail = \lfloor \frac{\alpha(S_u)}{\sum_{i=0}^{k}(\alpha(S_i'))+w(v)} \cdot n \rfloor$, where n is the total number of available subchannels. Since we assign subchannels in the descending order of $w(v)$, current v must be the vertex with the maximum weight. For each $u \in adj(v)$ and let $u \in S_u$, we assign $\lfloor \frac{\alpha(S_u)}{\sum_{i=0}^{k}(\alpha(S_i'))+w(v)} \cdot n \rfloor$.

4.5 Evaluation

In this section, we evaluate the performance of our interference identification and resource assignment through extensive simulations.

4.5.1 Interference Identification

The simulation steps are outlined as follows. First, we create an area and deploy a femtocell network $N = \{F, M\}$, where F and M are femtocells and MSs at random locations. Each femtocell is assigned with a random number of MS (between 1 to 4) and each MS is associated with exactly one femtocell. Second, according to their location and the path loss function, we compute the RSS between every femtocell-MS pair. We define an interfering threshold Γ, such that if the RSS of a femtocell f ($f \neq f(m)$) to an MS m is above Γ, f is regarded as a *real* interferer to m. The set of real interferers to m is denoted as *interferer(m)*. Third, in each frame, the simulator generates

a transmission pattern for each femtocell. Based on the RSS and the transmission patterns, the simulator computes the received patterns on each MS. Finally, from these transmission patterns and received patterns, we invoke the identification procedure to identify the interference relations in the network. The simulation setup and parameters follow closely to [93].

Note that the threshold Γ here is only used to "create real interferers." These real interferers are what we want to identify without knowing or utilizing Γ. We want to detect the real interferers using our interference identification method. In real-world, we do not need to "create" interferers. Any femtocell that is constantly causing interference to MSs is what we want to identify. We adopt WINNER II NLOS model as the indoor path loss function between a femtocell and an MS [87, 93].

$$PL = 46.4 + 20 \cdot log_{10}(R) + 20 \cdot log_{10}(\frac{f}{5}) + 3 \cdot n_w + F_L \tag{4.1}$$

where R is the distance between the femtocell and the MS in meters, f is the centeral frequency in GHz, which is 5.3 GHz in our experiments, and n_w is the number of walls between the femtocell and the MS. We follow the WINNER II NLOS model [93] and assume that there is a light wall every 5 meters and each wall has a penetration loss of 3 dB. F_L is defined as $17 + 4 \cdot (n_f - 1)$, where n_f is the number of floors between the femtocell, and the MS. In our simulations, each femtocell is randomly assigned to a floor between ground floor to the third floor (0 to 3).

4.5.1.1 Simulating the Patterns

In our simulation, for an MS m associated with its femtocell $f(m)$, m receives transmission pattern $p_{tx}(f(m))$, and the received pattern is affected by real interferers. For example, if $p_{tx}(f(m))$ = 000001111111111, $p_{rx}(m)$ should be 000001111111111. We set the interfering threshold to Γ to $-100\ dB$. For each real interferer f in *interferer(m)*, it causes interference on m, and we update $p_{rx}(m)$ according to $p_{tx}(f)$. Suppose m is experiencing interference from an interferer f with $p_{tx}(f)$ = 111111111000000. $p_{rx}(m)$ is updated to 000000000011111.

However, several factors affect wireless links, such as frequency selective fading. It is not sufficient to generate the received pattern merely by the aforementioned method. It is possible that

a real interferer to m is only affecting part of the subchannels of m because this real interferer is a weak interferer to m. It is also possible that the RB carried by subchannel m is dropped because the wireless link between m and $f(m)$ becomes weak (due to frequency selective fading), not because of any interferer. We model such uncertainty by applying an *error probability P* that randomly changes 1s to 0s or 0s to 1s in p_{rx}. Note that an MS m does not receive RBs that are not meant for it, and thus if $p_{tx}(f_m)[i]$ is 0, $p_{rx}(m)[i]$ should never be flip from 0 to 1.

4.5.1.2 Evaluation Metrics

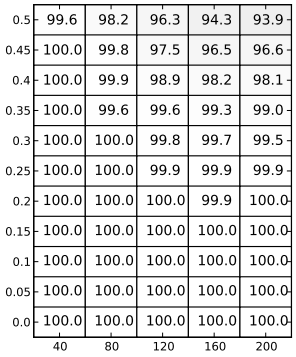
In a femtocell network N, the total number of real interferers is $\sum_{i=0}^{|M|-1}$ *interferer(m_i)* and the total number of suspicious interferers in N is $\sum_{i=0}^{|M|-1}$ *suspect(m_i)*. We define two metrics to evaluate the performance of our interference identification method.

$$detection\ rate : R_d = \frac{d}{\sum_{i=0}^{|M|-1} interferer(m_i)} \tag{4.2}$$

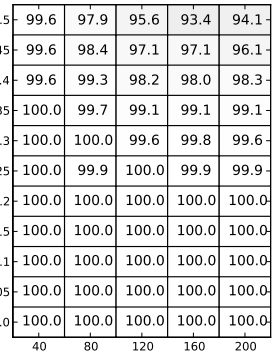
$$exclusion\ rate : R_e = \frac{e}{\sum_{i=0}^{|M|-1} irrelevant(m_i)} \tag{4.3}$$

where d is the total number of real interferers successfully detected and e is the number of non-interfering femtocells (denoted as *irrelevant(m_i) = suspect(m_i) − interferer(m_i)*) successfully identified by our method.

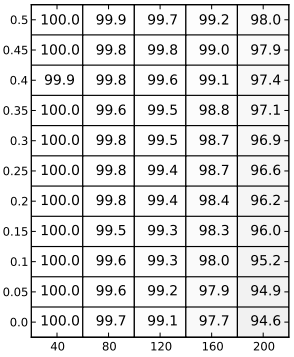
Both metrics are the higher the better, but detection rate (R_d) is more critical than exclusion rate (R_e). If R_d is not 100%, it means that some interferers are not detected correctly. Failing to detect all real interferers may result in interference to $f(m)$. On the other hand, it is less severe if R_e is not 100%. As mentioned in Section 4.3, the worst-case performance is bounded by identifying all $f \in$ *suspect(m)* as interferers to m. The worst case performance means $R_e = 0$, all of the non-interfering femtocells are misidentified as real interferers. It only falls back to identifying interferers based on the threshold of the RSS.



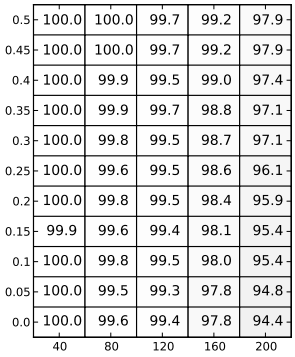
Detection Rate R_d, $|p| = 15$, $\theta = -102dB$



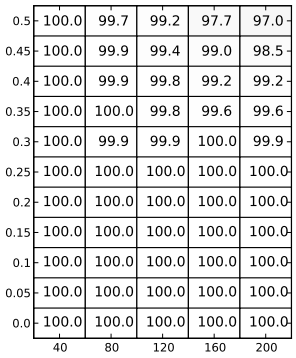
Detection Rate R_d, $|p| = 15$, $\theta = -104dB$



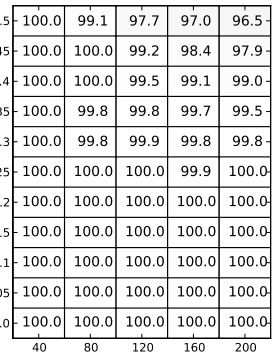
Elimination Rate R_e, $|p| = 15$, $\theta = -102dB$



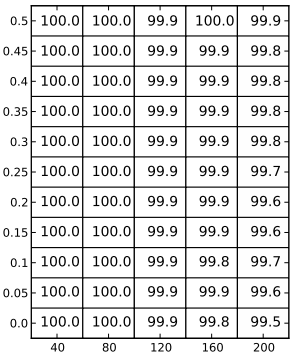
Elimination Rate R_e, $|p| = 15$, $\theta = -104dB$



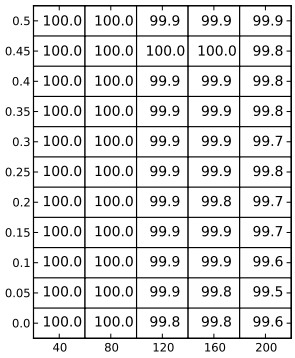
Detection Rate R_d, $|p| = 30$, $\theta = -102dB$



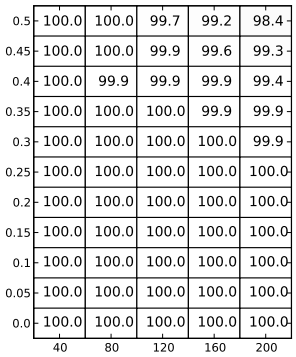
Detection Rate R_d, $|p| = 30$, $\theta = -104dB$



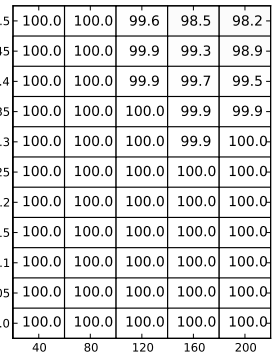
Elimination Rate R_e, $|p| = 30$, $\theta = -102dB$



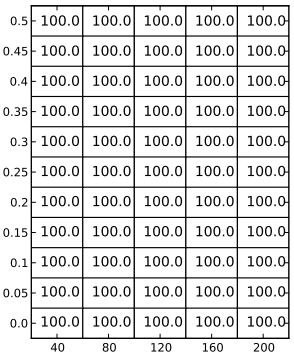
Elimination Rate R_e, $|p| = 30$, $\theta = -104dB$



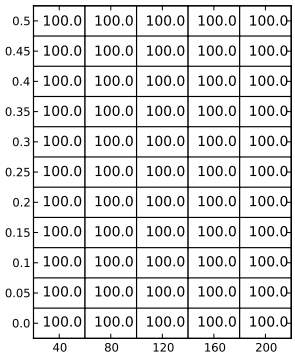
Detection Rate R_d, $|p| = 60$, $\theta = -102dB$



Detection Rate R_d, $|p| = 60$, $\theta = -104dB$



Elimination Rate R_e, $|p| = 60$, $\theta = -102dB$



Elimination Rate R_e, $|p| = 60$, $\theta = -104dB$

Figure 4.9: The *detection rate* R_d and *exclusion rate* R_e. θ is the *suspect threshold* and $|p|$ is the *pattern length*. X-axis denotes femtocell density settings in a ($500 \cdot 500\ m^2$ area) and y-axis denotes *error probability* P.

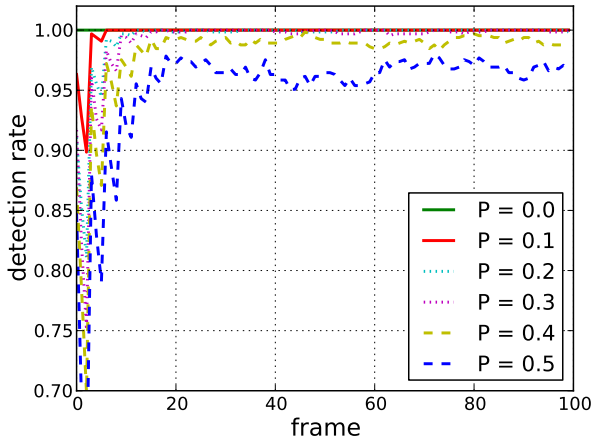
4.5.1.3 Evaluation of Interference Identification

We generate twenty random test cases in a $500 \cdot 500\ m^2$ area for each femtocell density setting (40 to 200 femtocells, incremented by 40). The system bandwidth in Mobile WiMAX can be 5, 10 or 20 MHz, which means the pattern length $|p|$ can be 15, 30 or 60. We simulate each of the test cases 10 times with different *error probability* P and $|p|$ for 100 frames. The *error probability* P varies from 0 to 0.5, increment by 0.05. The simulation results are shown in Figure 4.9. Each block denotes the average of all ten runs of ten test cases, and the darker the block is, the lower the R_e and R_d are.

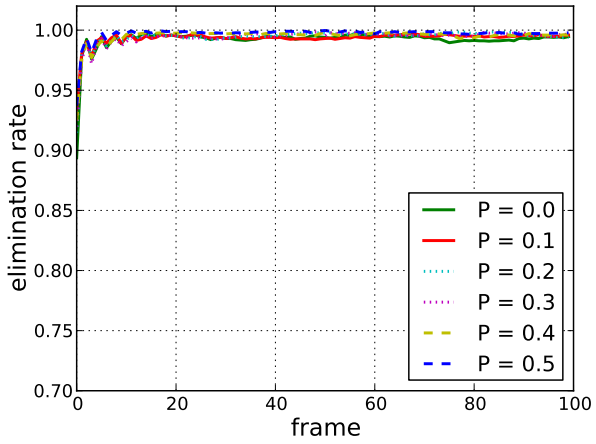
Consider the simulation of 200 femtocells, $(\theta, |p|, P) = (-102\ dB, 15, 0.0)$, R_e is 94.6% and R_d is 100%, which means that all real interferers are detected correctly and 94.6% of the non-interfering femtocells are identified. R_d is always 100% if P is below 0.2, which roughly means that if there are less than 20% of error bits in p_{rx}, we can identify *all* real interferers. It is inevitable that when P increases, the number of bad bits increases and R_d decreases. The pattern becomes less meaningful as P increases because the BDR difference is no longer mainly caused by interference. However, in our interference experiment in Section 4.2, the error probability is at most 10% when the system bandwidth is 1 MHz and 5% when the system bandwidth is 10 MHz. The proposed interference identification works well in these conditions.

One might notice that when there are more femtocells in the area, R_e increases along with P (vertical direction in each graph. It is more noticeable when $|p| = 15$). However, it does not mean better identification performance. We should also take the corresponding R_d into consideration for high femtocell density and high P. Considering both R_d and R_e, the identification performance is reduced slightly in those cases. With a higher error probability P, it starts to identify real interferers as non-interfering femtocells mistakenly.

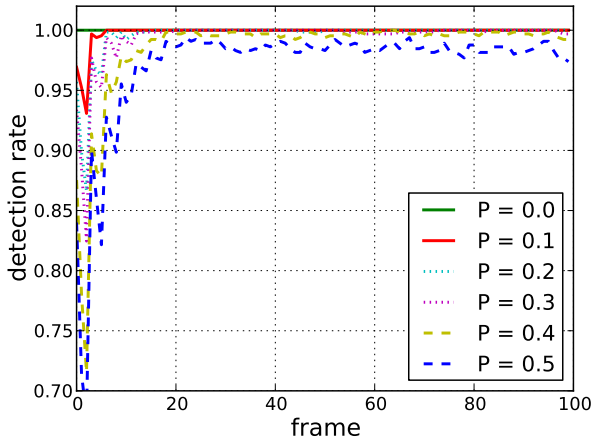
On the other hand, R_e decreases while the density of femtocell increases (horizontal direction in each graph) and the decrement of R_d is less significant. It is because there is a limited number of bits in a pattern. Higher femtocell density means more potential interferer and the possibility of collision is higher. If the RBs in all subchannels collided and $p_{rx}(m) = 0_n$, we cannot extract



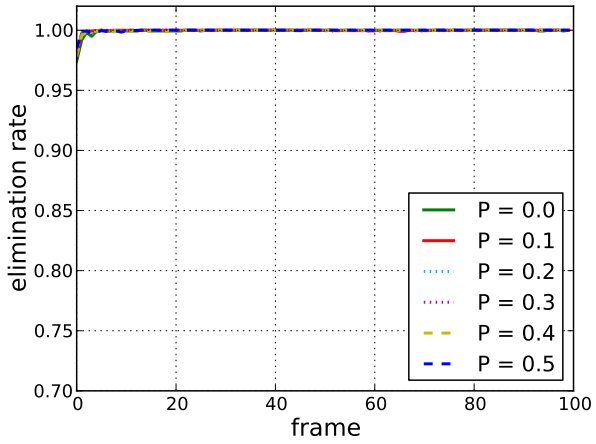
Detection rate R_d, $|F| = 200$, $\theta = -102\ dB$, $|p| = 30$.



Elimination rate R_e, $|F| = 200$, $\theta = -102\ dB$, $|p| = 30$.



Detection rate R_d, $|F| = 200$, $\theta = -102\ dB$, $|p| = 60$.



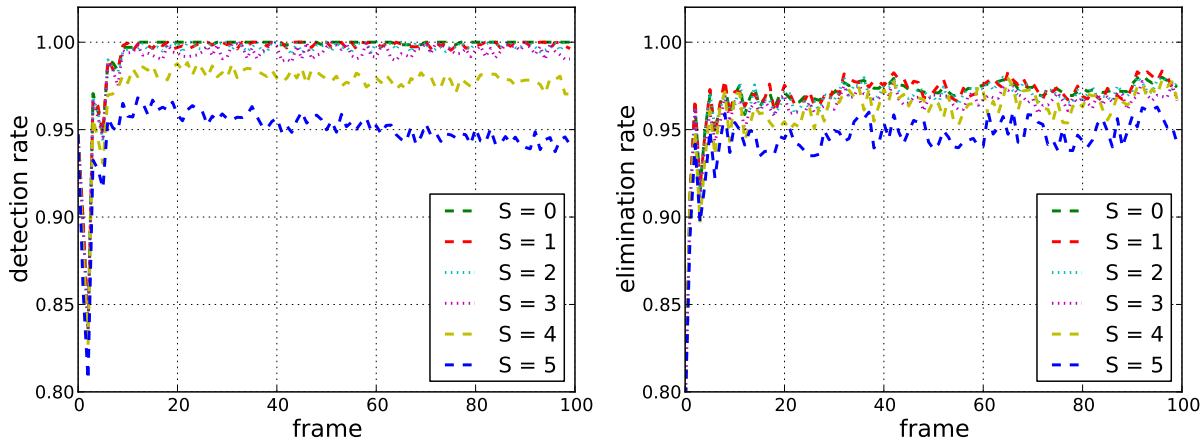
Elimination rate R_e, $|F| = 200$, $\theta = -102\ dB$, $|p| = 60$.

Figure 4.10: Interference Identification for 100 frames.

much useful information from 0_n. Recall that when there are bad bits in a pattern and p_{free} is not sufficient for our goal, we rely on p_{ux}, p_{cx} and p_{irr} to identify the interferer. As the density of femtocell increases, the chance that we have a useful p_{ux} decreases and we cannot extract p_{cx} and p_{irr}. It leaves us no choice but to assume that the suspicious interferer is a real interferer.

For $|p| = 15$, R_e drops as the femtocell density increases, which means more non-interfering femtocells are misidentified as interferers. However, note that our method still effectively identifies 94.4% of non-interfering femtocells in $(\theta, |p|, P) = (-104\ dB, 15, 0.0)$. The performance is expected

to be improved if there are more subchannels ($|p|$ gets longer). With more subchannels, the chance that a received pattern becomes 0_n becomes lower and more useful information can be extracted, and it is confirmed in Figure 4.9.



Detection rate R_d Elimination rate R_e

Figure 4.11: Interference identification performance when MS is moving in a fixed speed. $|F| = 200$, $(\theta, |p|, P) = (-102\ dB, 60, 0.2)$. MSs are moving in a fixed speed S of 1 to 5 m/s.

4.5.1.4 Time Consumption and Mobility

Figure 4.10 shows two sets of simulations of 200 femtocells with $|p| = 30$ or 60 for 100 frames. The interference identification quickly becomes stable in about 15 frames in both simulations. Although we set the length of the records to 40, our method only needs less than 15 frames to achieve a stable identification. The detection rates are always almost 100% in both experiments. The R_es of the experiments where $|p| = 30$ fluctuate more than that of $|p| = 60$, but still more than 98% of the non-interfering femtocells are eliminated when $|p| = 30$. The frame length of OFDMA systems varies from 2 to 20 ms, which means our method is able to achieve stable identification in less than 400 ms. Therefore, our method can handle the network dynamics and user mobility in cellular networks.

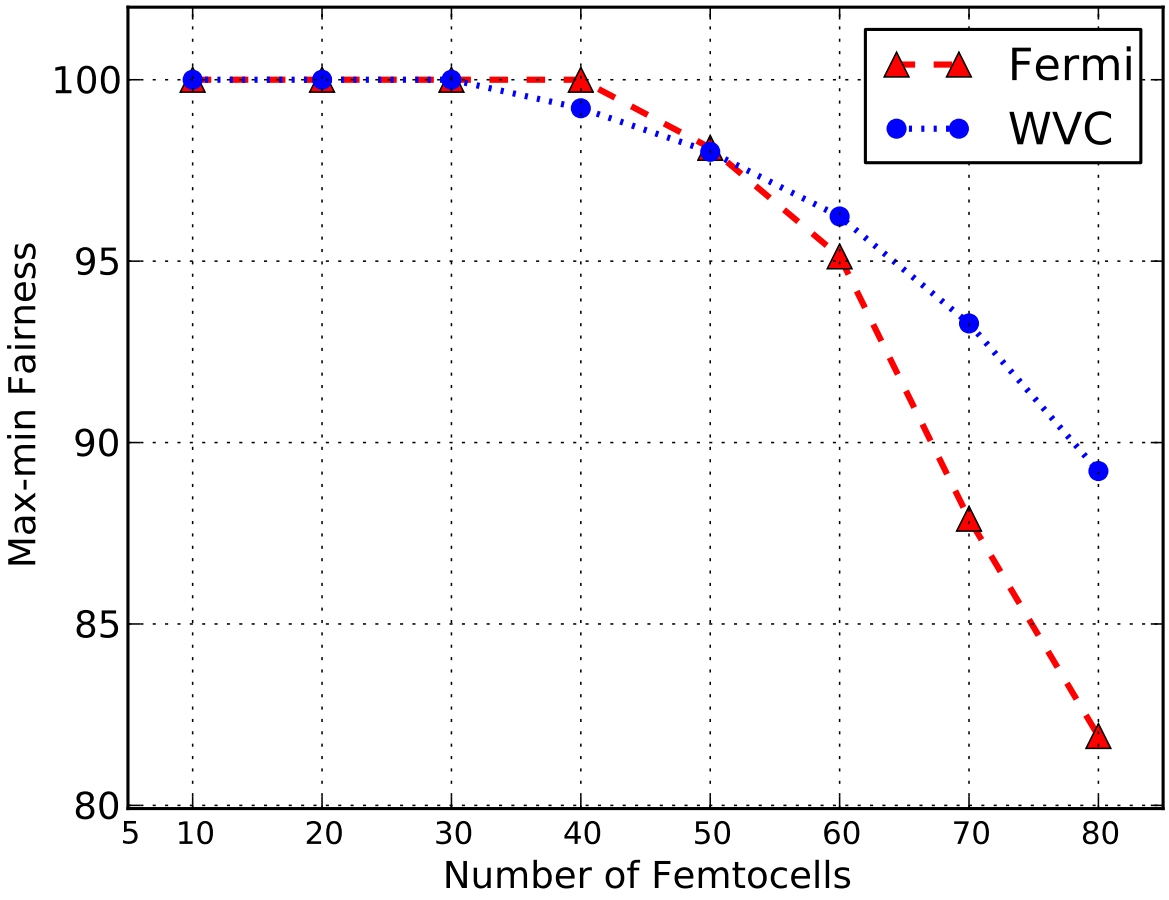


Figure 4.12: Max-min fairness

Figure 4.11 shows the results of mobility simulations consisting of 200 femtocells with $(\theta, |p|, P) = (-102\ dB, 60, 0.2)$ for 100 frames. Each MS is moving at a fixed speed S toward a fixed random direction. The performance is still stable and R_d is almost 100% when the MS moving speed is less than 2 m/s. When the MSs are moving in 3 m/s, R_d starts to fluctuate slightly, but in average R_d at 3 m/s is still stable and close to 100%. However, when the MSs are moving faster than 4 m/s, R_d starts to decrease. It is because our channel model considers a light wall every 5 meters. One frame is 20 ms in our simulation. 100 frames are 2 seconds and the MSs move 8 meters. The MSs starts to move across walls and brings significant penetration loss. However, the original purpose of femtocells is to improve indoor cellular coverage. It is possible that the users

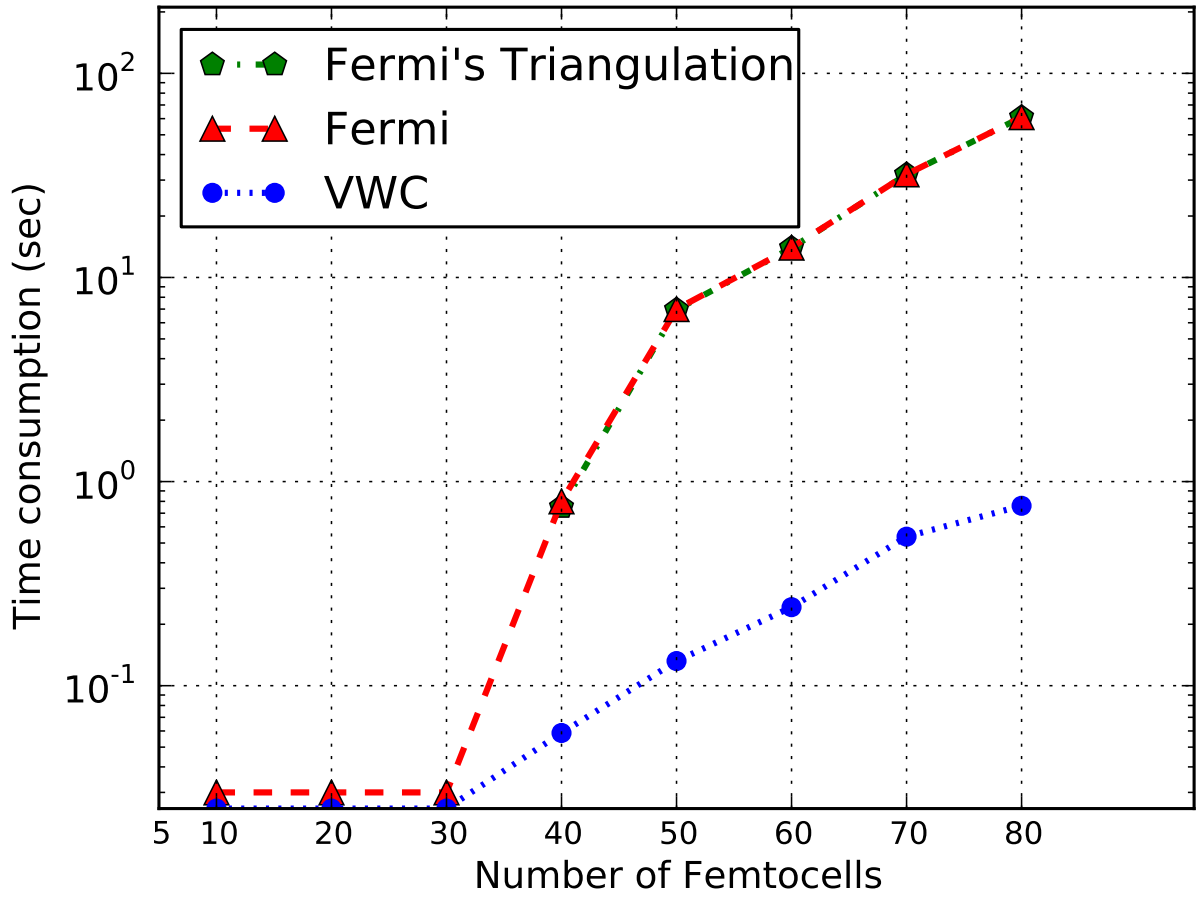


Figure 4.13: Time consumption

are mobile indoor, but moving at speed higher than 3 m/s is less likely. Therefore, our method is able to handle most of the possible indoor MS mobility.

4.5.2 Resource Allocation and Assignment

The central controller constructs the conflict graph as it identifies the interferers. The fairness and the time consumption for the test cases where $\Gamma = -100\ dB$, $\theta = -102\ dB$ and $|p| = 60$ are shown in Figure 4.12 and 4.13. In both figures, the blue and red curves represent WVC algorithm and FERMI, respectively. Each point on the curves is the average of all 10 test cases with certain femtocell density. For low femtocell density (10 - 30 femtocells), the femtocells are usually entirely

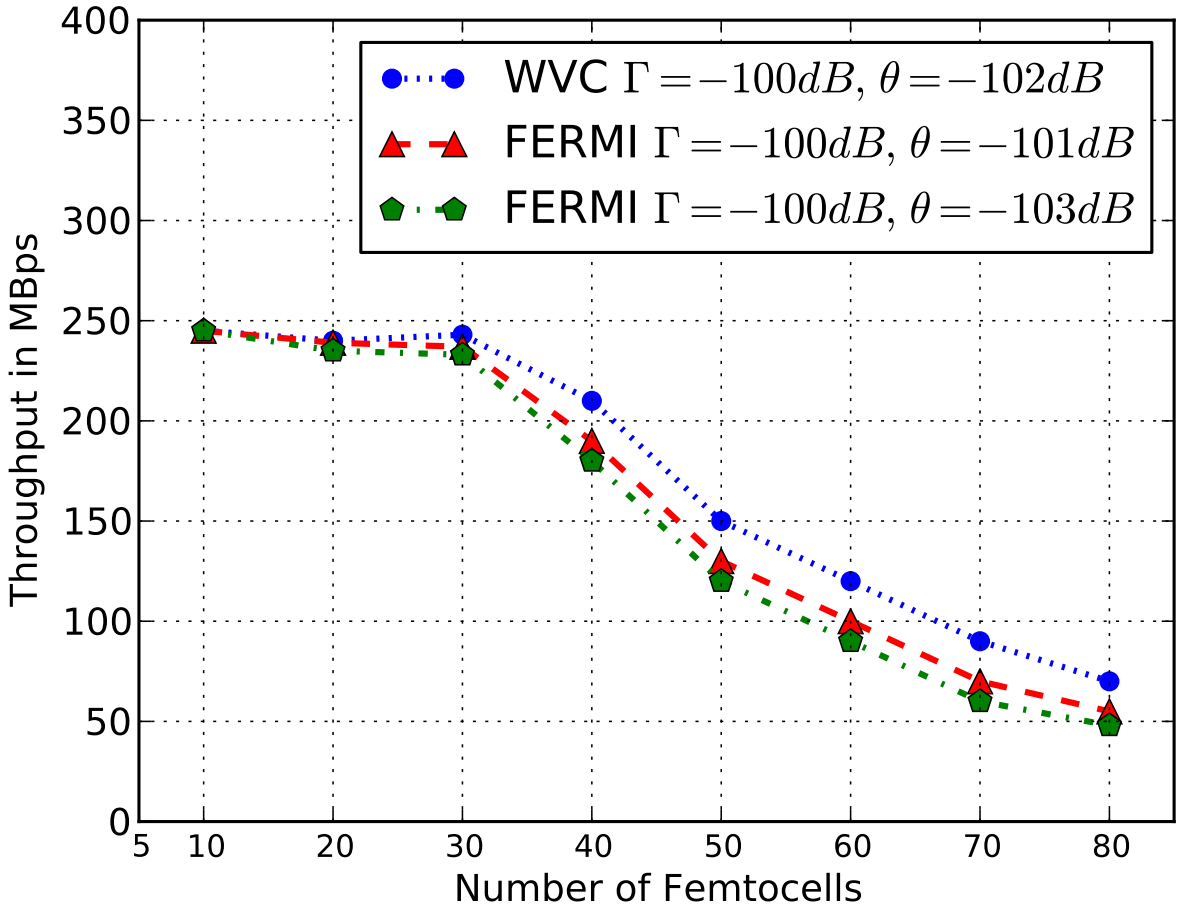


Figure 4.14: Throughput comparison

interference free. There is no conflict graph constructed and there is no resource isolation at all. The time consumption in those scenarios are trivial. When there are 40 femtocells in the network, the generated conflict graphs are usually simple chordal graphs, and FERMI can achieve optimal fairness assignment easily. However, the possibility that the input graph is non-chordal increases drastically along with the size of the graph. It requires triangulation [10] to add the fill-in edges to make the input graph chordal and brings two side effects. (1) First, the extra fill-in edges degrade the fairness significantly as the size of the graph grows. (2) Second, the triangulation algorithm [10] involves paths with special property and can be very time-consuming in large graphs. As we can see in Figure 4.13, the green curve represents the time consumption of the triangulation algorithm. The triangulation algorithm dominates the time consumption of FERMI. Our WVC algorithm in

Modulation	Coding Rate	Receiver SNR (dB)
BPSL	1/2	6.4
QPSK	1/2	9.4
QPSK	3/4	11.2
QAM-16	1/2	16.4
QAM-16	3/4	18.2
QAM-64	1/2	22.7
QAM-64	3/4	24.4

Table 4.1: Received SNR threshold [47].

low femtocell density has slightly lower fairness, but in overall our WVC algorithm outperforms maximal clique method in terms of efficiency.

The throughput results are shown in Figure 4.14. For each RB assigned to MS m, we compute an RSS from its femtocell $f(m)$ to m using the channel model. All other signals from *suspect(m)* are noises to m. An SINR to m is obtained. The simulator converts this SINR from $f(m)$ to m to useful bits using Table 4.1. Finally, the simulator sums up these useful bits and gets the throughput from $f(m)$ to m. The blue curve represents the throughput of WVC algorithm. The red and green curve represents the throughput of FERMI's algorithm under different parameters. Each point denotes the average of ten test cases with certain femtocell density. Higher femtocell density results in resource isolation and less throughput acquired by the MS. In simulations with $|p| = 30$, more than 96% non-interfering femtocells are identified, and the number of edges in our conflict graph decreases significantly. In contrast, FERMI can only conservatively set the femtocells with RSS higher than the suspect threshold θ as interferers and the performance is reduced due to those unnecessary edges in the conflict graph. Even θ is set to *-101 dB* the throughput is still lower than the proposed WVC algorithm with our interference identification.

4.6 Summary

Identifying interference in femtocell network is the most critical step that directly affects the performance of resource allocation. In this chapter, we propose an efficient method that takes advantage of the availability of multiple subchannels in OFDMA to identify the inter-femtocell

interference by generating received patterns on the MSs. These patterns are then collected at a central controller where interference relations are identified. We conduct experiments on USRPs to show the generations of received patterns on MSs. We simulate our interference identification method and demonstrate that if the error probability is less than 0.2, our method successfully identifies *all* real interferers and *at least 94%* of the non-interfering femtocells. Moreover, the method achieves a stable identification in less than fifteen frames, which means our algorithm can handle the dynamic and user mobility of cellular networks. We also propose a weighted vertex-coloring (WVC) based resource assignment algorithm that achieves better fairness for higher density networks in less time compared with FERMI. With the improved identification of real interferers, our algorithm not only assigns resources with better fairness but also achieves higher throughput than existing approaches. Moreover, the proposed interferer identification algorithm is not restricted to femtocells only. It can be applied to all types of cells in HetNets. However, resource isolation significantly reduces the amount of resources and achievable throughput on the MSs. We have discussed the interference identification problem in HetNets. We propose an interference mitigation approach in HetNets that precancels the interfering signal in the next chapter.

CHAPTER 5

INTERFERENCE PRECANCELLATION FOR RESOURCE MANAGEMENT IN HETEROGENEOUS CELLULAR NETWORKS

The next research problem is to mitigate the interference in HetNets, which means avoiding interference from occurring, eliminating the impact of interfering signals, or ensuring the signal is delivered to the intended receiver. Most existing interference mitigation techniques mitigated interference by assigning isolated resources to interfering cells [8, 73, 112, 120, 120, 126] or power control [21, 28, 80]. However, resource isolation leads to significantly fewer resources and power control sacrifices signal quality and coverage. In this chapter, if the exact interferers are identified, we propose an interference mitigation technique for HetNets, called interference precancellation. Note that the interference precancellation can operate with any exact interferer identification algorithm, including but not limited the technique proposed in Chapter 4.

The resources in OFDMA-based 4G cellular networks are either composed of subchannels in the frequency domain, time slices in the time domain or the combination of both domains, as aforementioned in Section 4.1. When multiple cells transmit using overlapped radio resources to an MS located in the overlapped coverage of these cells, the receiving MS is unable to decode the scrambled signal. The resource and the power spent on these transmissions are wasted. The cells that the receiving MS does not intend to receive from cause *interference* to the receiving MS (victim MS) and are the *interferers* to the receiver. However, multiple transmitting cells can reuse the same resource as long as they do not interfere with each other.

The major goals of resource management in HetNets are to enhance throughput by maximizing the degree of resource reuse while minimizing the inter-cell interference (ICI) and maximizing fairness between cells. Most interference mitigation techniques mitigated ICI by assigning isolated resources to interfering cells [8, 73, 112, 120, 120, 126] or power auto-configuration [21, 28, 80].

Techniques that attempt to extract useful messages from scrambled signals have been very popular in the research community. Successive interference cancellation (SIC) [105, 117] is a

technique that attempts to extract useful signal from the scrambled signal at a receiver. The basic idea of SIC is summarized as follows. When the receiver receives a scrambled signal composed of multiple signals, the receiver attempts to decode the strongest signal from the scrambled signal, subtract the strongest signal from the scrambled signal, and extract the weaker signals from the residue.

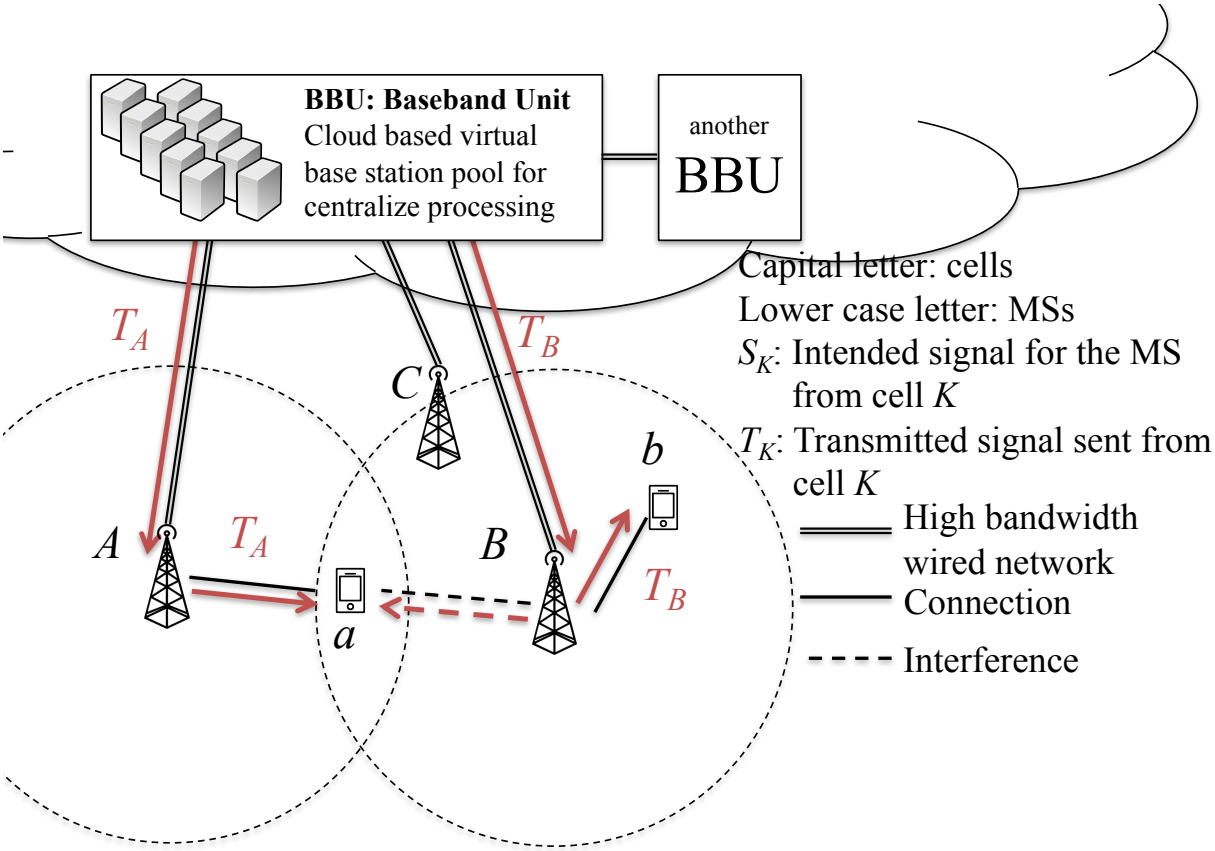


Figure 5.1: An overview to C-RAN and an example.

Although the signal processing consumes relatively less energy compared with other components on modern wireless devices, the signal processing of SIC still introduces extra power consumption that is undesired for the battery-powered mobile devices. Also, since an MS only communicates with one cell, decoding multiple signals is unnecessary to an MS. Moreover, legacy devices cannot take advantage of SIC and are still suffering from collisions and packet loss. Therefore, instead of canceling interference *afterward*, we propose to *precancel* known interference

before the signal is sent. No modification to legacy devices is required to take advantage of the interference precancellation technique.

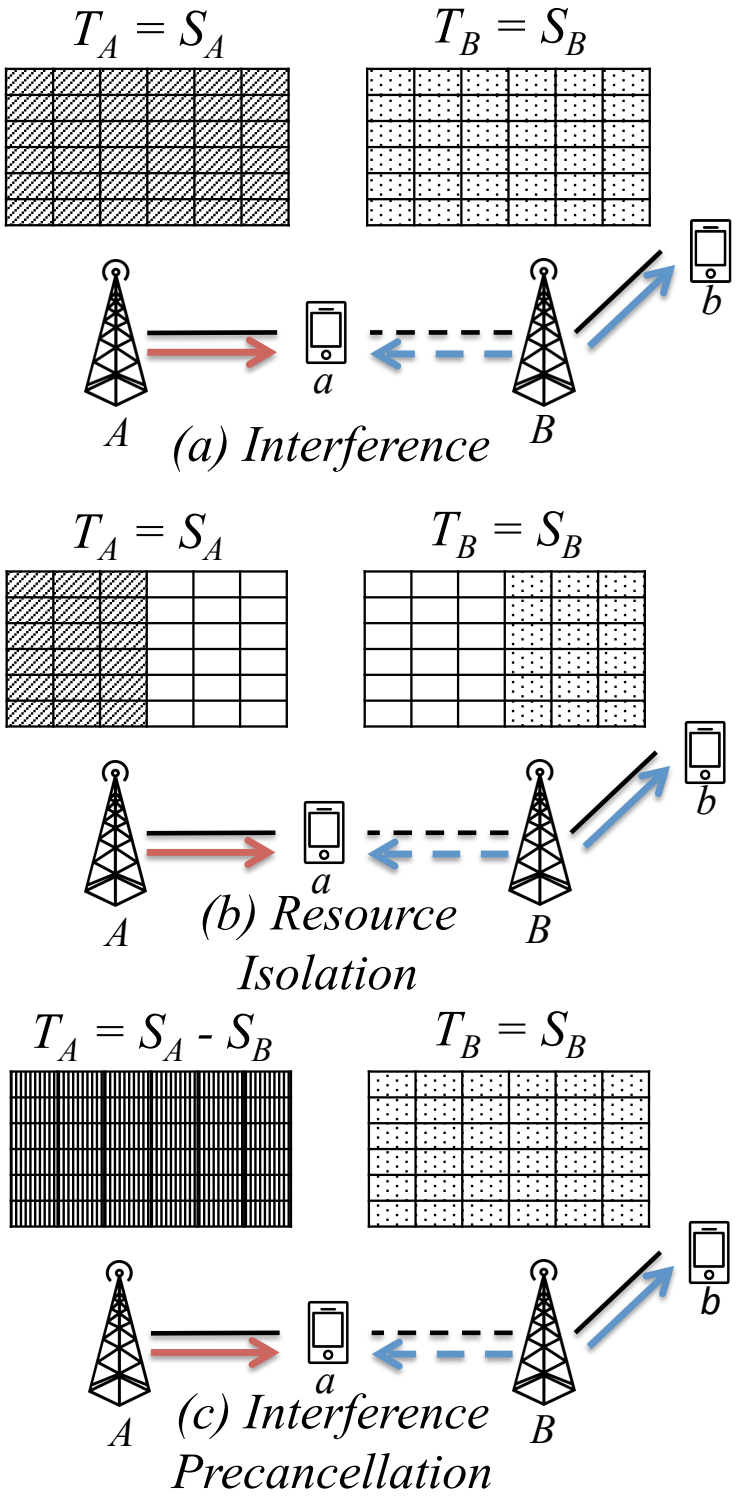


Figure 5.2: Different resource assignments for Figure 5.1.

Many advanced cellular functionalities such as handoff and interferer identification, requires certain cooperation between cells. In order to enable cooperation, several cloud-based architectures are proposed in recent years [11, 23, 41, 48, 111]. These cloud-based architectures are perfect candidates to perform interference precancellation. The cloud radio access network (C-RAN) [23, 48] is a promising candidate for performing interference precancellation.

If the *exact* interferers to an MS can be identified, we propose an *interference precancellation* technique to mitigate interference. In a cellular network, the data for the MSs are originated from the core cellular network. The data are encoded as signals and sent to the cells, and the cells transmit the signals over the air to the MSs. This signal intended to be received by an MS is the *intended signal* and is denoted as S_X, where X is the transmitting cell. As shown in Figure 5.1, the intended signals are sent to cell A and cell B through the high speed wired backhauls. Cell A and cell B then transmit the signal to the MSs via their wireless interfaces. Let us define the signal *actually sent* by cell X as *transmitted signal* and denote it as T_X.

Without any attempt at handling interference shown in Figure 5.2(a), the transmitted signal T_A from cell A to MS a is the original intended signal S_A, and the transmitted signal T_B from cell B to MS b is the original intended signal S_B. T_B *collides* with T_A at MS a, MS a receives $S_A + S_B$ and MS a cannot decode the scrambled signal. Because MS a connects to cell A and wishes to receive from cell A only, T_B is the *interference signal* to *victim MS* a and cell A is the *victim cell*. Therefore, a popular technique to mitigate interference is to assign *isolated resources* to interfering cells as shown in Figure 5.2(b). The interfering cells transmit using non-overlapping (isolated) resource to avoid interference. However, with resource isolation, each MS only gets half of the resources in a frame.

Since in cellular networks, the intended signal (S_A and S_B) being sent to the MSs are originated from the core cellular network, instead of letting $T_A = S_A$ and $T_B = S_B$, which results in a collision at MS a, we propose to let the C-RAN *precancel the interference* as shown in Figure 5.2(c) by letting $T_A = S_A - S_B$. Therefore, $S_A - S_B$ from cell A and S_B from cell B scramble at MS a. The signal received by MS a becomes $S_A - S_B + S_B = S_A$. In this case, the same set of resources

are utilized by both MS a and MS b and the achievable throughput can be significantly increased. Some of the preliminary results are presented in [72]. The key contributions of this chapter are summarized as follows.

- We propose an *interference precancellation* technique. If the exact interferer can be identified, the C-RAN sends the *interference precanceled signal*, which is the intended signal for the victim MS subtracts the interference signal, to the victim cell. The victim cell transmits the *interference precanceled signal* over the air to the intended receiving MS.

- The interference precanceled signal and the interference signal scrambles at the victim MS and becomes the signal intended for the victim MS. The computational burden lies on the C-RAN. No extra computation overhead is required on the MSs and legacy devices can take advantage of this technique.

- Some MSs might still require isolated resources because there may be interference-free MSs, MSs with known exact interferers and MSs with unidentified interferers. We propose a resource management with interference precancellation (RMIP) algorithm that jointly considers MSs that are experiencing a different level of interference.

The rest of the chapter is organized as follows. We discuss the related work and briefly introduce some related background in Section 5.1. Section 5.2 introduces the interference precancellation technique, and Section 5.3 presents RMIP algorithm that jointly considers MSs experiencing a different level of interference. We evaluate RMIP algorithm and report the results in Section 5.4. Finally, Section 5.5 summarizes this chapter.

5.1 Related Work and Background

In this section, we briefly introduce the background of the C-RAN architecture. We also discuss some interference mitigation techniques in 3G and 4G cellular networks and some interference suppression related work.

5.1.1 C-RAN Architecture

A traditional cellular base station (BTS) is an individual identity that serves an area and has its own processing unit, backhaul wired fiber, wireless interface and so on. Each BTS processes and transmits the signals to the MSs independently. The lack of cooperation between traditional BTSs prohibits more advanced functionalities to be implemented. For example, when BTSs are interfering with each other, assigning isolated resources to interfering BTSs to mitigate interference requires cooperation among the BTSs.

Many other advanced cellular functionalities such as handoff and interferer identification, also require certain cooperation between cells. In order to enable cooperation, several cloud-based architectures are proposed recently [11, 23, 41, 48, 111]. The cloud radio access network architecture (C-RAN) [23, 48] is one of the most promising candidates for enabling cooperation between cells. The C-RANs are expected to equip with virtualized baseband processing server pool on the centralized baseband unit (BBU) to deal with the computationally intensive tasks. The distributed wireless interfaces (remote radio heads, RRHs, such as EnodeB in LTE) that connect to the BBU via high-speed wired networks are only responsible for communicating with the MSs. The centralized BBU not only provides superior computational capability, but also enables the cooperation between RRHs.

A C-RAN is able to discriminate up to 504 cell identities. The macrocells, smallcells and the RRHs can all cooperate through the C-RAN. The C-RAN is promising for enabling more advanced functionalities such as interference identification and resource management for HetNets, and is a well-suited candidate for performing interference precancellation. In the following context, we include RRH into smallcells.

5.1.2 Interference Mitigation in Cellular Networks

The macrocells are deployed by wireless service providers with careful preplanning and inter-macrocell interference is minimized at coverage edges. OFDMA frequency preplanning for macrocells is discussed in [22, 96], but these approaches require a certain amount of knowledge of

the base stations, such as antenna configuration and location. Although the information can be obtained easily for macrocells and microcells, some smallcells such as femtocells and possibly picocells, possess little knowledge of the information, so frequency preplanning is not feasible for HetNets. Therefore, interference management is one of the most critical components for network densification of HetNets [128].

Macro-femto interference management has been widely studied. In earlier generation cellular networks (e.g., CDMA and GSM), only one carrier is available and frequency sharing is not available. Therefore, interference management solutions mainly rely on pure frequency isolation [120] or power auto-configuration [21, 28]. In [120], different frequency spectrum bands are assigned to macrocells and femtocells arbitrarily. However, this approach lacks flexibility. When either femtocell or macrocell is idle, the spectrum bands are wasted. On the other hand, power auto-configuration adjusts the transmission power, reduces the coverage of a cell and shrinks the overlapping of the cells. However, reducing transmission power also leads to reduced signal-to-noise ratio and, therefore, harms throughput. For OFDMA-based cellular networks, a centralized approach for cross-tier interference management is proposed in [112], which prevents femtocells from reusing the resources that are occupied by macrocells. In these papers, the knowledge of macrocells is relatively easier to obtain.

Inter-femtocell interference managements through transmission power adjustment is also discussed in the literature [51, 66, 80]. The most straightforward approach to determine inter-femtocell interference is based on the locations of the femtocells. In [66], the inter-femtocell interference is determined by a femtocell gateway using the location information provided by the femtocells. Based on the interference relations, the authors in [66] proposed a centralized adaptive fractional frequency reuse (FFR) algorithm to adjust the coverage area by controlling femtocells' transmit power. The authors in [80] propose that the femtocells need to be self-organizing and to adjust their transmission powers to avoid interference.

Interference mitigation through resource isolation is another popular technique. The work [51] proposes cognitive femtocell to sense interference and adopts a game-theory approach to allocate

isolated resources to avoid interference. In [73], the authors propose to identify interference by generating unique received patterns on the MSs and allocating isolated resources to interfering femtocells using a weighted vertex coloring algorithm. FERMI [8] introduces two dedicated measurement zones in the downlink of a frame to determine if an MS is experiencing interference. FERMI proposes a linear complexity resource allocation algorithm to assign isolated resources to the femtocells. If the interference conflict graph is chordal, FERMI is able to assign optimal fairness resources to the femtocells. A distributed hashing based inter-femtocell resource management is proposed in [112]. Femtocells assign resources to MSs by a random hashing scheme without any coordination. Collisions might occur if two interfering femtocells assign the same RB to their MSs. A collision resolution procedure is thus invoked to resolve the collision, rehashing another available RB to the MS. In RADION [126], instead of assigning resources randomly, a probing method is proposed for the femtocells to search for available subchannels opportunistically.

For base stations (BS) equipped with multiple-input-multiple-output (MIMO) antenna systems [84, 102], the BSs are able to serve multiple MSs using the same resources by zero-forcing (ZF) technique. ZF-based precoding can achieve near-optimal capacity when the number of antennas and the number of MSs are sufficient, even if the MSs have only one antenna. With sufficient amount of antenna elements, a linear single-user conjugate beamforming that is able to significant capacity improvement is proposed in [84]. However, ZF-based precoding requires precise channel coefficients to achieve near-optimal capacity improvement. On the other hand, the interference precancellation proposed in this chapter attempts to reuse the same resources in a more conservative way. Precise channel coefficient is also preferred in interference precancellation, but the regular (non-precancellation) signal is not affected. Moreover, in contrast to MIMO BSs precode the signals for each MSs locally, interference precancellation is performed on C-RAN with a global view to the HetNet.

5.2 Interference Precancellation

In this section, we formulate the interference precancellation technique for HetNets considering channel coefficients. Let us denote the *transmitted signal* from cell X as $T_X(t)$, and $y_m(t)$ denote the *received signal* on an MS m at time t. For a pair of transmitter cell X and receiving MS m, the signal $y_m(t)$ received by m at time t can be written as

$$y_m(t) = \sqrt{\rho_X} \times h_{X,m}(t) \times T_X(t) + n_m(t) \tag{5.1}$$

where ρ_X is the transmission power of cell X, $h_{X,m}(t)$ is the channel coefficient between cell X and MS m at time t and n_m denotes the additive white Gaussian noise (AWGN) on MS m at time t.

In the scenario shown in Figure 5.1, without any interference management, $T_A(t)$ is S_A and $T_B(t)$ is S_B. The scrambled received signal on MS a at time t, $y_a(t)$ can be expressed as

$$\begin{aligned} y_a(t) = &\sqrt{\rho_A} \times h_{A,a}(t) \times T_A(t) + \\ &\sqrt{\rho_B} \times h_{B,a}(t) \times T_B(t) + n_a(t) \end{aligned} \tag{5.2}$$

where ρ_A and ρ_B are the transmission power from cell A and cell B, $h_{A,a}(t)$ and $h_{B,b}(t)$ are the channel coefficients between cell A and MS a and cell B and MS a at time t and n_a denotes the AWGN on MS a at time t. The frame received by MS a is a scrambled signal composed of S_A and S_B and MS a cannot decode $y_a(t)$.

However, as described in Eq. 5.1, the channel coefficient on different communication links might be different and it will affect the received signal at the MS. The interference signal from cell B to MS a can be written as

$$\sqrt{\rho_B} \times h_{B,a}(t) \times T_B(t) \; and \; T_B(t) = S_B. \tag{5.3}$$

In order for the received signal $y_a(t)$ to be decodable on MS a, $y_a(t)$ must satisfy the following equation:

$$y_a(t) = \sqrt{\rho_A} \times h_{A,a}(t) \times S_A + n_a(t) \tag{5.4}$$

We want the signal $y_a(t)$ in Eq. 5.2 to be close to Eq. 5.4 .

$$\begin{aligned}\sqrt{\rho_A} \times h_{A,a}(t) \times S_A =& \sqrt{\rho_A} \times h_{A,a}(t) \times T_A(t) + \\ & \sqrt{\rho_B} \times h_{B,a}(t) \times S_B \\ \sqrt{\rho_A} \times h_{A,a}(t) \times T_A(t) =& \sqrt{\rho_A} \times h_{A,a}(t) \times S_A - \\ & \sqrt{\rho_B} \times h_{B,a}(t) \times S_B\end{aligned} \tag{5.5}$$

As we can see in the previous equation, if cell A wants MS a to be able to decode the received signal, cell A's transmitted signal T_A must be the *intended signal (S_A) for MS a precanceling the interference signal (S_B) from cell B*. The *interference precanceled signal* for MS a must satisfy the following equation.

$$T_A(t) = S_A - \frac{\sqrt{\rho_B} \times h_{B,a}(t)}{\sqrt{\rho_A} \times h_{A,a}(t)} \times S_B \tag{5.6}$$

Suppose MS a is interfered by multiple interferers and let I be the set of interferers to MS a. The precancellation of multiple interferers can be expressed as following equation.

$$T_A(t) = S_A - \sum_{X \in I} \left(\frac{\sqrt{\rho_X} \times h_{X,a}(t)}{\sqrt{\rho_A} \times h_{A,a}(t)}\right) \times S_X \tag{5.7}$$

However, when the size of I is large, the transmitted signal in Eq. 5.7 is very likely to exceed the power constraint of the wireless interfaces. Therefore, in this chapter, we only consider precanceling *one* interference signal.

The C-RAN is responsible for precanceling the interference on MS a using Eq. 5.6. The computational burden lies on the C-RAN and the MSs do not need to handle the signal processing. Moreover, legacy devices can also benefit from the interference precancellation technique. For each cell X, the C-RAN has control over the transmission power (ρ_X) and the intended signals (S_Xs) originates from C-RAN. The C-RAN can also obtain channel coefficients ($h_{X,m}$) from previous channel coefficient estimations reported from the MSs. However, since a victim MS m does not interact with the interferers, the channel coefficients ($h_{X,m}$) between the interferers and the victim MS m are unknown. A mechanism for estimating channel coefficient between the interferers and the victim MSs is needed.

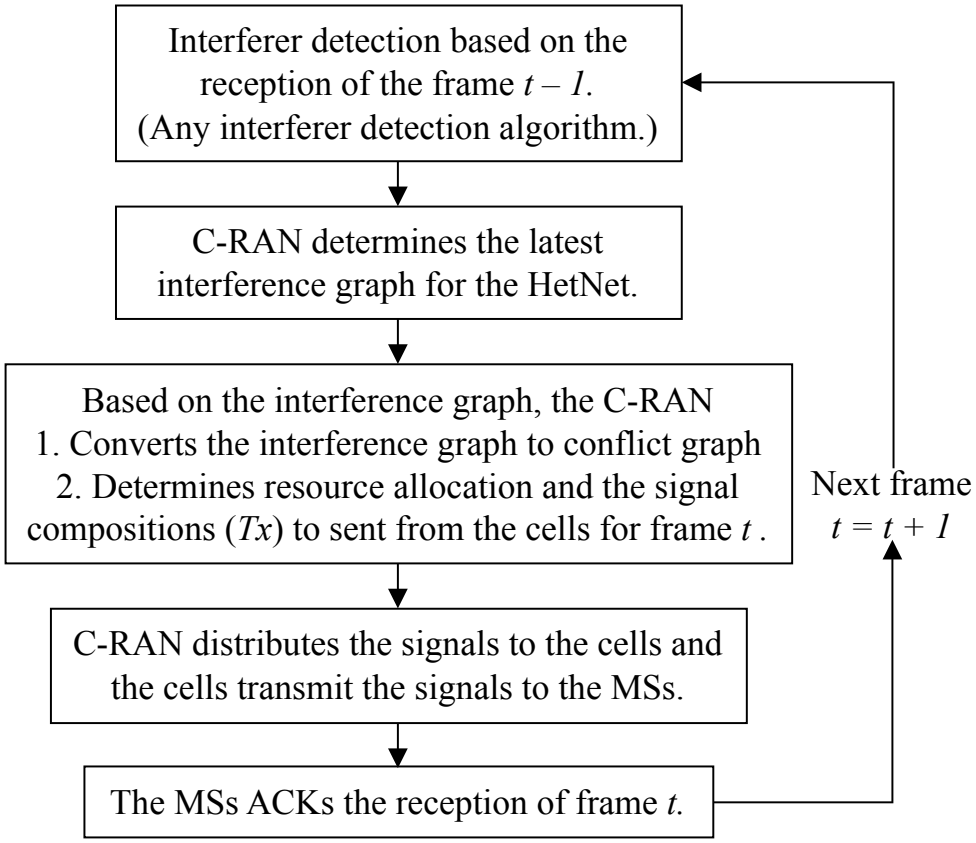


Figure 5.3: The flow of RMIP for frame t .

We have formulated the problem of interference precancellation for HetNets. We conduct experiments on GNU Radio [43] and USRP [37] to examine the behavior of interference precancellation. Estimating the channel coefficient between the interferers and the MS is challenging because the interferers and the victim MS do not communicate directly. We design a mechanism for estimating channel coefficient between the interferers and the MS. If the interference can be precanceled, same resources can be reused by interfering cellular stations and therefore the throughput is improved.

5.3 Resource Management with Interference Precancellation

In this section, we propose a resource management with interference precancellation (RMIP) algorithm for C-RAN enabled HetNets. In this chapter, the term *resource* denotes the wireless radio resources that are utilized by cells to carry data to the MSs (i.e. the white blocks in Figure 4.1). The

purpose of resource management at the C-RAN is to determine the following parameters. (a) The resource allocation for each cell that maximizes throughput, and (b) the signals (either interference precanceled signal or regular signal) for the cells to send to the MSs they serve.

The flow of RMIP algorithm is summarized as follows and is shown in Figure 5.3. First, the C-RAN attempt to determine the exact interferer to each MS and the interference relations in the network using any interferer detection algorithm as discussed in Section 5.3.1. Based on the interference relation, the C-RAN builds an interference graph as described in Section 5.3.2. The C-RAN then converts the interference graph to a conflict graph. The C-RAN uses the conflict graph to assign resources to the cells and to determine the transmitted signal composition as introduced in Section 5.3.3. Finally the transmitted signals are distributed to each cell and the cells transmit the signal to the MSs.

The granularity and performance of RMIP depends on the frequency of interference identification and management of a cellular system. The RMIP should be invoked and a new set of signal, either regular or precanceled signals, should be computed to handle the latest interference relation in a network.

5.3.1 Interference and interferer Detection

Various interference and interferer detection approaches have been proposed to mitigate either cross-tier or inter-femtocell interference. In [51, 80], the authors employ channel sensing to detect potential interference. The work [51] proposes to sense potential interference using cognitive technology and assigns isolated resources to mitigate interference. Detecting exact interferers using compressive sensing is proposed in [46]. Each cell in a HetNet sends unique training signal representing its physical layer cell identity. If an MS is within the transmission range of multiple cells, the signal received by the MS is the superposition of the training signals from these cells. Although there may be a large number of cells in a HetNet, any given MS is expected to experience interference from a relatively small subset of the cells. Therefore, based on the *spareness* of interferers, the authors in [46] propose to identify exact interferer using block sparse

signal reconstruction algorithms in compressive sensing framework.

The paper [115] proposes to take advantage of the received signal strength (RSS) to detect interference. Suppose cell C serves the MS m, MS m reports the other cells that have high RSS to MS m to cell A as potential handover candidates and potential interferer. FERMI [8] introduces two dedicated measurement zones (*free* and *occupied* zones) in the downlink of a frame. If an MS were experiencing interference, the data delivery rate in the *occupied* zone will be significantly lower than that of the *free* zone. In FERMI, a central controller infers the interference in the cell networks by comparing the data delivery rate in free and occupied zone. The authors in [73] propose to identify interference for HetNets by generating received pattern on the MSs. The patterns are generated by letting cell transmit data using a subset of subchannels. The different combinations of subchannels are called *patterns*. The cells transmit with different transmission patterns, which impact the subchannels the MSs can receive data from. The cells extract the received patterns from the ACKs from the MSs and detect the interference by examining these received patterns.

Interference detection is out of the scope of this chapter. Different interference detection algorithms have different performance and may yield different interferer detection results. In a HetNet, some of the MSs may be interference-free while the other MSs are experiencing interference. While an interferer detection algorithm may be able to detect the exact interferer to some victim MSs, it is possible that some victim MSs are indeed experiencing interference, but an interferer detection algorithm fails to detect the exact interferers. Interference precancellation technique can only be applied to victim MSs whose exact interferers have been detected. Therefore, the RMIP algorithm must jointly consider interference-free MSs, MSs with exact interferer and MSs experiencing interference with unidentified interferers.

5.3.2 Interference Graph

After the interference relations of a HetNet is determined, the C-RAN constructs an *interference graph* $G_i = (V_i, E_i)$, where the vertices V_i are the cells and the set of directed edges E denotes interference relations. The weight on the vertices denotes the number of MSs connected to the

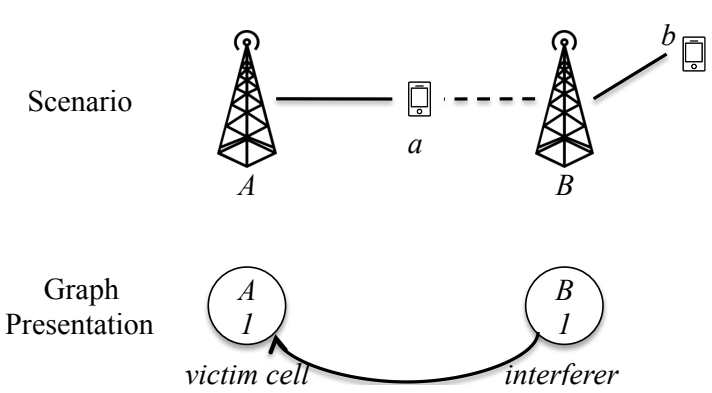


Figure 5.4: Single direction interference.

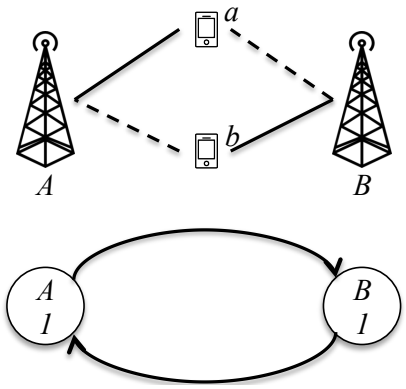


Figure 5.5: Double direction interference.

corresponding cell. For each interferer I and victim cell X pair, an edge from the interferer to the victim cell, denoted as (I, X), is added to G_i. As shown in Figure 5.4, cell B interferes with cell A and an edge (B, A) is added to G_i. If two cells are interfering with each other as shown in Figure 5.5, we decide to take a conservative approach to assign isolated resources to them.

Theoretically and ideally, it is possible to reuse the same set of resources for the scenario shown shown in Figure 5.5, if the precise channel coefficients are known. We want MS a to receive S_A and MS b to receive S_B. The received signals of a (y_a) and b (y_b) can be expressed as the following equations[1].

$$\begin{aligned} y_a &= h_{A,a} \times T_A + h_{B,a} \times T_B = S_A \\ y_b &= h_{A,b} \times T_A + h_{B,b} \times T_B = S_B \end{aligned} \tag{5.8}$$

By letting $y_a - \frac{h_{B,a}}{h_{B,b}} \times y_b$, we can derive the desired T_A as

$$T_A = \frac{h_{B,b} \times S_A - h_{B,a} \times S_B}{h_{A,a} \times h_{B,b} - h_{B,a} \times h_{A,b}} \tag{5.9}$$

T_B can also be derived using similar approach. However, the channel coefficients ($h_{X,m}$) are known to be fickle and difficult to estimate. In this scenario, the transmitted signal T_A in Eq. 5.9 and T_B depend on *four* channel coefficients, while the transmitted signal in Eq. 5.6 relies only *two* channel coefficients.

[1]For easy presentation, the parameter t, transmission power ρ and AWGN is omitted.

Considering the example shown in Figure 5.2(c), the whole frame is assigned to cell A, $T_A = S_A - S_B$ and $T_B = S_B$. In this example, one channel coefficient might affect T_B and two channel coefficients might affect T_A. The overall throughput of scenario in Figure 5.2(c) is expected to outperform that of Figure 5.2(b) because in (c), cell B gets the whole frame, while in Figure 5.2(b), cell A and cell B get half of the frame. Moreover, in Figure 5.2(c), cell A also gets a whole frame with interference precancellation and the resources MS a gets is purely additional.

In contrast, when applying interference precancellation to the example shown in Figure 5.5, both T_A (Eq. 5.9) and T_B are associated with four channel coefficients. The possibly imperfect estimations of the four channel coefficients may produce much worse effects. However, there is no possible interference precanceled signal for T_A and T_B shown in Figure 5.5, and isolated resources must be assigned to them in order to mitigate interference.

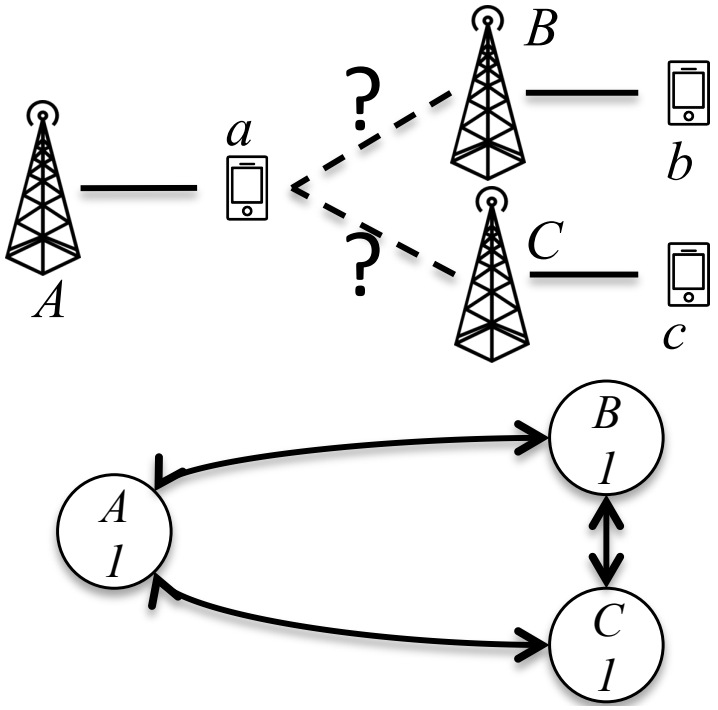


Figure 5.6: Unable to detect exact interferer.

Theorem 1. *If two cells are interfering with each other, there is no feasible interference precanlcelled signal for them.*

Proof. We prove this theorem by contradiction. It is natural to assume the signal to be delivered to MS a and MS b are different and $S_A \neq S_B$. Using Equation 5.4, we want the signal received at a to be $y_a(t) = \sqrt{\rho_A} \times h_{A,a}(t) \times S_A + n_a(t)$ and at b to be $y_b(t) = \sqrt{\rho_B} \times h_{B,b}(t) \times S_B + n_b(t)$. $y_a(t)$ must be different to $y_b(t)$, since $S_A \neq S_B$. We are determining the transmitted signals T_A and T_B, so that the received signals $y_a(t)$ and $y_b(t)$ are satisfied. Since MS a and MS b are in both cells' coverage, the signals from the cells scramble on both MS a and MS b. The actual received signals on MS a is $y_a(t) = \sqrt{\rho_A} \times h_{A,a}(t) \times T_A + \sqrt{\rho_B} \times h_{B,a}(t) \times T_B + n_a(t)$ and on MS b is $y_b(t) = \sqrt{\rho_A} \times h_{A,b}(t) \times T_A + \sqrt{\rho_B} \times h_{B,b}(t) \times T_B + n_b(t)$. However, the received signal on MS a and MS b is almost the same except for the respective channel coefficients ($h_{X,m}$) and the AWGN ($n_m(t)$), which contradicts to the premise that $S_A \neq S_B$ and $y_a(t) \neq y_b(t)$. □

Therefore, when two cells are interfering with each other, instead of risking the throughput by using signal combinations that depend on four channel coefficients, we employ the conservative resource isolation approach (Figure 5.2(b)) to mitigate interference. Similarly, when n cells are interfering with each other, isolated resources must be assigned to them. We call these cells *isolated cells*. On the other hand, as shown in Figure 5.6, it is possible that an MS is experiencing interference, but the interference detection algorithm is unable to determine the exact interferer. In this case, *all* interferers and the victim cell must be assigned with isolated resources.Therefore, for each cell X whose exact interferers is unknown, for each interferer I to cell X, two edges (X, I) and (I, X) are added to G_i. For each pair of cell I and cell J that are interfering with each other, (I, J) and (J, I) are added to G_i. Cell A, cell B and cell C are connected with each other in both directions, as shown in Figure 5.6. (For easy presentation, two directed edges are shown as one edge with arrowheads at both ends.)

5.3.3 Resource Management with Interference Precancellation

Unlike previous approaches that assign isolated resources to interfering cells, in the resource management with interference precancellation (RMIP) algorithm, the same set of RBs can be utilized by the interferer and the victim cell. The RMIP algorithm proposed in this chapter aims at

two major goals. Firstly, the C-RAN allocates and assigns RBs to each interferer and victim cell pair to maximize the utilization of RBs. The second goal is to determine the signal (either regular signal or interference precanceled signal) for each cell to transmit to its MSs.

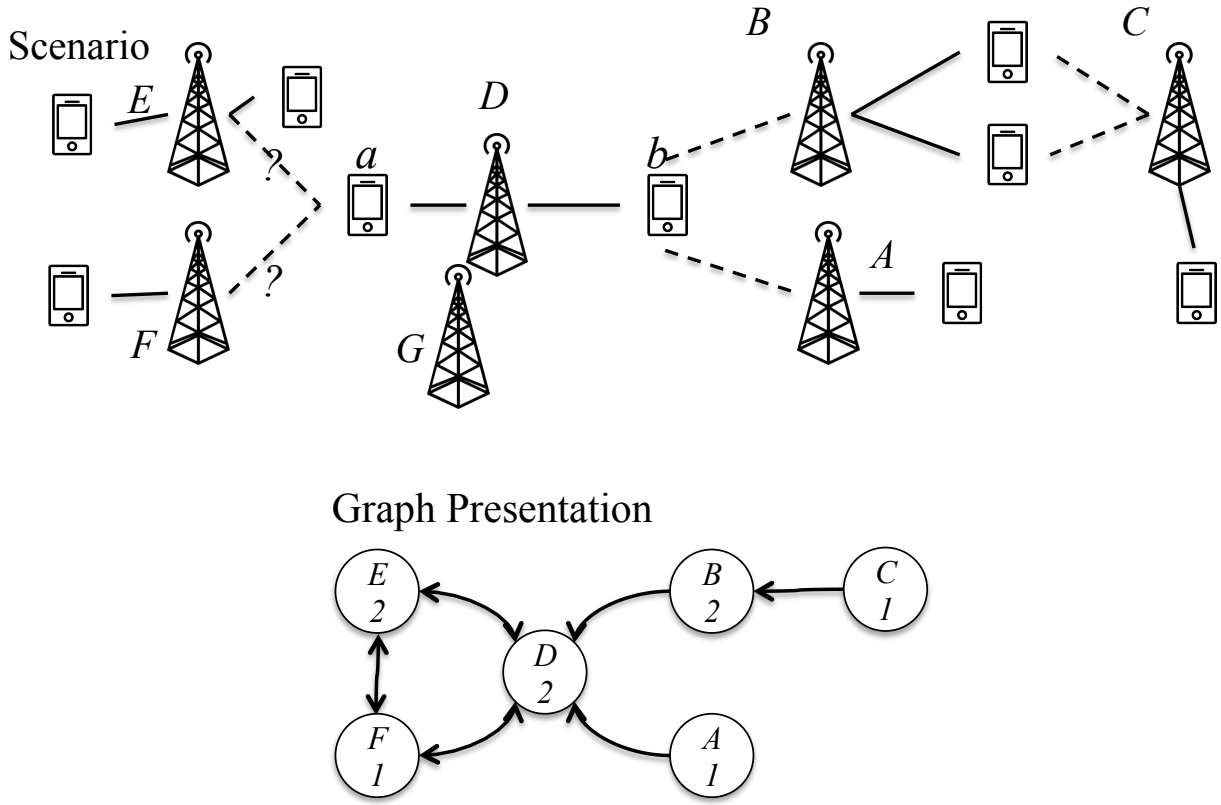


Figure 5.7: An example of 6 cells. Note that cell G does not serve any MS and is not shown in the graph.

Since we are assigning a set of resources that can be *shared by a pair of interferer and victim cell*, we define *resource sharing group* as follows.

Definition 7. *With interference precancellation technique, a resource sharing group (RSG) is group of interfering cells that is able to serve their MSs with the same set of resources. The same set of RBs is assigned to the interferer to carry regular signals and is assigned to the victim cell to carry interference precanceled signal.*

Given an interference scenario, we first build a directed interference graph $G_i = (V_i, E_i)$. Then, we convert the interference graph G_i to a conflict graph $G_c = (V_c, E_c)$, where the vertices V_c are the RSGs that will be assigned with resources. An example of G_i is shown on the right-hand side of Figure 5.7 and the corresponding G_c is shown in Figure 5.8.

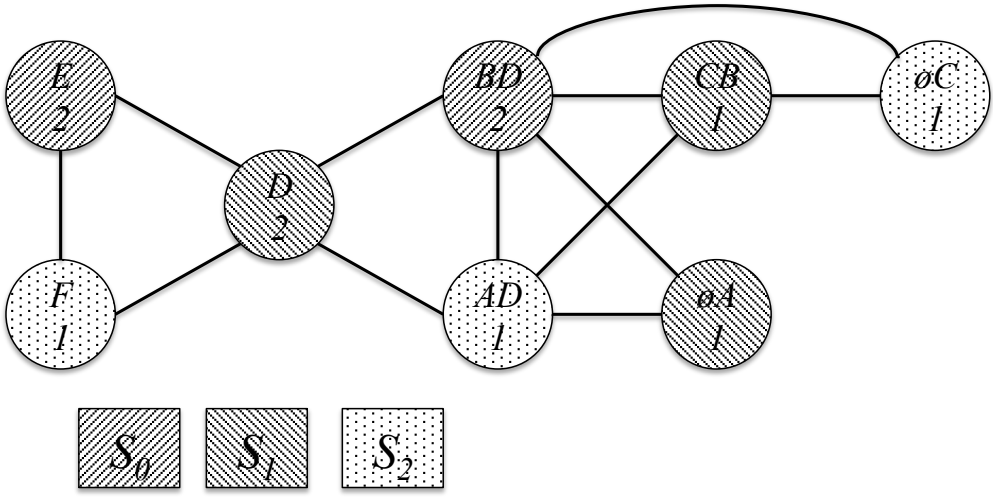


Figure 5.8: The conflict graph.

5.3.3.1 RSG Computation

There are three types of RSGs in the conflict graph.

- Type 1: Each interferer and victim cell pair form an RSG (i.e., $\{C, B\}$, $\{B, D\}$ and $\{A, D\}$ in Figure 5.8).
- Type 2: Isolated cells form an isolated RSGs (i.e., $\{D\}$, $\{E\}$, and $\{F\}$ in Figure 5.8).
- Type 3: The cells with no interferer also form RSG.

Type 1 RSGs will be assigned with RBs using interference precancellation and Type 2 RSGs will be assigned with isolated resources. Type 3 RSGs are designed to maintain resource assignment fairness because for the cells that are both interferer and victim (Type 1 RSG), they get a share of resources from being the victim cell and another share of resources from being the interferer. Take

cell B in Figure 5.7 as an example, cell B belongs to RSG $\{C, B\}$ because cell B is the victim of cell C, and cell B belongs to RSG $\{B, D\}$ because cell B is the interferer to cell D.

In contrast, cells such as cell A and cell C that are not experiencing interference only get resources from being interferer, which leads to fewer resources to cell A and cell C. Therefore, we also generate the RSGs for them $\{\emptyset, A\}$ and $\{\emptyset, C\}$ and they are Type 3 RSGs. Similarly, for cell X that is victim cell and is not interferers, we also generate the RSGs $\{X, \emptyset\}$ to allocate two shares of resources. The RSGs in Figure 5.7 are $\{\emptyset, A\}$, $\{\emptyset, C\}$, $\{C, B\}$, $\{B, D\}$, $\{A, D\}$, $\{D\}$, $\{E\}$, and $\{F\}$.

The RSG computation starts with leaf nodes (cells with no interferer) in the interference graph G_i. For each node $N \in V_i$, if N is a leaf node, the node forms a Type 3 RSG $\{\emptyset, N\}$ and N and each of N's children C form Type 1 RSGs. This can be done by visiting the child node of N. For each intermediate (non-leaf) node N, N also forms Type 1 RSG with each of its children nodes. This process can be done by performing breadth-first-search starting from each leaf nodes. If an intermediate (non-leaf) node N has been visited before, the search stops and the process continues with next leaf node. For each pair of nodes M and N, if both directed edges (M, N) and (N, M) are in E_i (the case in Figure 5.5), both N and M form a Type 2 RSG. The complexity of RSG computation is $O(|V_i| + |E_i|)$, where $|V_i|$ denotes the number of vertices and $|E_i|$ denotes the number of edges in G_i. Next step is to include the conflict relations between the RSGs.

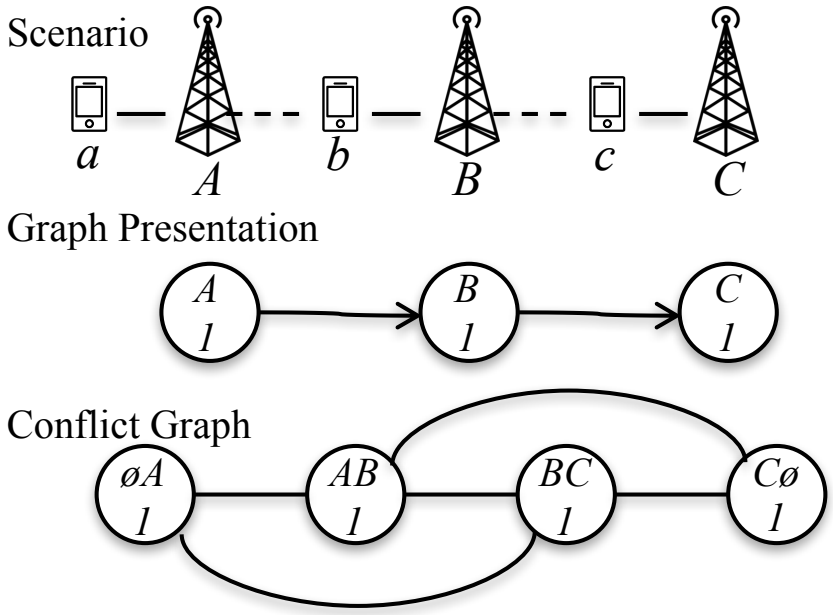


Figure 5.9: A simple chain topology.

5.3.3.2 RSG Conflict Relations

Neighboring RSGs must be assigned with isolated resources and conflict edges are added to E_c. In Figure 5.9, suppose a set of RBs (R_1) is allocated by the RSG $\{A, B\}$ (cell A interfere with cell B), cell A transmits regular signal using R_1 and cell B transmits the interference precanceled signal using R_1. The Type 3 RSG $\{\emptyset, A\}$ cannot reuse R_1 for any purpose because R_1 is already assigned to A. The Type 1 RSG $\{B, C\}$ cannot reuse R_1 because R_1 is already assigned to B to transmit interference precanceled signal. Both MS b and MS c will receive the signals in R_1. There is no possible way for C to reuse R_1. Therefore, any RSG containing *cell A, cell B and cell C* cannot reuse R_1. Conflict relations exist between RSG $\{A, B\}$, RSG $\{\emptyset, A\}$, between RSG $\{A, B\}$, RSG $\{B, C\}$ and between RSG $\{A, B\}$, RSG $\{C, \emptyset\}$.

Similarly, suppose another set of RBs R_2 is allocated to RSG $\{\emptyset, A\}$, as if cell A is the victim cell[2]. R_2 cannot be reused by B in any way because the signal (either regular signal or interference

[2] However, since this is a Type 3 RSG and there is not really an interferer, cell A still utilize R_2 with regular signal.

precanceled signal, but regular signal in this case) sent from cell A is interfering with MS b. Therefore, any RSG containing *cell A and cell B* cannot reuse R_2. Conflict relations exist between RSG $\{\emptyset, A\}$, RSG $\{A, B\}$ and between RSG $\{\emptyset, A\}$, RSG $\{B, C\}$.

In summary, for each RSG $\{I, X\}$ (interferer I and victim cell X) in V_c, we add a conflict edge between this RSG $\{I, X\}$ and any other RSG $\{I', X'\}$, if I can reach I' on a simple path of length 2 in the directed interference graph G_i. On the other hand, for cells that require isolated resources and formed Type 2 RSG ($\{D\}$, $\{E\}$, and $\{F\}$ in Figure 5.7), these RGSs form a complete subgraph, so that isolated resources can be assigned to them. The resulting conflict graph is guaranteed to be *chordal*. A chordal graph is an undirected graph that has no induced cycles of length more than 3 [42, 92, 113].

Theorem 2. *The conflict graph generated from interference graph is guaranteed to be chordal.*

Proof. For all paths $P = \{A, B, C\}$ of length 2 in G_i, the RSGs generated are $\{\emptyset, A\}$, $\{A, B\}$, $\{B, C\}$ and $\{C, *\}$ (no matter C is the leaf node or node C has a child node). For a set of resource R_i assigned to RSG $\{\emptyset, A\}$, R_i cannot be reused by any RSG containing node A and node B. Therefore, RSG $\{\emptyset, A\}$ and RSG $\{A, B\}$ have a conflict edge. RSG $\{\emptyset, A\}$ and RSG $\{B, C\}$ have a conflict edge. For a set of resource R_j assigned to RSG $\{A, B\}$, R_j cannot be reused by any RSG containing node A, node B and node C. Therefore, RSG $\{A, B\}$ and RSG $\{B, C\}$ have a conflict edge. RSG $\{B, C\}$ and RSG $\{C, \emptyset\}$ have a conflict edge. These edges can must form cycles of length 3. Suppose A has a parent node P, the RSG $\{\emptyset, A\}$ is replaced by $\{P, A\}$ and these RSGs can only form cycles of length 3 too. On the other hand, the subgraph formed by the Type 2 RSGs is complete graph, which is also chordal. Therefore, the resulting conflict graph is guaranteed to be chordal. □

For each RSG $\{I, X\}$, the algorithm examines all paths of length 2 in the interference graph. This can be done by performing by traversing all paths of length two starting from each node. The number of nodes of X is $|V_i|$. Therefore, the complexity of adding RSG conflict relations is also $O(|V_i|)$.

5.3.3.3 Resource Allocation

Our resource assignment algorithm is a two-step procedure:

- Step 1: Assign each RSG in G_c to independent set that neighboring RSGs do no belong to the same independent set, and
- Step 2: Allocate and assign groups of RBs to the RSGs based on the independent set assigned.

A proper coloring C for a weighted graph $G = (V, E)$ is $\{S_1, S_2, ..., S_k\}$, where S_i is a disjoint independent subset of V and $\cup_{j=1}^{k}(S_j) = V$. An independent set S_i in G is a set of pairwise nonadjacent vertices. The weight of a vertex set S_i is denoted as $\alpha(S_i)$ and is defined as the maximum weight vertex in that S_i. The weight of the proper coloring C is denoted as $W(C) = \sum_{i=0}^{k} \alpha(S_i)$. The goal is to minimize $W(C)$.

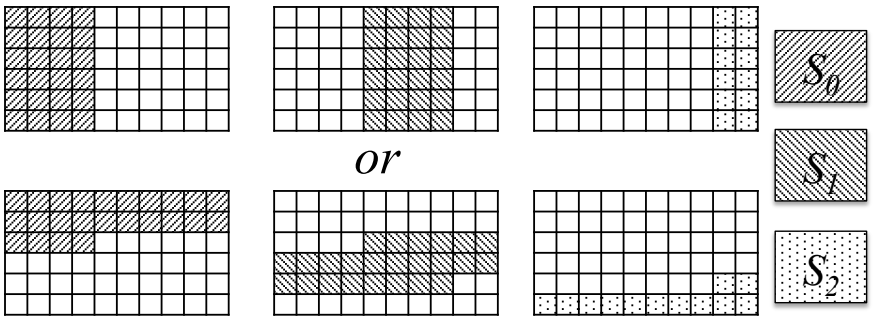


Figure 5.10: Two possible of RB assignments for Figure 5.8.

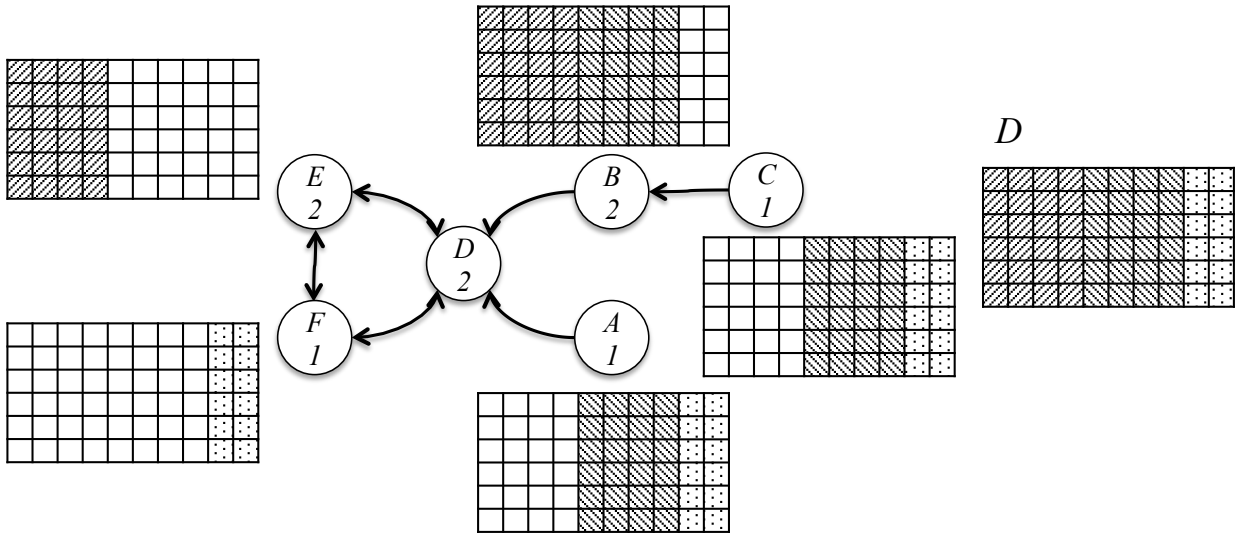


Figure 5.11: The resource assignment with interference precancellation.

Although graph coloring problem for general graphs is NP-complete, the *optimal minimum coloring* of chordal graphs can be solved in polynomial time [42, 92]. Therefore, we adopt the simple $O(|E_c|)$ greedy algorithm described in [42]. In worst case, $|E_c|$ is $|V_c|^2$ when each pair of nodes (RSGs) in the graph has an edge (conflict relation) between them. The greedy algorithm produces *optimal minimum coloring* if the algorithm assigns RSGs to independent sets in the simplicial elimination ordering of the RSGs. The simplicial elimination ordering of the RSGs can be obtained using the MCS procedure proposed in [113]. When assigning an RSG v, we assign v to the independent set S_i such that the increment of $W(C)$ is minimized. We check if S_i can include v in the order of descending $\alpha(S_i)$. If none of these independent sets is able to include v, a new independent set $\{v\}$ is created in C to include v. Because the conflict graph is chordal and we assign the RSGs in a simplicial elimination ordering of RSGs, $W(C)$ is guaranteed to be optimal. After the weight of the independent sets and the total weight $W(C)$ is determined, for each independent set S_i, the set gets the $\frac{\alpha(S_i)}{W(C)}$ of the total resources. Two possible RBs assignments to each independent set for the conflict graph in Figure 5.8 is shown in Figure 5.10.

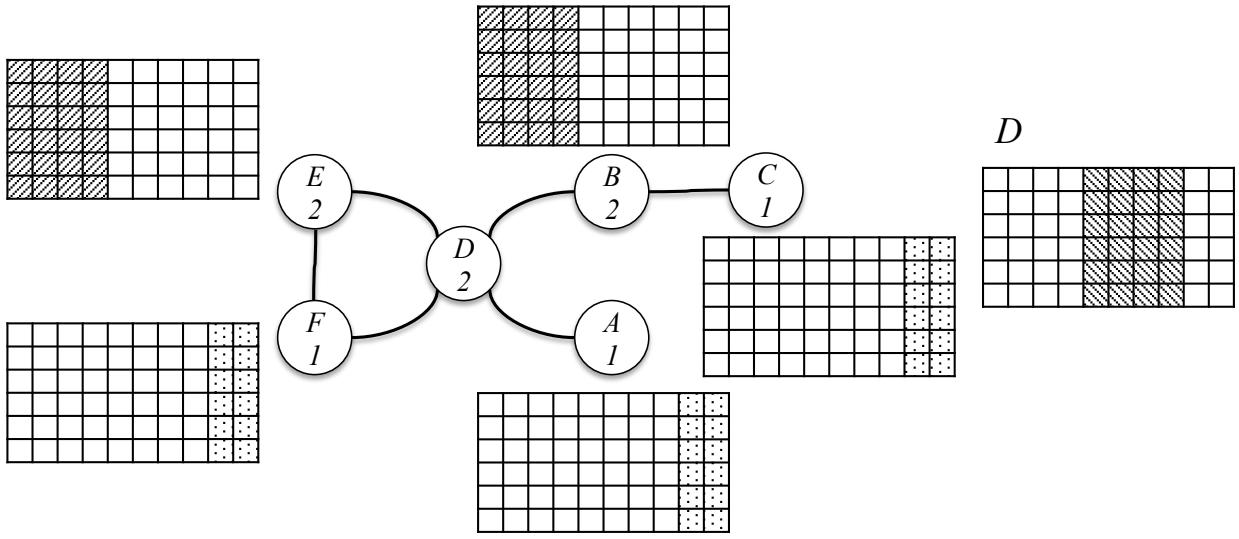


Figure 5.12: The traditional isolated resource assignment.

The next step is to assign the RBs to each cell. For each cell X in RSG that is assigned to the independent set S_i, the cell X is assigned with the RBs for S_i. Take cell B in Figure 5.7 as an example, cell B belongs to RSG $\{B, D\}$ and RSG $\{C, B\}$, cell B get the resources from S_0 and S_1. A possible resource assignment results are shown in Figure 5.11.

5.3.3.4 Determining the Signals

After the RBs have been assigned, the final step is to determine which cell gets regular signal and which cell gets the interference precanceled signal. For each RSG $\{I, X\}$, the interferer transmits the regular signal and the victim cell transmits the interference precanceled signal, so that the interference signal scrambles with the interference precanceled signal at the victim MS and becomes the intended signal for the victim MS. Take cell B in Figure 5.11 as example, the RBs B gets from the independent set S_0 is assigned to the RSG $\{B, D\}$, in which B is the interferer. In those RBs, cell B transmits S_B and cell D transmits $S_D - S_B$. On the other hand, the RBs cell B gets from the independent set S_1 are assigned to the RSG $\{C, B\}$, in which cell B is the victim cell. In those RBs, cell B transmits $S_B - S_C$ and cell C transmits S_C.

It is noteworthy that D gets all resources in a frame. The RBs D gets from S_1 are assigned to the RSG $\{D\}$, in which D is the isolated cell. In those RBs, cell D transmits S_D. The RBs D gets from S_0 are assigned to the RSG $\{B, D\}$, in which D is the victim cell. In those RBs, cell D transmits $S_D - S_B$. One may wonder if the cell D's transmission using RBs in S_0 will cause interference to cell E. Note that cell D is not interfering with any MSs connected to cell E. It is the other way around, cell E interferes with cell D's MS, so the RBs in S_0 is not interfering with cell E.

In contrast, as shown in Figure 5.7, E is interfering with cell D's MS a, which means cell D's RBs from S_0 will be interfered by E. Therefore, the RBs cell D obtained from RSG $\{B, D\}$ and RSG $\{A, D\}$ is used to serve MS b and the RBs cell D obtained from RSG $\{D\}$ are the isolated resources for serving MS a. Cell D tells MS a and MS b to receive from which RBs using the control frame indicator (Figure 4.1) in the LTE frame. To MS a, it does not care the RBs collided by cell E and cell F and it only cares about the RBs in S_0. The complexity for determining the signals is $O(|V_c|)$ because the procedure processes each RSG for exactly once.

5.3.3.5 Improvement over Isolated Resource Allocation

An optimal isolated resource allocation using traditional approach is shown in Figure 5.12 and RMIP's resource allocation result is shown in Figure 5.11. The RBs allocated to each cell are subsets to the RBs allocated using our approach. The extra RBs for the cells to send data using interference precancellation technique are entirely improvement. Even if the RBs sending interference precanceled signal has lower SINR due to signal scrambling and imperfect channel coefficient estimation. The interferer still uses the RBs to send regular signal (not precanceling any signal) in the same way as traditional approach. Therefore, the performance is guaranteed to improve compared with traditional resource isolation approach.

5.3.3.6 RMIP Complexity Analysis

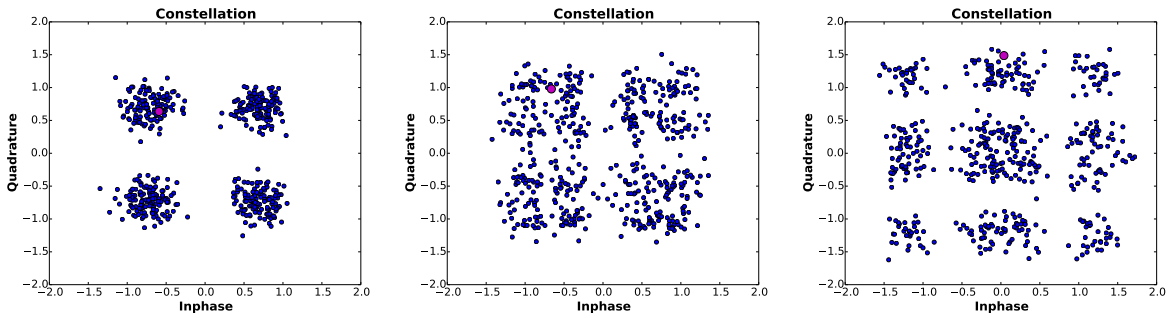
The RMIP algorithm first converts interference graph G_i to RSG conflict graph G_c. The complexity of RSG generation is $O(|V_i| + |E_i|)$. The upper bound of the number of edges $|E_i|$ is $O(|V_i|^2)$ because

in worst case, each pair of cell M and cell N in V_c has an edge between the cells. After the RGS generation is done, we have the conflict graph G_c containing RSGs with no edges. The upper bound of the number of vertices V_c in G_c is $O(|V_i|)$ only. It because for the worst case that every pair of cell M and cell N in G_i has an edge (interference relation) between them, the cells are interfering with each other. According to Theorem 1, the cells must form Type 2 RSGs for resource isolation, instead of Type 1 RSGs. Type 1 RSGs are only formed when the subgraph is a chain with no loop. The complexity of adding RSG conflict relations is $O(|V_i|)$ because for a node v in the interference graph G_v, we search for all offspring that can be reached from v in a path of length 2.

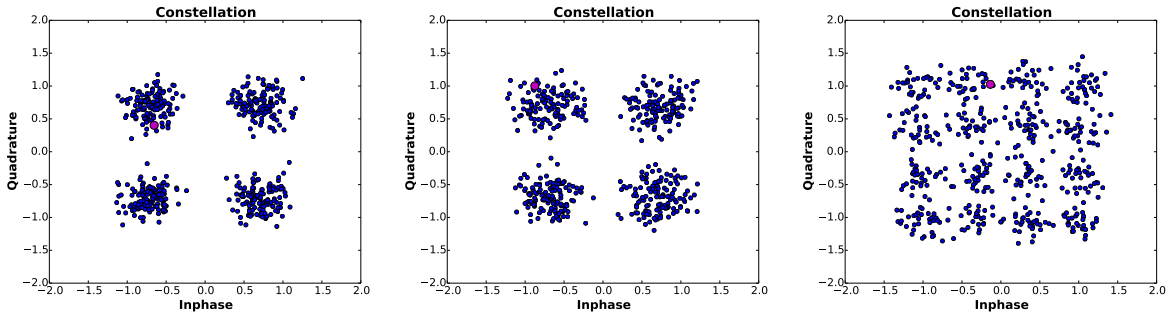
After the RSG conflict graph is built, RMIP performs resource allocation using the greedy algorithm proposed described in [42]. The complexity of the greedy resource allocation is $O(E_c)$. In worst case, $|E_c|$ is $|V_c|^2$ when each pair of nodes (RSGs) in the graph has an edge (conflict relation) between them. After the resource blocks are assigned to each RSG, RMIP determines the signal composition (regular or interference precanceled signal) for each cell by checking $|V_c|$ RSGs one by one. Therefore, the complexity of determining the signal is also $O(|V_c|)$. Finally, the overall complexity of the RMIP algorithm is $O(|V_i| + |E_i|) + O(|V_i|) + O(|E_c|) = O(|V_i| + |V_i|^2) + O(|V_i|) + O(|V_i|^2) = O(|V_i|^2)$.

5.4 Evaluation

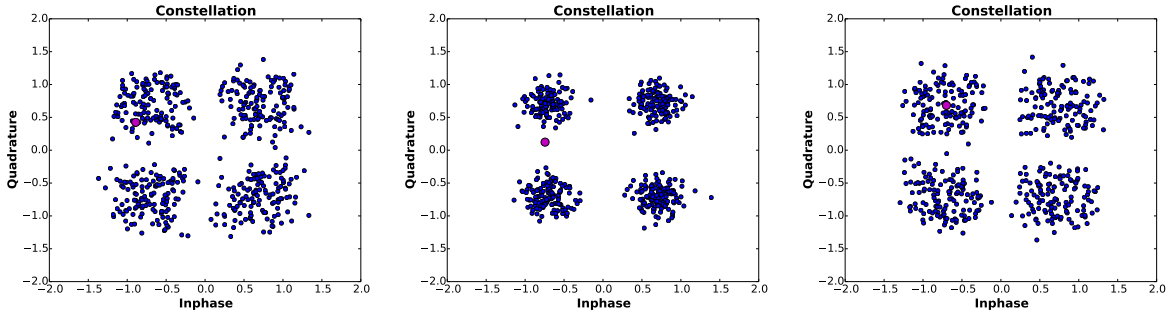
In this section, we validate the interference precancellation using GNU Radio [43] and USRP [37] and evaluate the performance of RMIP algorithm on large scale HetNets through simulation.



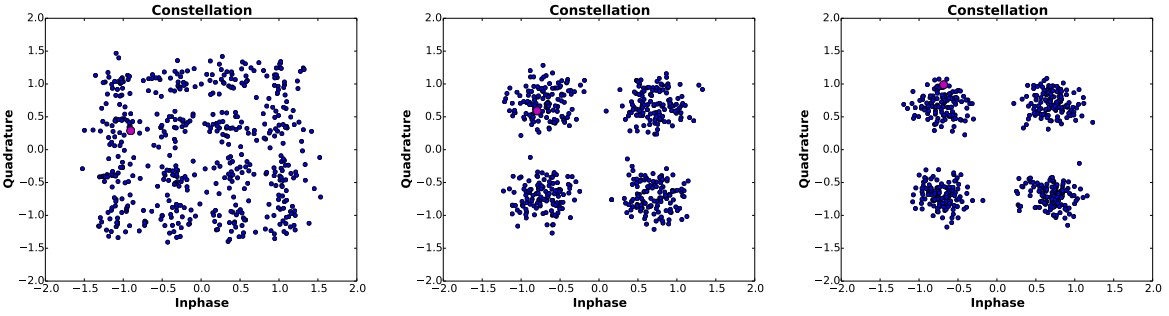
$k = 0.2, g = 9$ $k = 0.2, g = 12$ $k = 0.2, g = 15$



$k = 0.4, g = 9$ $k = 0.4, g = 12$ $k = 0.4, g = 15$



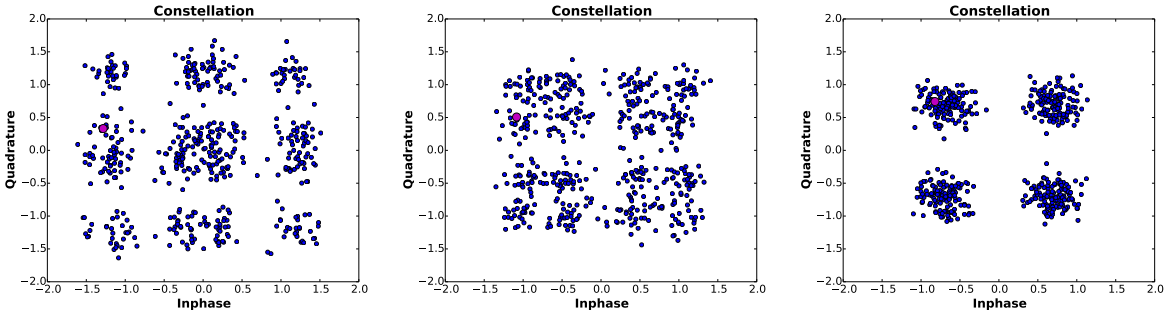
$k = 0.6, g = 9$ $k = 0.6, g = 12$ $k = 0.6, g = 15$



$k = 0.8, g = 9$ $k = 0.8, g = 12$ $k = 0.8, g = 15$

Figure 5.14: The scrambled signal constellation by alternating k and g.

Figure 5.14: (cont'd)



$k = 1.0, g = 9$ $k = 1.0, g = 12$ $k = 1.0, g = 15$

5.4.1 Interference Precancellation

We conduct experiments to examine the behavior of interference precancellation. The experiment is similar to the scenario depicted in Figure. 5.2(c). We synchronize two USRP N210s to simulate the interferer B and the victim cell A. Another USRP N210 is used as victim MS a to receive the scrambled signal. A, a and B are deployed in a line and the distance between them is around 1 meter. The center frequency is 4.0 GHz and the bandwidth is 10 MHz, which is divided into 64 subcarriers. The receiving gain on a is 15 dB. We measure the minimal gain for victim MS a to receive data successfully from the interferer B and the victim cell A for 10 times. The minimal gain of A for m to receive from A is 13 dB. Therefore, we let the transmission gain of A be 15 dB. The signals sent from cell B and interference precanceled signals sent from cell A are precomputed. These precomputed signals are converted to constellation symbols and stored on the laptops that drive the USRPs to minimize latency. The symbols are sent repeatedly and we observe the signal received by a.

From Eq. 5.6, the interference precanceled signal must take channel coefficient between cell B and MS a into account. The interference precanceled signal is rewritten as $T_A(t) = S_A - k \times S_B$, where k is the *precancellation ratio* and is $\frac{\sqrt{\rho_B} \times h_{B,a}(t)}{\sqrt{\rho_A} \times h_{A,a}(t)}$. Since it is impossible to control channel coefficient, we generate a set of interference precanceled signal with different $k = \{0.2, 0.4, 0.6, 0.8, 1.0\}$. We control the strength of interference by controlling the transmission gain of cell B, $g = \{9, 12, 15\}$.

Constellation: We let both the victim cell A and the interferer B transmit using QPSK. The bits 00, 10, 11 and 01 are mapped to $-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}j$, $\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}j$, $\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}j$ and $-\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}j$. If $k = 1$, the symbol $-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}j$ precanceling $\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}j$ becomes $-\sqrt{2} - \sqrt{2}j$. The scrambled signal received at the victim MS a is shown in Figure 5.14. When $k = 1.0$ and $g = 9$, the interference is *over-precanceled*, and the interference is not strong enough to make the scrambled signal to become the intended signal for a. On the other hand, when $k = 0.2$ and $g = 15$, the interference is *under-precanceled*. For other cases, if the precancellation ratio (k) and the interference strength (g) are not too different, the scrambled signals are usually decodable. Therefore, determining the value of k, which is composed of $h_{A,a}$ and $h_{B,a}$, is critical to the performance of interference precancellation.

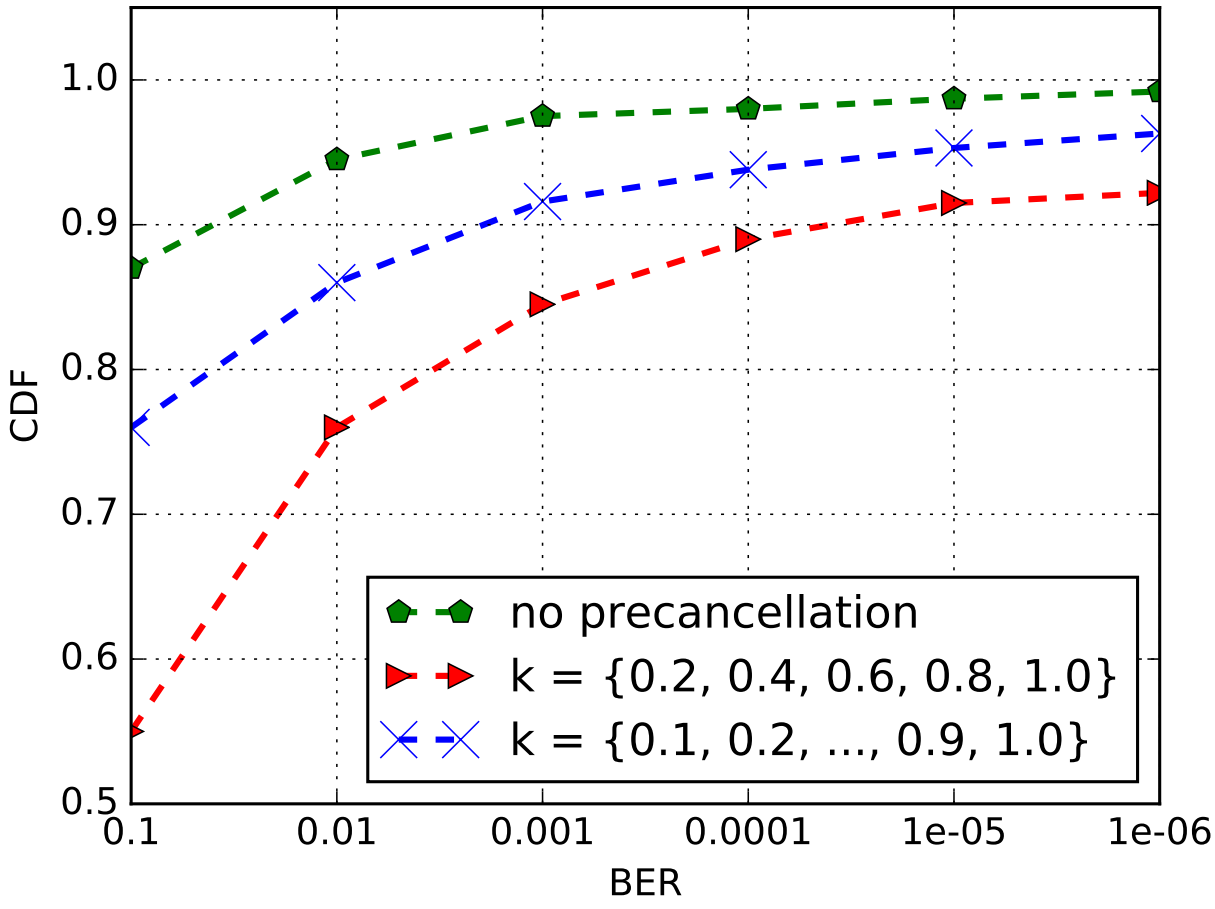


Figure 5.15: The CDF of the bit error rate of different k.

Channel Coefficient Estimation: We also implemented a prototype that adapts the level of precancellation considering the channel coefficients. We insert two pilot symbols at subcarrier 16 and 48 for estimating the channel coefficient. A symbol known to the victim MS a is sent over the pilot subcarriers. When receiving the frame, a responds with the channel coefficient a estimated in the ACKs. Currently, the N210 is able to send a frame every 100 ms and the receiver responds with an ACK after 10 ms for the switching between transmission and reception mode of the SBX daughterboard.

The victim cell A starts with sending regular signals S_A. The victim MS a reports the estimated channel condition ($h_{A,a}$) to A. After 100 frames (10 seconds), the interferer B starts to send the interfering signal (S_B). The channel coefficient reported by the MS to A becomes $h_{A,a} + h_{B,a}$. B stops after 100 frames. During the time when B is sending, A computes $h_{B,a}$ from the ACK and computes the latest k. In this prototype, the transmission gains of both A and B are 15 dB. Due to the USRP N210 latency limitation, we have to precompute the interference precanceled signal and store them in A's driver laptop. When A gets the latest k, A finds the interference precanceled signal that is closest to k and sends it to a. The bit error rates of no interference precancellation and two sets of ks are shown in Figure. 5.15. As shown in Figure. 5.15, higher resolution $k = \{0.1, 0.2, ...1.0\}$ achieves better bit error rate. Although the interference precancellation's bit error rate is still higher than that of no precancellation, it is noteworthy that the interference precancellation RBs are pure additional RBs as shown in Figure. 5.2(c).

The major causes of the latency are the architecture of USRP N-series and the limited computational capability of the driver laptops. The driver laptop connects to the USRP N210 via Ethernet, while in C-RAN, there should be no such latency. The switching between transmission and reception modes of the SBX daughterboards also have a latency of around 10 ms. These two latencies caused by the platform do not exist on real cellular stations. The computation of the interference precanceled signal might introduce some latency, but it should be less than the computation latency on the laptop because the C-RAN is expected to have advanced computational power.

5.4.2 Resource Management with Interference Precancellation

Our simulation setup and parameters follow the simulation guide proposed in [93] and the simulation steps are summarized as follows. We create an area of 500 x 500 m^2 and deploy a HetNet $N = \{V, M\}$, where V and M are cells and MSs at random locations. Each cell is serving a random number of MS (between 1 to 4). Currently, the cells have the same coverage of 100 m. The MSs are also deployed at random locations and are static. Each MS m is connected with exactly one cell, denoted as $c(m)$. Secondly, according to the MS locations and the path loss function, the simulator computes the RSS between every cell-MS pair. We define an interfering threshold Γ, such that if the RSS of a cell v ($v \neq c(m)$) to an MS m is above Γ, cell v is regarded as a *real* interferer to m. The set of real interferers to m is denoted as *interferer(m)*.

The threshold Γ here is only used to "create real interferers." In real-world, we do not need to "create" interferers. Any cell that is constantly causing interference to MSs is considered a true interferer. We adopt WINNER II NLOS model as the indoor path loss function between a femtocell and an MS [87, 93].

$$\begin{aligned} PL =& 46.4 + 20 \times log_{10}(R) + 20 \times log_{10}(\frac{f}{5}) + \\ & 3 \times n_w + F_L \end{aligned} \tag{5.10}$$

where R is the distance between the cell the and MS in meters, f is the center frequency in MHz, which is 900 MHz and n_w is the number of walls between the cell and the MS. We assume that there exists a light wall every 5 meters and the penetration loss of the light wall is 3 dB. F_L is defined as $17 + 4 \times (n_f - 1)$, where n_f is the number of floors between the cell and the Ms. In our simulations, each cell is randomly assigned to a floor between ground and the third floor. We employ the interference detection algorithm proposed in [73]. We perform RMIP algorithm using the interference relations detected. We compare our RMIP algorithm with FERMI [8] and the weighted vertex coloring (WVC) in [73].

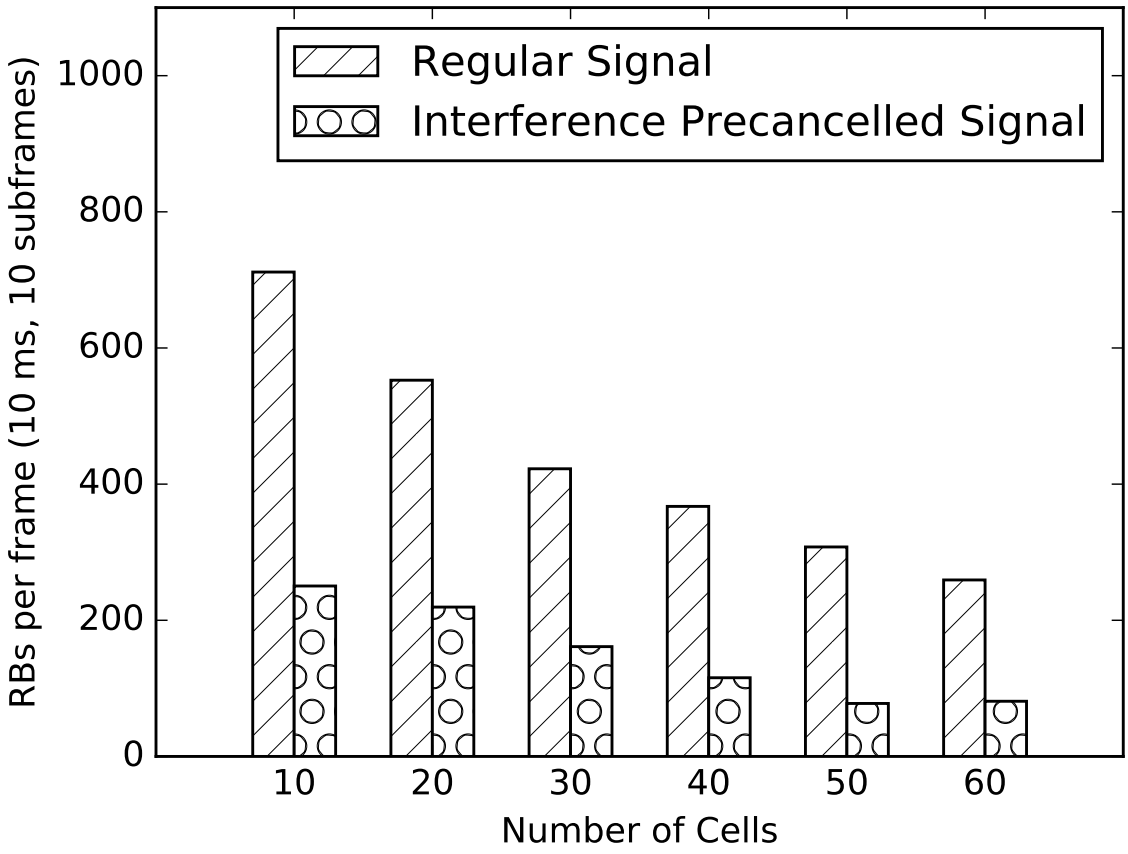


Figure 5.16: The number of RBs that carry interference precanceled signal and the number of RBs that carry regular signal.

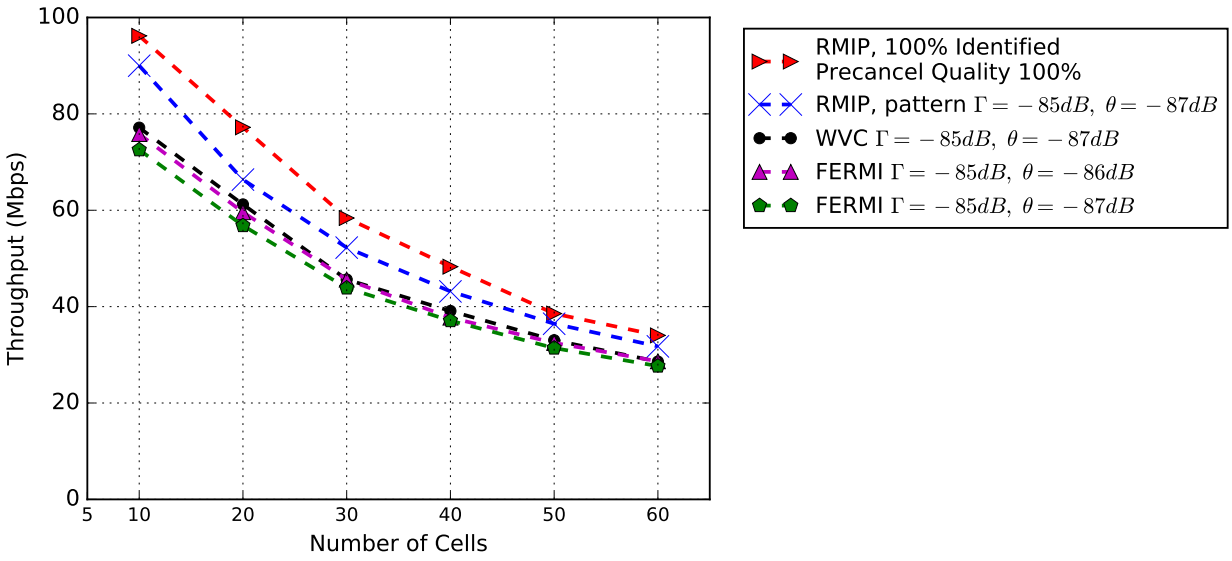


Figure 5.17: The average throughput at each MS.

5.4.3 Average Throughput

In our simulation, we consider an LTE channel of bandwidth 20 MHz and 10% of the bandwidth is used for guard bands. The channel is further divided into 1200 15 kHz subcarriers. An RB consists 12 subcarriers in the frequency domain and two time slots (2×0.5 ms) in the time domain. Therefore, a subframe in a channel of 20 MHz can be divided into 100 RBs. A frame consists of 10 subframes and therefore, a frame has 1000 RBs. The comparison of number of RBs that carry interference precanceled signal and the number of RBs that carry regular signal is shown in Figure 5.16.

In order to convert the number of RBs to actual throughput, we must obtain the SNR of each MS and. Each downlink RB has $12 \times 2 \times 7$ symbols per ms, which means a 20 MHz channel has $16.8M$ samples per second. If the modulation is 64-QAM (6 bits per symbol), the throughput is roughly $16.8M \times 6 = 100.8$ Mbps. For 4x4 MIMO systems, the throughput is 403.2 Mbps (100.8×4). However, in this chapter, in order to emphasize that RMIP algorithm can also apply to legacy devices, we only consider SISO systems.

For each scenario, the simulation flow is summarized as follows. First, for an MS m and the cell it connects to $c(m)$, we compute the signal strength using Equation 5.10. All other cells are considered interferer to m. If a cell n, $n \neq c(m)$, has a signal strength higher than a threshold to m, we consider n to be m's interferer. We use interferer detection algorithms [73] to detect interferers to each MS and build interference relation for the scenario. Based on the interference relation, we perform the resource allocation algorithms using RMIP, FERMI [8] or WVC[73] to obtain different RB allocations.

Using the signal strength from the cell and the interference signal strength, we compute the SNR of the RBs on each MS m and map the SNR to channel quality indicator (CQI) [86]. After getting the CQI to MS m, we convert the number of RBs to effective throughputs in Mbps using the MCS conversion table as shown in Table 7.2.3-1 in 3GPP LTE specification [2].

The average throughput at each MS is shown in Figure 5.17. The red curve denotes that all interferers are detected, and the RBs assigned with interference precancellation performs 100%

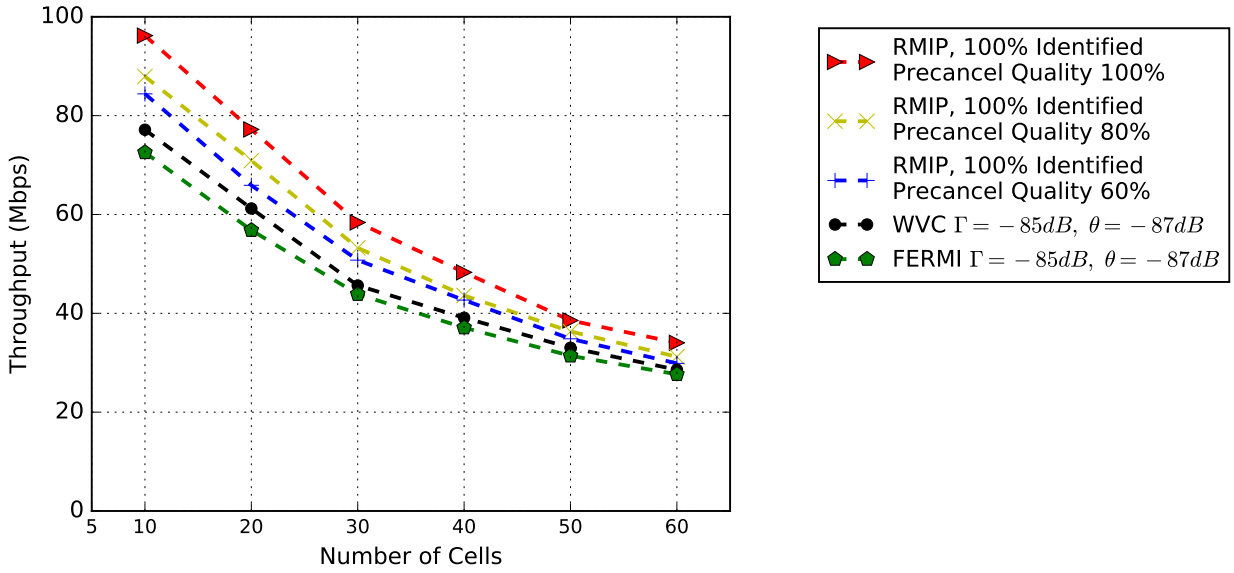


Figure 5.18: The average throughput at each MS, alternating the precancel quality.

throughput as regular RBs. When the density of cells is low, the average throughput allocated is significantly improved. It is because when there are few cells in the same area, the interference relations are simpler than that of more cells. There is more interferer and victim cells detected, and they can share the same set of RBs using interference precancellation technique. However, when the number of cells increases in the same area, many of the cells start to interfere with each other. These cells form Type 2 RSGs, and they require isolated RBs as well. On the other hand, the blue curve uses the interferer detection approach proposed in [73]. Around 95% of the exact interferers are identified, and all cells mitigate interference using resource isolation. With RMIP algorithm, the MSs can achieve 19.03% higher throughput than WVC if all interferers are identified and can achieve 11.05% higher throughput compared with WVC using the same interference detection algorithm.

However, in any communication systems, the estimation of channel condition between transmitter and the receiver may be imperfect and so does the channel condition estimation between the interferer and the victim MSs. The RBs assigned with interference precancellation performs may not be able to achieve the exact same performance as regular signals. In order to gain some insight into the impact of imperfect interference precancellation, we define a metric called *precan-*

cel quality. The calculated SNR of RBs assigned to interference precancellation are multiplied by the precancel quality before converted the CQI and the throughput is reduced. The results are shown in Figure 5.18. The blue and green curves are the same as the ones in Figure 5.17. The red, yellow and blue curve show the average throughput with the precancel quality 100%, 80% and 60%, respectively. The throughput still outperforms resource isolation techniques, even if the precancel quality is as low as 60%.

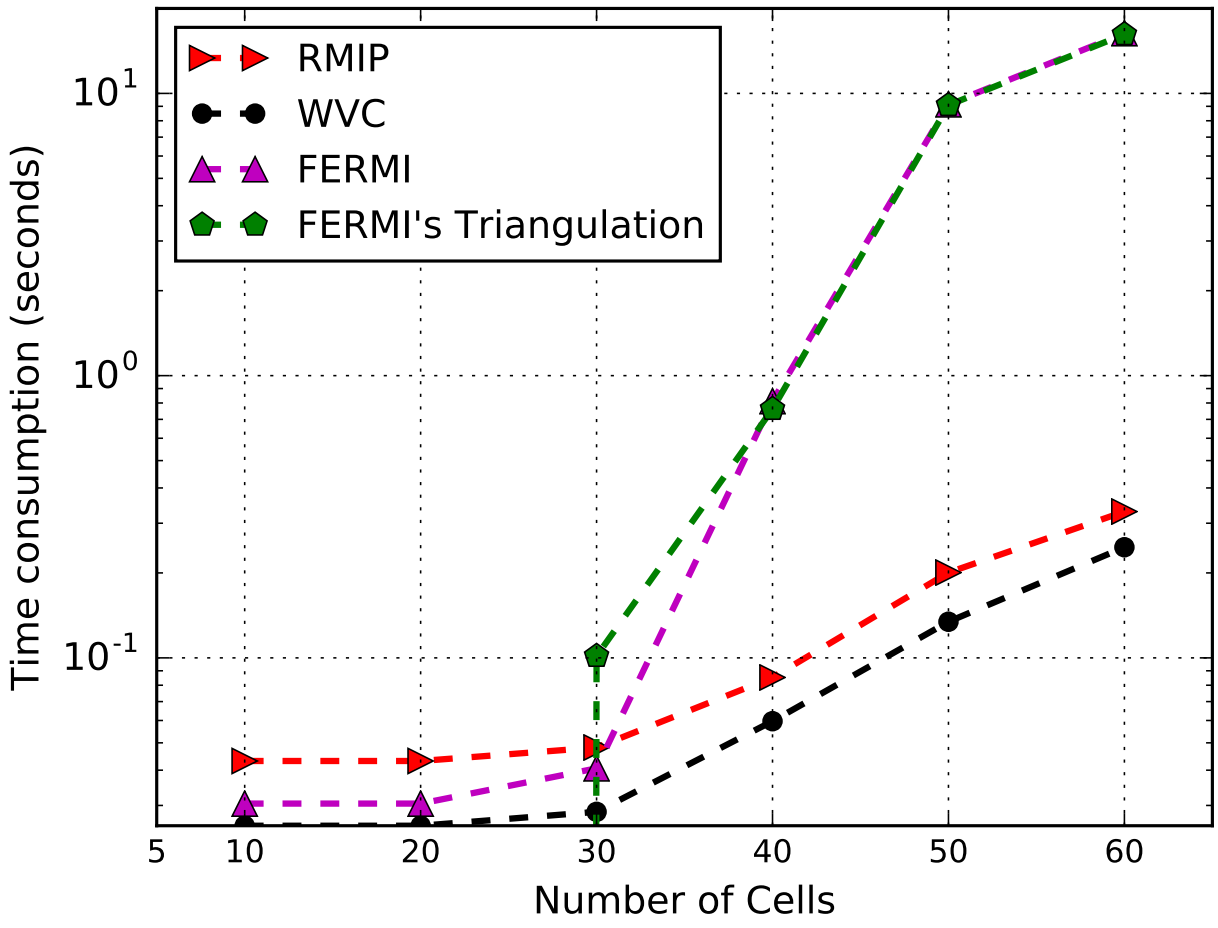


Figure 5.19: The time consumption of resource block assignment.

5.4.4 Time Consumption for Resource Allocation

The comparison of time consumption for assigning RBs is shown in Figure 5.19. Note that the conversion from RB to throughput is not included in this figure. The WVC consumes the least

time due to its simplicity. WVC builds conflict graph where the vertices are the cells and assigns resource to the cells directly. The WVC is a heuristic algorithm and cannot guarantee optimal resource allocation. On the other hand, RMIP algorithm requires the conversion from interference graph (V_i) to conflict graph (V_c) to perform resource allocation. Although our RMIP algorithm consumes less time when allocating resources to RSGs, the number of RSGs in G_c is $O(|V_i|)$, where $|V_i|$ is the number of cells. FERMI also allocates resources based on graphs where the vertices are the cells. Although FERMI is able to achieve optimal resource allocation when the graph is chordal when the graph is non-chordal, FERMI requires a triangulation procedure to chordalize the graph, and the triangulation procedure is extremely time-consuming. When the HetNet is sparse (10 and 20 cells in the area), the graph is chordal and therefore there is no triangulation time. When the HetNet is denser, the time consumption of FERMI is dominated by the triangulation process. Note that the y-axis of Figure 5.19 is logarithmic.

5.5 Summary

In this chapter, we propose an interference precancellation technique by taking advantage of the characteristic that the data for the cells and the MSs are all originated from the core cellular network. If the exact interferer to a victim cell is identified, when the interferer is transmitting regular signal using the RBs allocated to the interferer, the victim cell attempts to use the same RBs to serve its MSs with *interference precanceled signal*. The wireless capacity of the network can be improved. However, some MSs still require isolated resources because their exact interferer might not be identified and some MSs do not require interference management because they are not experiencing interference. Therefore, we propose a resource management with interference precancellation (RMIP) algorithm that jointly considers MSs that are experiencing a different level of interference. Simulation results show that even with imperfect interference detection algorithm in densely deployed HetNet, the MSs achieves at least 11.05% higher throughput than pure resource isolation approaches. If all exact interferers are identified, RMIP algorithm can achieve at least 19.03% higher throughput than previous studies.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter, we conclude our work in this dissertation and discuss some future research directions.

6.1 Conclusion

This dissertation addresses several challenging problems in heterogeneous wireless networks, including CRNs and HetNets.

6.1.1 Cognitive Radio Networks

For CRNs, a non-contiguous control channel (NCCC) establishment method that takes advantage of OFDM pulses is proposed. Most existing approaches for finding control channels are CH-based. The time consumption of CH-based approaches grows drastically along with the number of channels in the network, which is inapplicable to NC-OFDM based interfaces because the number of subcarriers in the spectrum is ten times or even a hundred times to the number of channels. In NCCC, the SUs attempt to probe their neighbors about their available channels and construct control channels accordingly. We demonstrate that the NCCC can form NCCCs in at least 19.60% less time compared with traditional CH-based approaches. Also, a common requirement of previous studies is that a channel across the network must be available to all SUs to serve as the control channel. However, such a channel may not exist in some CRNs. The use of NC-OFDM-based control interfaces for control purpose enables NCCC that aggregates non-contiguous spectrum fragments, which improves the control channel establishment rate by at least 10% even when the control interface can only access the spectrum bandwidth that is equal to one channel.

The second research problem deals with the potential interruptions to a routing path in the secondary networks when the PUs become active again. The k-protected routing protocol that builds routing paths that will not be interrupted even after k PU returns in CRNs is proposed. In the literature, the reactive approaches that seek alternative spectrum when PU returns suffer from

longer delay and possible interruption if an alternative cannot be found. A k-protected routing algorithm is discussed in this dissertation to build routing paths that can sustain from at most k PUs returns without interrupting the ongoing communication. We demonstrated that k-protected routes could sustain k PU returns active without being interrupted through simulation. The average interruption rate of a 1-protected route is never interrupted when one PU appear and is lower than 18% when two PUs return. Moreover, because of the preallocated backup resources, when any PU becomes active again, the backup activation time for the k-protected routes built by the centralized algorithm is about 20% to that of previous studies. The backup activation time for the k-protected routes built by the distributed algorithm is about 80% to that of previous studies.

6.1.2 Heterogeneous Cellular Networks

A significant challenge of HetNets is that some smallcells suffer from interference from the interferer that is hard to identify. Existing interference identification approaches often regard more cells as interferers than necessary. The false positive interferers significantly reduce the achievable throughput at the MSs. An inter-cell interference identification using received patterns on the MSs is proposed in this dissertation. Our interference identification method successfully identifies all real interferers and at least 94% of the non-interfering cells. Moreover, the method achieves a stable identification in less than fifteen frames, which means our algorithm can handle the dynamic user mobility of cellular networks. With improved interference identification, we propose a weighted vertex-coloring (WVC) based resource assignment algorithm that achieves better fairness with much lower time consumption for high-density networks than existing approaches.

Traditionally, the interference mitigation approaches in the literature are mainly trying to avoid the interference, such as resource isolation that leads to significantly fewer resources, or power control that sacrifices signal quality and coverage. If the exact interferer can be identified through any interferer identification algorithm, we propose to mitigate the interference by *precanceling* the interference signals. The same set of resources can be shared by cells, and the wireless capacity of the HetNet can be improved. However, some MSs still require isolated resources because their exact

interferer might not be identified and some MSs do not require interference management because they are not experiencing interference. We propose a resource management with interference precancellation (RMIP) algorithm that jointly considers MSs that are experiencing a different level of interference. The results show that even with imperfect interference detection algorithm in densely deployed HetNet, the MSs achieves at least 11.05% higher throughput than pure resource isolation approaches. If all exact interferers are identified, RMIP algorithm can achieve at least 19.03% higher throughput than pure resource isolation approaches.

6.1.3 Contribution

In this dissertation, we discuss several challenges in heterogeneous wireless networks including CRNs that balance the uneven utilization of different spectrum bands and HetNets that improve the spectrum utilization within the same spectrum bands. For CRNs, this dissertation contributes to designing efficient algorithms that help the SUs form an effective CRNs. The non-contiguous control channels establishment helps the SUs to find control channels efficiently in the licensed spectrum with a higher establishment rate than approaches for fixed-width channels. The k-protected routing protocol builds robust routing paths that are never interrupted even when k PUs become active again. This dissertation also contributes to combating the inter-cell interference, which is one of the most challenging issues in HetNets. The interference identification algorithm identifies the exact interferers efficiently and can achieve a stable identification quickly so that unnecessary interference mitigation can be avoided. The interference precancellation mitigates interference by precanceling the interference signal from known interferers and with interference precancellation, the same set of resources can be shared between MSs and the achievable throughput is improved. The mechanisms and algorithms proposed in this dissertation aim at breaking barriers for supporting heterogeneous wireless networks to improve the utilization of the precious and limited wireless spectrum.

6.2 Future Research Work

As discussed in this dissertation, the wireless demand has been skyrocketed in the past decade and will continue to grow in the future. In the year of 2021, video-related traffic is expected to grow threefold and the traffic originates from smartphones will surpass the traffic from personal computers [91]. Moreover, the wireless traffic will continue to multiply in the foreseeable future with new mobile applications such as live 3D/HD video, augmented reality, and virtual reality. Internet of Things is also a fast-growing field that will make a direct impact on wireless demands. Satisfying the even more overwhelming traffic demands remains one of the most critical challenges of future wireless networks.

In this dissertation, we have been focusing on improving spectrum utilization using heterogeneous wireless networks without allocating additional spectrum. Another straightforward approach to improve the capacity of wireless links is to aggregate more spectra. However, it is extremely challenging to allocate more spectra in the range from 300 MHz to 3 GHz to cellular networks, because the spectrum in the range is already saturated with the allocation to existing wireless technologies [64]. Employing higher radio frequencies to improve bandwidth is inevitable. Millimeter-wave bands (mmWaves) in the range from 30 to 300 GHz have large chunks of available spectrum and are promising for satisfying the traffic demands in future wireless networks.

In the past, the most common applications of mmWaves are short-range applications such as WPAN [17][108] because mmWaves suffer from severe air absorption and mmWaves have high attenuation against physical objects, such as precipitation, foliage or human activities [99][134]. Therefore, beamforming with directional antennas [45][44] is required to produce high gain to combat signal attenuation. Due to the severe congestion in the 2.4 GHz ISM band, IEEE 802.11ad that can operate on the 60 GHz band [1] has been published. IEEE 802.11ad boasts a theoretical maximum speed of 7 Gbps using the unlicensed 60 GHz.

Similarly, mmWave is also a promising candidate to improve the capacity of the cellular networks. In the past, the size of the directional antenna has hindered the use of mmWave on mobile devices. Fortunately, the advancement of the technologies in recent years has made small-

size high-gain steerable directional antennas for mobile devices (i.e., antenna arrays) affordable, and has made the research community to reconsider the viability of mmWave for cellular networks [59][101][100]. However, even with directional antennas, the coverage of mmWave-enabled cells is still smaller than that of sub-3GHz cells. The limited coverage ranges of the mmWave-enabled cells will densify the cellular network. Moreover, as discussed above, mmWave requires beamforming to combat signal attenuation, the interference also becomes directional. All these factors are expected to make the mmWave-enabled HetNets denser and more complicated than sub-3GHz HetNets.

The interferer identification in mmWave-enabled of HetNet is intrinsically different from that of omnidirectional wireless networks. Similarly, the signal directionality also makes the unlicensed opportunistic use of licensed spectrum different from that of sub-3 GHz CRNs. Since the SUs' interference is also limited to a specific direction, it might be possible that the SUs opportunistically access licensed spectrum without interfering with active PUs. The multitier heterogeneous wireless networks are still promising to deal with challenges introduced by the mmWave technology and to support the development of mmWave-enabled networks. We identify some of the most critical challenges of mmWave networks as follows.

- **Inter-cell Interference**: Similar to sub-3GHz cells with omnidirectional coverage, interference is a critical issue in mmWave-enabled HetNets. However, because the mmWave technology requires beamforming, the interference behavior of mmWave-enabled smallcells is intrinsically different from that of omnidirectional cells. In a mmWave-enabled HetNet, a beam direction will change along with the mobility of the MS it is serving. Two beams that were not interfering with each other might suddenly intersect at an MS due to mobility, and the intersection might make the beams to interfere with one another, which brings unexpected reduction to the overall performance. The directionality makes the interferences in mmWave-enabled HetNets more fickle than that of sub-3GHz HetNets. The interference detection needs to be done with high accuracy, and the mitigation needs to be activated faster.

- **Frequent Handover**: Even with directional antennas, the transmission range of mmWave

signals would still be shorter than that of sub-3GHz signals. The density of mmWave-enabled HetNets is expected to be even higher than sub-3GHz HetNets. Additionally, it is also possible that the changes of beam direction introduce physical objects in the line-of-sight of the beam and result in blockage of the beam[134]. The density of the cells, the mobility of the MSs, and the possible blockages will result in even more frequent handovers and more frequent user authentication on each cell, which demand both efficient handover algorithms and user authentication protocols.

- **Handover Candidate Cell Selection**: As aforementioned, the potential frequent handover is one of the challenges in mmWave-enabled HetNets, and the cell density is expected to be higher. The selection of candidate cell is critical to the performance of the HetNets. Unlike sub-3GHz cells, signal strength is no longer a reliable metric for measuring the quality of cells due to their directionality. A useful metric that is tailored for mmWave-enabled HetNets is desired and is critical to the performance of mmWave-enabled HetNets.

- **Blockage and Interference Prediction**: For cellphones and tablets that are carried by human beings, their mobility pattern is not purely random, and most of the obstacles are stationary. If history incidents information, such as incident type (e.g., blockage, interference), location, cell signal strength, can be recorded, applying machine learning or pattern matching techniques might be able to predict future incident. If blockage and interference incidents can be predicted, countermeasures can be taken before the incidents happen and the performance degradation can be minimized or even avoided. For example, for an MS at a location and its mobility pattern have matched to a frequent blockage incident in the past and blockage is predicted, the cell that is serving the MS can initiate a handover request before the blockage happens and reduce the handover delay time.

The challenges of mmWaves have formed a brand new research area. The development of mmWave-enabled wireless networks will continue to identify new challenges. More research efforts will be required to overcome these challenges, to break the barriers of mmWave-enabled

wireless networks, and to ensure the successful adoption of mmWave technology to future wireless networks.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Iso/iec/ieee international standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 3: Enhancements for very high throughput in the 60 ghz band (adoption of ieee std 802.11ad-2012). *ISO/IEC/IEEE 8802-11:2012/Amd.3:2014(E)*, pages 1–634, March 2014.

[2] 3GPP. Lte; evolved universal terrestrial radio access (e-utra); physical layer procedures (3gpp ts 36.213 version 8.8.0 release 8), 2009.

[3] F. M. Abinader, E. P. L. Almeida, F. S. Chaves, A. M. Cavalcante, R. D. Vieira, R. C. D. Paiva, A. M. Sobrinho, S. Choudhury, E. Tuomaala, K. Doppler, and V. A. Sousa. Enabling the coexistence of lte and wi-fi in unlicensed bands. *IEEE Communications Magazine*, 52(11):54–61, Nov 2014.

[4] I. F. Akyildiz, W. y. Lee, M. C. Vuran, and S. Mohanty. A survey on spectrum management in cognitive radio networks. *IEEE Communications Magazine*, 46(4):40–48, April 2008.

[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*, 50(13):2127 – 2159, 2006.

[6] A. Al-Ali and K. Chowdhury. Simulating dynamic spectrum access using ns-3 for wireless networks in smart environments. In *Sensing, Communication, and Networking Workshops (SECON Workshops), 2014 Eleventh Annual IEEE International Conference on*, pages 28–33, June 2014.

[7] J.G. Andrews. Seven ways that hetnets are a cellular paradigm shift. *Communications Magazine, IEEE*, 51(3):136–144, March 2013.

[8] M.Y. Arslan, Jongwon Yoon, K. Sundaresan, S.V. Krishnamurthy, and S. Banerjee. A resource management system for interference mitigation in enterprise ofdma femtocells. *Networking, IEEE/ACM Transactions on*, 21(5):1447–1460, Oct 2013.

[9] A Attar, V. Krishnamurthy, and O.N. Gharehshiran. Interference management using cognitive base-stations for umts lte. *Communications Magazine, IEEE*, 49(8):152–159, August 2011.

[10] Anne Berry, Jean R. S. Blair, Pinar Heggernes, and Barry W. Peyton. Maximum cardinality search for computing minimal triangulations of graphs. In *ALGORITHMICA*, pages 1–12. Springer Verlag, 2002.

[11] Sourjya Bhaumik, Shoban Preeth Chandrabose, Manjunath Kashyap Jataprolu, Gautam Kumar, Anand Muralidhar, Paul Polakos, Vikram Srinivasan, and Thomas Woo. Cloudiq: A framework for processing base stations in a data center. In *Proceedings of the 18th*

Annual International Conference on Mobile Computing and Networking, Mobicom '12, pages 125–136, New York, NY, USA, 2012. ACM.

[12] Kaigui Bian, Jung-Min Park, and Ruiliang Chen. A quorum-based framework for establishing control channels in dynamic spectrum access networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, MobiCom '09, pages 25–36, New York, NY, USA, 2009. ACM.

[13] Eric Blossom. Gnu radio: tools for exploring the radio frequency spectrum. *Linux J.*, 2004(122):4, June 2004.

[14] Michael Borokhovich, Liron Schiff, and Stefan Schmid. Provable data plane connectivity with local fast failover: Introducing openflow graph algorithms. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 121–126, New York, NY, USA, 2014. ACM.

[15] Vladimir Brik, Eric Rozner, and Suman Banerjee. Dsap: a protocol for coordinated spectrum access. In *In IEEE DySPAN*, pages 611–614, 2005.

[16] M.M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. Dimsumnet: new directions in wireless networking using coordinated dynamic spectrum. In *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pages 78–85, 2005.

[17] L.X. Cai, Lin Cai, Xuemin Shen, and Jon W. Mark. Rex: A randomized exclusive region based scheduling scheme for mmwave wpans with directional antenna. *Wireless Communications, IEEE Transactions on*, 9(1):113–121, January 2010.

[18] Eugene Chai, Jeongkeun Lee, Sung-Ju Lee, Raúl H. Etkin, and Kang G. Shin. Building efficient spectrum-agile devices for dummies. In *MOBICOM*, pages 149–160, 2012.

[19] A. Chandak and S. Ramasubramanian. Dual-link failure resiliency through backup link mutual exclusion. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, pages 258–267 Vol. 1, Oct 2005.

[20] V. Chandrasekhar, J. Andrews, and A. Gatherer. Femtocell networks: a survey. *Communications Magazine, IEEE*, 46(9):59–67, september 2008.

[21] Vikram Chandrasekhar and Jeffrey G. Andrews. Uplink capacity and interference avoidance for two-tier femtocell networks. *Trans. Wireless. Comm.*, 8(7):3498–3509, July 2009.

[22] Ronald Y. Chang, Zhifeng Tao, Jinyun Zhang, and C.-C. Jay Kuo. Dynamic fractional frequency reuse (d-ffr) for multicell ofdma networks using a graph framework. *Wireless Communications and Mobile Computing*, 13(1):12–27, 2013.

[23] A. Checko, H.L. Christiansen, Ying Yan, L. Scolari, G. Kardaras, M.S. Berger, and L. Dittmann. Cloud ran for mobile networks: A technology overview. *Communications Surveys Tutorials, IEEE*, 17(1):405–426, Firstquarter 2015.

[24] Geng Cheng, Wei Liu, Yunzhao Li, and Wenqing Cheng. Joint on-demand routing and spectrum assignment in cognitive radio networks. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 6499–6503, June 2007.

[25] Marco Chiesa, Ilya Nikolaevskiy, Slobodan Mitrovic, Andrei Gurtov, Aleksander Madry, Michael Schapira, and Scott Shenker. On the resiliency of static forwarding tables. *IEEE/ACM Trans. Netw.*, 25(2):1133–1146, April 2017.

[26] K.R. Chowdhury and I.F. Akyldiz. Ofdm-based common control channel design for cognitive radio ad hoc networks. *Mobile Computing, IEEE Transactions on*, 10(2):228–238, 2011.

[27] Asaf Cidon, Kanthi Nagaraj, Sachin Katti, and Pramod Viswanath. Flashback: decoupled lightweight wireless control. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '12, pages 223–234, New York, NY, USA, 2012. ACM.

[28] Holger Claussen. Co-channel operation of macro- and femtocells in a hierarchical cell structure. *International Journal of Wireless Information Networks*, 15(3-4):137–147, 2008.

[29] Federal Communications Commission. Amendment of part 15 of the commission's rules for unlicensed operations in the tv bands, repurposed 600 mhz band, 600 mhz guard bands and duplex gap, and channel 37, and amendment of part 74, et al.

[30] Federal Communications Commission et al. Unlicensed operation in the tv broadcast bands. *ET Docket*, (04-186), 2004.

[31] Shared Spectrum Company. General survey of radio frequency bands (30 mhz to 3 ghz). `http://www.sharedspectrum.com/papers/spectrum-reports/`, Vienna, Virginia, September 1 - 5 2009.

[32] Carlos Cordeiro and K. Challapali. C-mac: A cognitive mac protocol for multi-channel wireless networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 147–157, 2007.

[33] Claudia Cormio and Kaushik R. Chowdhury. A survey on mac protocols for cognitive radio networks. *Ad Hoc Netw.*, 7(7):1315–1329, September 2009.

[34] Claudia Cormio and Kaushik R. Chowdhury. Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping. *Ad Hoc Netw.*, 8(4):430–438, June 2010.

[35] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews. Modeling and analysis of k-tier downlink heterogeneous cellular networks. *IEEE Journal on Selected Areas in Communications*, 30(3):550–560, April 2012.

[36] Yong Ding and Li Xiao. Video on-demand streaming in cognitive wireless mesh networks. *Mobile Computing, IEEE Transactions on*, 12(3):412–423, 2013.

[37] Ettus. Universal software radio peripheral. `http://www.ettus.com/`.

[38] ET FCC. Docket no. 03-108. *Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies, FCC Report and Order adopted (March 10, 2005)*, 2005.

[39] Wei Feng, Jiannong Cao, Chisheng Zhang, Jun Zhang, and Qin Xin. Coordination of multi-link spectrum handoff in multi-radio multi-hop cognitive networks. *J. Parallel Distrib. Comput.*, 72(4):613–625, April 2012.

[40] A. B. Flores, R. E. Guerra, E. W. Knightly, P. Ecclesine, and S. Pandey. Ieee 802.11af: a standard for tv white space spectrum sharing. *IEEE Communications Magazine*, 51(10):92–100, October 2013.

[41] Chenfei Gao, G. Ozcan, Jian Tang, M. C. Gursoy, and Weiyi Zhang. R-cloud: A cloud framework for enabling radio-as-a-service over a wireless substrate. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–10, Nov 2016.

[42] Fanica Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.

[43] GNU Radio Website, accessed February 2012.

[44] L.C. Godara. Application of antenna arrays to mobile communications. ii. beam-forming and direction-of-arrival considerations. *Proceedings of the IEEE*, 85(8):1195–1245, Aug 1997.

[45] L.C. Godara. Applications of antenna arrays to mobile communications. i. performance improvement, feasibility, and system considerations. *Proceedings of the IEEE*, 85(7):1031–1060, Jul 1997.

[46] N. M. Gowda and A. P. Kannu. Interferer identification in hetnets using compressive sensing framework. *IEEE Transactions on Communications*, 61(11):4780–4787, November 2013.

[47] IEEE 802.16 Working Group et al. Ieee standard for local and metropolitan area networks, part 16: Air interface for fixed broadband wireless access systems. *IEEE Std*, 802:16–2004, 2004.

[48] M. Hadzialic, B. Dosenovic, M. Dzaferagic, and J. Musovic. Cloud-ran: Innovative radio access network architecture. In *ELMAR, 2013 55th International Symposium*, pages 115–120, Sept 2013.

[49] G. Hampel, K.L. Clarkson, J.D. Hobby, and P.A. Polakos. The tradeoff between coverage and capacity in dynamic optimization of 3g cellular networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 2, pages 927–932 Vol.2, Oct 2003.

[50] Thomas R Henderson, Mathieu Lacage, George F Riley, C Dowell, and JB Kopena. Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 2008.

[51] J.W. Huang and V. Krishnamurthy. Cognitive base stations in lte/3gpp femtocells: A correlated equilibrium game-theoretic approach. *Communications, IEEE Transactions on*, 59(12):3485–3493, December 2011.

[52] R. Irmer and F. Diehm. On coverage and capacity of relaying in lte-advanced in example deployments. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–5, Sept 2008.

[53] Raj Jain, Arjan Durresi, and Gojko Babic. Throughput fairness index: An explanation. In *TM Forum Document Number: ATM Forum / 990045*, Feb 1999.

[54] Klaus Jansen. The optimum cost chromatic partition problem. In *Algorithms and complexity*, pages 25–36. Springer, 1997.

[55] Juncheng Jia, Qian Zhang, and Xuemin Shen. Hc-mac: A hardware-constrained cognitive mac for efficient spectrum management. *Selected Areas in Communications, IEEE Journal on*, 26(1):106–117, 2008.

[56] Mahdi Pirmoradian Jonathan Rodriguez, Olayinka Adigun and Christos Politis. *Cognitive Radio for 5G Wireless Networks in Fundamentals of 5G Mobile Networks*. Wiley, 2015.

[57] C. Joo and N. B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. *IEEE/ACM Transactions on Networking*, 17(5):1481–1493, Oct 2009.

[58] Ahmed E. Kamal. 1 + n network protection for mesh networks: Network coding-based protection using p-cycles. *IEEE/ACM Trans. Netw.*, 18(1):67–80, February 2010.

[59] Taeyoung Kim, JeongHo Park, Ji-Yun Seol, Suryong Jeong, Jaeweon Cho, and Wonil Roh. Tens of gbps support with mmwave beamforming systems for next generation communications. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 3685–3690, Dec 2013.

[60] S. Kini, S. Ramasubramanian, A. Kvalbein, and A.F. Hansen. Fast recovery from dual-link or single-node failures in ip networks using tunneling. *Networking, IEEE/ACM Transactions on*, 18(6):1988–1999, Dec 2010.

[61] Murali Kodialam and Thyaga Nandagopal. Characterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom '03, pages 42–54, New York, NY, USA, 2003. ACM.

[62] Y.R. Kondareddy and P. Agrawal. Synchronized mac protocol for multi-hop cognitive radio networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 3198–3202, 2008.

[63] G. Kuperman and E. Modiano. Providing protection in multi-hop wireless networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 926–934, April 2013.

[64] M. Lazarus. The great spectrum famine. *Spectrum, IEEE*, 47(10):26–31, October 2010.

[65] Loukas Lazos, Sisi Liu, and Marwan Krunz. Spectrum opportunity-based control channel assignment in cognitive radio networks. In *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON'09, pages 135–143, Piscataway, NJ, USA, 2009. IEEE Press.

[66] Heui-Chang Lee, Dong-Chan Oh, and Yong-Hwan Lee. Mitigation of inter-femtocell interference with adaptive fractional frequency reuse. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5, May 2010.

[67] Husheng Li and Lijun Qian. Enhancing the reliability of cognitive radio networks via channel assignment: risk analysis and redundancy allocation. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6, March 2010.

[68] Qi Li, Mingwei Xu, Lingtao Pan, and Yong Cui. A study of path protection in self-healing routing. In *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pages 554–561. Springer, 2008.

[69] Yu-Shan Liang, Wei-Ho Chung, Guo-Kai Ni, Ing-Yi Chen, Hongke Zhang, and Sy-Yen Kuo. Resource allocation with interference avoidance in ofdma femtocell networks. *Vehicular Technology, IEEE Transactions on*, 61(5):2243–2255, Jun 2012.

[70] Shao-Yu Lien, Chih-Cheng Tseng, Kwang-Cheng Chen, and Chih-Wei Su. Cognitive radio resource management for qos guarantees in autonomous femtocell networks. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6, may 2010.

[71] Zhiyong Lin, Hai Liu, Xiaowen Chu, and Yiu-Wing Leung. Enhanced jump-stay rendezvous algorithm for cognitive radio networks. *Communications Letters, IEEE*, 17(9):1742–1745, September 2013.

[72] C. J. Liu and L. Xiao. Rmip: Resource management with interference precancellation in heterogeneous cellular networks. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–10, Nov 2016.

[73] Chin-Jung Liu, Pei Huang, Li Xiao, and A.-H. Esfahanian. Interference identification and resource management in ofdma femtocell networks. In *Networking Conference, 2014 IFIP*, pages 1–9, June 2014.

[74] Chin-Jung Liu, Pei Huang, Li Xiao, and Abdol-Hossein Esfahanian. Interference identification and resource management in ofdma femtocell networks. In *Proceedings of the 13th International IFIP TC 6 Conference on Networking*, NETWORKING'14, June 2014.

[75] Chin-Jung Liu and Li Xiao. k-protected routing protocol in multi-hop cognitive radio networks. In *Proceedings of the 2017 International Conference on Distributed Computing Systems, Applications and Experiences Track*, 2017.

[76] Hai Liu, Zhiyong Lin, Xiaowen Chu, and Y.-W. Leung. Jump-stay rendezvous algorithm for cognitive radio networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(10):1867–1881, 2012.

[77] Hai Liu, Zhiyong Lin, Xiaowen Chu, and Yiu-Wing Leung. Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, GREENCOM-CPSCOM '10, pages 755–760, Washington, DC, USA, 2010. IEEE Computer Society.

[78] Junda Liu, Aurojit Panda, Ankit Singla, Brighten Godfrey, Michael Schapira, and Scott Shenker. Ensuring connectivity via data plane mechanisms. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, pages 113–126, Berkeley, CA, USA, 2013. USENIX Association.

[79] Brandon F. Lo. A survey of common control channel design in cognitive radio networks. *Phys. Commun.*, 4(1):26–39, March 2011.

[80] D. Lopez-Perez, A Valcarce, G. de la Roche, and Jie Zhang. Ofdma femtocells: A roadmap on interference avoidance. *Communications Magazine, IEEE*, 47(9):41–48, September 2009.

[81] Report M.2243. Assessment of the global mobile broadband deployments and forecasts for international mobile telecommunications. `https://www.itu.int/pub/R-REP-M.2243-2011`, 11 2011.

[82] Liangping Ma, Xiaofeng Han, and Chien-Chung Shen. Dynamic open spectrum sharing mac protocol for wireless ad hoc networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 203–213, 2005.

[83] Rukun Mao and Husheng Li. Protecting cognitive radio networks against primary users: A backup path approach. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6, Dec 2011.

[84] Thomas L. Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas. *Trans. Wireless. Comm.*, 9(11):3590–3600, November 2010.

[85] Muriel Médard, Steven G. Finn, and Richard A. Barry. Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. *IEEE/ACM Trans. Netw.*, 7(5):641–652, October 1999.

[86] C. Mehlführer, M. Wrulich, J. C. Ikuno, D. Bosanska, and M. Rupp. Simulating the long term evolution physical layer. In *2009 17th European Signal Processing Conference*, pages 1471–1478, Aug 2009.

[87] Juha Meinilä, Pekka Kyösti, Tommi Jämsä, and Lassi Hentilä. Winner ii channel models. *Radio Technologies and Concepts for IMT-Advanced*, pages 39–92, 2009.

[88] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for wlans. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(3):19–31, July 2005.

[89] T. Nakamura, S. Nagata, A. Benjebbour, Y. Kishiyama, Tang Hai, Shen Xiaodong, Yang Ning, and Li Nan. Trends in small cell enhancements in lte advanced. *Communications Magazine, IEEE*, 51(2):98–105, February 2013.

[90] Richard van Nee and Ramjee Prasad. *OFDM for Wireless Multimedia Communications*. Artech House, Inc., Norwood, MA, USA, 1st edition, 2000.

[91] White Paper. Cisco visual networking index: Forecast and methodology, 2016-2021. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, 3 2017.

[92] Fernando Magno Quintao Pereira and Jens Palsberg. Register allocation via coloring of chordal graphs. In *Programming Languages and Systems*, pages 315–329. Springer, 2005.

[93] Shu ping Yeh, Shilpa Talwar, Nageen Himayat, and Kerstin Johnsson. Text proposal on hierarchical networks simulation methodology. http://www.ieee802.org/16/ppc/contrib/C80216ppc-10_0061.doc, September 2010.

[94] Chang Woo Pyo and Mikio Hasegawa. Minimum weight routing based on a common link control radio for cognitive wireless ad hoc networks. In *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing*, IWCMC '07, pages 399–404, New York, NY, USA, 2007. ACM.

[95] Qualcomm. Lte-u: Lte advanced in unlicensed spectrum. https://www.qualcomm.com/invention/technologies/lte/unlicensed.

[96] T. Quek, Zhongding Lei, and Sumei Sun. Adaptive interference coordination in multi-cell ofdma systems. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 2380–2384, sept. 2009.

[97] Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat. Learning to share: narrowband-friendly wideband networks. *SIGCOMM Comput. Commun. Rev.*, 38(4):147–158, August 2008.

[98] S Ramamurthy, Laxman Sahasrabuddhe, and Biswanath Mukherjee. Survivable wdm mesh networks. *Journal of Lightwave Technology*, 21(4):870, 2003.

[99] T. S. Rappaport and S. Deng. 73 ghz wideband millimeter-wave foliage and ground reflection measurements and models. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1238–1243, June 2015.

[100] T.S. Rappaport, Shu Sun, R. Mayzus, Hang Zhao, Y. Azar, K. Wang, G.N. Wong, J.K. Schulz, M. Samimi, and F. Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *Access, IEEE*, 1:335–349, 2013.

[101] W. Roh, Ji-Yun Seol, JeongHo Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and F. Aryanfar. Millimeter-wave beamforming as an enabling technology for 5g cellular communications: theoretical feasibility and prototype results. *Communications Magazine, IEEE*, 52(2):106–113, February 2014.

[102] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson. Scaling up mimo: Opportunities and challenges with very large arrays. *IEEE Signal Processing Magazine*, 30(1):40–60, Jan 2013.

[103] Adel AM Saleh and Jane M Simmons. Evolution toward the next-generation core optical network. *Journal of lightwave Technology*, 24(9):3303, 2006.

[104] Ashwin Sampath, Lei Yang, Lili Cao, Haitao Zheng, and Ben Y. Zhao. High throughput spectrum-aware routing for cognitive radio networks. In *PROC. OF INTERNATIONAL CONFERENCE ON COGNITIVE RADIO ORIENTED WIRELESS NETWORKS AND COMMUNICATIONS (CROWNCOM)*, 2007.

[105] S. Sen, N. Santhapuri, R.R. Choudhury, and S. Nelakuditi. Successive interference cancellation: Carving out mac layer opportunities. *Mobile Computing, IEEE Transactions on*, 12(2):346–357, Feb 2013.

[106] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001.

[107] Skype. How much bandwidth does skype need? `https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need`, 2018.

[108] In Keun Son, Shiwen Mao, M.X. Gong, and Yihan Li. On frame-based scheduling for directional mmwave wpans. In *INFOCOM, 2012 Proceedings IEEE*, pages 2149–2157, March 2012.

[109] Mechthild Stoer. *Design of survivable networks*. Springer, 1992.

[110] Hang Su and Xi Zhang. Cross-layer based opportunistic mac protocols for qos provisionings over cognitive radio wireless networks. *Selected Areas in Communications, IEEE Journal on*, 26(1):118–129, 2008.

[111] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan, and S. V. Krishnamurthy. Fluidnet: A flexible cloud-based radio access network for small cells. *IEEE/ACM Transactions on Networking*, 24(2):915–928, April 2016.

[112] Karthikeyan Sundaresan and Sampath Rangarajan. Efficient resource management in ofdma femto cells. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 33–42, New York, NY, USA, 2009. ACM.

[113] Robert E Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on computing*, 13(3):566–579, 1984.

[114] N.C. Theis, R.W. Thomas, and L.A. DaSilva. Rendezvous for cognitive radios. *Mobile Computing, IEEE Transactions on*, 10(2):216–227, 2011.

[115] S. Uygungelen, G. Auer, and Z. Bharucha. Graph-based dynamic frequency reuse in femtocell networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–6, May 2011.

[116] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis. The ieee 802.11g standard for high data rate wlans. *IEEE Network*, 19(3):21–26, May 2005.

[117] Sergio Verdu. *Multiuser detection*. Cambridge university press, 1998.

[118] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-layer wireless bit rate adaptation. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, pages 3–14, New York, NY, USA, 2009. ACM.

[119] C. X. Wang, F. Haider, X. Gao, X. H. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir. Cellular architecture and key technologies for 5g wireless communication networks. *IEEE Communications Magazine*, 52(2):122–130, February 2014.

[120] J.-S. Wu, J.-K. Chung, and M.-T. Sze. Analysis of uplink and downlink capacities for two-tier cellular system. *Communications, IEE Proceedings-*, 144(6):405–411, dec 1997.

[121] Baohua Yang, Junda Liu, S. Shenker, Jun Li, and Kai Zheng. Keep forwarding: Towards k-link failure resilient routing. In *INFOCOM, 2014 Proceedings IEEE*, pages 1617–1625, April 2014.

[122] Lei Yang, Wei Hou, Lili Cao, Ben Y. Zhao, and Haitao Zheng. Supporting demanding wireless applications with frequency-agile radios. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 5–5, Berkeley, CA, USA, 2010. USENIX Association.

[123] Lei Yang, Zengbin Zhang, Wei Hou, Ben Y. Zhao, and Haitao Zheng. Papyrus: A software platform for distributed dynamic spectrum sharing using sdrs. *SIGCOMM Comput. Commun. Rev.*, 41(1):31–37, January 2011.

[124] Sixing Yin, Dawei Chen, Qian Zhang, Mingyan Liu, and Shufang Li. Mining spectrum usage data: a large-scale spectrum measurement study. *Mobile Computing, IEEE Transactions on*, 11(6):1033–1046, 2012.

[125] Jongwon Yoon, Mustafa Y. Arslan, Karthikeyan Sundaresan, Srikanth V. Krishnamurthy, and Suman Banerjee. A distributed resource management framework for interference mitigation in ofdma femtocell networks. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 233–242, New York, NY, USA, 2012. ACM.

[126] Jongwon Yoon, M.Y. Arslan, K. Sundaresan, S.V. Krishnamurthy, and S. Banerjee. Self-organizing resource management framework in ofdma femtocells. *Mobile Computing, IEEE Transactions on*, 14(4):843–857, April 2015.

[127] Tevfik Yucek and Hüseyin Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys & Tutorials, IEEE*, 11(1):116–130, 2009.

[128] T. Zahir, K. Arshad, A Nakata, and K. Moessner. Interference management in femtocells. *Communications Surveys Tutorials, IEEE*, 15(1):293–311, First 2013.

[129] Jie Zhang and Guillaume de la Roche. *Femtocells: Technologies and Deployment*. Wiley Publishing, 2010.

[130] Yifan Zhang, Qun Li, Gexin Yu, and Baosheng Wang. Etch: Efficient channel hopping for communication rendezvous in dynamic spectrum access networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2471–2479, 2011.

[131] Jing Zhao and Guohong Cao. Robust topology control in multi-hop cognitive radio networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 2032–2040, March 2012.

[132] Jun Zhao, Haitao Zheng, and Guang-Hua Yang. Distributed coordination in dynamic spectrum allocation networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 259–268, 2005.

[133] Yongli Zhao, Xin Li, Jie Zhang, Shanguo Huang, and Wanyi Gu. K-dimensional protection structure (kdps) for multi-link failure in data center optical networks. *Optik - International Journal for Light and Electron Optics*, 125(19):5490 – 5493, 2014.

[134] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y. Zhao, and Haitao Zheng. Demystifying 60ghz outdoor picocells. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 5–16, New York, NY, USA, 2014. ACM.