TOWARD EFFICIENT SPECTRUM USE IN MULTICARRIER WIRELESS NETWORKS

By

Pei Huang

A DISSERTATION

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

Computer Science - Doctor of Philosophy

2014

ABSTRACT

TOWARD EFFICIENT SPECTRUM USE IN MULTICARRIER WIRELESS NETWORKS

By

Pei Huang

The last decade has witnessed growing interest in dynamic spectrum access, which is motivated by the observation that a large portion of the radio spectrum has been licensed but remains highly underutilized while a few small unlicensed bands that are open to anyone are getting more crowded due to the explosive expansion of wireless services. To provide more flexible access to radio spectrum, dynamic spectrum access is introduced to enable unlicensed users to opportunistically utilize vacant spectrum chunks (known as spectrum holes) in licensed frequency bands.

In dynamic spectrum access, non-contiguous orthogonal frequency division multiplexing (NC-OFDM) is widely adopted to efficiently utilize fragmented spectrum because it is convenient to keep silent on some spectrum fragments to avoid interference with licensed users. In NC-OFDM, a band of spectrum is divided into many orthogonal subcarriers and data are transmitted on a subset of them simultaneously. The subcarriers that interfere with the licensed users are deactivated. Because each subcarrier can be managed independently, this dissertation introduces a series of techniques that exploit the subcarriers to address problems in dynamic spectrum access.

When unlicensed users called secondary users (SUs) are granted the permission to operate in the licensed bands, they must ensure that the interference caused by them to licensed users known as primary users (PUs) is within a limit. Even without such a requirement, SUs should avoid as many collisions as possible. To improve spectrum hole extraction rate and reduce collision rate, we propose a spectrum occupancy prediction model that helps estimate the spectrum availability. It measures a wide band of spectrum with OFDM and groups subcarriers to subchannels based on spectrum use activities. In each subchannel, frequent spectrum occupancy patterns are identified and used to predict future channel states (i.e., busy or idle). With the prediction, SUs are able to make use of spectrum holes more aggressively without introducing undue interference to PUs. In the spectrum holes discovered above, a mechanism is needed to coordinate medium access between devices. Because devices opportunistically utilize spectrum holes, a device may experience severe contentions with devices from various applications. We propose a collision detection and bitwise arbitration (CDBA) mechanism that quickly identifies the winner in a contention using combined information from both the time domain and the frequency domain. It enables collision detection in the frequency domain by selectively deactivating subcarriers at each transmitter.

The CDBA assumes that all devices adopt the same channel width, but different radio technologies have different requirements on channel width. When heterogeneous radios coexist in a contention domain, wideband devices can hardly win medium access opportunities in contention with narrowband devices. To address the problem, we propose an adaptive channel bonding protocol in which a wideband device initiates transmission as long as there exist some idle narrow channels and gradually increases channel width during transmission whenever new narrow channels become available. To increase the chance to win some narrow channels, a wideband device contends on subcarriers of each narrow channel with a different priority.

After the contention problem is addressed, we study how to increase the transmission speed when a device is granted the permission to transmit. As wireless networks move toward wider channel widths, it is common that different subcarriers experience different fade. To cope with the frequency-selective fading, modulation scheme for each subcarrier should be selected based on the subcarrier channel quality. We exploit the modulation diversity in our modulation scheme coding to improve network throughput.

Besides unicast, broadcast is another fundamental mechanism in wireless networks. Because devices utilize spectrum fragments opportunistically, different receivers may have different vacant spectrum fragments at different locations. The broadcast is more challenging in dynamic spectrum access because the transmitter needs to consider the diversity of spectrum availability. We propose a spectrum fragment agile broadcast (SFAB) protocol to support broadcast under nonuniform spectrum availability. It encodes unique sequences on subcarriers of each spectrum fragment to achieve fast spectrum agreement between the transmitter and the receivers.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Professor Li Xiao for her continued support, guidance, and encouragement. Without her, I cannot complete this dissertation.

I would also like to express my deepest appreciation to the members of my dissertation committee, Professor Matt Mutka, Professor Guoliang Xing, and Professor Vidyadhar Mandrekar, for their valuable comments and guidance.

Special thanks to my collaborators and classmates, Chen Wang, Bo Wang, Yong Ding, Guokai Zeng, Chin-Jung Liu, Chowdhury Sayeed Hyder, Kanthakumar Pongaliur, Xi Yang, and Jin Chen for sharing their technical knowledge and invaluable assistance.

Finally, I am grateful to my caring, loving, and supportive parent and wife for supporting me all the time.

TABLE OF CONTENTS

LIST O	F TABI	LES
LIST O	F FIGU	RES
СНАРТ	ER 1	INTRODUCTION
1.1	Spectr	um Prediction
1.2	Spectr	um Contention
	1.2.1	Single-Channel Contention
	1.2.2	Multichannel Contention
1.3	Spectr	um Use
	1.3.1	Modulation Scheme Diversity
	1.3.2	Spectrum Availability Diversity
1.4	Disser	ation Organization
СНАРТ	TER 2	WIRELESS SPECTRUM OCCUPANCY PREDICTION BASED ON
		PARTIAL PERIODIC PATTERN MINING
2.1	Relate	1 work
2.2	Proble	m Formulation
2.3	Partial	Periodic Pattern Mining
	2.3.1	Frequent Pattern Enumeration
	2.3.2	Support Counting
	2.3.3	Candidate Pattern Pruning 18
		2.3.3.1 Confidence Constraint
		2.3.3.2 Backward-Extension Constraint
	2.3.4	Channel State Prediction
2.4	Predict	ion of Duration
	2.4.1	Duration Pattern Mining
	2.4.2	State Duration Prediction
2.5	Perform	nance Study
	2.5.1	Data Collection
	2.5.2	Preprocessing
	2.5.3	Macro Patterns
	2.5.4	Micro Patterns
	2.5.5	Impact of Utilization Change
	2.5.6	Impact of Training Data Set Size
	2.5.7	DSA with Prediction
2.6	Summ	ary
СНАРТ	TER 3	COLLISION DETECTION AND BITWISE ARBITRATION IN MUL-
		TICARRIER WIRELESS NETWORKS 40
3.1	Relate	d Work

3.2	Motiva	tion
	3.2.1	802.11 Medium Access Overhead
	3.2.2	High Collision Probability in CD
3.3	Archite	ecture and Design
	3.3.1	Arbitration Preamble Design
	3.3.2	Self-Interference Cancellation
	3.3.3	Collision Probe
	3.3.4	Bitwise Arbitration
		3.3.4.1 Time Domain without Cancellation
		3.3.4.2 Frequency Domain with Cancellation
3.4	CDBA	Design Details
	3.4.1	Noise and Interference
	3.4.2	Power Leakage
	3.4.3	System Delay and Asynchronous Contention
	3.4.4	Random Backoff
	3.4.5	Misdetection 61
	3.4.6	Hidden Terminal 62
	3.4.7	Message Prioritization 63
3.5	Testbe	d Results 64
0.0	3.5.1	Collision Detection Feasibility 65
	3.5.2	Throughput Gain
	3.5.3	Message Prioritization 71
3.6	Simula	tion Results
	3.6.1	Simulation Settings
	3.6.2	Impact of Contending Transmitters' Number on Throughput
	3.6.3	Impact of Data Rates on Throughput
	3.6.4	Fairness
3.7	Summa	ary
CHAPI	EK 4	ADAPTIVE CHANNEL BUNDING IN MULTICARRIER WIRE-
4 1	Dalata	LESS NET WORKS
4.1	Adonti	1 WOIK
4.2	Adapti	Outerrieure
	4.2.1	Overview
	4.2.2	A 2 2 1 Compound Describle Design
		4.2.2.1 Compound Pleamble Design
		4.2.2.2 Collision Detection
		4.2.2.5 Medium Access Control
	4 2 2	4.2.2.4 Case Sludy
	4.2.3	Spectrum Agreement
		4.2.3.1 Cross-correlation
	1 2 4	4.2.5.2 Partial Spectrum Correlation
	4.2.4	Synchronization Preamble Detection
4.2	4.2.3 D: f	Changing Channel width
4.3	Perform	nance Evaluation

	4.3.1	Medium Access Contention
		4.3.1.1 Reliability of Collision Detection
		4.3.1.2 Throughput Gain
	4.3.2	Spectrum Agreement
	4.3.3	Simulation Results
4.4	Summ	ary
СНАРТ	ER 5	EXPLOITING MODULATION DIVERSITY IN MULTICARRIER
		WIRELESS NETWORKS
5.1	Relate	d Work
5.2	Modul	ation Scheme Coding
	5.2.1	Challenge
	5.2.2	Splitting to Sets
	5.2.3	Identifying Modulation Scheme Code in a Set
5.3	Subcar	rier Nulling
5.4	Perform	mance Evaluation
	5.4.1	Modulation Scheme Code Abstraction
	5.4.2	Subcarrier Nulling Performance
	5.4.3	Bit Error Rate Reduction
	5.4.4	Throughput Improvement
5.5	Summ	ary
СНАРТ	TER 6	EFFICIENT BROADCAST ON FRAGMENTED SPECTRUM 131
6.1	Relate	d Work
6.2	Overvi	ew
	6.2.1	Broadcast Problem
	6.2.2	Challenges
6.3	Design	1
	6.3.1	
		Spectrum Agreement
		Spectrum Agreement
		Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification137
		Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation138
		Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144
	6.3.2	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145
	6.3.2 6.3.3	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148
6.4	6.3.2 6.3.3 Perform	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication149ance Evaluation149
6.4	6.3.2 6.3.3 Perform 6.4.1	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148nance Evaluation149Spectrum Agreement150
6.4	6.3.2 6.3.3 Perform 6.4.1 6.4.2	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148nance Evaluation149Spectrum Agreement150Simulation153
6.4	6.3.2 6.3.3 Perforn 6.4.1 6.4.2	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148nance Evaluation149Spectrum Agreement150Simulation1536.4.2.1Simulation Setup157
6.4	6.3.2 6.3.3 Perform 6.4.1 6.4.2	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148nance Evaluation149Spectrum Agreement150Simulation1536.4.2.1Simulation Setup1536.4.2.2Efficiency of Partial Spectrum Correlation156
6.4	6.3.2 6.3.3 Perform 6.4.1 6.4.2	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148mance Evaluation149Spectrum Agreement150Simulation1536.4.2.1Simulation Setup1536.4.2.2Efficiency of Partial Spectrum Correlation1566.4.2.3Broadcast156
6.4	6.3.2 6.3.3 Perfort 6.4.1 6.4.2 Summ	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148nance Evaluation149Spectrum Agreement150Simulation1536.4.2.1Simulation Setup1536.4.2.2Efficiency of Partial Spectrum Correlation1556.4.2.3Broadcast156ary159
6.4 6.5 CHAPT	6.3.2 6.3.3 Perfort 6.4.1 6.4.2 Summ	Spectrum Agreement1366.3.1.1Correlation1366.3.1.2Broadcast Intention and Broadcast Identification1376.3.1.3Partial Spectrum Correlation1386.3.1.4Respond to Broadcast Intention144Dividing a Packet and Interleaving145Establishing Communication148nance Evaluation149Spectrum Agreement150Simulation1536.4.2.1Simulation Setup1536.4.2.2Efficiency of Partial Spectrum Correlation1556.4.2.3Broadcast156ary159159CONCLUSIONS AND FUTURE WORK160

7.2	Future Wo	rk	 	•	•	•	 	•	•	•	 •	•	•	•	•	•	•	 •	•	•	•	•	•	•	 •	• •	164	
BIBLIC	OGRAPHY		 	•	•	•	 		•	•	 •				•	•		 •	•				•	•	 •	1	166	

LIST OF TABLES

Table 2.1: Time complexity with different training set sizes (2.3 GHz CPU 4G RAM Laptop)	36
Table 3.1: OFDM PHY characteristics	45

LIST OF FIGURES

Figure 2.1: Terms being used.	13
Figure 2.2: An illustration of the frequent partial periodic pattern tree	14
Figure 2.3: Rules can be fused under certain conditions	21
Figure 2.4: Measurement setup	26
Figure 2.5: Average power measured at $f_c = 1.91$ GHz	27
Figure 2.6: Noise power variation at channel 60	28
Figure 2.7: The duration prediction in the paging bands	29
Figure 2.8: The channel state prediction results under different rule confidence constraints. The total accuracy takes all missed slots as wrong predictions	31
Figure 2.9: Examples of frequent patterns	32
Figure 2.10: The channel state prediction results in the Personal Communication Service (PCS) bands	32
Figure 2.11: The channel state prediction results on different channel utilization periods $(\delta = 0.4)$. Either prior day's data or the same day's data are used for training.	34
Figure 2.12: Prediction accuracy with different training set sizes.	36
Figure 2.13: The spectrum extraction rate is significantly improved with PPPM-assisted prediction.	38
Figure 3.1: 802.11 DCF	45
Figure 3.2: Collision detection in the frequency domain	47
Figure 3.3: Current self-interference cancellation is sufficient to prevent ADC saturation	49
Figure 3.4: Arbitration phase is shortened with additional information obtained in the frequency domain.	51
Figure 3.5: The flowchart of CDBA.	54

Figure 3.6: FFT results of received signal when 2 nodes are transmitting with 64-point IFFT. One node occupies subcarriers 15, 21, and 51; the other node occupies 27, 39, and 45. The peak at subcarrier 32 is the DC offset	56
	20
Figure 3.7: Ensuring a complete arbitration preamble by repetition	59
Figure 3.8: Different amount of an arbitration preamble is included in the FFT	60
Figure 3.9: Comparing the signal from the original transmitter with the signal from a relaying node in the time domain.	63
Figure 3.10: Results of performing 64-point FFT on received signals. The peak in the middle of each frame is the DC offset. Left frame contains no transmission. Right frame contains transmissions from two nodes	66
Figure 3.11: Taking a certain percent of the highest magnitude in a FFT frame to determine active subcarriers.	67
Figure 3.12: Collision detection performance in CDBA	68
Figure 3.13: Throughput gain of CDBA over 802.11.	69
Figure 3.14: Channel access overhead.	70
Figure 3.15: Message prioritization in CDBA.	71
Figure 3.16: Impact of contending nodes' number on throughput (CDBA-CN overlaps with CDBA, WiFi-Nano-NAV-CN overlaps with WiFi-Nano-NAV)	74
Figure 3.17: Impact of data rates on throughput under light contention.	76
Figure 3.18: Impact of data rates on throughput under heavy contention	77
Figure 3.19: Impact of contending nodes' number on fairness.	79
Figure 3.20: Impact of data rates on fairness.	80
Figure 4.1: Binary code mapping with NC-OFDM for collision detection.	89
Figure 4.2: The initial three phases in adaptive channel bonding.	92
Figure 4.3: Time domain signal correlation.	95
Figure 4.4: Cross-correlation of received signal with expected IFFT results	98
Figure 4.5: Results of performing 64 point FFT on received signal (left: no transmission; right: transmitting preamble)	01

Figure 4.6: Setting noise threshold slightly higher than the noise floor to increase detection accuracy
Figure 4.7: The misdetection rate is reduced with higher SINR
Figure 4.8: Throughput improvement with bitwise arbitration
Figure 4.9: Partial spectrum correlation for channel 1 at SINR of 0 dB
Figure 4.10: Partial spectrum correlation for channel 2 at SINR of 0 dB
Figure 4.11: Setting threshold for correlation peak indication
Figure 4.12: Probability of missing detections of occupied channels
Figure 4.13: Throughput comparison with different number of contending transmitters 108
Figure 4.14: The throughput of a wideband device is affected by the number of interference- free narrow channels
Figure 4.15: The throughput of a wideband device is affected by narrowband devices' activities
Figure 5.1: OFDM Transmitter
Figure 5.2: The code length for a modulation scheme impacts the shift in bit allocation 117
Figure 5.3: Constellation Diagram with Subcarrier Nulling
Figure 5.4: The percentage of bonus bits and the percentage of total bits that can be represented by modulation scheme codes
Figure 5.5: There are more opportunities to use the modulation scheme code when a low- density modulation scheme is used
Figure 5.6: The percentage of total bits represented by modulation scheme codes when more than one code is used for a high-density modulation scheme
Figure 5.7: The percentage of total bits represented by modulation scheme codes when d- ifferent modulation schemes are used for different subcarriers (BPSK:QPSK:16- QAM:64-QAM)
Figure 5.8: At the same SNR, the detection error rate is much lower than the bit error rate 127
Figure 5.9: The BER is reduced due to the known bits introduced by MSC
Figure 5.10: The throughput gains introduced by MSC

Figure 6.1: Broadcast on fragmented spectrum. Shaded channels are occupied	. 134
Figure 6.2: A extended example of broadcast on fragmented spectrum	. 136
Figure 6.3: The cross-correlation value is maximized when the base sequence is aligned with itself at a certain time lag of the received signal.	. 138
Figure 6.4: Construct a broadcast intention OFDM symbol through IFFT	. 139
Figure 6.5: A rectangular function in the frequency domain matches a sinc function in the time domain. The Duality property of the Fourier transform provides that the reverse is also true.	. 141
Figure 6.6: The receiver assumes that the base broadcast intention sequence is present in each channel.	. 142
Figure 6.7: Identify whether or not a broadcast intention sequence is present in a narrow channel.	. 142
Figure 6.8: Identify the best channel combination to cover all nodes.	. 147
Figure 6.9: An example to show channels in which a node should receive data. 'x' means the channel is unavailable at the node	. 148
Figure 6.10: An example of partial spectrum correlation for broadcast intention at 0 dB	. 152
Figure 6.11: An example of partial spectrum correlation for spectrum notification at 0 dB	. 154
Figure 6.12: The control overhead has a significant impact on the throughput in high speed transmissions.	. 156
Figure 6.13: Throughput comparison with different number of narrowband interference sources.	. 157
Figure 6.14: Throughput comparison with different data rates of narrowband interference sources.	. 158

CHAPTER 1

INTRODUCTION

In most countries, the radio spectrum is divided into frequency bands and assigned to users through administrative licensing. Some frequency bands are licensed to certain services whereas any device can transmit in unlicensed frequency bands as long as it respects certain communication parameter limits. Due to the explosive expansion of wireless services, we are lack of spectrum to allocate to new services. Many wireless services are crowed in unlicensed frequency bands. The spectrum scarcity problem calls for more flexible access to radio spectrum.

Recent spectrum utilization measurements have revealed that while the unlicensed bands are getting more crowded, many licensed bands are highly underutilized. There exist a large number of spectrum holes (i.e., vacant multidimensional regions within frequency, time, and space) in licensed frequency bands. The spectrum scarcity problem is actually caused by the static and rigid frequency allocation, which fails to consider variations of spectrum use in both spatial and temporal domains. Motivated by the observations, dynamic spectrum access is introduced to enable unlicensed users to opportunistically utilize spectrum holes.

However, licensed users have a higher priority to use the licensed frequency bands. Unlicensed users may be hard to find a contiguous band of vacant spectrum, especially a wide band of vacant spectrum to support high speed transmission. An efficient way to utilize fragmented spectrum is to employ non-contiguous orthogonal frequency division multiplexing (NC-OFDM), which divides a band of spectrum into many closely spaced orthogonal subcarriers and transmits data on a subset of them in parallel. Since subcarriers that interfere with the licensed users are deactivated, licensed users are willing to open their licensed bands for additional return on investment. This dissertation thoroughly exploits subcarriers to address problems that emerge with dynamic spectrum access.

1.1 Spectrum Prediction

Dynamic spectrum access is made possible by recent development of cognitive radio (CR), which is an intelligent radio that can autonomously sense and learn from its surrounding environment and adaptively adjust its operating parameters to meet users' communication needs.

In a cognitive radio network (CRN), licensed users are called primary users (PUs) and unlicensed users are called secondary users (SUs). SUs are responsible for detecting spectrum holes and they must vacate the channel as soon as the PUs begin transmitting. A spectrum hole is defined as a time slot in which a licensed frequency band is not utilized by any PU at a specific geographic location. Since SUs need to check whether or not PUs are active, the information can be used to study the behaviors of PUs. If prediction models can be created, SUs will be able to reduce the frequency of checking medium state, increase spectrum hole extraction rate, and reduce interference to PUs.

To this end, we propose a spectrum occupancy prediction model to help estimate the availability of spectrum holes. OFDM is employed to measure a wide band of spectrum and subcarriers with similar spectrum use activities are grouped as one subchannel. In each subchannel, partial periodic pattern mining is employed to identify frequent patterns that may be irregular due to noise, sensing errors, and unpredictable behaviors. With the mined frequent patterns, future channel states (i.e., busy or idle) and the length of high/low utilization period are predicted. Based on the prediction, SUs are able to make use of spectrum holes more aggressively without introducing significant interference to PUs.

1.2 Spectrum Contention

As more than one device may have data to transmit and they may have the same prediction on spectrum holes, it is critical to address the contention problem between devices. If two devices start transmission at a similar time, a collision happens and both transmissions may fail. Therefore, it is important to have a medium access control (MAC) protocol to coordinate the medium access.

1.2.1 Single-Channel Contention

In contention-based wireless networks, carrier sense multiple access / collision avoidance (CS-MA/CA) is widely adopted to reduce collisions between contending nodes. Each node defers its transmission for a random number of slots to avoid collision. Each slot must be long enough to provide accurate medium state assessment and leave sufficient time for radio transmitting/receiving (TX/RX) mode switch.

The medium access overhead does not affect the efficiency significantly in low-speed transmissions. However, advancements in QAM, OFDM, MIMO, and channel bonding have increased the wireless physical layer (PHY) data rates by hundreds of times. The medium access overhead becomes a major factor that prevents high PHY data rates from being translated to commensurate throughput gains. For example, the transmission time for a 1500 byte packet in Wi-Fi has been reduced from 12 ms at 1 Mbps to 20 μ s at 600 Mbps in a dozen years. With more spectrum resources introduced by dynamic spectrum access, the actual transmission time of a packet will be further shortened. The throughput, however, can hardly be increased due to the medium access overhead. In 802.11n, the average medium access overhead is 101.5 μ s, which is five times of the 20 μ s data transmission time at 600 Mbps [1]. The high overhead is the reason why high PHY data rates cannot be translated to high throughput.

To improve the communication efficiency in high-speed wireless communications, we investigate the practical issues of collision detection in the frequency domain and introduce a binary mapping scheme to reduce the collision probability. According to the binary mapping, subcarriers are selectively deactivated to facilitate collision detection in the frequency domain. When a collision is detected, a bitwise arbitration mechanism is activated to try to make only one transmitter survive to data transmission. The throughput is significantly improved because of the low collision probability and fast arbitration for medium access. Because collisions are unlikely to happen, unfairness caused by capture effect of radios is also reduced. Further, the collision detection and bitwise arbitration (CDBA) can be set to let high priority messages get through unimpeded, making it suitable for real time prioritized communication.

1.2.2 Multichannel Contention

The CDBA focuses on the contentions between devices in a single channel. When multiple channels are available, a wideband device that tries to utilize multiple channels may be hard to win medium access opportunities if devices that use different channel widths coexist in a contention domain. Two narrowband devices may work independently in two different narrow channels. If the wideband device has to defer its transmission to the time when all of the narrow channels are idle, it has much fewer medium access opportunities than the narrowband devices. Unfair medium access contention prevents wideband devices from actually benefiting from wider channels.

To increase medium access opportunities for wideband devices, we propose an adaptive channel bonding (ACB) protocol, in which a wideband channel is viewed as an aggregation of multiple narrow channels that can be evaluated independently. A node is allowed to start transmission as long as there exist some idle narrow channels and gradually increases channel width during transmission whenever new narrow channels become available. The strategy gives a wideband device fair medium access opportunities in each narrow channel when contending with narrowband devices. In addition, a wideband device contends on subcarriers of each narrow channel with a different priority to increase the chance to win some narrow channels.

1.3 Spectrum Use

After contentions between devices are addressed and a device is granted the permission to transmit, we study new features and new problems introduced by modulation scheme diversity and spectrum availability diversity.

1.3.1 Modulation Scheme Diversity

As wireless networks move toward wider channels, frequency diversity can no longer be ignored. Because subcarriers span across a wide band of spectrum, it is common that different subcarriers experience different fade.

4

To cope with different channel conditions, each subcarrier may be modulated in a different way in data transmission. Since each subcarrier adopts different modulation schemes, the diversity itself can be used to represent useful information. We propose a modulation scheme coding (MSC) method that exploits the diversity to deliver more data than that can be carried by a modulation symbol on a subcarrier. In addition, the MSC helps improve decoding performance with the known bits represented by the modulation scheme codes. The throughput is improved by shortened transmission time and reduced bit error rate.

1.3.2 Spectrum Availability Diversity

Dynamic spectrum access encourages opportunistic use of spectrum. A device may have to utilize spectrum fragments instead of a contiguous band of spectrum. This complicates the communication because the combinations of spectrum fragments are unknown at the receiver. The receiver is unable to decode the transmission if it does not know from which spectrum fragment it should expect data. In addition, different receivers may obtain different spectrum fragments to use at different locations in broadcast-based communication. The broadcast needs to consider the diversity of spectrum availability at receivers.

To support broadcast under nonuniform spectrum availability, we propose a spectrum fragment agile broadcasting (SFAB) protocol. In SFAB, the sender is able to inform receivers of a broadcast intent and obtain receivers' spectrum availability efficiently through encoding specially designed sequences on subcarriers of each available narrow channel. With the collected information, the sender determines how to divide a packet into segments and map them to different narrow channels in a way that all receivers can obtain a complete packet. The SFAB significantly increases the broadcast efficiency because the sender does not need to broadcast multiple times in different narrow channels when there is no common channel to all receivers.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. Chapter 2 discusses the spectrum prediction and our spectrum occupancy prediction model based on partial periodic pattern mining. Chapter 3 presents our collision detection and bitwise arbitration mechanism that addresses contentions between nodes in a more efficient way than the widely adopted CSMA/CA. Chapter 4 introduces the adaptive channel bonding which enables wideband devices to contend with narrowband devices fairly. Chapter 5 studies the benefits of modulation scheme coding on transmission and Chapter 6 presents broadcast support under nonuniform spectrum availability. We conclude the dissertation and discuss future work in Chapter 7.

CHAPTER 2

WIRELESS SPECTRUM OCCUPANCY PREDICTION BASED ON PARTIAL PERIODIC PATTERN MINING

The rapid growth of wireless applications has resulted in a dense allocation of frequency bands. The traditional static frequency allocation does not consider variations of spectrum use in both spatial and temporal domains. Recent measurements have revealed that many licensed frequency bands are underutilized [2], whereas many services such as WLAN, Bluetooth, and ZigBee are confined to crowded unlicensed bands. To exploit the temporally unused spectrum chunks, the concept of Cognitive Radio (CR) is introduced. In a CR network (CRN), unlicensed users known as secondary users (SUs) sense the licensed channels and use the slots (called spectrum holes) that are unused by primary users (PUs). The dynamic spectrum access (DSA) increases the spectrum resources that are available to SUs, but SUs must immediately vacate the channels upon PUs' return [3].

Dynamic behaviors of PUs require accurate sensing and fast switch strategies to convince service providers to open their licensed bands. Due to the difficulty of accurate sensing for low power signals, FCC decided to adopt the geo-location and database access method, eliminating the spectrum sensing requirement for unlicensed TV band devices [4]. However, it is hard to determine the accurate TV signal reception range due to the tropospheric propagation [2]. In addition, some service bands (e.g., personal communication service bands and paging bands) have lots of short and frequent sessions, preventing the use of database access method, which may incur high overhead for frequent update. Moreover, even between SUs, spectrum sharing is a tough problem. Due to power asymmetry or different physical layer standards, some SUs may act as virtual PUs toward the other SUs. With spectrum occupancy prediction, CR devices can aggressively utilize spectrum holes and effectively avoid using time slots that have high collision probabilities. If the collision rate can be upper bounded by prediction-assisted DSA, more service providers will be willing to open their licensed bands for additional return on investment. Wireless communications are designed to conform to certain protocols. Patterns are expected to be observed in spectrum use. Recently, a Frequent Pattern Mining (FPM) method has been introduced to find frequent patterns and shows encouraging prediction performance [5]. To discover more useful patterns, this chapter introduces a Partial Periodic Pattern Mining (PPPM) based spectrum occupancy prediction (i.e., channel state prediction) model. Different from full periodic patterns, partial periodic patterns identify regularity of behaviors at some but not all points of time [6]. For example, a pattern disclosing that the channel utilization of a channel is high during certain hours every day but not regular in other time is a partial periodic pattern.

A spectrum occupancy pattern is affected by various factors such as noise, sensing errors, and irregular user behaviors. Therefore, the periodicity may be partially observable. PPPM algorithm is tailored for identification of real patterns that are irregular in nature. Using PPPM, more occupancy patterns are identified, leading to extra channel state prediction rules that can reduce the number of unpredictable slots.

Besides accurate prediction, timely estimation of spectrum availability is also critical. Hence, we speed up the mining process in two phases. First, for each candidate frequent pattern, we reduce the number of subsequences that need to be examined in a time series by introducing an index list structure. Using the index list structure, counting the occurrences of a pattern does not need to scan the entire database, improving mining expedition. Second, we reduce the number of candidate frequent patterns by identifying patterns that cannot yield longer frequent patterns so as to stop mining on them early. Moreover, by checking whether or not a pattern will be absorbed by another pattern, we avoid redundant mining on two branches that produce identical frequent patterns. Finally, we extend the prediction mechanism to predict the remaining time of high/low utilization in different channels. This helps to optimize channel switch strategies of CR devices. In summary, the contributions of this work are as follows.

• An efficient Partial Periodic Pattern Mining (PPPM) algorithm is proposed to mine rules for predicting the channel state in the next time slot.

- An index list structure along with an Apriori-like property and a backward-extension rule is introduced to speed up the mining process.
- An efficient channel utilization duration mining algorithm is developed for predicting the high/low utilization duration of a channel.

We compare our PPPM with an FPM model [5] and a hidden Markov model (HMM) [7] using real-world network activities and data collected in the Personal Communication Service (PCS) bands. The results show that the proposed PPPM-based prediction can predict more slots than the FPM-based prediction, indicating that many spectrum occupancy patterns exhibit partial periodicity only. PPPM also outperforms the fixed order HMM because it uses variable lengths of observations for prediction and it skips uncertain slots. In addition, we particularly observe that distinguishing low utilization periods from high utilization periods and mining rules in corresponding utilization periods will substantially improve the prediction performance.

In order to show the advantages of integrating prediction in DSA, we compare our predictionassisted DSA with a statistical knowledge-based DSA. The results show that the prediction of channel state significantly improves the **spectrum extraction rate**, which is defined as the ratio between the number of idle slots that are actually obtained by SUs and the total number of available idle slots left by PUs [8].

2.1 Related work

Various models have been developed to provide prediction for signal power, duty cycle, and channel state. We give a brief review on them in this section and also discuss related work on partial periodic pattern mining.

Several papers [9] [10] [11] model the variation of signal power in a channel. In [9], a secondorder autoregressive (AR-2) process is used to model the channel variation. Once the parameters of the AR-2 model are computed, the signal power can be predicted and the channel state can be estimated via a Kalman filter. In [10], a moving average model (MA) is introduced and it is integrated with the AR model to create an autoregressive moving average model (ARMA). The results show that the time series of all TV channels fall into the moving average model. An ARMA model requires that the time series is stationary, which means the statistical properties of a time series are similar to those of time shifted series [12]. An autoregressive integrated moving average (ARIMA) model is introduced in [12] to handle non-stationary time series. It models the change of band occupancy, which is defined as the ratio between the number of busy channels and the total number of channels in a given band. An inconvenience of these models is that a time series must be converted to a stationary and periodic time series before analyzing it.

Instead of predicting signal power, many papers consider that the channel is either detected as occupied or unoccupied and the channel states constitute a binary time series. Most studies assume that a Markov chain exists in such a binary time series. Assuming that the PU's traffic follows a Poisson process, a hidden Markov model (HMM) based DSA scheme is introduced in [7]. In [13], several deterministic traffic patterns are studied for the HMM-based channel state predictor. A more detailed analysis of the HMM-based predictor is presented in [14], where a multilayer perceptron (MLP) based channel state predictor is also analyzed. The existence of Markov chain in spectrum occupancy caused by PUs has been validated in [15] with data collected in the paging bands. Introducing one more state named "fuzzy" (unknown availability), a 3-state higher-order HMM channel state predictor is introduced in [16]. Instead of using a fixed length of observations, a variable order Markov model is introduced in [17]. Recently, a frequent pattern mining (FPM) based predictor [5] demonstrates that it provides better channel state prediction performance over the first-order HMM-based predictor. In this chapter, we show that PPPM can generate more prediction rules, leading to fewer unpredictable slots.

Mining patterns with gap constraints has been studied for sequence databases [18] [19] [20] [21]. Their methods, however, are not suitable for our problem. The number of entries in a sequence database is fixed. On the contrary, the number of sequences in a time series is changing with the length of a sequence. Partial periodic pattern mining in time series databases has been studied in [6] [22], but the algorithm aims at enumerating all partial periodic patterns of a single

period. It has to construct a complex max-subpattern tree for a single period. For a set of periods, the max-subpattern hit algorithm incurs high overhead for looping over the single period mining. In spectrum occupancy prediction, there is no need to enumerate all partial periodic patterns of a single period. Many of them are inappropriate for use because they include too many uncertain symbols. A pattern with a lot of uncertain symbols has very low prediction power. Therefore, we use a gap constraint (introduced in Section 2.2) to filter out useless patterns. We also introduce conditional entropy (in Section 2.3) to provide incremental partial periodic pattern generation, i-dentifying patterns of unknown lengths without using complex structures. We further study how to extract prediction rules from mined patterns and utilize them to improve communication efficiency.

The formalization of our problem uses terms (e.g., symbol, alphabet, pattern) that are used in language theory [23] [24], but our study is not to parse a sentence and derive the meaning. Our work is to check whether or not there exist frequent patterns in spectrum use even when signals of multiple devices are mixed.

In duration prediction, many studies focus on finding a good statistical model to specify the distribution of idle durations [5] [25] [26]. However, the assumed underlying distribution may not hold in unknown traffic and thus the parameters may be hard to estimate accurately. Moreover, the distribution may be different when different patterns are observed. Our mining recognizes the subtle changes of idle duration distribution in different periods.

2.2 Problem Formulation

In CRNs, SUs are responsible for sensing the channel before commencing any transmission so as to guarantee that the interference caused by them to PUs is under a certain limit. Multiple PUs can be viewed as a single virtual PU. If the spectrum use of the virtual PU exhibits some patterns, SUs can learn from the sensing results and utilize the frequent patterns to predict the availability of spectrum holes.

Let '1' denote busy and '0' represent idle. The channel states in a period can be represented by a binary time series. Let *I* be the alphabet of all possible values that occur in the time series, we

have $I = \{0, 1\}$. A wild card (denoted by a single *) is a special character that matches any value in *I*, which means it does not specify the channel state of a slot because the uncertainty in the slot is high. A **gap** is a sequence of wild cards, and the *size* of a gap is the number of wild cards in it. We use g^m to represent a gap whose size is within the range [0,m], which means the wild card * can be repeated at most *m* times in a gap. Here, *m* is the **gap constraint**. If a pattern has too many wild cards, many observation sequences can match the pattern. The prediction power of the pattern is weak. Therefore, we stop developing a pattern if it violates the gap constraint. Same as other PPPM algorithms [18] [19] [20] [21], it is a user-defined parameter in the mining algorithm.

A symbol in a pattern is a value in *I* or a wild card. Given a pattern *P*, we use P[i] to represent the *i*th symbol of *P* and p_i to represent the *i*th value of *P*. A pattern $P = p_1 g^m p_2 g^m \dots p_{q-1} g^m p_q$ is a set of values and gaps. We define the *length* of a pattern P, l = |P|, as the number of symbols in *P*, which means the wild cards are also counted toward the pattern's length. If $l \ge 2$, the substring that contains the first l - 1 symbols is called the *prefix* of *P* (denoted by *prefix(P)*), and the substring that contains the last l - 1 symbols is the *suffix* of *P* (denoted by *suffix(P)*).

A pattern $P = P[1]P[2]...P[l_1]$ is a *subpattern* of $C = C[1]C[2]...C[l_2]$ if $l_2 > l_1$ and there exists an integer $1 \le i \le l_2 - l_1 + 1$ such that $P[1] = C[i], P[2] = C[i+1], ..., P[l_1] = C[i+l_1-1]$. *C* is a *superpattern* of *P*. An example is illustrated in Fig. 2.1. Note that $\langle 1000 \rangle$ is not regarded as a subpattern of $\langle 11 * *0 \rangle$ because it violates the Apriori property in data mining (i.e., the support of a pattern cannot exceed the support of any of its subpatterns [27]). The pattern $\langle 11 * *0 \rangle$ may have more matches than $\langle 1000 \rangle$ in a time series and thus $\langle 1000 \rangle$ cannot be regarded as a subpattern of $\langle 11 * *0 \rangle$.

Our goal is to identify frequent patterns in a time series $S = s_1 s_2 \dots s_n$ where s_i is the *i*th value of *S*. We need to define the term *frequency* and how often a pattern should occur before we can consider it to be *frequent* in *S*. We define the *frequency* of a pattern *P* by the probability of observing *P* if we randomly pick a subsequence of length *l* in *S*. The time series *S* can be divided into N_l subsequences of equal length *l* that start at different starting points: $S_i = s_i s_{i+1} \dots s_{i+l-1}$ where $1 \le i \le n - l + 1$ and $N_l = n - l + 1$. A subsequence S_i matches a pattern *P* of length *l* if



Figure 2.1: Terms being used.

for each position $1 \le j \le l$, $s_{i+j-1} \in P[j]$ (a wild card matches any value). The **support** of *P* in *S*, sup(P), is the number of subsequences in *S* that match the pattern. The **confidence** of *P*, conf(P), is defined as the division of its support by the total number of subsequences of length *l* in *S*, which reflects the probability of observing *P* in *S*.

We say that a pattern is a *frequent partial periodic pattern* in a time series if its confidence exceeds a user-specified threshold ρ_c , which is defined as the **confidence constraint**. For example, if a pattern $P = \langle 0 * 1 \rangle$ appears 5 times in a time series $S = \langle 001100110001 \rangle$, the confidence of *P* is $conf(P) = sup(P)/N_l = 5/10 = 1/2$. If $\rho_c = 0.5$, *P* can be considered as a frequent pattern. We use confidence instead of support to measure the frequency because the supports of long patterns are usually lower than that of short patterns. This is because the total number of sequences of length *l* is monotonically decreasing with the increasing length *l*. The support is thus normalized by N_l to correctly reflect the frequency.

2.3 Partial Periodic Pattern Mining

Frequent pattern mining is to enumerate candidate frequent patterns and count the support of each pattern to determine whether or not a pattern is frequent. The general steps of mining include *generating candidate patterns, counting the support,* and *pruning candidate patterns.* To adapt to different applications, different techniques are designed to discover different useful patterns. This chapter introduces conditional entropy to determine when a wild card is needed in partial



Figure 2.2: An illustration of the frequent partial periodic pattern tree.

periodic pattern generation. An index list structure is designed to speed up support counting. In addition, an Apriori-like property is derived for confidence-based candidate pattern pruning and a backward-extension checking is introduced to further make the pruning process efficient.

2.3.1 Frequent Pattern Enumeration

The complete search space of frequent partial periodic patterns forms a tree as shown in Fig. 2.2. Starting at a root node that is labeled with Φ , we can recursively extend a node of level *L* on the tree by concatenating a value in *I* or a wild card that satisfies the gap constraint to get a child node on the next level *L*+1. Whether a parent node *Q* will develop a child node by concatenating a wild card largely depends on the conditional entropy H(X|Q) defined in Definition 2.

Entropy is a measure of the uncertainty inherent in the distribution of a random variable. When the entropy of a random variable is large, it implies that the uncertainty as to the value of that random variable is large. We hence consider it as a noise, but the noise should not affect the discovery of long patterns. We give the definitions of entropy and conditional entropy as follows.

Definition 1. The entropy of a discrete random variable X is defined as $H(X) = -\sum_{x \in I} p(x) \log_2 p(x)$

where p(x) is the occurrence probability of x and $I = \{0, 1\}$.

Definition 2. For a given node Q, the conditional entropy of a discrete random variable X after node Q is defined as $H(X|Q) = -\sum_{x \in I} p(x|Q) \log_2 p(x|Q)$ where p(x|Q) denotes the fraction of subsequences with prefix Q in S that match $P = Q \oplus x$ and \oplus denotes concatenation.

If the conditional entropy of a node H(X|Q) is higher than a user-defined threshold ρ_e , the node will develop a child node by extending itself with a wild card if both the gap constraint and the confidence constraint are not violated and at least one child with a value extension cannot exceed the confidence threshold. The ρ_e specifies how users define the uncertainty. In our case, if no value (i.e., either 0 or 1) appears after a pattern with a probability higher than 75%, we consider the next slot following the pattern is uncertain. Consequently, a child with the extension of * may be generated and it is the *collateral child* of the pattern. The other two children with a value extension are considered as the *lineal children* of the pattern. The children of a node are generated and arranged according to a lexicographical order. If a node is not frequent, we stop developing its subtree. To save memory, we grow a pattern in a depth-first search way so that we can remove a branch when it is done. Algorithm 1 summarize the mining.

The introduction of the collateral child has two advantages. First, channel sensing is not perfect. Misdetections will reduce the observed occurrences of a pattern. The wild cards can save some frequent patterns that are near the decision boundary. For example, two patterns $\langle 00011 \rangle$ and $\langle 00010 \rangle$ may occur 987 and 324 times, respectively. However, the confidence constraint requires that a pattern of length 5 must occur at least 1000 times to be considered frequent. If most occurrences of the two patterns are followed by a '0', we miss an important frequent pattern $\langle 0001 * 0 \rangle$ and any frequent pattern that has the prefix of $\langle 0001 * 0 \rangle$. Noise and sensing errors may make the number of occurrences of a pattern fall below the support threshold. Therefore, when the lineal children of a node cannot pass the confidence constraint while its collateral child can, we go on developing the subtree of this node so as to avoid missing any possible frequent pattern. Mining on the subtree will terminate soon due to the violation of gap constraint or confidence constraint. In the example, the support of $P = \langle 0001 * \rangle$ is as low as its parent pattern $\langle 0001 \rangle$ (i.e., 1311). If

Algorithm 1: Partial periodic pattern mining.

```
let C be a stack;
C.push(1);
C.push(0);
while C is not empty do
    P \leftarrow C.pop();
    P_0 \leftarrow P \oplus 0;
    P_1 \leftarrow P \oplus 1;
    P_2 \leftarrow P \oplus *;
    if P_0 or P_1 is infrequent and H(X|P) > \rho_e then
         if P<sub>2</sub> meets gap constraint then
             C.push(P_2);
         end
    end
    if P<sub>1</sub> is frequent then
     C.push(P_1);
    end
    if P_0 is frequent then
        C.push(P_0);
    end
end
```

we extend the P with a value in I, the support is reduced as it splits into two paths. Mining on the subtree will terminate in several levels when superpatterns fail to meet the confidence constraint. If we keep extending the P with wild cards, the support will not be reduced as a wild card matches both 1 and 0. However, we do not allow m consecutive wild cards if the gap constraint is m. Mining on the subtree will terminate in m levels due to the violation of gap constraint.

Second, even when one lineal child of a node is frequent, we may want to develop an additional pattern that accounts for the irregularity of behaviors. Suppose one of the two patterns described above passes the confidence constraint with some irregular slots (e.g., $\langle 00010 \rangle$ occurs 1328 times and $\langle 00011 \rangle$ occurs 964 times). The irregularity will be detected by the entropy of their parent (i.e., $H(X \mid 0001)$). An additional pattern $\langle 0001* \rangle$ is developed along with $\langle 00010 \rangle$. The reason is that the regularity of behaviors during a certain period of time may not be identified. For example, people make phone calls in a pure random manner. We may not be able to identify regular channel activities during certain periods of time but it is desirable to discover frequent patterns like $\langle 0001* \rangle$

***0001 so that we can predict channel states regardless of observed channel states in some time slots. These partial periodic patterns specify regularity of behaviors at some but not all points of time. Considering that partial periodicity exists ubiquitously in real life, more patterns can be discovered and more useful prediction rules can be extracted.

If all of the lineal children are frequent, we consider them as two different frequent patterns and do not generate the collateral child even though the entropy of their parent is higher than ρ_e .

2.3.2 Support Counting

For each candidate pattern, we have to check whether or not it is frequent by counting its support. This requires us to scan the entire time series S. Some algorithms traverse the entire time series as many times as needed [6] [21]. Since we perform depth-first search, we can use an index list structure to reduce the number of subsequences that need to be examined.

Given a time series *S* and a pattern *P* of length *l*, the *head index list* of *P* (denoted by HIL(P)) is a list of indices where the *P* appears in *S*. For example, if $S = \langle 00110100010 \rangle$ and $P = \langle 0*1 \rangle$, then HIL(P) = 1, 2, 8. Given HIL(P), it is easy to count the supports of patterns that have the prefix *P*. A pattern *Q* of length l - 1 has three children: $P_0 = Q \oplus 0$, $P_1 = Q \oplus 1$, and $P_2 = Q \oplus *$. Their

HILs can be compute	d by	y examining su	bsequer	ices that	t start at j	positions s	pecified b	y HIL	(Q).
---------------------	------	----------------	---------	-----------	--------------	-------------	------------	-------	----	----

Algorithm 2: Head index list construction.								
for $x \in HIL(Q)$ do								
if $(x+l-1) \le n$ then								
if $S[x+l-1] == 0$ then								
Insert x in $HIL(P_0)$;								
$sup(P_0) \leftarrow sup(P_0) + 1;$								
else								
Insert x in $HIL(P_1)$;								
$sup(P_1) \leftarrow sup(P_1) + 1;$								
end								
end								
end								

For each head index x in HIL(Q), if there exists a subsequence of length l that starts at x in S,

we check the last value of the subsequence. If the value is equal to '0', the head index *x* is inserted to $HIL(P_0)$; otherwise, the head index *x* is inserted to $HIL(P_1)$. The HIL of the collateral child P_2 is given by

$$HIL(P_2) = HIL(Q). \tag{2.1}$$

Since we do depth-first search, the length of the HIL is monotonically decreasing until we reach a pattern that is not frequent and we stop developing the subtree. The method speeds up the mining process quite a lot in practice because the number of subsequences that need to be examined is significantly reduced and a single character comparison is much faster than a string comparison.

2.3.3 Candidate Pattern Pruning

A common problem in data mining is that the number of candidate patterns is huge. There exist about 3^{l} candidate patterns for a certain length l in our problem. It is infeasible to enumerate all candidate frequent patterns and count their supports. In this section, we introduce two constraints for efficient candidate pattern pruning.

2.3.3.1 Confidence Constraint

For efficient pruning, most mining algorithms adopt the *Apriori* property, which states that any subpattern of a frequent pattern must also have the minimum support [27]. We can extend the *Apriori* property to the concept of confidence if the total number of sequences is fixed. However, the number of sequences of length *l* in a time series is changing with *l*. For example, in a time series $S = \langle \underline{00}1 \underline{00}1 \rangle$, a pattern $P = \langle 001 \rangle$ and its subpattern $Q = \langle 00 \rangle$ both have sup(P) = sup(Q) = 2, but $N_3 = 4$ and $N_2 = 5$. As a result, $conf(P) = \frac{2}{4}$ and $conf(Q) = \frac{2}{5}$, which shows that the confidence of a pattern may exceed the confidence of its subpattern. Therefore, we cannot prune a candidate pattern even though its confidence is lower than the user-specified threshold ρ_c ; otherwise, we are unable to discover some long patterns. To achieve efficient pruning, we derive an *Apriori-like* property in Theorem 1.

Theorem 1. Given a length-l pattern P and a length-m (m < l) subpattern Q = P[i]P[i+1]...P[i+m-1] of P, where $1 \le i \le l-m+1$, we have $conf(Q) \ge \frac{N_l}{N_m}\rho_c$.

Proof. Because Q is a subpattern of P, we have $sup(Q) \ge sup(P)$. If a length-l pattern P is frequent, then by definition, we have $conf(P) = sup(P)/N_l \ge \rho_c$. Now, consider a length-m (m < l) subpattern Q of P, we have

$$conf(Q) = \frac{sup(Q)}{N_m} \ge \frac{sup(P)}{N_m} \ge \frac{N_l}{N_m} \rho_c = \lambda_{l,m} \cdot \rho_c$$
(2.2)

Theorem 1 implies that any length-*m* subpattern *Q* of a frequent length-*l* pattern *P* must retain a confidence that is greater than or equal to $\lambda_{l,m} \cdot \rho_c$. This Apriori-like property allows us to prune a large number of candidate patterns from consideration. Suppose the length of the longest frequent pattern in the time series *S* is l_m . If the confidence of a length-*i* pattern is less than $\lambda_{l_m,i} \cdot \rho_c$, we can stop growing the pattern. This requires that the user has a rough idea about the value of l_m . We can guarantee that all frequent patterns of length less than or equal to l_m will be discovered. There exist some periodicity detection methods [28] [29] that can be used to determine the length of the longest pattern. Users can also estimate the channel state immediately, the rule set cannot be too large. A length of tens of bits is usually sufficient. Section 2.4 also introduces a duration mining method that can be used to estimate the l_m .

2.3.3.2 Backward-Extension Constraint

To further reduce the number of candidate frequent patterns that need to be examined, we identify patterns that can be absorbed by other patterns. Suppose P = P[1]P[2]...P[l] is a pattern of length lin a time series S. Given another pattern $C = p_0P$ where $p_0 \in I$, if sup(P) = sup(C), we say p_0 is a *backward-extension item* of P. If P has a backward-extension item p_0 , it can be absorbed by C. For example, if there is a '0' before any occurrence of '110', the pattern $\langle 110 \rangle$ can be absorbed by $\langle 0110 \rangle$ because any pattern developed under the tree of '110' has the prefix '0110'. It is redundant to develop both of them (i.e., '110' and '0110').

To check the backward-extension item of a pattern *P*, we extract the head indices from HIL(P)and examine whether the values located one symbol before all occurrences of *P* are the same value in *I*. Since $I = \{0, 1\}$, we use summation to check the condition.

Algorithm 3: Backward-extension check.	
$Cnt \leftarrow 0;$	
boundary \leftarrow False;	
for $x \in HIL(P)$ do	
if $x \ge 2$ then	
Cnt \leftarrow sum (Cnt, $S[x-1]$)	
else	<pre>/* reach boundary */</pre>
boundary \leftarrow True	
end	
end	
if $Cnt == 0$ or $Cnt == len (HIL(P))$ or $(Cnt == len (HIL(P)))$)-1 and boundary) then
<i>P</i> can be safely pruned;	
end	

As shown in Algorithm 3, when we count the support of a pattern *P*, we also add up the value that is one symbol ahead of each matched subsequence. If the sum is 0, it implies that there is a '0' before any occurrence of *P*. If the sum is equal to len(HIL(P)), it implies that there is a '1' before any occurrence of *P*. If there exists a x = 1 in HIL(P), we reach the boundary and we can regard that there is a '1' before any occurrence of *P* if the sum is equal to len(HIL(P)) - 1. In any of the three cases, the pattern *P* can be absorbed by another pattern and we skip developing the subtree of *P*. The pruning method is very efficient because if we use a simple candidate-maintenance-andtest method, in a case where $\langle 10 \rangle$ can be absorbed by $\langle 110 \rangle$, $\langle 110 \rangle$ must be mined before $\langle 11 \rangle$ as shown in Fig. 2.2. Therefore, a candidate-maintenance-and-test method will fail to remove the redundancy. On the contrary, in our method, we can discover that $\langle 10 \rangle$ can be absorbed by $\langle 110 \rangle$ ahead of $\langle 11 \rangle$. The redundant development of $\langle 10 \rangle$ can be avoided as desired.

$$R_{1}: \langle 01010 \rangle \rightarrow 1$$

$$R_{2}: \langle 0^{*}010 \rangle \rightarrow 1$$

$$R_{3}: \langle 010^{*}0 \rangle \rightarrow 0$$

Figure 2.3: Rules can be fused under certain conditions.

2.3.4 Channel State Prediction

The mining algorithm outputs frequent patterns and generates prediction rules at the same time. A prediction rule is defined as $P \Rightarrow C$, where *C* is a superpattern of *P*. The confidence of a rule is defined as $conf(P \Rightarrow C) = sup(C)/sup(P)$. For example, if $\langle 0001 \rangle$ appears 1000 times and $\langle 00010 \rangle$ appears 926 times, the confidence of the prediction rule $\langle 0001 \rangle \Rightarrow \langle 00010 \rangle$ is 92.6%, which declares that the channel state is predicted to be idle in the next slot with a confidence of 92.6% when channel states $\langle 0001 \rangle$ are observed. In the following discussions, we present a rule $P \Rightarrow C$ as $P \Rightarrow c$, where $C = P \oplus c$.

Both partial periodic patterns and full periodic patterns are discovered in PPPM. To improve prediction efficiency by reducing the number of rules, we examine whether or not a rule can be absorbed by another rule. Suppose two rules $P_1 \Rightarrow c$ and $P_2 \Rightarrow c$ yield the same prediction of c. If they have the same length, and $P_1[i] \in P_2[i]$ for each position i, and $conf(P_1 \Rightarrow c) \leq conf(P_2 \Rightarrow c)$, the first rule can be absorbed by the second rule. Fig. 2.3 shows an example. Note that the fusion cannot be performed if $conf(R_1) > conf(R_2)$; otherwise, if there exist a rule $conf(R_3) >$ $conf(R_2)$, we will incorrectly adopt the rule R_3 when $\langle 01010 \rangle$ is observed because R_3 has a higher confidence than R_2 . We should keep the rule R_1 so that we can get the correct prediction of '1' from R_1 instead of '0' from R_3 .

In the prediction phase we start with patterns of the longest length. If observed channel states match the pattern *P* in a rule $P \Rightarrow c$, we predict that the channel state in the next slot is *c*. Because a wild card * matches any value, it is possible that patterns from multiple rules are matched. In such

a case, we adopt the rule with the largest number of matched values. If several patterns have the same number of matched values, we adopt the rule with the highest confidence. We stop searching for rules of a shorter length once a prediction can be made. The reason is that a longer match usually leads to a more accurate prediction. A shorter pattern may provide a prediction with a higher confidence, but it includes contributions from multiple long patterns. For example, $\langle 010 \rangle$ appears whenever $\langle 1010 \rangle$ or $\langle 0010 \rangle$ appears. When channel states '1010' are observed, it is better to use the rule based on $\langle 1010 \rangle$ instead of $\langle 010 \rangle$.

If no pattern can match the observed channel states, we do not make any prediction and count it as a miss. We define **prediction accuracy** as the ratio between the number of correctly predicted slots and the total number of predicted slots. The ratio between the number of unpredictable slots and the total number of slots is defined as the **miss rate**.

2.4 Prediction of Duration

The channel state prediction predicts the channel state in the next slot. The prediction accuracy drops quickly when we try to predict more slots because we use predicted slots to perform further prediction. Therefore, channel state prediction is limited to answer whether or not the next slot is available. To view a bigger picture, we can make slot size larger. However, this sacrifices spectral efficiency as any activity detected in a slot marks the slot as occupied. We thus convert a binary time series of small slots to a duration sequence.

We count the number of consecutive ones and use a positive number to represent it. Similarly, we count the number of consecutive zeros and use a negative number to represent it. As values besides 0 and 1 are contained in the converted sequence, we extend our 3-state (i.e., busy, idle, uncertain) mining algorithm to an *N*-state mining algorithm as follows.

2.4.1 Duration Pattern Mining

Because any number may appear in the converted duration sequence, a specific pattern usually has a very low support. We cannot use a wild card that matches any duration to address the variation because two durations of a great disparity may belong to two different patterns (e.g., we cannot regard $\langle 3, -14, 7 \rangle$ and $\langle 3, -14, 170 \rangle$ as one pattern $\langle 3, -14, * \rangle$). Therefore, we define a variation threshold ε to aggregate similar sequences, making frequent patterns distinguishable quickly.

When we count the support for a pattern *P* of length *l*, we consider that a subsequence S_i matches *P* as long as the value difference in each position is less than ε . For example, if $\varepsilon = 1$, subsequence $S_i = \langle 3, -14, 7 \rangle$ matches the pattern $\langle 3, -14, 8 \rangle$. Due to the variation tolerance, several close values have the similar supports. As an example, when we are developing the subtree of a pattern $\langle 3, -14 \rangle$, we may find that the value of the next position can be either '7' or '8'. Because we count occurrences of '7' as occurrences of '8' and vice versa, $\langle 3, -14, 7 \rangle$ and $\langle 3, -14, 8 \rangle$ will have the same support. To avoid redundancy, we only develop the pattern with the largest actual number of occurrences. For example, if $\langle 3, -14, 7 \rangle$ appears 10 times and $\langle 3, -14, 8 \rangle$ appears 26 times, we only develop the pattern $\langle 3, -14, 8 \rangle$. The ε allows some patterns to pass the confidence threshold with one representative.

In duration pattern mining, we stop growing a pattern if its occurrences are less than min_sup . Because each value in a duration pattern represents a number of slots, even a short duration pattern can span across a long period of time. In a finite period of observation, duration patterns that can be observed multiple times are usually short patterns. Assuming a sufficiently large value (e.g., $l_m = 100$) for the length of the longest duration pattern, we have $min_sup \ge N_{l_m} \times \rho_c$, which ensures that the longest frequent pattern ($l \le l_m$) will be identified. The mining outputs all frequent duration patterns that are less than or equal to l_m . The sum of the absolute values in a duration pattern gives the period of a possible micro pattern. The largest sum can be used as the l_m defined in Theorem 1 for micro channel state pattern mining.

Algorithm 4 summarizes the duration pattern mining. The algorithm scans the duration sequence *S* to identify all length-1 frequent patterns. In the *i*th iteration, the algorithm generates a
length-(i+1) candidate pattern *P* by joining a length-*i* frequent pattern *Q* with a length-1 frequent pattern *p*. Because the depth-first search is used, the algorithm just checks the subsequences where the length-*i* pattern *Q* occurs. If the (i+1)th value s_{i+1} of a subsequence is within the variation threshold of *p* (i.e., $||s_{i+1}| - |p|| \le \varepsilon$), the subsequence is a match for *P*.

Algorithm 4	4:	Duration	pattern	mining.
	••	Daration	parcern	

```
scan S to find all length-1 frequent patterns;

F_1 \leftarrow the set of all length-1 frequent patterns;

let C be a stack;

for each pattern p in F_1 do

| C.push(p);

end

while C is not empty do

| Q \leftarrow C.pop();

for each pattern p in F_1 do

| P \leftarrow Q \oplus p;

if P is frequent then

| C.push(P);

end

end

end
```

2.4.2 State Duration Prediction

With the mined duration patterns, the remaining idle time of a channel can be predicted. A CR device can adapt its packet size according to the estimation so as to reduce short yet high interference upon PUs' return.

The duration pattern mining can also be used to predict high/low utilization duration. A CR device may want to know the remaining low utilization duration of its host channel so that it can plan its channel switch before the host channel becomes busy. A CR device may also want to know the low utilization duration of a candidate channel before it switches to the channel. To maintain statistical information about backup channels, a CR device can check backup channels periodically or use a secondary radio to sweep a large band of spectrum. Because two subsequent measurements in a backup channel can be several milliseconds or seconds apart, we cannot de-

tect activities between two subsequent measurements, but we can use the rough measurements to estimate the channel utilization.

Suppose a channel is considered to be good for use when the channel utilization is below a certain threshold δ . We calculate the channel utilization of a channel every minute. If the channel utilization is greater than δ , we output a '1'; otherwise, we output a '0'. The binary time series is then converted to a duration sequence and the duration patterns are mined as described above.

Because there are too many possible values following a duration pattern, we cannot predict a specific value with a high confidence. Therefore, we predict the longest duration whose confidence is higher than a threshold ρ_d . The confidence of a certain duration x is defined as the probability of obtaining such a duration, which is $P(X \ge x) = \sum_{|x_i| \ge x} P(X = x_i) = \sum_{|x_i| \ge x} p(x_i)$. For example, following $\langle 3, -14 \rangle$, '7' appears 3 times, '8' appears 60 times, and '9' appears 7 times. The probability of obtaining a duration of 9 is $P(X \ge 9) = 10\%$, the probability of obtaining a duration of 8 is $P(X \ge 8) = 96\%$, and the probability of obtaining a duration of $\langle 3, -14 \rangle$ because it is the longest duration whose confidence is higher than the ρ_d . Since a CR device is able to estimate the low utilization duration of a candidate channel, it can choose the channel with the longest low utilization duration to use. This reduces the frequency of channel switch.

2.5 Performance Study

In this section, we compare PPPM with an FPM model [5] to show that more prediction rules are extracted from the patterns mined by PPPM and thus more slots are predicted. We also compare PPPM with an HMM model [7] to demonstrate better prediction performance. Further, comparing PPPM-based spectrum access strategy with an optimal statistical knowledge-based spectrum access strategy [30], we show that the spectrum extraction rate is significantly improved with prediction.



Figure 2.4: Measurement setup.

2.5.1 Data Collection

There are various techniques for spectrum sensing including energy detection, matched filter detection, cyclostationary feature detection, covariance-based detection, and wavelet-based detection [16]. To detect spectrum holes, we employ the energy detection method because the method does not require the knowledge of signal features in a channel. The measurement equipment is the Universal Software Radio Peripheral (USRP), which is an popular hardware platform for building software radios using general purpose computing platforms.

Fig. 2.4 shows that a bandwidth of 4 MHz is continuously sensed. Because there are half-band filters implemented for digital down conversion (DDC) and anti-aliasing, these is a pretty gradual roll-off that causes strong curvature at two edges. To reduce the effect, we discard 0.5 MHz at both edges. Therefore, only the middle 3 MHz is recorded. We conducted measurements on the upper paging bands from 929 MHz to 932 MHz and part of the personal communication service (PCS) downlink bands from 1.981 GHz to 1.984 GHz, recording the maximum value every 20 ms, which results in 180000 values on each channel per hour. We also did measurements on the PCS bands from 1.965 GHz to 1.986 GHz.

2.5.2 Preprocessing

Due to the nonlinear frequency responses of various components in USRP1, no single noise threshold is appropriate for all channels in a band. We need identify a noise threshold for each channel. To compute noise thresholds, we first found a band of 4 MHz that is vacant and measured it for one



Figure 2.5: Average power measured at $f_c = 1.91$ GHz.

hour. We calculated the average power for each of the 128 channels (as 128-point FFT is used). Only the middle 97 channels' average power are collected as shown in Fig. 2.5 because 0.5 MHz are dropped at each edge to avoid the severe signal distortion. We also measured the same band for an hour by connecting the USRP1 to another USRP1 with a coaxial cable. There is no signal input. All measured power are electronic noises. The shape of the average power is similar to that in Fig. 2.5 with lower values. It proves that the 4 MHz spectrum band only includes white noise. The high power at the center frequency is caused by the DC offset, which is hard to be completely eliminated.

By examining the variations of the noise power on each channel, we observed that the maximum noise power will not exceed the average value by more than 4 dB on most of the channels as illustrated in Fig. 2.6. Therefore, we add 4 dB to the average value of each channel and use it as



Figure 2.6: Noise power variation at channel 60.

the noise threshold for each channel. Based on the noise thresholds, we get a binary time series of channel state for each spectrum channel of 31.25 kHz wide with a time resolution of 20 ms. Because the signal strengths measured in the PCS bands are strong, the effect of misdetection is less than that of false alarm. We did the same noise threshold identification at $f_c = 900$ MHz for signal power measured in the paging bands.

2.5.3 Macro Patterns

To mine duration patterns, we use the slowly changing signals measured in the paging bands for study. These signals appear for tens to hundreds of milliseconds and vanish for up to tens of seconds. The slow changing signals are used to simulate patterns that may span across a long period of time (e.g., utilization changes). One day's data are available for training and one day's



Figure 2.7: The duration prediction in the paging bands.

data are used for test. We check whether there exist frequent patterns and whether they can be identified with data collected in several hours. The prediction performance based on mined rules is presented in Fig. 2.7.

The prediction rules are as follows. When a pattern is observed, we predict that there is a length l_e of consecutive busy or idle states. If the actual length is l_t and it is equal to or greater than the predicted l_e , we count it as a correct estimation because there exists such a length (l_e) of consecutive busy or idle states. If the actual length l_t is longer than the predicted l_e , we count it as a nuderestimation rate is defined as $(l_t - l_e)/l_t$. If the actual length l_t is shorter than the l_e , we count it as an error and an overestimation with an overestimation rate of $(l_e - l_t)/l_e$. The underestimation and overestimation rates indicate the degree of errors; otherwise, prediction accuracy can be increased by preferring to predict shorter lengths. If no rule can be found, we count it as a miss and an error.

A pattern may span across several seconds or minutes. The number of occurrences of a pattern is small in a short period of observation. Therefore, only a few frequent patterns can be identified with data collected in several hours. Because the number of prediction rules is small, the miss rate is high. Since a miss is also counted as an error, the prediction accuracy is low. The underestimation rate and the overestimation rate are low because only a few predictions are made.

If the observation time is long enough, frequent patterns may have appeared many times and made them distinguishable. With data collected for more than 18 hours, most rules can be mined and the prediction accuracy becomes stable at 92% with an underestimation rate of 8%, an overestimation rate of 10%, and a miss rate of 5%. We can add more training data, but the impact is trivial as shown in Fig. 2.7.

Although we need data of at least 18 hours here, only the short duration sequence instead of the long binary time series is kept. The memory requirement is not high. The duration pattern mining gives us a way to predict the idle duration of a channel. It is also useful for discovering the maximum pattern length of micro patterns. Any macro pattern $\langle a, b, c \rangle$ indicates the existence of a micro pattern of length l = |a| + |b| + |c|. By examining macro patterns, we can set an appropriate value for l_m that is needed in Theorem 1.

2.5.4 Micro Patterns

To get real-world network activities, we use the network traffic trace collected in the SIGCOM-M'08 [31] as a reference. We divide the time into slots of 20 ms and convert the trace into a binary time series of channel states: if channel activities are detected in a slot, we output a '1'; otherwise, we output a '0'. There is a beacon every 100 ms and thus 5 bits cover a beacon interval. Depending on the application, the slot size may vary. When a CR device utilizes a spectrum hole, it may expect that the slot is long enough for at least one transmission. Therefore, from a practical point of view, an entire slot may be considered busy if channel activities are detected. The duration pattern mining shows that it is sufficient to set $l_m = 30$ in micro pattern mining, which covers six beacon intervals.



Figure 2.8: The channel state prediction results under different rule confidence constraints. The total accuracy takes all missed slots as wrong predictions.

To investigate the prediction accuracy and the miss rate, we use one day's data for training and one day's data for test. Fig. 2.8 summarizes the prediction results under different rule confidence constraints (R_c). In order to bound the prediction accuracy, only rules with confidences that are higher than R_c are used. Generally, PPPM reduces the miss rate by around 5% ~ 9% with a sacrifice of about 1% on prediction accuracy. The lower miss rate implies that more useful rules are extracted from the patterns mined by PPPM, which validates that some patterns exhibit partial periodicity only and they cannot be discovered by FPM.

Fig. 2.9 shows some frequent pattern examples. The 1st example is a full periodic pattern. The 2nd example shows that the beaconing is quite regular regardless of channel activities. However, FPM may miss these patterns as they have uncertainty in some slots. The 3rd example shows that the beaconing may not occur exactly every 100 ms. It may be delayed by some activities. If this does not happen often, FPM may not consider the offsetted beaconing as a frequent pattern. When $\langle 0000100000 \rangle$ is observed, there is no rule for prediction. In PPPM, a partial periodic pattern $\langle 000010000 \times 1 \rangle$ may provide a correct prediction.

Fig. 2.8 also shows that the miss rate is significantly reduced when rules with lower confidences

P 00001000010000110	S0	0.939169	len 17
P 00001 <u>1010</u> 1 <u>1***</u> 1 <u>***0</u>	S1	0.765957	len 19
P 00001 <u>0000</u> *10***	S1	0.814815	len 15

Figure 2.9: Examples of frequent patterns.



Figure 2.10: The channel state prediction results in the Personal Communication Service (PCS) bands.

are used. It implies that most rules have confidences between 0.7 and 0.9. The prediction accuracy thus will be pulled down to around 80% if most slots are predicted. Because the unpredictable slots cannot be utilized, we may define the accuracy as the ratio between the number of correctly predicted slots and the total number of slots (instead of predicted slots). HMM uses a fixed length of observations for prediction and it always gives a prediction. If the probability of obtaining 1 is higher than that of 0, the channel state is predicted to be busy; otherwise, the channel state is predicted to be idle. The corresponding **total accuracy** shows that PPPM is the best as it masks the irregularity with wild cards, whereas both FPM and HMM try to capture the channel state in every slot.

Fig. 2.10 summarizes the prediction results in the PCS bands. Different from Wi-Fi channels, there is no explicit periodicity of channel activities due to measurement granularity and different

spectrum access strategies. Few rules have confidences of prediction higher than 0.9 and we have to loosen the rule confidence constraint. The miss rate is reduced by around 40% if rules with confidences between 0.7 and 0.8 are allowed to use. This indicates that there exist some patterns in activities that seem to be random. They are just obscure and hard to identify. PPPM shows its advantage in discovering patterns in real-world traffic. The miss rate of PPPM is $16\% \sim 18\%$ lower than that of FPM. The total accuracy of PPPM is higher than that of FPM because PPPM predicts many more slots with only a slight drop on accuracy.

2.5.5 Impact of Utilization Change

The major reason for prediction errors and high miss rates is the changing utilization, which is defined as the ratio between the number of slots that are utilized by PUs and the total number of slots. A channel may be busy in some periods of time and is underutilized in the other periods of time. Using all data together makes the prediction power of patterns weak.

The same pattern *P* may have a high probability to be followed by a '1' during high utilization periods but may have a low probability to be followed by a '1' in low utilization periods. If we mix data of high and low utilization periods, pattern *P* may not be able to give any prediction, causing high miss rate. Another possibility is that, in a global view, we may get a rule $P \Rightarrow 1$. In low utilization periods, however, *P* is more frequently followed by a '0' rather than a '1'. The rule $P \Rightarrow 1$ results in many prediction errors during low utilization periods. Therefore, rules mined in high utilization periods may not be applied to low utilization periods and vice versa. Mixing them can cause errors in their counterparts. It is desirable to distinguish low channel utilization periods from high channel utilization periods and mine rules in corresponding utilization periods.

Because the patterns observed in high utilization periods are different from the patterns observed in low utilization periods. We divide the SIGCOMM trace into two sets based on channel utilization in every 10 minutes. If the channel utilization is above a certain threshold δ , we accumulate data in the high utilization set. If the channel utilization drops below δ , we accumulate data in the low utilization set. Rules mined from the two sets are used to predict channel states



Figure 2.11: The channel state prediction results on different channel utilization periods ($\delta = 0.4$). Either prior day's data or the same day's data are used for training.

in corresponding utilization periods. With $R_c = 0.7$, the prediction performance is summarized in Fig. 2.11. In low channel utilization periods, the regularity of spectrum occupancy is easy to identify (there is a beacon every 100 *ms*). The prediction performance is much better than that in high utilization periods.

In high utilization periods, the behaviors of a single device are easy to predict as wireless communications are designed to conform to certain protocols. When multiple devices are contending, the regularity becomes obscure as each device acts in response to others. Since the aggregate behaviors of multiple devices may not repeat a pattern perfectly, PPPM outperforms FPM by identifying partial periodicity in spectrum use. The miss rate is reduced by $9\% \sim 12\%$ in high utilization periods.

When we use one day's data to predict channel states in the next day, the spectrum occupancy patterns may have changed. We thus change to do 10-fold cross validation on one day's data. There is a slight improvement as demonstrated in Fig. 2.11. The results show that the spectrum occupancy patterns are similar for the same wireless service. We may not need to perform pattern mining frequently and we can perform mining offline. A potential system structure is to use a

central unit to collect sensing results from SUs and create the model. SUs contribute sensing results and periodically get model update from the central unit. If each SU prefers to perform channel modeling independently, we study whether a small training data set is sufficient for the modeling.

2.5.6 Impact of Training Data Set Size

To make the mining practical on memory-constrained devices, we study whether we can perform mining on the data collected in the latest short period of time and predict channel states in the next short period of time. The method adapts to transient channel state changes. Mining on one day's data will take hours to finish, but micro patterns only span across tens to hundreds of milliseconds. A few seconds of observations may be sufficient to discover micro patterns that occur frequently.

Table 2.1 shows that it takes milliseconds to mine rules on data collected in the latest several seconds. The mining has been expedited by several components in the design as introduced before.

First, when counting the support of a pattern, only subsequences that are indexed by HIL are examined and only the last character of each subsequence is checked. Suppose the support of a pattern *P* is sup(P). The overhead of counting the supports of all its children patterns is O(sup(P)) as the Algorithm 2 runs once to obtain the supports of all children patterns.

Second, many candidate patterns are ruled out by checking the Apriori-like property and the backward-extension. The FPM uses less time to finish mining because it does not need to develop partial periodic patterns. However, this also leads to fewer prediction rules. For the same data, HMM needs seconds for the convergence of the transition probability matrix. The computations are more complex and consume more time than simple character comparisons.

Note that we identify a frequent pattern based on the probability of observing it when randomly pick up a sequence of the same length in the observations. Therefore, the support threshold is dynamically changed with the training data set size. Even with a short period of observation, we are able to identify a frequent pattern even though its number of occurrences is small. However, some patterns may be missed. Therefore, we need to evaluate the time complexity along with the

training set size	FPM	PPPM	HMM
1 s	0.01 ms	0.01 ms	2.22 s
2 s	0.03 ms	0.04 ms	4.62 s
3 s	0.04 ms	0.08 ms	7.29 s
5 s	0.06 ms	1.1 ms	12.74 s
10 s	1.2 ms	2.6 ms	30.23 s
20 s	4.1 ms	9.3 ms	69.76 s
30 s	10.5 ms	21.3 ms	103.73 s

Table 2.1: Time complexity with different training set sizes (2.3 GHz CPU 4G RAM Laptop)



Figure 2.12: Prediction accuracy with different training set sizes.

resulted prediction accuracy.

Fig. 2.12 shows that with data collected in 10 seconds in Wi-Fi traffic, the prediction of PPPM becomes stable. The average time consumed to do mining is around 3 ms. Therefore, it is affordable to repeat mining on data collected in the latest short period of time to keep up with channel state changes. The similar discoveries are also observed in the PCS bands.

2.5.7 DSA with Prediction

We study how dynamic spectrum access can benefit from the prediction. When licensed bands are open for SUs' use, the PUs often impose a collision probability constraint η . In our PPPM-assisted DSA, we use the prediction rules mined by PPPM to estimate the collision probability. When a sequence of channel state observations matches a frequent pattern, the decision of channel access is made based on the prediction. If the prediction is busy, we do not access the channel in the next slot even if the channel is sensed to be idle at the beginning of the slot. On the other hand, we utilize the next slot if the collision probability is below the collision constraint η and the channel is detected to be idle at the beginning of the slot. We compare the performance of our PPPM-assisted DSA with an optimal DSA introduced in [30]. The optimal DSA is a statistical knowledge based access (SKA) strategy that estimates the risk of accessing the channel based on the probability density function (PDF) of idle durations $f(\cdot)$, the cumulative distribution function (CDF) of idle durations $F(\cdot)$, and the PU collision probability constraint η . Study in [8] again demonstrates that the SKA strategy improves the spectrum extraction rate by 2-3 times over no knowledge based access with data measured in various bands.

In previous work, the PDF and the CDF are calculated from a global view of the entire training set. The subtle variations of PDF and CDF in different periods are overlooked. In light traffic periods, the probability of having a long idle duration is larger than that in periods of slightly heavier traffic loads. The CDF obtained in a global view only reveals the synthesized effect.

In PPPM, frequent patterns are identified. The confidence of predicting '0' and '1' is no longer given based on a global view of the idle duration distribution. Instead, it is given based on latest observed patterns. Each prediction is updated with the channel state that is just observed in the last slot. As shown in Fig. 2.13, PPPM-assisted DSA can utilize more idle slots with a slight increase on the collision rate. PPPM improves the spectrum extraction rate by 50% over SKA when the constraint of collision probability is tight (i.e., 10%) in Wi-Fi. The reason is that SKA is too conservative for a tight collision constraint. PPPM, however, opportunistically utilizes some slots that have high collision probabilities in a global view but safe for use in low utilization periods.



Figure 2.13: The spectrum extraction rate is significantly improved with PPPM-assisted prediction.

For example, the probability of having an idle slot after a beacon may not be high in a global view because transmissions may occur after a beacon. A SU cannot utilize the slot if the collision constraint is tight. However, in PPPM, if a pattern $\langle 0000100001 \rangle$ is observed, the SU can be more aggressive because the channel is in a low utilization period. On the other hand, if a pattern $\langle 0110110101 \rangle$ is observed, the SU will be more conservative for using the slot after a beacon.

The significant improvement on the spectrum extraction rate supports the importance of identifying frequent patterns if there exists regularity in spectrum use; otherwise, even though there are many underutilized frequency bands, the actual number of spectrum holes that SUs can utilize are few due to the tight collision probability constraint.

2.6 Summary

In this chapter, a spectrum occupancy prediction model based on Partial Periodic Pattern Mining (PPPM) is introduced. The mining takes into account the irregularity of spectrum use and thus is more suitable for real-world traffic. The proposed PPPM algorithm combines the gap-constrained

pattern growth, the head index list structure, the Apriori-like property, and the backward-extension pruning to achieve fast and reliable partial periodic pattern mining. To estimate channel utilization duration, the three-state mining algorithm is extended to an N-state mining algorithm.

The partial periodicity in spectrum use and the performance of PPPM are validated with Wi-Fi network activities and data collected in the PCS bands. PPPM extracts more channel state prediction rules, leading to a significant reduction on miss rate in spectrum occupancy prediction in comparison with traditional FPM. PPPM also outperforms HMM for being more robust to irregularity. We particularly observe that distinguishing low utilization periods from high utilization periods and mining rules in corresponding utilization periods will substantially improve the prediction performance.

Although there are many underutilized spectrum bands, the actual number of spectrum holes that SUs can utilize are few due to the PUs' tight interference constraints. We compare the PPPM-assisted DSA with a statistical knowledge based DSA to demonstrate that the prediction of channel state significantly improves the spectrum extraction rate without introducing significant interference to PUs. We also demonstrate that the prediction of high/low channel utilization duration reaches an accuracy of 92% with a miss rate of 5% using data collected in the paging bands.

With the spectrum prediction, the interference to PUs is reduced. However, all nodes have the same prediction on spectrum holes. When multiple nodes have data to send, it is important to coordinate their medium access with medium access control (MAC) protocols. The next two chapters address the medium access contention issues.

39

CHAPTER 3

COLLISION DETECTION AND BITWISE ARBITRATION IN MULTICARRIER WIRELESS NETWORKS

The previous chapter discusses how to identify spectrum holes. In these spectrum holes, a medium access control (MAC) protocol is needed to coordinate medium access between contending transmitters. However, due to the fast increasing physical layer (PHY) data rates, the overhead incurred by current MAC protocols has risen from a minor factor to a major factor that affects the efficiency of contention-based wireless communication. Taking Wi-Fi as an example, the transmission time for a 1500 byte packet has been reduced from 12 *ms* at 1 Mbps to 20 μ *s* at 600 Mbps in a dozen years. The transmission time will be further shortened with higher Gbps PHY data rates supported in ongoing standardization of 802.11ac and 802.11ad. The average medium access overhead, however, stays at the same order of magnitude as before. In 802.11n, the average medium access overhead is 101.5 μ *s*, which is about 5 times of the data transmission time at 600 Mbps [1]. Although frame aggregation can reduce the relative impact of medium access overhead, it does not work well for non-bulk data flows or traffic that has stringent deadline requirements (e.g., short HTTP flows and VoIP packets).

In contention-based wireless networks, carrier sense multiple access / collision avoidance (CS-MA/CA) is widely adopted to reduce collisions between nodes. Each node defers its transmission for a random number of slots to avoid collisions and each slot must be long enough for accurate detection of channel state. The random backoff is necessary because collision detection was thought to be impossible in wireless communication. The reason is that even if a node could listen on the channel while transmitting, the strong self-interference would mask all other signals on the air. However, recent advances in self-interference cancellation [32] [33] [34] have made full duplex wireless communication possible. This allows transmitters to parallelize transmission and detection. By designing different preambles before data transmission, different contention resolution schemes have been proposed.

WiFi-Nano [1] relies on correlation to compute the start time of a preamble transmission. It still needs random backoff to spread out nodes' transmissions but the random backoff time is shortened by reducing the random backoff slot duration to 800 *ns*. Although the adoption of tiny time slots in random backoff helps reduce medium access overhead, the random backoff can no longer de-synchronize hidden terminals and they may keep colliding with each other. In addition, a weak signal can be discovered only after all stronger transmissions are aborted, but transmissions initiated in the same time slot are aborted probabilistically. Therefore, the time used to resolve a collision is uncertain and it is possible that all nodes abort their transmissions in a contention.

Full duplex also motivates researchers to migrate random backoff from the time domain to the frequency domain [35] [36] using the emerging non-contiguous OFDM (NC-OFDM). Current frequency domain backoff designs make a node occupy only one randomly selected subcarrier for preamble transmission. Contending nodes learn each other's random number and backoff accord-ingly. In practice, the collision probability is high because only a few subcarriers are available for contention resolution and there is no complementary mechanism like binary exponential backoff (BEB) that can be used to address the issue.

In this chapter, we investigate the practical issues of collision detection (CD) in the frequency domain and introduce a binary mapping scheme for preamble design. The binary mapping scheme ensures that the probability of choosing the same subcarriers by two nodes is low and thus a collision between them can be easily detected. When a collision is detected, a bitwise arbitration (BA) mechanism is devised to resolve the collision within a short bounded period of time. When multiple nodes initiate arbitration preamble transmission at a similar time, usually only one node will be granted the permission to start data transmission at the end of the arbitration phase. Because collisions are expected to be resolved before data transmission, the benefits are two folds. First, the throughput is improved as the channel is not occupied by collided transmissions. Second, unfairness caused by capture effect of radios is reduced since collisions are unlikely to happen. In addition, the CDBA enables message prioritization. Nodes with higher priority messages can obtain more medium access opportunities. This feature makes CDBA suitable for real time prioritized

communication. In summary, we make the following contributions:

- We investigate practical issues of collision detection in the frequency domain, devising methods that make collision detection reliable and practical.
- We introduce a binary mapping scheme for arbitration preamble design to provide low collision probability.
- We introduce a bitwise arbitration mechanism that resolves collision before data transmission with much lower overhead than that incurred by collision avoidance.
- We introduce a way to prioritize messages so that nodes with higher priority messages can obtain more medium access opportunities.
- We integrate these mechanisms as a complete system and prototype it on the USRP / GNU Radio platform, validating its correctness and efficiency through both experiments and extensive simulations.

The rest of the chapter is organized as follows. Section 3.1 summarizes related work. Section 3.2 presents the motivation and Section 3.3 presents an overview of the architecture design. The design details are presented in Section 3.4. The performance of CDBA is evaluated through both experiments in Section 3.5 and extensive simulations in Section 3.6. We conclude this chapter in Section 3.7.

3.1 Related Work

Full duplex wireless [32] [33] studies how to eliminate self-interference so that a node can transmit and receive simultaneously. Current solutions include nulling antenna [33], analog interference cancellation [37], digital interference cancellation [38] [39], and a combination of them [32]. They are efficient when two nodes have data for each other. However, if all communications are in one direction, MIMO performs better because it doubles throughput with two antennas. The proposed CDBA may use two antennas for collision probe and medium access arbitration, but once the arbitration ends, both antennas are available for data transmission and thus it is compatible with MIMO.

As full duplex wireless communication becomes practical, CSMA/CN [40] lets the receiver notify the transmitter of any collision. The collision notification is the receiver's unique signature. CSMA/CN shows that by continuously correlating the receiver's signature with the incoming signal, a node can detect the signature even under strong self-interference. Because the collision notification is not specific for a particular transmitter, all transmitters that are transmitting to the receiver have to abort their transmissions. In CDBA, transmitters gradually abort their transmissions but at least one node will survive and proceed to data transmission.

It has been shown that correlation is able to identify a known pattern in an incoming signal under strong interference [40]. WiFi-Nano [1] reduces the backoff time slot in CSMA/CA to 800 *ns*. Because nodes can no longer avoid collisions with such a short random backoff, a node must listen on the channel while it is transmitting the correlation preamble. Through correlation, a node is able to detect other preamble transmissions while it is transmitting its own preamble. They calculate which node starts transmission first and backoff accordingly. Because a weak signal can be discovered only after all stronger transmissions are aborted, a node needs some time to detect the earliest transmission because colliding transmitters abort their transmissions probabilistically. It is possible that all transmitters abort their transmissions and a new contention begins. To avoid persistent collisions between hidden terminals, the channel state may have to be set to busy when a correlation preamble is detected. Channel utilization is decreased when all transmitters aborted their transmissions.

Back2F [36] proposes a backoff mechanism in the frequency domain for access points. An AP maps its selected random number to a subcarrier and APs learn each other's random number in the frequency domain. Due to power leakage, the available subcarriers that can be used for contention resolution is very few. Suppose only 8 subcarriers are available, only 8 random numbers can be used. The probability that two nodes choose the same number is high. Increasing FFT

points with more samples does not solve the power leakage problem, and it increases the time for collecting samples. In addition, increasing the number of subcarriers decreases the subcarrier spacing, making Back2F more sensitive to frequency offset.

3.2 Motivation

This section presents the motivation of CDBA by highlighting the limitations of collision avoidance (CA) in 802.11 and high collision probability in current collision detection (CD) proposals. Although Wi-Fi is used as an example, the idea can be generalized to other multicarrier wireless networks.

3.2.1 802.11 Medium Access Overhead

The fundamental medium access method of IEEE 802.11 MAC is a distributed coordination function (DCF) that relies on the CSMA/CA. Due to physical constraints (e.g., processing delay, TX/RX turnaround time), a gap is inserted between contiguous frames in CSMA/CA. A node thus must sense that the medium is idle for at least such a length of duration before attempting to transmit. As shown in Fig. 3.1, the CSMA/CA algorithm mandates that a node must verify that the medium is idle for a duration of DIFS (DCF interframe space), which comprises a SIFS (short interframe space) and 2 slots. After the DIFS, a node shall defer its transmission for a random number of slots. The random number is drawn from a uniform distribution over the interval [0,CW], unless the random number generated in the last contention has not been decreased to 0. It is possible that two nodes defer their transmissions to the same slot, leading to a collision. The two nodes are unaware of the collision until they complete their transmissions. Without a CD mechanism, they can only infer a collision from their unacknowledged data packets. To reduce the collision probability, the CW takes an initial value of aCWmin and then is increased exponentially every time an unsuccessful transmission occurs, until the CW reaches the value of aCWmax. The CW cannot be reset to aCWmin unless the node successfully transmits a packet. The process makes the



Figure 3.1: 802.11 DCF.

Table 3.1: OFDM PHY characteristics

aSlotTime	9 µs	aSIFSTime	16 µs
aRxTxTurnaroundTime	$< 2 \ \mu s$	aCCATime	$< 4 \ \mu s$
aAirPropagationTime	$\ll 1 \ \mu s$	aCWmin	15
aMACProcessingDelay	$< 2 \ \mu s$	aCWmax	1023

CW vary with the load, quickly switching to a large size when the channel contention is severe.

Table 3.1 summarizes the parameters defined in IEEE 802.11 standard [41]. To ensure accurate detection of busy state and leave enough time for radio receive-transmit switch, the slot size defined in IEEE 802.11 standard is close to the minimum feasible value [1]. Even if there is no collision and the CW stays at the aCWmin, the average backoff time incurred by CSMA/CA is aDIFS + aCWmin/2 * aSlotTime = aSIFSTime + (2 + aCWmin/2) * aSlotTime = 16 μs + (2 + 7.5) * 9 μs = 101.5 μs for the OFDM PHY, but the data rate with the OFDM PHY can reach 600 Mbps in 802.11n, leading to a transmission time of 20 μs for a 1500 byte packet. The medium access overhead becomes a major factor that causes the inefficiency of Wi-Fi.

3.2.2 High Collision Probability in CD

With the emerging full duplex wireless communication and NC-OFDM, it is possible to achieve collision detection in the frequency domain. OFDM divides a band of wireless spectrum into multiple narrowband channels known as subcarriers. Each subcarrier carries a data stream in parallel with others. With NC-OFDM, a node can map data streams to a subset of subcarriers. A node can

easily detect a collision if it observes activities on subcarriers that are not used by itself. Based on the idea, recent CD proposals [35] [36] let a node transmit on one randomly selected subcarrier. Contending nodes learn each other's random number in the frequency domain and the node with the largest random number wins. In practice, however, only a few subcarriers are available for contention resolution. The probability that two nodes choose the same subcarrier is high. As a result, the collision cannot be detected and both nodes will start data transmission. Therefore, although medium access overhead is reduced, retransmission overhead may be increased due to collisions.

In this chapter, we introduce a binary mapping from a random number to active/inactive subcarriers to decrease the collision probability. The winner in a contention is selected through a bitwise arbitration mechanism, which forces low priority nodes to abort their transmissions so that only one node survives and proceeds to data transmission. The bitwise arbitration is very efficient because it ends immediately when the collision is resolved. The arbitration phase is on average shorter than the designed length.

3.3 Architecture and Design

The proposed CDBA precedes each data transmission with a collision probe phase and an arbitration phase. However, the bitwise arbitration is activated only when a collision is detected in the collision probe phase; otherwise, the arbitration phase is skipped and the data transmission is initiated immediately after the collision probe. This is optimized for single transmitter scenarios. The data transmission uses the same PHY convergence procedure defined in 802.11 [41] to estimate the channel impulse response and indicate the start of a data frame.

3.3.1 Arbitration Preamble Design

To make nodes be able to detect collisions, we let nodes generate arbitration preambles that are distinguishable from each other in the frequency domain. The method is to let a node draw a random number from a uniform distribution over $[1, 2^k - 1]$ before each contention. The random number



Figure 3.2: Collision detection in the frequency domain.

represents the node's priority in this round of contention. Here k is the number of subcarriers and the probability that two nodes choose the same number decreases exponentially with the k. Because a node can use subcarriers selectively in NC-OFDM, we use a k bit binary code to represent the random number and the binary code is mapped to subcarriers with '1' indicating active and '0' indicating inactive as shown in Fig. 3.2.

To implement the binary mapping, a node simply feeds 0 instead of modulation symbols (e.g., 1+0i, -1+0i in BPSK) to the inactive subcarriers, leading to zero power on them. We design an arbitration preamble as one OFDM symbol, which consists of *k* samples when using a *k*-point inverse fast Fourier transform (IFFT) algorithm. IFFT is employed by OFDM to efficiently modulate multiple subcarriers at the same time. Suppose a 64-point IFFT is used to modulate 64 subcarriers. The output of the IFFT consists of 64 samples that are referred to as one OFDM symbol. The

arbitration preamble incurs low overhead because it takes only 3.2 μs to transmit the 64 samples at 20 MHz in Wi-Fi.

3.3.2 Self-Interference Cancellation

We expect that a node is able to receive signals that are transmitted by other nodes during its own transmission in the same frequency band. Recent studies on self-interference cancellation have made significant progress on full duplex wireless communication [32] [33] [34]. Even with one antenna, a node is able to transmit and receive simultaneously [42]. Our MAC protocol only requires that the self-interference cancellation is sufficient to prevent ADC saturation.

A 14-bit ADC has a theoretical dynamic range of 86 dB where the dynamic range is calculated as $DR(dB) = 6.02 \times n + 1.76$ for n = 14 [42]. The dynamic range indicates the ability to accurately measure weak signals in the presence of strong signals. Considering a noise floor of -90 dBm (1 picowatt) in Wi-Fi, the strongest signal that can be input to the radio should not exceed -4 dBm so that the weak signals can be measured simultaneously. The highest transmission power on a Wi-Fi 2.4 GHz antenna, however, can be around 30 dBm [42], which is strong enough to bury other weak signals. This is why half-duplex communication is long assumed in wireless system designs.

Recent self-interference cancellation design reports at least 45 dB cancellation even with one antenna for simultaneous TX and RX [42]. It is sufficient to prevent ADC saturation as shown in Fig. 3.3. A better self-interference cancellation scheme may provide real full-duplex wire-less communication through antenna separation or other methods. In our design, a lightweight self-interference cancellation algorithm is sufficient because each node sets some subcarriers to inactive, leading to zero power on them. As long as the ADC is not saturated, a node can detect signals on inactive subcarriers if there is a collision. The CDBA design does not impose stringent requirements on self-interference cancellation.

Some self-interference cancellation designs [32] [33] may require TX/RX antenna separation, but in our design all antennas are available for data transmission once arbitration ends. As MIMO is a trend in wireless communication, multiple antennas are expected to be available on most devices

Transmitter



Figure 3.3: Current self-interference cancellation is sufficient to prevent ADC saturation.

in the near future. The proposed CDBA is compatible with MIMO.

3.3.3 Collision Probe

In our design, a node is allowed to start transmission immediately when the medium is sensed to be idle. To check whether this bold attempt will cause a collision, a node first broadcasts its arbitration preamble to perform collision probe.

When a node is transmitting its arbitration preamble, it also analyzes received signals in the frequency domain by performing FFT. The FFT result presents magnitude on each subcarrier. The node checks the magnitudes on inactive subcarriers to determine whether or not there exists a collision.

When there is only one node is transmitting, it should not detect high magnitudes on subcarriers that are deactivated by itself. In such a condition, it assumes that there is no collision and proceeds to data transmission directly. However, suppose a node observes high magnitudes on subcarriers that are not used by its own transmission. It learns that there is a collision as shown in Fig. 3.2.

A collision cannot be detected if two nodes choose the same binary code because we do not

assume that perfect self-interference cancellation is provided. However, the collision probability is decreased exponentially with the k where k is the number of subcarriers used for contention resolution. Compared with other frequency backoff algorithms [35] [36] that use one of the k subcarriers to represent a number, the probability that two nodes choose the same number is significantly reduced ($[0, 2^k - 1]$ versus [0, k - 1]).

It is possible that the binary code of one node may be masked by the code of another node (e.g., 110100 vs. 110110). One node assumes that there is no collision and initiates data transmission while the other node enters in the arbitration phase. To prevent the arbitration from interfering with the data transmission, we double the standard 802.11 short OFDM training symbol which occupies 12 subcarriers with an interval of four zero-power subcarriers. If a node detect this, it immediately aborts its transmission.

3.3.4 Bitwise Arbitration

When multiple nodes have data to transmit, it is possible that two nodes detect idle medium and initiate arbitration preamble transmission at a similar time. They will detect the collision during the collision probe phase and enter in the arbitration phase as shown in Fig. 3.4b. The figure shows that the binary code selected by a node has dual functions: *it is not only used for constructing the arbitration preamble in the frequency domain, but also used to determine whether or not they can continue transmission in the time domain for arbitration.*

To see the benefits introduced by self-interference cancellation and spectrum analysis, we first examine how bitwise arbitration will work in the time domain.

3.3.4.1 Time Domain without Cancellation

Before we have self-interference cancellation, we have to rely solely on time domain information for bitwise arbitration. A node can be equipped with a fast-switching radio or two radios, one for transmitting and one for receiving. Suppose nodes' contentions are implicitly synchronized by the



Figure 3.4: Arbitration phase is shortened with additional information obtained in the frequency domain.

busy medium. When the medium is detected to be idle, nodes start to transmit their arbitration preambles.

As introduced in Section 3.3.1, a node chooses a random number from a uniform distribution over $[1, 2^k - 1]$ to be its priority in a contention. The number is represented in the form of a k bit binary code. Each node goes through its binary code starting from the most significant bit. If the *i*th bit is '1', the node transmits its arbitration preamble in the *i*th OFDM symbol slot (note that the arbitration preamble is only one OFDM symbol); if the *i*th bit is '0', the node pauses its transmission for one OFDM symbol long. If a node detects signal when it pauses its transmission, it loses the contention and aborts its transmission as shown in Fig. 3.4a. Because contending nodes gradually abort their transmissions, only one node will survive at the end of the arbitration phase as long as nodes choose different random numbers.

In this design, a node cannot learn that it has won the medium access in a slot because it is unable to determine whether or not there is any other transmission during its own transmission. As a result, a node has to go through all bits to determine the winner. To reduce the probability that two nodes choose the same random number, the k should be increased and thus the length of the arbitration phase is increased. This makes bitwise arbitration less appealing than the CSMA/CA. If we can obtain perfect self-interference cancellation, a node is able to end the arbitration phase early (e.g., the fifth slot in the example). However, this imposes a high requirement on the self-interference cancellation.

3.3.4.2 Frequency Domain with Cancellation

If self-interference cancellation is sufficient to prevent ADC saturation, frequency domain information can be introduced to help detect collision. The arbitration preamble introduced in Section 3.3.1 occupies subcarriers selectively according to a node's binary code. If two nodes choose different binary codes, they can detect collisions between them in the frequency domain as illustrated in Fig. 3.2. If two nodes choose the same binary code and the code happens to be the highest priority among all contending nodes, they both initiate data transmission. The binary mapping from a random number to active/inactive subcarriers aims at reducing the probability.

In addition, because a node is able to examine whether or not a subcarrier is occupied by any node, all idle OFDM symbol slots can be skipped. In the bitwise arbitration phase, a node goes through its binary code and pauses its transmission in the *i*th OFDM symbol slot if its *i*th bit is 0. Suppose the *i*th bits of all contending nodes are 0s. All nodes will pause their transmissions in the *i*th OFDM symbol slot and it is useless to determine the winner. Because each node uses its binary code for active/inactive subcarrier mapping, none of the contending nodes occupies the *i*th subcarrier. Nodes detect this in the frequency domain and skip the corresponding bits to speed up the arbitration as shown in Fig. 3.4b. The winner is determined in the 3rd OFDM symbol slot in the example.

Because the arbitration phase may end earlier without traversing all bits, the total length of the arbitration phase can be designed to be slightly longer so that the collision probability is reduced with the increased k. The actual impact on the medium access overhead is much smaller than that in the time domain design. Fig. 3.5 summarizes the medium access control in CDBA.

3.4 CDBA Design Details

In this section we discuss the feasibility and practical issues of various components in the implementation of CDBA.

3.4.1 Noise and Interference

The collision detection relies on detecting signals on subcarriers used for arbitration. Noise and interference may cause incorrect detections. In CDBA, several subcarriers are selected to do arbitration. If transient high noise and interference do not affect these subcarriers, the arbitration is not disturbed. If they are under interference, arbitration time is increased or the winner cannot be determined. However, if the strong interference is not transient, it is reasonable to abort the node's transmission because the following data transmission will not succeed under the strong



Figure 3.5: The flowchart of CDBA.

interference.

The most harmful interference comes from neighboring nodes' transmissions because they use the same subcarriers for arbitration. Same as other wireless system designs [35] [1] [36], a node is allowed to access the medium only when the node is the winner in all contention domains. Enabling parallel transmissions if they do not interfere with each other at the receivers will be our future work.

3.4.2 Power Leakage

For an infinitely long sine wave of a single frequency, the frequency spectrum is an infinitesimally narrow peak (i.e., a delta function). Because we cannot bring an infinite long signal into a digital computer, we usually sample the input signal at a sample rate of f_s and divide the discrete samples into segments. Suppose a *N*-point FFT algorithm is used. We usually analyze a short segment of *N* samples at one time. The FFT algorithm outputs magnitudes and phases at *N* analytical frequencies $f(m) = \frac{mf_s}{N}$, where m = 0, 1, 2, ..., N - 1.

Note that the original signal is implicitly truncated by a rectangular window because only *N* samples are extracted from the original signal while others are set to zero. An important Fourier transform property is that multiplication in one domain corresponds to convolution in the other domain [43] [44]. Therefore, the corresponding frequency responses of the signal and the window function are convolved in the frequency domain. Since the original spectrum of the sine wave is a delta function, the spectrum of the windowed signal appears as the spectrum of the window function shifted to the location of the peak. As shown in Fig. 3.6a, the frequency response of a rectangular window function has numerous side lobes. The spectrum of the windowed sign appears because we use only a portion of the original time domain signal.

The frequency response of a continuous signal is a continuous curve in the frequency domain. When the continuous signal is sampled in the time domain, the effect is same as sampling the corresponding continuous curve in the frequency domain [45]. Because we usually use *N* samples



(a) Frequency response of window function (Rectangular (b) window).



(c) Directly taking samples of received signal for 64- (d) For the same segment of samples on the left, a point FFT may give a spectrum of good shape.



point FFT may give a spectrum of bad shape.



Frequency response of window function (Blackman-Harris window, $\alpha_0(0.35875), \alpha_1(0.48829),$ $\alpha_2(0.14128), \alpha_3(0.01168)).$



Blackman-Harris window is applied.



(e) Directly taking samples of received signal for 64- (f) For the same segment of samples on the left, a Blackman-Harris window is applied.

Figure 3.6: FFT results of received signal when 2 nodes are transmitting with 64-point IFFT. One node occupies subcarriers 15, 21, and 51; the other node occupies 27, 39, and 45. The peak at subcarrier 32 is the DC offset.

in a *N*-point FFT algorithm, the small number of samples on the spectrum's continuous curve results in irregular tails depending on where the samples occur. If the input segment contains energy precisely at integral multiples of the fundamental frequency f_s/N , the samples occur exactly at the valleys in the spectrum and there is no tail as shown in Fig. 3.6c. However, if the input segment has a signal component at an intermediate frequency between two analytical frequencies of mf_s/N , the samples occur somewhere along the peaks and valleys, resulting in power leakage to other frequencies. The obtained power spectrum may mislead a node to believe that there are signals on some inactive subcarriers as shown in Fig. 3.6e. The collision detection is thus not reliable, making nodes abort transmissions for nonexistent contentions.

To make CD reliable, we apply a window function that shapes the samples more gently instead of using a rectangular window that suddenly set all samples outside of the window to a value of zero. The use of a proper window function is to reduce the order of discontinuity at the boundary, which is the cause of side lobes. As shown in Fig. 3.6b, a Blackman-Harris window function yields much lower magnitude tails compared with the rectangular window. The underlying mathematics are well explained in [46] [44] [43] [45]. Fig. 3.6f shows that for the same time domain signal segment, the shape of the FFT result is much better. If we continue to take more samples, the shape is more clear, but without a proper window function the improvement is limited. Accumulated energy should make active subcarriers and inactive subcarriers more distinguishable. However, if the power leakage is not suppressed by a proper window function, inactive subcarriers may still accumulate enough energy to cause incorrect detection. Therefore, it is impractical to use all subcarriers for contention resolution.

Although windows like the Blackman-Harris window suppress the tail, they have a wider main lobe compared with the rectangular window, leading to wider peaks. The spectrum resolution is reduced. Because two points at each side of a peak have observable magnitudes as shown in Fig. 3.6d, we use one subcarrier every six subcarriers. This large separation ensures that an inactive subcarrier will not be detected as active due to power leakage. There are many other window functions available. In general a window function that produces lower side lobes has a wider main lobe as compared in [47]. There is a trade-off between main lobe width and side lobe levels. We choose the Blackman-Harris window because it is able to suppress the side lobes by more than 92 dB with a main lobe width that is acceptable in our design. The Blackman-Harris window is also identified as a top performer in many applications [46].

For a 64-point FFT, we have 64 subcarriers available. In USRPs, these is a pretty gradual roll-off in filter implementation, which causes strong curvature at two edges and there is a DC offset in the middle. Therefore, we avoid using 5 subcarriers at each edge and 6 subcarriers in the middle, leading to 48/6 = 8 subcarriers for binary code mapping (4 at each side of the DC offset). With only 8 subcarriers available for contention resolution, current CD designs will introduce high collision probability because if two nodes choose the same subcarrier, they do not know that they will have a collision in data transmission. It is possible to reduce the collision probability by increasing the number of subcarriers, but increasing the number of subcarriers means a N-point FFT consumes more samples each time. It takes more time to collect samples and the arbitration time is increased. Further, the reduced subcarrier spacing makes a protocol more sensitive to frequency offset. In CDBA, however, even if only eight subcarriers are available, there are $2^8 - 1 = 255$ unique binary codes (0 is excluded) available. The collision probability is same as setting the backoff window size to 255 in CSMA/CA, where a device will on average wait about 254/2 = 127 slots before transmission. It corresponds to an average backoff of 1.143 ms. On the contrary, CDBA takes only one FFT frame for collision probe and at most 8 FFT frames for bitwise arbitration. Suppose a 64-point FFT is used in a 20 MHz channel. The maximum overhead is $9 \times 3.2 \ \mu s = 28.8$ μs , which is a significant reduction on medium access overhead.

3.4.3 System Delay and Asynchronous Contention

As shown in Fig. 3.7, when the CCA algorithm indicates that the channel is clear, a node randomly selects a number and instructs the transmitter to send the corresponding arbitration preamble. To meet the stringent time requirement of MAC, we implement the CDBA on FPGA of USRP E110. However, we observe that samples collected at the same time do not contain a complete arbitration



Figure 3.7: Ensuring a complete arbitration preamble by repetition.

preamble due to some system delays. Therefore, we repeat the arbitration preamble to guarantee that the second FFT window after the sending instruction includes a complete arbitration preamble as shown in Fig. 3.7. It ensures that a node at least can obtain the non-distorted power spectrum for its own arbitration preamble.

The results of odd FFT frames are discarded and the results of even FFT frames are used to determine whether or not a collision exists. A disadvantage is that the incurred overhead is doubled. A node will defer its data transmission for at least two FFT frames even when there is no collision.

Contending nodes are implicitly synchronized by busy medium. However, if a node gets data to send after the medium becomes idle, its contention may not be synchronized with other contending nodes. As shown in Fig. 3.7, its CCA window contains a portion of preambles sent by other contending nodes. Fig. 3.8 shows the difference when different amount of an arbitration preamble is included in the FFT.

Even with half of an arbitration preamble, the FFT result can clearly show the difference between active and inactive subcarriers because all power of a transmitter is concentrated on several active subcarriers selected by the transmitter. Therefore, if a node is allowed to transmit, its CCA window should contain less than half of an arbitration preamble sent by another node. Its second FFT window should contain the major part of an arbitration preamble sent by another node, allowing it to detect the collision.


Figure 3.8: Different amount of an arbitration preamble is included in the FFT.

3.4.4 Random Backoff

To improve the efficiency when contention is heavy, we intentionally desynchronize contending nodes. In large deployment simulations we discover that the arbitration phase can be shortened by reducing the number of nodes that participate in the bitwise arbitration. We thus do not rely solely on bitwise arbitration to determine the winner in a contention. A short random backoff is introduced. Because half of an arbitration preamble is able to indicate busy, we defer nodes' transmissions by a random number of slots with a slot duration that is half of an arbitration preamble (e.g., $3.2 \ \mu s / 2 = 1.6 \ \mu s$ at 20 MHz). The random backoff is much shorter than that is defined in 802.11 [41]. With a short random backoff, we can reduce the number of contending nodes

that enter in the collision probe phase, which in turn helps to shorten the arbitration phase. For example, suppose we need to traverse 4 bits to resolve collisions between 5 nodes. With random backoff, only 2 nodes may enter in the collision probe phase immediately while the other 3 nodes back off due to detected busy state. It is possible that the collision between the two nodes is able to be resolved in the first bit in the arbitration phase. The arbitration phase is thus shortened. Note that the two nodes enter in the collision probe phase without any backoff. The random backoff mechanism thus does not introduce any overhead if any contending node chooses a backoff of 0.

When the contention is light, the random backoff is an overhead because the probability that some nodes will start collision probe without any backoff is low. If the saved time in arbitration phase is less than the introduced backoff, the random backoff mechanism should be disabled. Therefore, we let a node monitor the length of the arbitration phase. If it on average needs to traverse more than half of total bits to determine the winner, it enables the random backoff; otherwise it disables the random backoff. This helps to maintain the efficiency of the bitwise arbitration.

3.4.5 Misdetection

Due to frequency-selective fading, the power on some subcarriers may be weak. If a node chooses a binary code with only one bit set to '1', its arbitration preamble may not be detected by another contending node due to imperfect self-interference cancellation. We thus split the 8 bit binary code into two 4 bit binary codes. Each 4 bit binary code represents the subcarrier occupation on each side of the DC offset. A node now occupies at least one subcarrier on two different parts of the frequency spectrum. The collision detection is more reliable because usually at least one of them will be detected by other contending nodes. The collision probability, however, is slightly increased. We have reserved 1111 1111 for beacons or urgent control packets sent by access points (APs). The code 0000 0000 is excluded. The number of unique binary codes is reduced from $2^8 - 2 = 254$ to $(2^4 - 2)^2 = 196$. If more subcarriers are available, the collision probability will be reduced exponentially.

Note that when more than one node initiates arbitration preamble transmission at a similar

time, it is not destructive to the collision detection. If two nodes occupy different subcarriers, the arbitration preamble from each other is detectable as long as the SNR is above the detection threshold. If two nodes occupy the same subcarriers, the interference actually helps the third node detect collision for the high aggregated power on the shared subcarriers.

3.4.6 Hidden Terminal

In 802.11 standard [41], the OFDM PLCP (physical layer convergence procedure) preamble is BPSK-OFDM modulated with coding rate 1/2. The 802.11 standard thus mandates that the CCA mechanism should indicate busy for the start of a valid OFDM transmission at a power level equal to or greater than the minimum modulation and coding rate sensitivity (e.g., -82 dBm for 20 MHz channel spacing). If the preamble portion was missed, the CCA mechanism should signal busy for any signal 20 dB above the minimum modulation and coding rate sensitivity (e.g., -62 dBm for 20 MHz channel spacing). The high threshold ensures that a node does not back off for noise, but may make two nodes invisible to each other. Because the arbitration preamble shows unique characteristics in the frequency domain, it is easier to be recognized than the data transmission. The detection range is larger than the carrier sensing range. Therefore, most hidden terminals can detect each other's arbitration preamble.

In addition, we can let the AP rebroadcast whatever it received in the collision probe and bitwise arbitration phase. To check the delay of rebroadcasting, we conduct the following experiment with three USRP E110. We implement the rebroadcast function at the very front of the receiving chain. Node A is connected with both node B and node C through cables. Node B rebroadcasts whatever it received from node A. Node C compares signals directly received from node A with signals received from node B. Measurements show that the rebroadcasted signal is only one sample delayed as shown in Fig. 3.9. With the rebroadcast at the receiver, collision detection is guaranteed for hidden terminals in the collision probe phase and the arbitration phase. However, once the data transmission begins, all antennas on the receiver are used for receiving. Hidden terminals may think that the medium is idle because of the silence at the receiver. To avoid collisions,



Figure 3.9: Comparing the signal from the original transmitter with the signal from a relaying node in the time domain.

CDBA lets a node set the medium state to busy for a data transmission period when it detects an arbitration preamble transmission. Since usually one node survives to data transmission, the channel utilization is not reduced. Collisions caused by hidden terminals are avoided because hidden terminals defer their transmissions for a complete data transmission period once they lose a contention.

3.4.7 Message Prioritization

The BA allows us to assign different priorities to messages. For example, we can reserve the most significant bit for urgent messages. If a node needs to send an urgent message, it selects a binary code that sets the most significant bit to 1 (1xxx xxxx). For common messages, a node selects bina-

ry codes from the remaining pool (0xxx xxxx). When two nodes attempt to send at the same time, they will detect a collision and enter in the arbitration phase. The node with a common message will pause its transmission because the most significant bit is 0. On the contrary, the node with an urgent message will win the medium access immediately. This allows high priority messages to get through unimpeded, making CDBA suitable for real time prioritized communication. We can set several levels of priorities with the first *m* most significant bits.

We can also reserve 1111 1111 for beacons and implement a time slotted network without time slot assignment. Usually in time division multiple access (TDMA), each node is assigned a slot and all the slots constitute a time frame to repeat. Wi-Fi does not adopt this scheme because the traffic is dynamic. If we assign a slot to a node, time slots are wasted if the owner has no data to send. In CDBA, all time slots are open to any node. A node can stay in the low power sleep mode if it has no data to send. It may periodically wake up in some slots to check whether or not it has data to receive. When it has data to send, it initiates the collision probe at the beginning of a slot. If there is no collision, it transmits the data. If there exists a collision, it enters in the arbitration phase to resolve the collision. No slot is reserved for any node and there is no underutilization issue.

3.5 Testbed Results

Some delay measurements show that the delay between GNU Radio and FPGA is about hundreds of microseconds on USRP platforms [48]. If we implement the CDBA in GNU Radio, the system delay dominates the total delay before data transmission. To meet the stringent time requirement of MAC, we implement the collision probe and the bitwise arbitration on FPGA while packet encoding/decoding are implemented in GNU Radio because they are more complex than spectrum analysis. When a packet is ready to send, it is buffered on FPGA and triggers the collision probe. If there is no collision, the packet is sent. If there is a collision, the bitwise arbitration is activated. Once the collision is resolved, the buffered packet is sent. Because we do not have much space for buffer, we set packet size to 100 bytes.

3.5.1 Collision Detection Feasibility

We first use two USRP E110 equipped with WBX transceivers to check the feasibility of collision detection. The WBX transceiver has local oscillators (LOs) that work independently for transmitting and receiving chains. In the receiving chain, we implemented a 64-point FFT algorithm to perform spectrum analysis. Before the FFT, a Blackman-Harris window function is applied on received signals. The outputs of the FFT algorithm are used for active subcarrier detection and collision detection. The results of collision detection are fed to the bitwise arbitration block. The TX and RX antennas are separated by about 2 feet to reduce self-interference. Many optimizations are available for reducing the distance [1] [32] [33] [34] [42], but it is beyond the scope of this chapter. With better self-interference cancellation techniques, the performance of collision detection is expected to be improved.

When there is no transmission, the power spectrum only indicates a peak in the middle (DC offset) as shown in Fig. 3.10. Magnitudes on all other subcarriers are low. If we use them to derive a noise threshold, some subcarriers may be regarded as active when there are transmissions because of transmitter noise and power leakage. Therefore, the noise threshold cannot be obtained by sampling the channel when there is no transmission.

To identify active subcarriers, we search for the highest magnitude in each FFT frame. If it is located at the DC offset, the channel is regarded as idle. If it is located at a subcarrier other than the DC offset, we use it as a reference. We regard that a subcarrier is active if the magnitude on the subcarrier reaches a certain percent of the highest value. This may prevent a node from detecting a node that is far away. If a node is receiving an arbitration preamble from a nearby device, it may not detect a weak signal from a farther node because the signal strength falls out of the dynamic range. However, as proximate devices detect each other and abort transmissions, devices that are farther away will soon become detectable. If the farther device has a high priority, it may have not quit because it has many 1s. A node aborts its transmission only if it detects signals when it pauses its transmission, which means it has a lower priority compared to another node.

We must make sure that a node can start data transmission if there is no other transmission.



Figure 3.10: Results of performing 64-point FFT on received signals. The peak in the middle of each frame is the DC offset. Left frame contains no transmission. Right frame contains transmissions from two nodes.

Therefore, we let one node work in the full duplex mode. Fig. 3.11 shows the detection accuracy in accordance with different noise thresholds. The result is based on 10,000 FFT frames. When we take 60% of the highest magnitude in a FFT frame as the noise threshold, we can avoid detecting an unoccupied subcarrier as active. If we use a lower noise threshold, some unoccupied subcarriers may be occasionally detected as active, then a node cannot start data transmission immediately because it believes that there is a collision. If the noise threshold is too high, the detected active subcarriers do not match the binary code selected by the node. It is also counted as an incorrect detection here.

When there are other transmissions, it is easy to identify a collision if they are strong. However, depending on the adopted self-interference cancellation technique, transmissions from another



Figure 3.11: Taking a certain percent of the highest magnitude in a FFT frame to determine active subcarriers.

node may be too weak to be detected. We examine the efficacy of the collision detection at various SNR values by conducting the following experiments. We separate two nodes by various distances. Because the noise floor levels at a node increase when the node is simultaneously transmitting and the increment varies depending on the adopted self-interference cancellation technique, we compute the SNR at a node by setting the node to receive only. Once the SNR is measured, we set the node to full-duplex and check whether it can detect collision. Fig. 3.12 shows that the collision detection probability is 100% if the SNR is above 9dB. Note that we do not need to decode anything, we do not require a signal to be located exactly at a subcarrier, and an arbitration preamble occupies at least two subcarriers. The collision detection is thus reliable as long as the transmission from another node is not too weak. A better self-interference cancellation technique can further



Figure 3.12: Collision detection performance in CDBA.

improve the collision detection performance.

3.5.2 Throughput Gain

Since a node can start data transmission immediately if there is no collision or the BA mechanism gets involved to resolve the collision in several FFT frames, the channel utilization is significantly improved. We let three USRP E110 devices keep sending packets to another E110. The USRP E110 uses 64 Mega samples per second (Msps) ADC and the decimation rate can be set to 2^k . We test different PHY data rates by setting the sampling rate to 0.5 Msps, 1 Msps, 2 Msps, 4 Msps, and 8 Msps. Using dense modulation schemes requires good channel condition, we thus increase the data rate by increasing the channel width. Fig. 3.13 shows that the improvement of CDBA over 802.11 increases as the data rate increases, where medium access overhead plays an important role



Figure 3.13: Throughput gain of CDBA over 802.11.

in determining the efficiency.

In 802.11, two transmissions may collide with each other due to the small CW [0, 15]. The two nodes then increase their window sizes exponentially, leading to longer backoff. CDBA uses 8 bit binary codes. The medium access arbitration is settled down within 2*8=16 FFT frames. As analyzed in Section 3.4.5, two nodes have a probability of $(\frac{1}{196})^2$ to choose the same binary code. To obtain the same low collision probability, 802.11 needs a fixed window size of 196. Fig. 3.13 shows that the throughput of 802.11 is improved a little bit with a larger contention window but the throughput is still lower than that obtained in CDBA under the same condition. In addition, although the larger CW decreases the collision probability when there are multiple nodes contend for sending, it reduces efficiency when there is only one node that has data to send.

In CDBA, if there is only one node intends to send, it will not observe collision during the collision probe phase. It starts data transmission 2 FFT frames after the medium is sensed to be



Figure 3.14: Channel access overhead.

idle. Two FFT frames consume 2 * 64 = 128 complex samples. The time used to get 128 samples decreases as the data rate increases. The collision probe takes $128/8 = 16 \ \mu s$ at 8 Msps. However, in 802.11 the slot size is fixed at 9 μs . Even with the smallest CW [0, 15], the average backoff is $7.5 * 9 = 67.5 \ \mu s$. With a larger CW [0, 195], the average backoff is $97.5 * 9 = 877.5 \ \mu s$, which is nearly an order of magnitude greater than the collision probe.

When the data rate is low, the random backoff in 802.11 does not affect the efficiency of Wi-Fi significantly. As shown in Fig. 3.14, actual packet transmission dominates the air time in a successful transmission at low data rates. The throughput of 802.11 is similar to that of CDBA at 0.5 Msps. However, when the data rate increases, the time used for transmitting a packet is reduced but the time wasted on backoff stays the same. The random backoff soon becomes the dominant portion of a transmission. As a result, the medium access overhead becomes the major factor that prevents the high PHY data rates from being translated to high throughput.

Because the packet size is small here (i.e., 100 bytes), packet transmission time is around 288 μs at 8 Msps with BPSK. The largest backoff with a contention window [0, 195] is $195 * 9 = 1755 \ \mu s$, which can accommodate several packet transmissions at high data rates. A node may grab the channel when others back off. The throughput of 802.11 under contention is thus higher than that can be achieved by a single node when the data rate is high. To achieve the same collision probability as CDBA, 802.11 needs a large contention window that is undesirable in single node



Figure 3.15: Message prioritization in CDBA.

transmission scenarios.

3.5.3 Message Prioritization

CDBA allows high priority messages to get through the arbitration unimpeded. In this experiment, we let node A and node B choose binary codes from the pool of (0xxx xxxx) and let node C choose binary codes from the pool of (1xxx xxxx). Node A and node B are placed at different distances from the receiver to examine the capture effect of radios between near-and-far terminals. First, node A was turned on and then node B was turned on. Fig. 3.15 shows that they share the channel fairly with a slightly lower aggregated throughput than that is achieved by a single node. This is because the bitwise arbitration is activated. Fig. 3.15 also indicates that CDBA is not significantly biased toward node B, which is closer to the receiver. Soon node C was turned on. Node A and node B have to back off for node C. When they enter in the arbitration phase, only node C can

transmit because the most significant bit of its binary code is '1'. Nodes A and B have to abort their transmissions because they detect signal when they pause their transmissions. Although the bitwise arbitration is involved after collision probe, the collision is resolved in one bit and thus the throughput is similar to that can be achieved by a single node.

3.6 Simulation Results

In this section, we extend our evaluation of CDBA in larger deployments with simulations. We compare CDBA with WiFi-Nano [1] to show advantages of CDBA over WiFi-Nano in the presence of hidden terminals. CSMA/CN [40] is also included for addressing the hidden terminal problem. CSMA/CN uses the cross-correlation technique to notify transmitters of a collision. If a receiver detects a collision, it broadcasts a pseudo-random sequence called the receiver's signature. Through continuously correlating for the receiver's signature in the incoming signal, nearby transmitters will notice the collision notification. All transmitters that are sending data to the receiver abort their transmissions upon receiving the collision notification. The channel is thus free for other transmissions.

3.6.1 Simulation Settings

We implement CDBA, WiFi-Nano [1], and CSMA/CN [40] in ns-2.35. We simulate 802.11n data rates in a 20 MHz channel with 800 ns guard interval. To be fair with WiFi-Nano, we use the same wireless LAN settings as WiFi-Nano. Multiple nodes communicate with an AP with a transmission power of 20 dBm. A path loss exponent of 2.27 is used so that the transmission range for the 6.5 Mbps modulation is 100 m. Different from WiFi-Nano settings, we do not avoid hidden terminals with planned topologies. Nodes are randomly scattered within the AP's communication range.

3.6.2 Impact of Contending Transmitters' Number on Throughput

We first investigate how the number of contending transmitters impact the achievable throughput. We increase the number of contending transmitters from 1 to 30, simulating both light contention in a home environment and heavy contention in a conference room. All nodes generate fully backlogged CBR traffic with packet size of 1500 bytes. For some full-duplex wireless designs, at least two antennas are required. Suppose both antennas are used for data transmission after the arbitration. The maximum data rate is 130 Mbps using 64-QAM modulation and 5/6 coding rate as defined in 802.11 standard [41].

Fig. 3.16 compares the throughput when more and more nodes are deployed with a data generation rate of 130 Mbps. If there is only one node that has data to send, the throughput of CD-BA and WiFi-Nano is 30% higher than that of 802.11 because both CDBA and WiFi-Nano have considerably reduced the medium access overhead. In 802.11, the average random backoff is $7.5 \times 9\mu s = 67.5 \ \mu s$ with the CW [0, 15]. In CDBA the medium access overhead is a duration of $2 \times 3.2\mu s = 6.4 \ \mu s$ for collision probe. In WiFi-Nano, the average backoff time is $7.5 \times 0.8\mu s = 6 \ \mu s$. With a $4 \ \mu s$ preamble after the random backoff, the average medium access overhead is $10 \ \mu s$. The difference on medium access overhead leads to the 30% improvement on throughput. If the data rate is further increased, the improvement on throughput will be greater because more bits can be transmitted with the time saved on medium access contention.

When more than one node is deployed, hidden terminals may appear in some scenarios. The probability of observing hidden terminals in a scenario increases with the number of deployed nodes. Fig. 3.16 shows that the adoption of tiny backoff slots introduces severe contention in the presence of hidden terminals. The reason is that WiFi-Nano cannot spread out nodes' transmissions with tiny backoff slots and thus hidden terminals may keep colliding with each other.

In 802.11, a collision will increase the CW exponentially. Because the backoff slot size is 9 μs , a larger CW leads to a longer backoff time interval. The long backoff interval is able to desynchronize hidden terminals. For example, when the CW reaches the maximum value 1023, nodes' transmissions are distributed in an interval of 9.207 *ms*. Even if there are hidden terminals,



Figure 3.16: Impact of contending nodes' number on throughput (CDBA-CN overlaps with CDBA, WiFi-Nano-NAV-CN overlaps with WiFi-Nano-NAV).

nodes can get opportunities to deliver their packets successfully because the time used to transmit a packet of 1500 bytes is 1.846 ms at the lowest rate 6.5 Mbps. However, with the 800 ns slot size defined in WiFi-Nano, the longest backoff is 818.4 μ s. Hidden terminals will keep colliding with each other. The throughput of WiFi-Nano drops quickly when more and more nodes are deployed. As shown in Fig. 3.16, CSMA/CN improves the throughput of WiFi-Nano by 3 times, confirming the large number of collisions introduced by the tiny backoff slot.

We assume that a pair of transceivers is reserved for collision notification and thus a collision can be detected and notified whenever it happens. The ideal case may not hold if all antennas are used for data transmission once the correlation preamble transmission is completed in WiFi-Nano. Note that even with the help of CSMA/CN, the throughput of WiFi-Nano is lower than that of 802.11 in the presence of hidden terminals. This is because even if nodes are informed of collisions through CSMA/CN, their transmissions are still crowded in a small backoff interval. The essential problem is that nodes' transmissions cannot be spread out with the tiny 800 ns backoff slots. On the contrary, there are fewer collisions caused by hidden terminals in 802.11 because of a larger backoff time interval. The throughput gain achieved by CSMA/CN over 802.11 is thus smaller.

Because the pseudo-random preamble in WiFi-Nano can be detected at a very low SNR through correlation, two nodes that are out of carrier sensing range may detect each other's correlation preamble. Therefore, we let a node set the medium state to busy when it detects a correlation preamble. It is similar to the virtual carrier sensing mechanism known as network allocation vector (NAV) in 802.11. We present this variant of WiFi-Nano as WiFi-Nano-NAV. With the modification, it is rare that hidden terminals will cause a collision. Collision notification has minor improvements on throughput of WiFi-Nano-NAV, which is hard to observe in Fig. 3.16.

Although virtual carrier sensing helps address the hidden terminal problem, channel utilization is decreased when all nodes abort their transmissions. In WiFi-Nano, when *k* nodes initiated preamble transmission in the same slot, they keep transmitting with a probability of 1/k. It is possible that all of them abort transmissions. Without virtual carrier sensing, some nodes will grab the channel soon after all nodes in the last round of contention aborted their transmissions. However, virtual carrier sensing is needed to address the hidden terminal issue. With virtual carrier sensing, all nodes have to defer their transmissions for one packet transmission duration as they set the medium state to busy when they detect a correlation preamble. As more nodes are deployed, it is more likely that several nodes initiate preamble transmission in the same slot. The throughput of WiFi-Nano-NAV decreases as shown in Fig. 4.13. Therefore, WiFi-Nano faces a dilemma in addressing hidden terminal issue. It may need virtual carrier sensing to avoid persistent collisions between hidden terminals, but virtual carrier sensing reduces channel utilization in WiFi-Nano when all nodes happen to abort their transmissions.

In CDBA, the medium state is also set to busy when an arbitration preamble is detected. Because at least one node will survive to data transmission, preventing other nodes from transmission



Figure 3.17: Impact of data rates on throughput under light contention.

is reasonable. The problem is that the collision cannot be resolved if two nodes choose the same binary code. Considering that the random number range is large with the binary mapping, the collision probability is low. As shown in Fig. 3.16, CSMA/CN has a marginal improvement on the throughput of CDBA, indicating a low ratio of collisions to the total number of transmissions.

3.6.3 Impact of Data Rates on Throughput

Both CDBA and WiFi-Nano aim at reducing the medium access overhead. The saved time allows nodes to transmit more data within the same period of time. The throughput gain of CDBA and WiFi-Nano over 802.11 is expected to increase along with the data rates. In the following simulations, we do not include CSMA/CN. Fig. 3.17 and Fig. 3.18 compare the throughput achieved at different data rates when there are 5 transmitters and 30 transmitters, respectively.

When five transmitters are deployed, the throughput gain of CDBA and WiFi-Nano-NAV over



Figure 3.18: Impact of data rates on throughput under heavy contention.

802.11 becomes larger and larger as the data rate increases as shown in Fig. 3.17. This is because the medium access overhead of 802.11 takes a great proportion of the total time in a transmission at high data rates.

When thirty transmitters are deployed, the throughput gain of WiFi-Nano-NAV over 802.11 is not significant as shown in Fig. 3.18. With thirty contending nodes, it is highly possible that all contending nodes abort their transmissions in a contention. The throughput of WiFi-Nano-NAV drops quite a lot compared to five-node scenarios. In addition, as the data rate increases, the packet transmission time decreases, hence there are more rounds of contentions in the same period of time. More rounds of contentions imply more collisions. The throughput gap between WiFi-Nano-NAV with 5 nodes and WiFi-Nano-NAV with 30 nodes becomes larger and larger.

When thirty transmitters are deployed, the throughput of CDBA drops a little bit because the probability that two nodes choose the same binary code increases, but the throughput achieved

by CDBA is still 40% higher than that can be achieved by WiFi-Nano-NAV and 802.11. In both WiFi-Nano-NAV and 802.11, the CW starts from the minimum value. If more than one node defers the transmission to the same slot, there is a collision in 802.11 and there may be no transmission in WiFi-Nano-NAV. Although a collision in CDBA incurs the same cost of one packet transmission duration, CDBA has lower collision probabilities because of the large random number range. Therefore, CDBA provides a great throughput improvement over WiFi-Nano and 802.11.

3.6.4 Fairness

When a collision happens, many off-the-shelf transceivers are able to decode one of the colliding packets depending on the relative SINR at the receiver. The capture effect unfairly favors nodes that are closer to an AP. However, if collisions can be detected and resolved before data transmission, unfairness caused by capture effect of radios can be reduced. Fig. 3.19 shows how the fairness index [49] changes along with the number of contending transmitters.

When more and more nodes are deployed, the collision probability increases. If a node is close to the AP, its packet may be decoded in a collision. On the contrary, if a node is at the edge of the network, its packets may never be captured in a collision. Therefore, capture effect of radios helps increase throughput but introduces unfairness between near-and-far terminals. Fig. 3.19 shows that the fairness in 802.11 decreases quickly. CSMA/CN terminates a collision early. As a result, there are more rounds of contentions in the same period of time. The unfairness between near-and-far terminals is more severe.

Both CDBA and WiFi-Nano resolve collisions before data transmission, near-and-far terminals get equal chances to deliver their packets. However, if hidden terminals ignore the collision resolution, both throughput and fairness will be compromised. In WiFi-Nano, the correlation preamble can be detected at a very low SNR. Collisions are likely to be resolved before data transmission and only one node starts data transmission. However, because data transmission has no special characteristics in the time or frequency domain, it cannot be recognized at a very low SNR. S-ince WiFi-Nano uses tiny backoff slots, a hidden terminal will soon check the medium again and



Figure 3.19: Impact of contending nodes' number on fairness.

sense that the medium is idle. The hidden terminal may start a transmission and cause a collision. Only nodes that are close to the AP may success in data transmission. The fairness index of WiFi-Nano drops quickly as shown in Fig. 3.19. With virtual carrier sensing, once collision is resolved, contending nodes will backoff long enough to allow the ongoing transmission to complete. The fairness index of WiFi-Nano-NAV and CDBA is thus high , both are close to 1.

When data rates increase, a packet transmission can be completed in a shorter time. Because the backoff time range in 802.11 is large, a short transmission is unlikely to be corrupted by a transmission initiated later. Even if a node is far away from the AP, it has some opportunities to deliver its packets successfully. Therefore, the fairness of 802.11 is improved with higher data rates as shown in Fig. 3.20. CDBA and WiFi-Nano-NAV have resolved collisions before data transmission. The capture effect of radios thus does not play an important role in affecting fairness.



Figure 3.20: Impact of data rates on fairness.

3.7 Summary

Collision avoidance (CA) is an effective method to reduce collisions in contention-based wireless communication, but random backoff used in the method incurs unnecessary underutilization of channel. This chapter makes collision detection (CD) practical in wireless communication, allowing nodes to transmit immediately when the channel is sensed to be idle. If multiple nodes have data to transmit, a bitwise arbitration (BA) mechanism is activated to resolve the collision before data transmission. As long as each node chooses a unique binary code, only one node will be granted the permission to start data transmission at the end of the arbitration phase. High throughput is offered with low collision probability achieved in a short bounded arbitration phase. The BA mechanism also enables message prioritization. Through testbed experiments and extensive simulations, we show that CDBA is more efficient than current CA and CD designs.

Although CDBA is efficient in coordinating medium access between devices that use the same

channel width, it does not considered the scenarios where heterogeneous radios may coexist in a contention domain. In the past few years, wireless devices have been densely deployed. For example, in a digital home, various wireless devices are simultaneously active to support wireless LAN, wireless gaming, wireless display, and home security. As these devices target at different applications, they usually have different traffic loads and thus different requirements on channel widths. The radio heterogeneity imposes special challenges on medium access coordination. The next chapter studies the contention issues between wideband devices and narrowband devices.

CHAPTER 4

ADAPTIVE CHANNEL BONDING IN MULTICARRIER WIRELESS NETWORKS

Applications such as video conferencing and multimedia streaming demand low-lag gigabit speeds. The quest for high speed is pushing wireless communication toward wider channels. Channel bonding, an efficient method that increases data rate regardless of other technologies in use, has been introduced to the 802.11 standard. However, inefficiency and unfairness issues arise when heterogeneous radios coexist in a contention domain. When a device operates in a wideband channel that spans across multiple narrow channels, it has to defer its transmission to the time when all of the narrow channels are idle. This is inefficient because the device cannot utilize the other narrow channels when only one narrow channel is occupied. Further, when there are narrowband devices in more than one narrow channel, the devices that use channel bonding are harder to win medium access opportunities because narrowband devices may work independently in different narrow channels.

A natural solution is to revert to narrow channel operations. A recent work WiFi-NC [50] introduces a compound radio design that splits a channel into multiple narrow channels and use them independently. The strategy is efficient when a device has small packets to multiple devices. However, when a device has a bulk of data to one receiver, it is more efficient to use multiple narrow channels as one wide channel to increase spectral efficiency by removing the guard bands between contiguous narrow channels. Although WiFi-NC reduces the guardbands with sharp filters, signal spreading introduced by a sharp filter will increase the inter symbol interference (ISI) [50]. Addressing the ISI requires a tradeoff between spectral efficiency and capability of tolerating frequency offset. Further, it demands higher processing power and more resources to support multiple parallel signal processing flows.

The inefficiency and unfairness issues can be addressed if a device does not need to wait for all narrow channels to be idle to initiate a transmission. Recent spectrum-agile designs [51] [52] [53]

have shown that it is practical to aggregate noncontiguous narrow channels as one channel. The flexibility of spectrum use is thus comparable to WiFi-NC [50]. However, current spectrum-agile designs are frame-based. Therefore, when a new channel becomes available, it cannot be utilized until the next frame. An unfairness issue arises as not all nodes get contention opportunities for the new channel. The new channel is invisible to nodes that are in transmission. Some nodes may never be able to acquire the channel.

In addition, current spectrum-agile designs are lack of an efficient mechanism to address the multiple access issue. When a narrow channel becomes available, several nodes may attempt to acquire it at the same time. Leaving contention resolution to the carrier sense multiple access with collision avoidance (CSMA/CA) may require a large contention window to reduce collisions. The high overhead is hard to cut because low collision probabilities are desired. A transmission may fail even when two nodes have a collision only in a small fraction of the used spectrum.

After a transmitter has won some narrow channels, it needs to inform the receiver of the used spectrum; otherwise, the receiver is unable to decode packets sent by the transmitter. Using control packets to achieve spectrum agreement can introduce high overhead because of medium access contention and physical layer convergence procedure. A better spectrum agreement method is desired.

To address the inefficiency and unfairness issues in channel bonding, this chapter introduces an adaptive channel bonding (ACB) protocol in which a node is allowed to initiate a transmission as long as there exist some idle narrow channels and the node gradually increases channel width during transmission when new narrow channels are sensed to be idle. The design imposes three challenges.

First, when some narrow channels become idle, the medium access contention is severe because a transmitter has to contend with nodes that are within the same band of spectrum and nodes whose spectrum is partially overlapped with its spectrum. Even if two nodes collide only in a small fraction of the used spectrum, their transmissions may fail. Therefore, it is critical to address the contention issue in wideband spectrum sharing, which is the key to ensure that nodes will benefit from channel bonding. In this chapter, a compound preamble is designed to probe collisions in all channels at the same time and a parallel bitwise arbitration mechanism is introduced to resolve the collisions. Nodes are allowed to contend for different channels simultaneously with different priorities. A node is unlikely to lose all channels in a contention.

Second, ACB aggregates all narrow channels obtained by a transmitter as one wide channel. A challenge is the communication over uncertain channel combinations. If the receiver is unaware of the channels used by the transmitter, the receiver cannot decode any packet sent by the transmitter. To achieve spectrum agreement between the transmitter and the receiver, we design a partial spectrum correlation that encodes the receiver's unique signature in the frequency domain in all channels used by the transmitter. The receiver calculates the expected results when its signature is present in each channel. Through correlating received signal with all expected results, the receiver is able to identify channels used by the transmitter. Although there are n expected results when the target wide band is divided into n narrow channels, the searching for the signature in all channels can be parallelized.

Third, the transmitter was unable to monitor the medium state while it is transmitting. Therefore, current designs [51] [35] [52] [53] [54] are frame-based. Each frame will use the available spectrum fragments detected before transmission. This raises an unfairness issue for sharing channels. Because a node cannot contend for a channel that becomes available during transmission, the channel may be acquired by another node. It is possible that channels used by two nodes are never available to each other because their contentions are not synchronized. With advances in self-interference cancellation [42], it becomes feasible to detect new spectrum availability even during transmission without using another antenna. This allows ACB to break the frame-based structure, changing spectrum use during transmission.

This chapter addresses the three challenges and integrates all components as one complete system. We prototype it on the GNU Radio / USRP platform to validate its effectiveness. Experimental results show that the proposed ACB is practical. Extensive simulations in ns-2 further demonstrate that the ACB can significantly improve the efficiency of wideband spectrum sharing.

4.1 Related Work

Because different applications have different channel width demands and energy efficiency requirements, a variety of channel widths have been defined in different standards. On the one hand, radio spectrum is a finite resource and certain parts are preferable due to desirable propagation properties and antenna design constraints. On the other hand, no device will keep transmitting. Therefore, spectrum sharing is unavoidable and reasonable. Since channel widths are different, it is critical to address the inefficiency and unfairness issues caused by heterogeneous radio coexistence.

A natural solution is to revert to narrow channel operations based on two observations. First, wideband devices cannot transmit even when only a small part of the target spectrum is occupied [50] [53]. The availability of narrow channels is much higher than that of wide channels due to the presence of many uncoordinated narrowband devices [54]. It is thus attractive to use spectrum in the unit of narrow channels. Second, high PHY data rates achieved by wide channels increase the proportion of medium access overhead in data transmission when a node has a small amount of data to transmit [35]. Motivated by these observations, various designs of using narrow channels have been proposed.

WiFi-NC [50] introduces a compound radio design where each narrow channel is evaluated and used independently. It implements multiple receiving chains and transmitting chains in a radio, which demands a large FPGA. To reduce overhead incurred by setting guard bands between narrow channels, multiple sharp filters are used. Since a filter relies on spreading (smoothing) signal in time to shape the spectrum, a sharp filter spreads a sample to several subsequent samples [50]. This increases the inter symbol interference (ISI). WiFi-NC adopts the orthogonal frequency-division multiplexing (OFDM) for transmission. A common method to address the ISI is to use the cyclic prefix (CP). Due to the spreading, WiFi-NC needs a longer CP for each OFDM symbol. The spectral efficiency is reduced as the CP does not carry data. One way to address the issue is to increase the number of subcarriers, but it reduces the subcarrier spacing, making WiFi-NC sensitive to frequency offset and ill-suited for mobile scenarios where Doppler effect is obvious. Further, it duplicates multiple signal processing flows that would be redundant if data in all narrow channels are from the same transmitter to the same receiver.

In a unicast transmission, it is more efficient to bond narrow channels as one wide channel. This removes guard bands between contiguous narrow channels and reduces the complexity of wireless communication. However, with a wide channel, the data transmission efficiency decreases because random backoff for collision avoidance is no longer efficient for small packets in high-speed transmission. FICA [35] thus allocates channel width according to the typical frame size and the physical layer (PHY) data rate achieved at a certain channel width. OFDM is employed in FICA to obtain the flexibility of allocating spectrum resources. In OFDM, a band of spectrum is divided into multiple closely spaced orthogonal subcarriers. Each of them carries a data stream in parallel with others. FICA groups subcarriers as subchannels and assigns subchannels to different nodes. Although in FICA multiple nodes send simultaneously in orthogonal narrow channels, each narrow channel is a part of a wide band. The entire wide band must be verified to be idle before a new transmission can be commenced. The method is thus subject to the adverse impact of heterogeneous radios.

In OFDM we can deactivate some subcarriers by feeding 0 instead of modulation symbols (e.g., 1+0i, -1+0i for BPSK) to the corresponding subcarriers. The non-contiguous OFDM (NC-OFDM) technique allows a node to use some but not all subcarriers. As a result, instead of waiting for the entire wide band to be idle, a node can utilize spectrum fragments that are currently available. However, communication over uncertain subcarriers imposes a spectrum agreement challenge. The transmitter interleaves bits across all used subcarriers and the receiver reconstructs the original packet by extracting bits from all used subcarriers. If the receiver does not know what subcarriers are used, it may get insertions or deletions of bits from the data stream. The unknown insertions and deletions of bits make it impossible for the receiver to decode the packet sent by the transmitter. Therefore, it is critical to achieve spectrum agreement between the transmitter and the receiver.

To address the issue, early spectrum-agile designs [51] [52] use control packets. This means a dedicated control channel must be available, or the control packets can be send only when the entire wide band or a specific channel becomes available, which works against the spectrum-agile feature. The exchange of control packets also introduces high overhead because of the random backoff in collision avoidance and the preamble overhead in physical layer convergence procedure (PLCP).

RODIN [54] divides a wide band of spectrum into *n* nonoverlapping narrow channels and reshapes the spectrum of a frame to n_f (< *n*) of these narrow channels. RODIN supports direct connection to commercial off-the-shelf (COTS) devices but it has overhead of setting guard bands similar to WiFi-NC [50]. Because narrow channels are isolated in RODIN, the transmitter transmits a set of sequences simultaneously in all used channels. Each sequence is unique in a narrow channel. The receiver can identify the channels used by the transmitter through time domain signal correlation. RODIN needs multiple filters to isolate each channel, and it is not easy to determine how many narrow channels are used by the transmitter. RODIN thus fixes the number to n_f . The receiver must identify the n_f narrow channels in each transmission. The method does not adapt to a node's traffic where fewer or more channels are needed.

4.2 Adaptive Channel Bonding

Adaptive channel bonding (ACB) imposes three challenges: *how to detect newly available channels during transmission, how to contend for the available channels, and how to achieve spectrum agreement between transmitter and receiver.* In this section, we present our design that addresses these challenges.

4.2.1 Overview

We assume that a wide band of spectrum is divided into *n* narrow channels of equal size for independent evaluation (e.g., 6 MHz channels in the TV bands). In the remaining part of the chapter, a channel means a narrow channel unless otherwise stated. Before commencing a transmission, the transmitter checks what narrow channels are available in the target wide band. The clear channel assessment (CCA) can be done through checking the power spectral density (PSD) of the wide

band [52] [53]. Once available channels are identified, the transmitter needs to contend for these channels. Because there may exist many contending transmitters in a wide band of spectrum, medium access contention based on carrier sense multiple access with collision avoidance (CSMA/CA) may need a large contention window (CW) to ensure low collision probabilities. However, a large contention window leads to a long average backoff time, significantly reducing the efficiency. We thus introduce a parallel bitwise arbitration mechanism to reduce the medium access overhead and provide low collision probabilities. In addition, the bitwise arbitration allows a node to contend for different channels with different priorities. A node is unlikely to lose all channels in a contention.

After a transmitter wins some narrow channels, it needs to inform the receiver of the used channels; otherwise, the receiver is unable to decode packets sent by the transmitter. To achieve fast spectrum agreement between the transmitter and the receiver, we introduce partial spectrum correlation. With the partial spectrum correlation, the receiver is able to identify channels used by the transmitter and filter out unwanted signals in other channels, allowing the receiver to restore standard preamble detection on clean signal from the transmitter.

Once a node starts transmitting, it was unable to receive other signals simultaneously with the same RF front end and antenna because the self-interference is so strong that it buries other signals. However, recent advances in self-interference cancellation [32] [42] show that new radio designs will allow simultaneous transmission and reception even with the same antenna. Therefore, a transmitter can monitor the medium state even when it is transmitting. This allows all nodes to perceive a newly available channel. All nodes get equal chances to acquire the new channel. The contention is resolved through bitwise arbitration. The winner piggy-backs the new spectrum use via data to inform the receiver of increased channel width and then bonds more channels for subsequent data transmission.

4.2.2 Spectrum Contention

ACB employs OFDM to opportunistically utilize spectrum fragments in a wide band of spectrum. Besides contending transmitters in the same band of spectrum, there are transmitters contending for



Figure 4.1: Binary code mapping with NC-OFDM for collision detection.

partially overlapped spectrum as shown in Fig. 4.1 (difference applications may choose different center frequencies and bandwidths). A collision in a small fraction of the used spectrum may cause the entire transmission to fail. The cost is high and thus a large contention window may be needed in CSMA/CA to reduce the collision probability. However, using a large contention window increases the average backoff time. A parallel bitwise arbitration mechanism is introduced in this chapter to provide low collision probabilities with low overhead.

The bitwise arbitration relies on collision detection. When a node has data to send, it first checks what channels are available. It then transmits a compound preamble that occupies all of the channels but the preamble is designed to set some subcarriers to inactive as shown in Fig. 4.1. First, some channels are busy, hence all subcarriers in these channels are set to inactive. Second, we intentionally deactivate some subcarriers in available channels for collision detection.

4.2.2.1 Compound Preamble Design

In the frequency domain, a compound preamble occupies the entire wide band with some subcarriers set to inactive as shown in Fig. 4.1. In the time domain, a compound preamble is one OFDM symbol that consists of *N* samples when a *N*-point IFFT is used. Suppose the target wide band is composed of *n* narrow channels. Each channel contains k = N/n subcarriers. A unique sequence of

k symbols $\{p_1p_2p_3...p_k\}$ is assigned to each node. For collision detection, the actual sequence is of no meaning. We thus postpone the discussion about the sequence to the introduction of spectrum agreement. Here we focus on how to create high magnitudes at some but not all subcarriers.

If a node repeats its unique sequence by *n* times ($k \times n = N$ symbols) and performs *N*-point IFFT, it generates a compound preamble that causes high magnitudes at all *N* subcarriers. To deactivate some subcarriers for collision detection, a node draws a random number from a uniform distribution over the interval ($0,2^k$) for each available channel. For busy channels, the number 0 is used. Because each number can be represented by a *k* bit binary code, the binary code is bitwise AND with the node's unique sequence. As we feed some 0s to the IFFT, not all subcarriers in each available channel are used and all subcarriers in busy channels are set to inactive as shown in Fig. 4.1. The binary representation of each number is essentially a map of active and inactive subcarriers in the corresponding channel with '1' indicating active and '0' indicating inactive.

With the compound preamble design, a collision is detectable if a node observes high magnitudes at subcarriers that are deactivated by it. The collision detection fails only when two nodes choose the same random number, but the probability is low as it decreases exponentially along with the increased subcarrier number k. For example, assuming a 64-point IFFT is performed on a 20 MHz band that comprises four 5 MHz channels, each channel contains 16 subcarriers. With the binary mapping, in total $2^{16} - 1$ random numbers can be represented (0 is excluded) in each channel. The probability that two nodes choose the same number for a channel is low. To get the same low collision probability in CSMA/CA, the contention window (CW) will be too large to be practical.

Because collisions are detectable in the frequency domain, instead of deferring nodes' transmissions randomly for collision avoidance, we allow a node to transmit immediately when some channels are identified as idle. To check whether the bold attempt will cause a collision, a node performs spectrum analysis on received signal while it is transmitting its compound preamble. The simultaneous transmission and reception relies on recent advances in self-interference cancellation.

4.2.2.2 Collision Detection

As discussed in the previous chapter, recent studies in self-interference cancellation have reached at least 45 dB cancellation in practice [42] [32]. The self-interference cancellation is sufficient to prevent ADC saturation. Although we need more self-interference cancellation to enable simultaneous transmission and reception of packets in the same channel, our collision detection does not require such a perfect cancellation. In our design, each node sets some subcarriers to inactive, leading to zero power at these subcarriers. As long as the self-interference does not cause ADC saturation, a node can detect signals at inactive subcarriers if there is a collision. The collision detection does not incur additional cost of an additional antenna. To enable real full duplex wireless communication, the transmitting antenna and the receiving antenna may be separated to obtain more self-interference cancellation [32].

4.2.2.3 Medium Access Control

As collisions are detectable, it is important to determine which node is the winner of a narrow channel in a contention. Note that each node generates different random numbers for different channels. This gives a node higher chances to win some channels. Fig. 4.2 shows an overview of medium access procedure in ACB. Although the example shows three nodes, the principle applies to any number of contending nodes.

There are two phases: *collision probe* and *bitwise arbitration*. When a node has data to send, it checks what channels are available. Once the CCA is done, the node transmits its first compound preamble to check whether there is a collision in any of the channels. We call this phase *collision probe*.

If there is no collision in a channel, the node wins the channel immediately. It uses all subcarriers in the channel to transmit the destination's unique sequence, which will be discussed soon. If there is no collision in all contending channels, the bitwise arbitration is skipped. However, if there is a collision in any of the channels, the bitwise arbitration is triggered.





In the bitwise arbitration phase, the compound preamble of a node is updated to consider three conditions. First, for busy channels, all subcarriers are set to inactive. Second, for a channel that is won by the node, the destination's unique sequence is mapped to all subcarriers. Third, for a channel in which the winner has not been determined, the *i*th bit of the corresponding binary code is checked for constructing the *i*th compound preamble in the bitwise arbitration phase.

The compound preamble is used to contend for all channels at the same time. If we focus on one arbitrary channel, a node traverses the corresponding binary code starting from the most significant bit. If the *i*th bit is 0, all subcarriers in the channel are set to inactive in the *i*th compound preamble. If the *i*th bit is 1, the node maps its unique sequence to active subcarriers according to the binary code.

While a node is transmitting its compound preamble, it also performs spectrum analysis on received signal. If it sets all subcarriers in a channel to inactive but observes high magnitudes at subcarriers in the channel, it loses the channel. On the contrary, if the channel is idle, the winner has not been determined and it needs to check the next bit. If a node uses some subcarriers in a channel and the collision still exists (detecting high magnitude at subcarriers that are deactivated by it), it proceeds to check the next bit too. On the contrary, if the frequency spectrum of the channel matches its corresponding binary code, the collision has been resolved and the node wins the channel. Once a node wins a channel, all subcarriers in the channel are used for the destination's unique sequence. The node should occupy the channel to prevent other nodes from taking it. When collisions in all channels are resolved, the node initiates the spectrum agreement.

The time length of a compound preamble is equivalent to one OFDM symbol, which contains N samples when a N-point IFFT algorithm is used. Meanwhile, the spectrum analysis needs N samples when a N-point FFT algorithm is used. The simultaneous transmission and reception of a compound preamble takes N/B seconds where B is the bandwidth. For example, assuming a 64 point IFFT/FFT algorithm is adopted, it takes $3.2 \ \mu s$ to complete the preamble transmission at 20 MHz. In the same $3.2 \ \mu s$, the receiving chain completes the collection of 64 samples for the spectrum analysis.

4.2.2.4 Case Study

Fig. 4.2 shows how bitwise arbitration gradually determines the winner of each channel. Collided transmissions are implicitly synchronized by the collision. Nodes detect collisions in the 1st compound preamble. In the 2nd OFDM symbol, the compound preamble sent by a node is updated according to the first bit of each binary code. For node A, the first bit for channel 1 is 1. It occupies some subcarriers according to the binary code. On the other hand, the first bit of the binary code for channel 3 is 0. All subcarriers in channel 3 are set to inactive. Following the same principle, node B and node C get a binary map where all subcarriers are set to inactive because the first bit of their binary codes is 0.

While nodes are transmitting the compound preamble, they also perform FFT on received signal. For channel 1, node A notices that there is no collision and thus it wins the channel. Node B and node C notice that channel 1 is busy when they have no active subcarrier in the channel. They lose the channel. The arbitration for channel 1 is done. For channel 3, all nodes detect an idle channel. The winner has not been determined. They must check the next bit to determine the winner.

In the 3rd OFDM symbol, node *A* has won channel 1 and it uses all subcarriers in channel 1 for the destination's unique sequence. In channel 3, both node *A* and node *C* have no active subcarrier because the 2nd bit of their binary codes is 0. Node *B* and node *C* have lost channel 1. They set all subcarriers in channel 1 to inactive. The 3rd compound preamble sent by node *B* occupies some subcarriers in channel 3 according to its binary code. Since there is no collision, node *B* wins channel 3.

The example shows that as long as two nodes do not choose the same binary code for a channel, the collision will be resolved within k OFDM symbols where k is the number of subcarriers used for medium access contention in a narrow channel. Note that the bitwise arbitration may end earlier without traversing all bits. Therefore, the overhead is upper bounded by k but most of the time it is lower.



Figure 4.3: Time domain signal correlation.

4.2.3 Spectrum Agreement

After the spectrum contention, a transmitter has won some narrow channels. However, the receiver is unable to decode any packet sent by the transmitter if the occupied spectrum is unknown. To achieve spectrum agreement, we introduce partial spectrum correlation.

4.2.3.1 Cross-correlation

In signal processing, cross-correlation is an efficient way to measure the similarity of two waveforms. The transmitter transmits a sequence of complex symbols that is known at the receiver. The receiver simply correlates the received signal with the complex conjugate of the known sequence. If a high correlation peak occurs, it means the known sequence is present in the received signal.

The cross-correlation can be used as an in-band signaling. Let the transmitter transmit a known sequence *s* of length *L*. The receiver searches for the known sequence by correlating the received signal *y* with the complex conjugate of the known symbol sequence, which is denoted by *s**. The cross-correlation value $C(s, y) = \sum_{k=1}^{L} s^*[k]y[k]$ is low if *s* is not present in *y*. The reason is that the noises and the other signals are supposed to be independent of *s*, and thus the similarity identified by the cross-correlation should be low. As shown in Fig. 4.3, the *L* samples of *y* are extracted by a sliding window. The cross-correlation value stays low until the sliding window is aligned with the *s*. When all samples of *s* are included in the sliding window, we get the highest correlation value $C(s, y)_{peak} = H \sum_{k=1}^{L} |s[k]|^2$ where *H* is the channel impulse response.
When a node wins m channels out of n narrow channels, it may transmit the destination's unique sequence in the m narrow channels. The receiver keeps correlating received signal in each of the n channels with its own unique sequence. If a high correlation value occurs, the receiver knows that a channel is used by a node to send data to it. The method, however, requires that each channel is isolated by multiple filters. We use all available channels as one channel, aiming at removing the guard bands. We thus exploit the orthogonality between subcarriers in OFDM to remove the need of using sharp filters.

4.2.3.2 Partial Spectrum Correlation

Each node has been assigned a unique sequence as its signature. The signature is known to its neighbors through neighbor discovery (routing also requires neighbor information).

To indicate channels that are used for the receiver, the transmitter encodes the receiver's signature in the frequency domain. A node maps the receiver's signature to subcarriers of each channel that is occupied by it. Suppose a narrow channel contains L subcarriers. We design a signature sequence of length L for each node. The signature of a node cannot result in high correlation values with other nodes' signatures. Some polyphase codes possess such good correlation properties and we use the Zadoff-Chu (ZC) sequences [55], which is currently employed in LTE PHY layer for synchronization. The transmitter uses the signature of the receiver to modulate all L subcarriers in each obtained channel while in other channels it feeds 0s to the corresponding subcarriers.

In OFDM, the mutlicarrier modulation is efficiently accomplished through IFFT, which yields the sum of all modulated subcarriers. If the receiver is synchronized with the transmitter and knows the start of each OFDM symbol, the ZC sequence in each channel can be recovered by performing FFT on the right OFDM symbol. However, the receiver cannot be synchronized with the transmitter without knowing the occupied spectrum. We thus let the receiver generates the time domain signal by calculating the expected IFFT result when its signature is presented in a channel.

If the used channels are known, the receiver can construct the exactly same OFDM symbol generated by the transmitter. A high correlation value should be observed when the sliding win-

dow is aligned with the right OFDM symbol because they are similar to each other. However, the receiver does not know the channels used by the transmitter. We thus let the receiver calculate the expected IFFT results assuming its signature is present in each of the *n* channels. The receiver then correlates received signal with all expected IFFT results in parallel. When the sliding window is aligned with the right OFDM symbol, a correlation peak will be observed for each used channel but the value is 1/m of the correlation peak observed when all m channels are correctly used to reconstruct the OFDM symbol. The reason is that the expected result is only partially similar to the transmitted signal. We normalize the correlation value of the *i*th channel to $C(s_i, y) / \sum_{k=1}^{L} |s_i[k]|^2$. Experiments in performance evaluation show that when the signature is indeed present in a channel, correlating the received signal with the corresponding expected result will yield a high peak. Although the received signal is the sum of all subcarriers, the orthogonality means that the crosstalk between subcarriers sums up to zero. Only the interference in the channel that we are trying to identify has a significant impact on the partial spectrum correlation of the channel. However, the interference in the channel should be low because the transmitter has won it through bitwise arbitration. The transmitter would not choose to use the channel if the interference is strong in the channel. With the partial spectrum correlation, the receiver can identify channels used by the transmitter.

4.2.4 Synchronization Preamble Detection

Cross-correlation works without knowing the boundary of each OFDM symbol. To decode a packet sent by the transmitter, however, the receiver must synchronize to the start of each OFDM symbol and estimate the frequency offset. Encoding synchronization preamble over non-contiguous bands imposes a challenge of preamble detection at the receiver. OFDM is designed to work in a contiguous band. The preamble detection is based on a time domain delayed correlation property embedded in the preamble [56] [57]. Because some narrow channels in the wide band are occupied by other transmissions, the delayed correlation property may not hold on this mixed signal. Therefore, it is important to filter out unwanted signals in channels that are not used by the transmitter



Figure 4.4: Cross-correlation of received signal with expected IFFT results.

so that the preamble detection can be successful on a clean signal.

As shown in Fig. 4.2, once the transmitter wins a narrow channel, it uses all subcarriers in the channel for the receiver's signature. The receiver learns that there is an incoming transmission through high correlation peaks. These correlation peaks also indicate the channels used by the transmitter. The arbitration phase ends when collisions in all narrow channels are resolved. One more OFDM symbol is used to complete the spectrum agreement so that the transmitter can inform the receiver of the last gained channel. After the spectrum agreement, the transmitter transmits a reserved ZC sequence in all gained channels to indicate the end of the arbitration. If the receiver gets a high correlation value for this end-of-arbitration signaling, it starts to filter out signals that are not transmitted by the transmitter. This allows the receiver to detect the OFDM synchronization preamble and then it follows the standard OFDM transmission procedure.

4.2.5 Changing Channel Width

As discussed before, the node can monitor the medium state even during transmission as long as the ADC is not saturated. Therefore, when new channels become available, the node performs the medium access contention in new channels without interrupting the current transmission. Since the receiver filters out signals that are not in currently used channels. Even if the transmitter is sending some symbols in other channels for contention, the current reception will not be affected. When the transmitter wins some other channels, it inserts a control message in the current data transmission. The control message specifies the new spectrum use. The receiver then changes to receive data in a wider bonded channel.

4.3 Performance Evaluation

We evaluate the adaptive channel bonding (ACB) through both GNU Radio experiments and ns-2 simulations. The first step is to evaluate the collision detection and bitwise arbitration. We compare them with CSMA/CA to demonstrate that the bitwise arbitration is more efficient in medium

access contention. We then evaluate the spectrum agreement design, verifying that the receiver can be informed of used channels through partial spectrum correlation. Finally we evaluate the performance of ACB using simulations over various network configurations.

4.3.1 Medium Access Contention

There is a delay up to hundreds of microseconds between GNU Radio and USRP platforms as measured in some studies [48]. We can hardly see the difference between bitwise arbitration and random backoff if the system delay dominates the total delay. Therefore, we implemented the collision detection and bitwise arbitration in Verilog / VHDL on the FPGA of USRP E110. Because packet encoding and decoding are more complex, we implement them in GNU Radio. A packet is buffered on FPGA before transmission. Due to the limited FPGA resources, each packet is set to 200 bytes and the modulation scheme is QPSK.

To obtain full duplex wireless communication, we use the WBX transceiver, which has independent local oscillators (LOs) for transmitting and receiving chains. To reduce self-interference, we simply separate the transmitting and receiving antennas by about 2 feet. The performance of collision detection will be improved with better self-interference cancellation techniques [42] [32].

4.3.1.1 Reliability of Collision Detection

When a node is transmitting the compound preamble, it should not falsely detect noises as collisions; otherwise it may defer its transmission unnecessarily. We set one USRP E110 to the full duplex mode. Fig. 4.5 shows that when the node is transmitting, the noise floor is increased. If a node measures the noise floor during the interval when there is no transmission and uses the noise floor to determine whether there is a signal at a subcarrier, it may incorrectly regard that there is a collision. To reduce false alarm rate, the noise threshold is increased slightly.

A node first measures the noise floor when the channel is idle. Fig. 4.5 shows that there is a DC offset at the middle of the spectrum in USPR. If the DC offset has the highest magnitude, we regard that the channel is idle. When only one node is transmitting, it should not identify



Figure 4.5: Results of performing 64 point FFT on received signal (left: no transmission; right: transmitting preamble).

inactive subcarriers as active. We set the noise threshold slightly higher than the measured noise floor. A subcarrier is considered active if its magnitude is above the threshold. To get the detection accuracy, we compare the indices of detected active subcarriers with the indices of actually used subcarriers. As shown in Fig. 4.6, if the noise threshold is increased by 7 dB from the measured noise floor, it is unlikely to mistake an inactive subcarrier as an active subcarrier.

Once the noise threshold is determined, we study whether a node can detect the collision from a weak signal. We start another USRP E110 and gradually increase the transmission power. Fig. 4.7 shows that a node is unlikely to miss the detection of another node's compound preamble if the SNR is above 9 dB.

The design of the compound preamble makes it easy to detect. Normally, when two trans-



Figure 4.6: Setting noise threshold slightly higher than the noise floor to increase detection accuracy.

missions collide, the weak signal is buried under the strong signal. However, when two compound preamble transmissions collide, they do not affect each other because they use different subcarriers. As long as the SNR is above 9 dB, a collision is detectable. If two nodes use the same subcarriers, the aggregated power increases the collision detection probability at a third node. A collision is undetectable only if two nodes choose the same binary code, but the probability is reduced exponentially with the length of the binary code. If the self-interference cancellation is good enough, the collision is detectable even when two nodes choose the same binary code.

4.3.1.2 Throughput Gain

The bitwise arbitration reduces the medium access overhead, leading to higher throughput. We make three USRP E110 keep sending packets to a E110. The sampling rate of ADCs in a E100 is



Figure 4.7: The misdetection rate is reduced with higher SINR.

64 Mega samples per second (Msps). Because the decimation rate can be set to the multiples of 2, we evaluated the throughput at data rates of 0.5, 1, 2, 4, and 8 Msps. To compare with CSMA/CA, we use the parameters defined in 802.11n [41]. Fig. 4.8 shows the improvement of ACB over 802.11n.

We implemented a 64 point IFFT/FFT algorithm on FPGA of USRP E110. Because the middle part is affected by the DC offset, we only use 4 subcarriers at each side of the DC offset for medium access contention. With 8 subcarriers, the collision probability between any pair of nodes in bitwise arbitration is $C(n,2) \times (\frac{1}{2^8-1})^2 = C(n,2) \times (\frac{1}{255})^2$. On the contrary, in CSMA/CA the collision probability may be high with the small initial CW [0, 15]. To obtain the same low collision probability, 802.11 needs a fixed CW size of 255. Fig. 4.8 shows that the throughput of 802.11 is improved a little bit with the larger contention window, but the throughput is still low



Figure 4.8: Throughput improvement with bitwise arbitration.

because the average backoff time is increased with a larger contention window.

The throughput improvement of ACB over 802.11 is more significant at high speeds as shown in Fig. 4.8. At high speeds, the medium access contention is a large portion of a transmission where the actual data transmission is completed in a short time. The overhead of medium access contention in 802.11 is invariable to the physical layer (PHY) data rate. The throughput is thus constrained by the medium access overhead. On the contrary, in the bitwise arbitration, the time used to collect samples is reduced with higher data rates. The collision detection and bitwise arbitration is upper bounded by (1 + k) FFT frames when k bit binary codes are used for medium access contention. Suppose each FFT frame consumes a vector of 64 samples. When the data rate is 4 Msps, each FFT frame takes 16 μs . When the data rate is increased to 8 Msps, each FFT frame takes only 8 μs . The channel access overhead is thus reduced with higher data rates, allowing high PHY data rates to actually improve the throughput.

4.3.2 Spectrum Agreement

Once a node wins some narrow channels, it needs to inform the receiver of the occupied channels. To validate the effectiveness of partial spectrum correlation, we use 128 point IFFT/FFT on a band of 4 MHz. The band is divided into four channels of 1 MHz and each contains 32 subcarriers. The transmitter encodes the receiver's signature in channel 1 and 3. Fig. 4.9 and Fig. 4.10 show the correlation results when the receiver expects its signature to be present in channel 1 and channel 2, respectively. Because the receiver's signature is indeed present in channel 1, each time the sliding window is aligned with the signature, the receiver identifies a correlation peak. On the contrary, the receiver's signature is not encoded in channel 2. There is no significant correlation peak. Experiments prove that although the time domain samples are the sum of all modulated subcarriers, the receiver is still able to identify whether a subset of subcarriers are modulated with a known sequence.

The partial spectrum correlation takes all signals in the entire band for process. Although the transmitter does not encode the receiver's signature in channel 2, the correlation value may be high when the interference transmission in channel 2 is strong. This is because the correlation value $C(s,y) = \sum_{k=1}^{L} s^*[k]y[k]$ is also related to the received energy of y[k]. When the interference transmission is strong, the correlation value is increased but no obvious pulse is present. To detect a real peak in correlation, we calculate the moving average while performing the correlation. A threshold that is *x* dB above the moving average is used to determine whether a real peak occurs. Fig. 4.11 shows that 5 dB can exclude false detections of correlation peaks.

With the threshold discovered above, we check the required SINR for detecting the correlation peaks. We reduce the transmitter's transmission power to obtain the desired SINR. Each experiment checks the number of missed detections in 1000 compound preambles. An accurate detection should indicate that both channel 1 and channel 3 are used. Fig. 4.12 shows that the detection of used channels is reliable even at low SINR. Note that the transmitter has won these channels in bitwise arbitration phase, the interference from other nodes is expected to be low.



Figure 4.9: Partial spectrum correlation for channel 1 at SINR of 0 dB.



Figure 4.10: Partial spectrum correlation for channel 2 at SINR of 0 dB.



Figure 4.11: Setting threshold for correlation peak indication.



Figure 4.12: Probability of missing detections of occupied channels.



Figure 4.13: Throughput comparison with different number of contending transmitters.

4.3.3 Simulation Results

We evaluate the performance of ACB using simulations in ns-2.35. We simulate a scenario where several pairs of nodes contend for a band of 40 MHz and the band is divided into 8 channels of 5 MHz. With one antenna, the maximum data rate is 135 Mbps using 64-QAM modulation and 5/6 coding rate as indicated in 802.11 standard [41]. To study the difference between bitwise arbitration and random backoff, we do not add narrow channel interferers at the beginning. We increase the number of wide band transmitters from 1 to 30. All transmitters generate fully backlogged CBR traffic with packet size of 1500 bytes. Fig. 4.13 shows the throughput gain of ACB over 802.11.

If there is only one pair of nodes, the transmitter can transmit immediately after collision probe in ACB. Performing 128 point FFT needs 128 samples per frame. It takes 3.2 μ s to collect 128 samples at 40 MHz. The channel access overhead is a duration of 3.2 μ s. In 802.11, the average



Figure 4.14: The throughput of a wideband device is affected by the number of interference-free narrow channels.



Figure 4.15: The throughput of a wideband device is affected by narrowband devices' activities.

channel access backoff is $7.5 \times 9\mu s = 67.5 \ \mu s$ with the contention window [0, 15]. The difference in channel access overhead leads to the difference in throughput.

When more nodes are contending for medium access opportunities, collisions and enlarged contention window contribute to the decrease of throughput in 802.11. A node can infer a collision only after it completes the transmission. To enhance 802.11, we implemented the collision notification (CN) mechanism introduced in [40]. We assume an ideal case where a collision can be detected and notified whenever it happens. Fig. 4.13 shows that CN improves the throughput of 802.11 by 30% but has a marginal improvement over ACB. The result shows that the number of collisions in ACB is few due to the binary mapping and efficient bitwise arbitration.

When all nodes have the same channel width, they have equal chance to win in medium access contention. When some nodes change to use narrow channels, they get more medium access opportunities. To study the impact of narrow channel interference, we add narrowband devices to the 8 channels one by one. Because narrowband devices work independently in the 8 channels, they are likely to win medium access as they do not contend with devices in other channels. On the contrary, a channel bonding device in 802.11 has to contend with all narrowband devices. If any channel is occupied, the channel bonding device cannot transmit. As shown in Fig. 4.14, when all eight narrow channels are occupied by narrowband devices, the channel bonding device nearly has no chance to transmit.

In frame-based channel bonding, a wideband device keeps counting down the random number as long as one channel is available. When the random backoff is completed, the wideband device transmits in all channels that are still idle. Fig. 4.14 shows that the method allows a wideband device to survive in a network where many uncoordinated narrowband devices exist. The spectrum resource, however, is not fully exploited. ACB lets a wideband device monitor the state of the entire band even when the device is transmitting. Whenever a channel becomes available, the wideband device will participate in the contention and it may win the channel for further bonding. In joint with the efficient parallel bitwise arbitration, the throughput of channel bonding is the highest under the severe narrow channel interferences. When there exist narrowband devices in all eight narrow channels, the 802.11 channel bonding can still provide service if the narrowband devices are not very active as shown in Fig. 4.15. However, if narrowband devices always have data to transmit, the wideband device can hardly find a gap when all narrow channels are idle. ACB allows a wideband device to initiate a transmission as long as some channels are available. The chance to deliver a packet is higher. ACB also bonds other channels whenever they become available. The throughput is thus higher than frame-based channel bonding where new channels can be used only in the next frame.

4.4 Summary

In this chapter we introduce an adaptive channel bonding (ACB) protocol to address the inefficiency and unfairness issues caused by coexistence of heterogeneous radios. The ACB introduces a parallel bitwise arbitration mechanism to address the severe contention in wideband spectrum sharing and a partial spectrum correlation method to achieve fast spectrum agreement. Experiments on testbed validate the effectiveness of ACB. Extensive simulations show that ACB allows a wideband device to obtain medium access opportunities even under intense narrowband interferences and the throughput is improved by 20% over current frame-based channel bonding.

Since the contention issue between devices is addressed, we move forward to improve data transmission when a node is granted the permission to transmit. As nodes contend for wide channels, it is unavoidable that they will experience frequency diversity. The next chapter discusses rich information brought by diversities.

CHAPTER 5

EXPLOITING MODULATION DIVERSITY IN MULTICARRIER WIRELESS NETWORKS

The previous chapter shows that a trend in wireless communication is to opportunistically aggregate unused spectrum fragments to obtain wider spectrum bands for higher data rates. However, a rapidly modulated wideband signal is subject to severe intersymbol interference (ISI).

Orthogonal frequency division multiplexing (OFDM) addresses the ISI problem by dividing available frequency spectrum into many closely spaced orthogonal subcarriers and transmitting data on each of them at a lower symbol rate in parallel. The low symbol rate makes the guard interval between symbols affordable because the guard interval is relatively short compared with the symbol duration. Because OFDM is robust to harsh channel conditions such as narrowband interference and frequency-selective fading, it is employed in many wireless communication systems.

As wireless networks move toward wider channels, it is common that each subcarrier experiences a different fade. Some protocols propose to adapt rate and power based on subcarrier channel quality [58] [59]. Since there are several different modulation schemes, the modulation diversity contains rich information to exploit.

In this chapter, we propose encoding modulation schemes to utilize the modulation diversity to increase network throughput. In an OFDM system, data bits are allocated to subcarriers and mapped to modulation symbols (e.g., 00 is mapped to $-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}j$ in QPSK). Different modulation schemes can deliver different number of bits with a single modulation symbol. We encode each modulation scheme with a few bits (e.g., BPSK: 1001, QPSK:1100). When data bits are allocated to a subcarrier, we first check whether the bits to be transmitted match the code defined for the modulation scheme on the subcarrier. If it is true, we do not map bits to a modulation symbol on the subcarrier. Instead, we alternate the subcarrier state to inform the receiver of the condition. If the receiver detects the subcarrier state change, the receiver knows that a predefined sequence of

bits should be inserted into the data stream. The receiver does not need to estimate the symbol transmitted by the transmitter.

One challenge is to find the optimal code for each modulation scheme so that the code can be used multiple times in the process. Suppose k bits can be represented by a symbol in a modulation scheme (e.g., 2 bits per symbol in QPSK) and the modulation scheme is encoded with l bits. Each time the code is used, the speed on the corresponding subcarrier is increased because more bits are transmitted if l > k. In addition, the detection of a subcarrier state change is much reliable than decoding a modulation symbol. Therefore, the known sequence of bits helps eliminate incorrect paths in decoding convolutional codes, leading to a lower bit error rate (BER). Because fewer error bits need to be retransmitted, the throughput is increased.

The rest of the chapter is organized as follows. Section 5.1 summarizes related work. Section 5.2 introduces the modulation scheme coding and Section 5.3 presents how to utilize the modulation scheme codes. The evaluation results are reported in Section 5.4 and we finally conclude the chapter in Section 5.5.

5.1 Related Work

Many measurements [58] [60] [61] have demonstrated the existence of frequency diversity in RF spectrum. FARA [58] exploits the frequency diversity by adapting bitrate based on the subcarrier SNR reported by the receiver. It calibrates each node to find the minimum required SNR for a particular bitrate. When a node receives a packet, it estimates the SNR and selects an appropriate bitrate for each subcarrier. The node tells the transmitter to move up or move down to the next bitrate or stay at the current bitrate by using the ACK. The method, however, needs a lot of work for calibration.

JPRA [59] further combines rate adaption with power allocation based on the error vector magnitude (EVM) measured on each subcarrier. It performs calibration measurements to find the relationship between power change and average EVM change for a particular link. To reduce the EVM on a subcarrier, JPRA allocates more power to the subcarrier under the constraint that the

total power budget is fixed. JPRA proposes an algorithm to ensure that the minimum supported bitrate can be increased after the power redistribution. It also proposes an algorithm to minimize the transmission time by concentrating power on some subcarriers to support higher bitrates. Same as FARA, the calibration for each link makes it suitable only for static networks.

Bhartia et al. [62] leverage the Channel State Information (CSI) to find reliable subcarriers and map important symbols to them. This helps increase throughput because more useful bits are received. This work also leverages CSI for partial FEC group recovery. Even when an FEC group fails, it extracts data symbols whose SNR is sufficiently high. In addition, they add an extra MAClayer FEC on top of physical layer FEC. One limitation is that the work is only compatible with block FEC schemes.

The diversities on subcarriers motivate us to exploit them to increase data rate and improve channel decoding. Several protocols have been proposed to exploit source redundancy or protocol signature to improve the performance of channel decoding such as SoftCast [63], ParCast [64], and LEAD [65]. Different from these schemes that search for useful bits between packets, we utilize useful bits within a packet.

5.2 Modulation Scheme Coding

Many studies have revealed that different subcarriers exhibit very different signal-to-noise ratios (SNRs) due to frequency-selective fading [58] [59] [62]. Rate adaption [58], power allocation [59], and bit interleaving [62] are developed to be smart on subcarrier SNR variation. The diversity brings another dimension to deliver data.

In a typical OFDM system, data bits are allocated to subcarriers and mapped to modulation symbols. The number of bits that can be represented by a modulation symbol depends on the adopted modulation scheme. One subcarrier may adopt BPSK (1 bit per symbol) while the other subcarrier may adopt 64-QAM (6 bits per symbol). The choice is usually determined based on the subcarrier SNR [58].

To represent these modulation schemes, we encode each modulation scheme with l bits. When allocating bits to a subcarrier, we first check whether the next l bits match the code defined for the modulation scheme on the subcarrier. If it is false, k bits are allocated to the subcarrier and mapped to a modulation symbol as usual. On the contrary, if it is true, we define it as a hit on the code. Instead of mapping k bits to a symbol on the subcarrier, l bits are allocated to the subcarrier for transmission. The l bits are transmitted through subcarrier state change as introduced in Section 5.3. Because l - k more bits are transmitted, the speed is increased. The improvement lies on the difference between l and k and the frequency of having a hit on the code. A longer modulation scheme code has a lower hit probability. Therefore, there is a tradeoff between modulation scheme code length and hit probability.

5.2.1 Challenge

The challenge is to find the optimal code for each modulation scheme. To increase the hit probability, we need to find bit patterns (i.e., sequences of bits) that appear frequently in data bits. A problem is that when a bit pattern is assumed to be the code for a modulation scheme, the allocation of data bits to subcarriers is completely changed.

An example is illustrated in Fig. 5.1. In a typical OFDM system, the first two bits are assigned to the first subcarrier because it adopts QPSK. The third bit is assigned to the second subcarrier, which uses BPSK. Suppose "1100" appears the most often when allocating bits to a subcarrier that uses QPSK. We may encode QPSK as "1100". The encoding means that 4 bits instead of 2 bits will be assigned to the first subcarrier if we employ the modulation scheme coding (MSC). The consequence is that all of the bit allocations have to be changed. The fifth bit instead of the third bit should be assigned to the second subcarrier. If previously "0011" appears the most often when allocating bits to a subcarrier that uses BPSK, the discovery may not hold any more. Due to the shift in bit allocation, the code identified for each modulation scheme may not be the optimal one based on the new assignment. Therefore, assuming a bit pattern will be the code for a modulation scheme will change the entire assignment and impact the result discovered earlier.



Figure 5.1: OFDM Transmitter.

5.2.2 Splitting to Sets

The code selected for a modulation scheme impacts the code choice for the other modulation schemes, which in turn puts doubt on the code choice for the current modulation scheme. To address the problem, we spit data bits into sets. Data bits that are assigned to subcarriers with the same modulation scheme are organized in the same set. The code for each modulation scheme is then selected independently.

Taking Fig. 5.1 as an example, no matter which code will be used for QPSK, the third bit is assigned to the second subcarrier that uses BPSK. The code used for QPSK will not affect the code choice for BPSK.

In decoding, the receiver understands that the additional bits represented by modulation scheme code are taken from subcarriers that use the same modulation scheme. For example, suppose l = 4 bits are assigned to a subcarrier that uses QPSK. When the receiver decodes 4 bits instead of 2 bits from the subcarrier, it knows that the first two bits are from the current subcarrier and the additional 2 bits are taken from the next subcarrier that use the same modulation scheme. It uses the right order to organize bits into bytes.

5.2.3 Identifying Modulation Scheme Code in a Set

After data bits are split into sets, the bit pattern that appears the most often in a set is identified as the code for the corresponding modulation scheme. Suppose the number of bits that can be



Figure 5.2: The code length for a modulation scheme impacts the shift in bit allocation.

represented by a symbol in a modulation scheme is k. We require that the length of the code must be an integer multiple of k. This ensures that the shift in bit allocation is an integer multiple of k, and thus the bits assigned to subcarriers do not change. The same segment of bits just move from one subcarrier to another subcarrier.

Fig. 5.2 shows an example for QPSK. Normally two bits are assigned to a subcarrier. Suppose the bit pattern "0011" appears the most often. It is selected as the code for QPSK. The decision does not change the pairing of bits. If a pair of bits is assigned to the *i*th subcarrier, once the modulation scheme code is determined, the two bits are shifted to another subcarrier together. The shift does not change the fact that "0011" appears the most often when allocating bits to subcarriers with the same modulation scheme.

However, if the code length is not an integer multiple of 2, the bits assigned to subcarriers will change and the selected code may not be the optimal one any more. In the example illustrated in Fig. 5.2, the bit pattern "001" appears when allocating bits "00" to the second and the sixth subcarriers. It appears twice in total. Suppose "001" is selected to be the code for QPSK. The pairing of bits is changed. The bit pattern "001" only appears one time when allocating bits to subcarriers, which implies that the selected code may not be the optimal one after the change of bit allocation. Therefore, the code length constraint enables us to find the optimal code without examining too many possibilities.

The modulation scheme code length and the hit probability together affect the total number of bits that can be exploited. If the code length is long, the improvement on speed is larg in a single hit (l - k more bits are transmitted). However, the hit probability is lower for a longer code. Therefore, the code length is selected to maximize the total number of additional bits that can be transmitted through the modulation scheme coding. It is to find a length-*l* bit pattern that appears the most often in a set of bits. This can be obtained by scanning the set of bits to count the number of occurrences of patterns. The pattern mining algorithm is summarized in Algorithm 5.

```
Algorithm 5: Frequent Bit Pattern Mining
Input : A set of bits S
 Output: A bit pattern P
 k \leftarrow number of bits represented by a symbol;
L \leftarrow size(S);
m \leftarrow 1;
n, n' \leftarrow 0;
                // number of additional bits represented by modulation scheme
 code
 P' \leftarrow empty string;
 while n' >= n \operatorname{do}
     n \leftarrow n':
     P \leftarrow P':
     C \leftarrow empty key/value container;
     m \leftarrow m + 1;
     l \leftarrow k * m;
                                     // code length must be an integer multiple of k
     i \leftarrow 0;
     while i < L - l do
         s \leftarrow S[i:i+l];
         if s in C then
             if s not overlap with the previous segment that has the same bit pattern then
                C[s] \leftarrow C[s] + 1;
              end
         else
            C[s] \leftarrow 1;
         end
         i \leftarrow i + k;
     end
     sortedC \leftarrow sort(C);
     P' \leftarrow the key of sortedC[0];
     n' \leftarrow (l-k) * C[P'];
 end
 return P;
```

5.3 Subcarrier Nulling

Fig. 5.1 shows that when data bits flow into an OFDM system, they are allocated to each subcarrier according to the adopted modulation scheme. On each subcarrier, the bits are mapped to a modulation symbol (e.g., 0 is mapped to -1+0j in BPSK, 00 is mapped to $-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}j$ in QPSK). When all subcarriers get the symbols, an inverse discrete Fourier transform (IFFT) algorithm is used to obtain orthogonality between subcarriers. The input of the IFFT block is *N* modulation symbols from the *N* subcarriers, and the output is *N* complex numbers that are referred to as one OFDM symbol.

After modulation schemes are encoded, subcarrier nulling is employed to inform the receiver that the code is used on a subcarrier in a particular OFDM symbol. When the transmitter allocates bits to a subcarrier, it may detect that the next *l* bits match the modulation scheme code. Instead of mapping *k* bits to a symbol on the subcarrier, the transmitter nulls the subcarrier by inputting the complex symbol 0+0j to the subcarrier. Since no signal is transmitted on the subcarrier, the power on the subcarrier is 0. If the receiver detects the state change, it inserts the known *l* bits in the data stream instead of trying to decode a symbol on the subcarrier.

When a radio signal radiates away from the transmitting antenna, it spreads out in different directions. Parts of the spreading wave may reflect off objects and arrive at the receiving antenna from different paths. The received signal is thus subject to multipath interference. The received symbol is not identical to the transmitted symbol on a subcarrier. Both the power and the phase are changed due to the multipath interference. In addition, because the transmitter and the receiver may not be perfectly synchronized on the frequency, the frequency offset also contributes to the accumulated phase shift of the received symbol. Fig. 5.3 shows that the received symbol may be closer to a constellation point that is not the one transmitted. Error bits are introduced.

On the contrary, if no signal is transmitted on a subcarrier, the receiver just detects white noise. It is less susceptible to multipath interference. In decoding, the receiver will notice that the received symbol on the subcarrier is very close to the origin of the constellation diagram. It thus knows that the subcarrier is deactivated and a known sequence of bits should be inserted. Although the nulling



Figure 5.3: Constellation Diagram with Subcarrier Nulling.

is subject to interference from other transmissions, we consider that the contention issue has been addressed by the MAC protocols [35] [1] [36] [66]. We reduce error bits caused by multipath interference and attenuation.

5.4 Performance Evaluation

In this section, we first study whether there are bit patterns that can be used as modulation scheme codes in real-world network traffic. We then validate the efficacy of subcarrier nulling in informing the receiver of the need to insert a known sequence of bits. We finally evaluate the benefits introduced by using the modulation scheme codes.

5.4.1 Modulation Scheme Code Abstraction

The first question is how many bits can be utilized with the modulation scheme coding (MSC). To obtain real-world traffic data, we use the SIGCOMM'08 trace [31] that records wireless network activities in the conference. Because the trace does not contain the complete content of a packet, we use Wireshark network protocol analyzer [67] to capture Wi-Fi traffic in a home network. The measurement collects traffic caused by video streaming, online music, web browsing, email service, wireless printer, and etc. in a home network. The captured packets are scrambled and encoded with a convolutional encoder to get the raw bits to be transmitted at the physical layer.

Although frequency diversity has motivated some frequency-aware designs, current implementations specified by the IEEE standard [41] use the same bitrate for all subcarriers and evenly distribute the transmission power across all subcarriers. Therefore, although some subcarriers may support higher density modulation schemes, all subcarriers have to adopt the modulation scheme that is supported by all subcarriers. If one subcarrier can only support BPSK, all subcarriers have to adopt BPSK. In such a system, Fig. 5.4 shows that there is a potential to transmit $7\% \sim 26\%$ more bits through MSC when a low density modulation scheme is used. When a high-density modulation schemes is used, the number of bonus bits introduced by MSC is small. The reasons are twofold.

First, as shown in Fig. 5.5, before bits are assigned to a subcarrier, a segment of bits is extracted to check whether or not it matches the modulation scheme code. If it is false, only one bit is assigned to the subcarrier in BPSK but four bits are assigned to the subcarrier in 16-QAM. The window is then moved by a different number of bits. It is obvious that there are more opportunities to have a hit on the modulation scheme code in low-density modulation schemes. Second, when we have a hit on the modulation scheme code, l - k more bits are transmitted in comparison with the symbol modulation. Given a code length of l, the number of bonus bits is larger when k is smaller in a lower density modulation scheme. For instance, suppose the code length is five. If we have a hit on the modulation scheme code, 5 - 1 = 4 more bits are transmitted in BPSK whereas only 5 - 4 = 1 more bit is transmitted in 16-QAM. Due to the two reasons, the number of bonus



Figure 5.4: The percentage of bonus bits and the percentage of total bits that can be represented by modulation scheme codes.



Figure 5.5: There are more opportunities to use the modulation scheme code when a low-density modulation scheme is used.

bits that can be introduced by MSC is small in high-density modulation schemes.

Although a transmission may not benefit from the bonus bits when a high-density modulation scheme is used, the transmission can benefit from the known bits that are represented by the modulation scheme codes. When the subcarrier state changes, the receiver knows that a predefined sequence of bits, which is the corresponding modulation scheme code, is transmitted by the transmitter. The known bits help improve decoding performance.

Fig. 5.4 shows that there are abundant known bits when low-density modulation schemes are used because of the high probability of having a hit on the modulation scheme code. In high-density modulation schemes, the opportunities to use modulation scheme codes are few. The number of known bits that can be exploited is small. To increase the number of known bits, we use more than one code for a high-density modulation scheme.

Instead of trying to deliver more bits when having a hit on the modulation scheme code, we increase the probability of having a hit on the modulation scheme code in high-density modulation schemes (i.e., any scheme higher than QPSK). If k bits are represented by a symbol in a high-density modulation scheme, we fix the code length to k for the modulation scheme. On each subcarrier that adopts the modulation scheme, a code is selected with a target to maximize the number of bits that can be represented by the code. Fig. 5.6 shows that the number of bits that can be represented to above 10 percent even in high-density modulation schemes.



Figure 5.6: The percentage of total bits represented by modulation scheme codes when more than one code is used for a high-density modulation scheme.

When modulation schemes are selected independently for each subcarrier, the number of bits that can be exploited through MSC is impacted by the ratio between high-density modulation schemes and low-density modulation schemes. The 802.11 standard [41] specifies 48 data subcarriers for the 20 MHz channel spacing. Fig. 5.7 shows that when most subcarriers adopt the low-density modulation scheme, more bits can be exploited. When high-density modulation schemes dominate, there are still at least 10% bits that can be exploited by MSC.

In summary, when channel quality is good and high-density modulation schemes are used on most subcarriers, MSC introduces at least 10% known bits to help decoding. When the channel quality degrades and low-density modulation schemes are used, more bits are available to improve decoding and the transmission time is shortened because a modulation scheme code represents more bits than a symbol in a low-density modulation scheme.

5.4.2 Subcarrier Nulling Performance

The subcarrier state change is indicated through subcarrier nulling. It is the method to inform the receiver that a known sequence of bits should be inserted. If the receiver fails to recognize a nulled subcarrier, it may result in deletions of bits from the data stream. Conversely, if the receiver recognizes a modulation symbol as a nulled subcarrier, more bits may be inserted to the data stream. The unknown insertions and deletions together with the error bits introduced by the subcarrier nulling will make the error correction harder to implement. It is thus important to recognize nulled subcarriers reliably.

Because the noise is scaled by the channel impulse response estimation, a lower SNR will lead to a higher error rate in detecting nulled subcarriers. To get the minimum required SNR for reliable detection of nulled subcarriers, we conduct experiments on USRP1 with WBX transceiver. We gradually increase the SNR by increasing the transmitting and receiving gains. Fig. 5.8 shows that when the detection error rate is reduced to below 0.001, the bit error rate is still above 0.1 at the same SNR. Therefore, when the subcarrier nulling becomes a reliable way to inform the receiver of subcarrier state change, there is still a large potential gain in reducing bit error rate.



Figure 5.7: The percentage of total bits represented by modulation scheme codes when different modulation schemes are used for different subcarriers (BPSK:QPSK:16-QAM:64-QAM).



Figure 5.8: At the same SNR, the detection error rate is much lower than the bit error rate.

5.4.3 Bit Error Rate Reduction

To reduce the BER, we use the proposed MSC to assist decoding. On the one hand, some bits are reliably transmitted to the receiver by subcarrier nulling. On the other hand, the known bits are able to eliminate some incorrect decoding paths, and thus more bits can be decoded correctly.

To study the decoding performance, we compare the BERs of packets decoded by the standard BCJR algorithm [68] and MSC. Fig. 5.9a shows that MSC reduces more than 90 percent bit errors when BPSK is adopted. This is because there are many opportunities to utilize the modulation scheme code. When high-density modulation schemes are used, the MSC is still able to reduce 30 percent to 50 percent bit errors. When all modulation schemes are mixed, the performance is averaged to around 40 percent reduction in BER.

Usually when the BER is too high, the rate adaption algorithms will adjust the modulation to a lower density modulation scheme. To study the MSC performance in a rate adaption system, we



(a) BER reduction at SNRs when the detection error rate for nulled subcarriers is below 0.001



(b) BER reduction at SNRs when the BER is controlled to around 0.01

Figure 5.9: The BER is reduced due to the known bits introduced by MSC.

increase the SNR to maintain the BER around 0.01. Fig. 5.9b shows that 30 percent to 80 percent bit errors are reduced depending on adopted modulation schemes.

5.4.4 Throughput Improvement

The MSC introduces some bonus bits for low-density modulation schemes, which enable the transmitter to shorten the transmission time. Each time when the modulation scheme code is used, more bits are transmitted in an OFDM symbol. Since a modulation symbol in a low-density modulation scheme can only represent one or two bits, a few bonus bits transmitted through subcarrier nulling can reduce a large number of OFDM symbols. For example, there are 48 data subcarriers in 802.11 for 20MHz channel. One OFDM symbol can transmit 48 bits in BPSK and 48 * 2 = 96 bits in QP-SK. Therefore, if there are 192 bits are transmitted as bonus bits in MSC, the transmission time is shortened by two OFDM symbol duration (i.e., 8 *us*) in QPSK but four OFDM symbol duration (i.e., 16 *us*) in BPSK.

For high-density modulation schemes, the transmitter mainly benefits from the error correction introduced by MSC. Because fewer error bits need to be corrected, the throughput can be increased. We integrate MSC with Maranello [69], which is a block retransmission based partial packet recovery protocol. In Maranello, a packet is divided into blocks and the receiver uses NACKs to inform the transmitter of corrupted blocks. The transmitter retransmits only the corrupted blocks instead of the entire packet.

Fig. 5.10 shows that the lower BER in MSC leads to fewer blocks to be retransmitted. The throughput is improved by 4 to 7 percent when QAM is used. When BPSK and QPSK are used, the throughput gains are 1.59x and 1.27x, respectively. Part of the gain is from the shortened transmission time. In low-density modulation schemes, a number of bonus bits are transmitted through subcarrier nulling. If they were modulated as symbols, a few more OFDM symbols are needed and the transmission time would be longer.



Figure 5.10: The throughput gains introduced by MSC.

5.5 Summary

This chapter views the opportunity of utilizing modulation scheme diversity as another dimension to increase network throughput. We propose a method to find the optimal code for each modulation scheme. The subcarrier nulling is employed to inform the receiver of the subcarrier state change so that the receiver understands it needs to insert the modulation scheme code to the data stream. The real 802.11 traffic data are used to validate that there exist chances to utilize modulation scheme diversity for throughput gain. The performance study shows that the modulation scheme coding method increases the throughput by reducing both transmission time and bit errors.

So far we assume that the spectrum availability is consistent between the transmitter and the receiver. In dynamic spectrum access, it is highly possible that the transmitter and the receiver have different available spectrum fragments. The next chapter studies the spectrum disparity problem, making the transmitter use the spectrum fragments that are also available at the receiver side.

CHAPTER 6

EFFICIENT BROADCAST ON FRAGMENTED SPECTRUM

Current spectrum-agile designs [51] [52] [54] assume that the spectrum availability is the same between the transmitter and the receiver. We consider that the spectrum resources at the two communication sides may be different, especially in broadcast-based communications where different receivers may obtain different spectrum fragments to use at different locations. The nonuniform spectrum availability imposes special challenges on broadcast.

In a network, broadcast is a fundamental mechanism for route discovery, code update, disseminating control information and alarm to all nodes. In a traditional single-channel or multi-channel network, a control channel can be reserved and all nodes tune to the control channel for broadcast. In a cognitive radio network (CRN), a common control channel is not guaranteed to be available due to the nonuniform spectrum availability. As a result, a node may have to broadcast multiple times in different channels to reach all neighbors. The scheduling of broadcast usually targets at minimum latency to reach all nodes [70] [71] [72]. If the channel to which the target receiver listens is unknown, channel rendezvous designs are also necessary [73] [74].

When a device operates in a wideband channel, it is beneficial to view the wideband channel as an aggregation of multiple narrow channels. Fast growing deployment of wireless devices urges heterogeneous radios to coexist with each other. It is difficult to obtain a wide band of spectrum for exclusive use. Dividing a wideband channel into multiple narrow channels for independent evaluation can increase the flexibility of spectrum use. As long as the available spectrum fragments in a wideband channel can meet the bandwidth demands, nodes can stay in the channel. This avoids frequent channel switch and the associated overhead of blind channel rendezvous. However, the dynamic spectrum access requires that the transmitter and the receiver can efficiently reach an agreement on how to use the spectrum fragments.

Moreover, the transmitter should be aware of the receiver's spectrum availability. In a broad-
cast, the spectrum resources at different receivers may be different. Most of existing studies assume that a dedicated control channel can be used to collect the information or channel hopping patterns can be designed to guarantee rendezvous. It is not always practical to have a common control channel, and the broadcast based on channel rendezvous is not efficient. This chapter introduces a Spectrum Fragment Agile Broadcast (SFAB) protocol that exploits the partial spectrum correlation to achieve spectrum agreement between the transmitter and the receivers efficiently. Based on the collected spectrum availability information, the transmitter determines how to divide a packet into segments and allocate them to different narrow channels to maximize the throughput. In summary, our contributions are:

- We introduce an efficient handshake mechanism for fast spectrum agreement between the transmitter and the receivers.
- We introduce an efficient interleaving mechanism for maximizing the bandwidth used for broadcast to improve the throughput.
- We evaluate the performance of the Spectrum Fragment Agile Broadcast (SFAB) through both experiments and simulations.

The rest of the chapter is organized as follows. In Section 6.1, we discuss related work in broadcast and spectrum-agile designs. Section 6.2 outlines the problem and challenges in supporting broadcast on fragmented spectrum. The detailed design is presented in Section 6.3. Following the performance evaluation in Section 6.4, we conclude this chapter in Section 6.5.

6.1 Related Work

Most studies on broadcast in CRNs consider a channel as a contiguous band of spectrum for exclusive use. In [70], the topology and the available channels of each node are assumed to be known. By introducing the concept of minimal neighbor graph, the work selects the minimal set of channels to reach all neighbors. As a result, the transmitter only needs to broadcast a packet in each of the channels sequentially instead of broadcast the same packet in all channels. However, the work does not consider the possibility that the neighbor may not listen to the channel selected by the transmitter. A jump-stay sequence is proposed in [75] to ensure channel rendezvous in broadcast.

To deliver a message to all nodes with the minimum latency in a multihop CRN, the work in [76] formulates the Minimum-Latency Broadcast Scheduling (MLBS) problem as an Integer Linear Programming (ILP) problem and proposes two polynomial-time heuristic solutions. A recent study in [77] further reduces the latency and redundancy in delivering a message to all nodes in a multihop CRN.

For CR devices that operate in a wideband channel, several spectrum-agile designs have been proposed to efficiently utilize the spectrum fragments [51] [52] [54]. Both SWIFT [51] and Jello [52] exploits the non-contiguous orthogonal frequency division multiplexing (NC-OFDM) to support communication over fragmented spectrum. However, control packets are needed to let the transmitter and the receiver agree on what spectrum fragments will be used. RODIN [54] introduces 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 achieve fast spectrum agreement.

Cross-correlation is an efficient technique to identify the presence of a signal buried under noise. It has also been used as a way to indicate collisions [40] and replace traditional control messages [78]. In these unicast communication designs, the spectrum availability between the transmitter and the receiver is assumed to be the same. In broadcast-based communication, multiple receivers may obtain different spectrum fragments to use as they are distributed at different locations. A spectrum fragment agile broadcast protocol is thus desired.

6.2 Overview

Many spectrum-agile designs have demonstrated that it is feasible to communicate over fragmented spectrum [51] [52] [54]. The communication, however, gets complicated when available spectrum



Channels

Figure 6.1: Broadcast on fragmented spectrum. Shaded channels are occupied.

resources are different at different nodes. This is highly possible in broadcast-based communications in CRNs where spectrum is opportunistically utilized.

6.2.1 Broadcast Problem

Fig. 6.1 illustrates the challenges of supporting broadcast over fragmented spectrum. Suppose nodes can operate in a wideband channel. We divide the wideband channel into multiple narrow channels for independent clear channel assessment (CCA). The CCA for multiple narrow channels can be done simultaneously through checking the power spectral density (PSD) of the entire wideband channel [52]. In the remaining part of the chapter, a channel means a narrow channel unless otherwise stated.

Fig. 6.1 shows that there exists a common channel. The case is trivial if we are only interested in finding one narrow channel to broadcast the message. However, node *A* can divide a packet into three segments (e.g., P_1 , P_2 , and P_3) and transmit them simultaneously to increase the throughput. Node *A* can broadcast P_1 in channel 3, P_2 in channel 1 and channel 5, P_3 in channel 2 and channel 4. All nodes can receive all of the segments and the throughput is tripled if each receiver knows from which channel it should decode the packet. In addition, the common channel may not exist. The transmitter has to rely on channel combinations.

When the transmitter broadcasts in a channel that is not available to all nodes, the transmission may cause interference to the original channel occupant at some receivers' locations. Power control and other interference-cancellation techniques [79] can be used to limit the interference. For

example, the work in [79] successfully enables Wi-Fi to work properly under the interference of baby monitors and cordless phones. In addition, beamforming can be effectively used to reduce the interference to the original channel occupant.

6.2.2 Challenges

To provide spectrum fragment agile broadcast, three major challenges need to be addressed. First, the sender needs to inform the receivers of the channels that are available at the sender side. Second, each receiver should respond with the narrow channels in which it is able to receive data without interference. Third, the sender needs to determine how to divide a packet and allocate them to different narrow channels. The sender should find the best channel combination scheme in which all receivers can receive data with the maximum channel width.

In the example illustrated in Fig. 6.1, node *A* may first identify channels that cover all receivers (i.e., channel 3). Further, Node *A* may identify the next channel that covers the most number of nodes (e.g., channel 1) and check which channel can be used as a complement to cover all receivers. In the example, although both channel 4 and channel 5 can be used along with channel 1 to cover all nodes, channel 5 should be selected in order to obtain the best channel combination scheme. The cost of taking a channel at a certain step includes the effect on other possible channel combinations.

In the example illustrated in Fig. 6.1, using either channel 4 or channel 5 contributes to deliver data to one node (i.e., node C) but wastes spectrum for other nodes (note that they can obtain data from channel 1). The profit/cost ratio seems to be equal for using channel 4 or 5, but we cannot draw the conclusion without examining the effect on all other possible channel combinations.

In fact, a channel that does not cause any waste of spectrum should be selected first. Fig. 6.2 shows a extended example where it is better to use channel 6 along with channel 1. However, similar to the example in Fig. 6.1, we may need to consider the consequence if we use the channel at this step. An efficient broadcast mechanism in CRNs has to identify spectrum availability efficiently and utilize them wisely.



Figure 6.2: A extended example of broadcast on fragmented spectrum.

6.3 Design

This section presents the design of the Spectrum Fragment Agile Broadcast (SFAB) protocol.

6.3.1 Spectrum Agreement

To establish communication, the sender and the receiver need to agree on narrow channels that will be used. This is vital for unicast in CRNs too. SFAB achieves the efficient spectrum agreement through partial spectrum correlation. We first introduce a method that is easy to understand but requires more resources to implement, and then discuss how partial spectrum correlation can reduce the complexity.

6.3.1.1 Correlation

Correlation is an efficient technique to detect a known waveform in random noise. Let t[m] denote the target signal that contains N values, and x[n] denote the received signal. Correlation uses the two signals to produce a third signal y[n], which is called the cross-correlation of the two input signals [45]. The cross-correlation y[n] can be written as

$$y[n] = \sum_{m=0}^{N-1} t^*[m]x[m+n]$$
(6.1)

where $t^*[m]$ is the complex conjugate of t[m].

If a node is expecting a known pseudo-random noise (PN) sequence, it may keep correlating the known sequence with the received signal to check the presence of the known sequence. If the known sequence is not present in the received signal, low cross-correlation values should be observed because other signals are supposed to be independent of the sequence and the similarity identified by the cross-correlation should be low. If a neighboring node indeed transmits the sequence, the receiver can detect the presence of the sequence when a high cross-correlation peak is detected. The cross-correlation value is maximized when the signal of interest is aligned with the matching waveform in the received signal. The technique can be used to achieve fast spectrum agreement.

To detect potential simultaneous broadcasts from different nodes, we need a set of sequences to differentiate broadcast intentions from different nodes. Each broadcast intention sequence should have a high correlation value with itself and low correlation values with others. One candidate sequence is the Zadoff-Chu (ZC) sequence [80], which is currently employed in LTE PHY layer for synchronization.

A ZC sequence $s_{\gamma}[m]$ is defined as [80]

$$s_{\gamma}[m] = e^{\left(-j\frac{\gamma\pi m(m+(L \mod 2))}{L}\right)}$$
(6.2)

for m = 0, 1, ..., L - 1, where *L* is the length of the sequence, and γ is a positive integer (called the root index) relatively prime to *L*. Given a ZC sequence, we can create L - 1 more ZC sequences by cyclically shifting the sequence. The cross-correlation of the original ZC sequence and any of the L - 1 cyclically-shifted versions is zero. Only when the ZC sequence is correlated with itself, a high correlation value of *L* occurs. The root index γ can be adjusted to create more ZC sequences, but the cross-correlation of two prime length ZC sequences is increased to $1/\sqrt{L}$, which is small if *L* is large.

6.3.1.2 Broadcast Intention and Broadcast Identification

Given a root index γ , a set of cyclically shifted ZC sequences $\{bs_0, bs_2, ..., bs_{L-1}\}$ are predefined as broadcast intention sequences. When a node has data to broadcast, it randomly selects a se-



Figure 6.3: The cross-correlation value is maximized when the base sequence is aligned with itself at a certain time lag of the received signal.

quence to use. In transmission, the transmitter extends the selected sequence to the length of 2L by repeating the sequence.

At receivers, the base broadcast intention sequence bs_0 is correlated with the received signal. No matter which broadcast intention sequence is transmitted, a high correlation peak will be observed. Keep in mind that any broadcast intention sequence is a cyclically shifted version of bs_0 . The bs_0 will align with itself at a certain time lag of the received signal as shown in Fig. 6.3. If the transmitter does not repeat the sequence, the receiver will still observe correlation spikes. The shape, however, is confusing because the tail part is aligned with noise. The transmitter thus repeats the sequence in transmission.

6.3.1.3 Partial Spectrum Correlation

SFAB utilizes multiple narrow channels at the same time. The aforementioned method requires the transmitter to implement multiple transmitting chains to transmit the broadcast intention sequences independently in each channel. The receiver also needs to rely on sharp filters and guardbands to isolate each narrow channel. To reduce the complexity and overhead, we employ NC-OFDM to transmit data in multiple channels at the same time and encode the broadcast intention sequence in the frequency domain instead of the time domain.

To utilize a wideband channel with NC-OFDM, a node employs an N-point IFFT/FFT algorithm to modulate/demodulate N subcarriers that span the entire wideband channel. Suppose a wideband channel is divided into M narrow channels. Each narrow channel contains L = N/M



Figure 6.4: Construct a broadcast intention OFDM symbol through IFFT.

subcarriers. When a node tries to avoid causing interference to an occupied narrow channel, it maps zeros to all subcarriers in the channel, leading to zero power on them.

When a node has data to broadcast, it identifies channels that are available at its side and randomly chooses a broadcast intention sequence bs_k to map to subcarriers of each available channel as shown in Fig. 6.4. All subcarriers in unavailable channels are fed with 0s. After performing the *N*-point IFFT, the OFDM symbol that consists of *N* complex numbers is transmitted with a cyclic prefix of itself (in total 2*N* values).

To extract sequences from each channel, the receiver needs to obtain a complete OFDM symbol. Since the transmitter and the receiver have not been synchronized, the receiver needs to perform the *N*-point FFT each time a new sample is obtained and then correlate the decoded sequence with the base sequence bs_0 , which incurs high overhead.

To reduce the overhead, adaptive channel bonding introduced in chapter 4 uses correlation to calculate similarity between two signals. Here, we employ a common Fourier transform pair: the rectangular function and the sinc function. Given a general rectangular function X[m] in which N samples contain K unity-valued samples from index $-n_o$ to $-n_o + (K-1)$ in the frequency domain, we can get the corresponding time domain expression through the inverse discrete Fourier

transform (IDFT):

$$x[n] = \frac{1}{N} \cdot \sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}} X[m] \cdot e^{j\frac{2\pi mn}{N}}$$

$$= \frac{1}{N} \cdot \sum_{m=-n_{o}}^{-n_{o}+(K-1)} 1 \cdot e^{j\frac{2\pi mn}{N}}$$

$$= \frac{1}{N} \cdot \frac{\sin \frac{\pi nK}{N}}{\sin \frac{\pi n}{N}} \cdot e^{j\frac{2\pi n}{N}(\frac{K-1}{2}-n_{o})}.$$
 (6.3)

The function has a main lobe that is centered at n = 0 as shown in Fig. 6.5b. The ratio of sine terms is the amplitude of x[n] and the exponential term is the phase angle of x[n].

To get a spike that is similar to the sinc function in the time domain, we need to create a rectangular function in the frequency domain. The method is to assume that the complex conjugate of the base broadcast intention sequence bs_0 is present in a channel and calculate the expected IFFT result as shown in Fig. 6.6. The receiver then correlates the expected OFDM symbol with the received signal.

A trick is that the base broadcast intention sequence bs_0 is symmetric. Flipping the sequence left to right does not change the sequence. The correlation of the expected OFDM symbol and the received signal is the same as the convolution of the expected OFDM symbol and the received signal. Because the convolution in the time domain corresponds to the multiplication in the frequency domain [45] [44], we actually obtain a rectangular function in the frequency domain as shown in Fig. 6.7 when performing the correlation.

Suppose the transmitted broadcast intention sequence in channel *i* is bs_0 , we have $bs_0^*[m] \cdot bs_0[m] = e^{j0}$. Because the multiplication forms a rectangular function in the frequency domain, the correlation in the time domain results in a sinc function that provides a detectable spike. The assumption of having a broadcast intention sequence in channel *i* is thus confirmed.

Suppose the transmitted broadcast intention sequence is any cyclically shifted version of bs_0 .



(b) The corresponding sinc function in the time domain.

Figure 6.5: A rectangular function in the frequency domain matches a sinc function in the time domain. The Duality property of the Fourier transform provides that the reverse is also true.



Figure 6.6: The receiver assumes that the base broadcast intention sequence is present in each channel.



Figure 6.7: Identify whether or not a broadcast intention sequence is present in a narrow channel.

When L is an even number, we have

$$bs_{0}^{*}[m] \cdot bs_{k}[m] = e^{(j\frac{\gamma\pi m^{2}}{L})} \cdot e^{(-j\frac{\gamma\pi (m-k)^{2}}{L})}$$

$$= e^{j\frac{\gamma\pi (m^{2} - (m-k)^{2})}{L}}$$

$$= e^{j0} \cdot e^{j\frac{\gamma\pi (2km-k^{2})}{L}}.$$
 (6.4)

Equation 6.4 shows that the rectangular function is multiplied by a linear phase shift $e^{j\frac{\gamma\pi(2km-k^2)}{L}}$ (note that *m* is the variable and *k* is a constant for a given bs_k). When *L* is odd, it can be proved that the phase shift is also linear. The shifting theorem of DFT states that a constant phase shift of *X*[*m*] corresponds to a circular shift of the output sequence *x*[*n*] [44]. Therefore, the receiver can still observe a spike during correlation. The spike of the sinc function is just shifted in the time domain.

Consequently, we let the receiver assume that the complex conjugate of bs_0 is present in each narrow channel and calculates the expected OFDM symbol. For all of the expected OFDM symbols, the receiver correlates them with the received signal in parallel. The receiver learns that a broadcast intention sequence is transmitted in a channel when a spike occurs. In this way, all receivers learn the channels that are available at the sender side.

Suppose two nodes broadcast at the same time and the signal strengths are similar when the two transmissions arrive at a receiver. The receiver can observe two spikes in L correlation values if the two transmitters choose different broadcast intention sequences. The receiver learns that there is a collision. If they choose the same broadcast intention sequence, an aggregated spike with a higher peak value will be observed. If the peak value exceeds the maximum normalized value of 1.0, the receiver learns that there is a collision. Even if the aggregated spike does not exceed 1.0, because the two transmitters may not collide in all channels as they may have different available channels, a peak value difference between an aggregated spike and a normal spike can be detected. The receiver learns that there is a collision. The receiver should respond with a special collision notification sequence to let transmitters back off.

6.3.1.4 Respond to Broadcast Intention

Through the partial spectrum correlation, a node learns channels that are available at the sender side. Each receiver needs to confirm the channels that are also available at the receiver side.

To uniquely identify a node, each node v is assigned a unique unicast sequence (uc_v) of length L at least locally. The unicast sequence of a node should not create a sinc function in the time domain when using the partial spectrum correlation with another node's unicast sequence. We can adjust the root index of ZC sequences, γ , to meet the requirement. If two nodes use ZC sequences that have different root indexes, the phase shift is no longer linear as shown below (assuming L is even).

$$s_{\gamma_{0}}^{*}[m] \cdot s_{\gamma_{1}}[m] = e^{(j\frac{\gamma_{0}\pi m^{2}}{L})} \cdot e^{(-j\frac{\gamma_{1}\pi(m-k)^{2}}{L})}$$

$$= e^{j\frac{\pi(\gamma_{0}m^{2}-\gamma_{1}(m-k)^{2})}{L}}$$

$$= e^{j\frac{\pi(\gamma_{0}-\gamma_{1})m^{2}+\gamma_{1}(2km-k^{2})}{L}}$$
(6.5)

Because the phase shift is no longer linear, the multiplication does not correspond to a shifting of the sinc function in the time domain. The spike no longer exists. Similar to having a routing table, we assume that a node maintains the unicast sequences of its neighbors (e.g., the unique node ID may be used as the root index).

For a simpler case, unicast, the sender can transmit the target receiver's unicast sequence in each available channel and the receiver correlates its unicast sequence with the received signal to detect any transmission intention to it. Upon detecting the transmission intention, the receiver responds with its unicast sequence in each channel that is available at its own side. Due to the temporal relationship, the transmitter correlates the receiver's unicast sequence with the received signal once it finishes the transmission of the target receiver's unicast sequence. Through the partial spectrum correlation, the transmitter learns the channels that are confirmed by the receiver. The normal data transmission can be initiated.

In broadcast, however, the SINR is too low to recognize some receivers' responses if all receivers respond at the same time. Given a node, we give each neighbor a different priority to respond (e.g., based on node ID). A node thus maintains more entries in its routing table. For any neighbor, it knows its priority to respond. A broadcast initiator transmits its unique unicast sequence immediately after it sends a broadcast intention sequence. Once a node detects a broadcast intention sequence in a channel, it checks which neighbor is the broadcast initiator. If the node has the highest priority to respond, it responds with its unicast sequence in each available channel immediately; otherwise, it should defer its response by k * 2 OFDM symbols if it should respond in the *k*th slot. If the broadcast initiator detects a collision notification sequence, the broadcast initiator should back off randomly and re-initiate the broadcast later.

6.3.2 Dividing a Packet and Interleaving

After the sender obtains the node list in each channel, it decides how to divide a packet. It needs to find the best channel combination scheme that provides the maximum channel width for broadcast. When a channel is considered for combination, the sender needs to estimate the profit/cost ratio of using the channel and the effect on other channel combinations.

We consider that the profit of using a channel is the number of uncovered nodes that can be covered by the channel and the cost is the number of nodes that can receive data in the channel. To maximize the profit/cost ratio of using a channel, the sender needs to group channels with the minimum overlap. Before determining the best channel combination, we first need to find the channel combinations that can cover all nodes. It is a set cover problem to solve.

The set cover problem is: Given a set of elements $U = \{u_1, u_2, ..., u_n\}$ and a collection *S* of subsets of $U, S = \{S_1, S_2, ..., S_m\}$, find a least-cost sub-collection *C* such that $\bigcup_{S_i \in C} S_i = U$. In an unweighted set cover problem, the cost of a collection *C* is the number of sets contained in it. A constrained variation is to find a sub-collection *C* of size *k* such that their union contains as many elements as possible. Both problems are NP-hard [81] [82] [83].

In our application, U is the set of nodes to be covered, $U = \{u_1, u_2, ..., u_n\}$. We first identify channels that can be used by all nodes. They will definitely be used in a broadcast. Each remaining channel can cover a subset of nodes. We have $S = \{ch_1(u_i, ..., u_j), ..., ch_m(u_p, ..., u_q)\}$. Same as

the set cover problem, the problem is to select k channels from S such that every node is reachable through at least one of the k channels and the goal is to minimize the value of k.

The possible minimum set cover size is k = 2. To identify 2-channel combinations that can cover all nodes, we may solve the set cover problem with the constraint k = 2, which is NP-hard. There may exist multiple solutions. We need to decide which two channels can be combined to get the best profit/cost ratio. The selected channels are then removed from *S*, and the constraint is relaxed to k = 3. The procedure continues until there is no more channel to use or k = |S| (i.e., all remaining channels are combined together). Each phase includes a NP-hard set cover problem to solve without considering how to choose between the solutions. Therefore, we introduce a greedy algorithm to find the channel combinations.

As shown in Fig. 6.8, the sender organizes neighbors and available channels into a matrix. For each channel, the sender identifies nodes that can receive data in the channel and marks the corresponding positions with 1. The sender counts the number of nonzero values in a column and the sum of all values in a column.

Assuming there are *n* neighbors to cover, we have a constraint COUNT = n. All single channels that can meet the requirement are put into a valid channel grouping set F_g and marked as unavailable in the matrix M_1 . Let M_k represent the node-channel matrix where *k* channels are grouped.

After single channels that cover all nodes are picked up, the sender needs to identify channel combinations that cover all nodes. It first obtains candidate channel combinations of 2 channels (i.e., M_2) based on M_1 . For each available channel in M_1 , the sender combines it with any other available channel in M_1 to construct the M_2 .

In M_2 , the sender identifies all valid channel combinations and puts them in the valid channel grouping set F_g in a decreasing order in terms of profit/cost ratio. When identifying valid channel combinations in the M_k , all channel combinations that meet the constraint COUNT = n are valid. However, we want to maximize the profit/cost ratio of using a channel (i.e., channels with the minimum overlap). Therefore, among all valid channel combinations, the one with the minimum



Figure 6.8: Identify the best channel combination to cover all nodes.

sum is selected. All related channel combinations are removed from M_k and all related channels are marked as unavailable in M_1 . The sender then finds the next channel combination that has the minimum sum until there is no more channel combination that holds COUNT = n. If the M_k still contains more than one column, the sender proceeds to the next level. Candidate channel combinations with k + 1 channels (i.e., M_{k+1}) are created based on M_k and M_1 . The search ends when all columns are removed from a matrix M_k or there is only one column remains in M_k and it fails to meet the requirement COUNT = n. It means that even all remaining channels are combined together, the combination cannot cover all nodes.

Finally, a packet is divided into $|F_g|$ segments where $|F_g|$ is the number of sets in F_g . The channel grouping ensures that each segment will be delivered to all nodes. To decode the packet,



Figure 6.9: An example to show channels in which a node should receive data. 'x' means the channel is unavailable at the node.

each node needs to know in which channel it should receive the data and how to de-interleave all segments. For example, in the example illustrated in Fig. 6.9, the node B should decode the packet from the channels 3, 1, 2 while the node C should decode the packet from the channels 3, 5, 4.

6.3.3 Establishing Communication

Once the scheme for the packet segment interleaving is determined, the broadcast initiator needs to let each receiver know the channels in which the receiver should decode the packet. The sender constructs the selective broadcast sequence (*sbs*) of each used channel. Essentially, the *sbs_i* is the combination of unicast sequences of all nodes that should receive a segment from the *i*th channel. If a channel is not used for broadcast, a length-L vector of 0s is used as the selective broadcast sequences are mapped to subcarriers of corresponding channels. After performing IFFT, the time domain symbols are transmitted as a preamble before the OFDM synchronization preamble.

Each node correlates its unicast sequence with the received signal using the partial spectrum correlation. If a spike is detected in a channel, the node learns that it should receive data in the channel. Once the channels to which it should listen are identified, it filters out signals in other channels so that it can detect the NC-OFDM synchronization preamble. Because unwanted signals are filtered out, the communication is the same as standard NC-OFDM communication. Once the

receiver is synchronized with the transmitter, the receiver is able to get bits from OFDM symbols. The bits in all channels are grouped together to get bytes. In SFAB, the receiver needs to know how to organize bits from different channels.

In the standard NC-OFDM communication, the receiver can follow a general rule to organize bits. For example, the receiver can organize bits in an OFDM symbol from left to right (lower index channel to higher index channel). In SFAB, however, the sender interleaves bits according to the collected spectrum availability information. The receiver needs to know the rule to deinterleave the bits. In the example illustrated in Fig. 6.9, node *B* should put bits extracted from channel 3 before bits extracted from channel 1. We let the sender use the first OFDM symbol after the synchronization to indicate the order.

As shown in Fig. 6.4, a number of bits can be extracted from one OFDM symbol. Suppose each channel contains *L* subcarriers, βL bits can be extracted from each channel in one OFDM symbol where β is determined by the modulation scheme. The value of *L* is usually large enough to create at least one byte. The first byte in each channel is used to represent the index of the channel for deinterleaving. Once the receiver learns the indexes, the receiver can organize bits in the right order in each OFDM symbol. The overhead incurred by diving a packet into segments is thus one OFDM symbol.

6.4 Performance Evaluation

We use the GNU Radio/USRP platform to validate the efficacy of partial spectrum correlation. The partial spectrum correlation is the basis for identifying a broadcast intention sequence that may appear in any channel. It is also used to confirm channels that are available at the receiver side. After the spectrum availability information is collected, it is used to inform the receivers of the final decision.

With collected spectrum availability information, the sender uses a greedy algorithm to find the best channel combination scheme to divide a packet and interleave bits. To demonstrate the advantages of performing spectrum fragment grouping, we need many nodes with different spectrum

resources. The spectrum diversity is simulated in ns-2 where we study the impact of fragmented spectrum on broadcast.

The baseline is the traditional broadcast paradigm in which a node waits for the entire wideband channel to be idle to broadcast a packet. An improvement is to use current spectrum-agile designs in which a node is able to use spectrum fragments that are available at broadcast time. However, they do not consider that the spectrum availability may be different between the transmitter and the receiver. We compare SFAB with them to show how broadcast can benefit from the spectrum agreement and the spectrum fragment grouping.

6.4.1 Spectrum Agreement

To validate the efficacy of partial spectrum correlation, we conduct experiments with 128 point IFFT on a band of 4 MHz. The band is divided into four channels of 1 MHz and each channel contains 31 subcarriers, leaving 2 unoccupied in the middle and 2 unoccupied at two edges. As introduced in Section 6.3.1, 31 cyclically shifted ZC sequences are created as the broadcast intention sequences. One USRP2 randomly selects a broadcast intention sequence and transmits it in two of the four channels. To create a collision, we let another USRP2 randomly select a sequence and transmit it in one of the two occupied channels. One USRP2 is transmitting data in all channels as noise and one USRP2 is used as the receiver.

To detect the broadcast intention sequence in a channel, the receiver assumes that the complex conjugate of the base sequence is mapped to the subcarriers of the channel and calculates the expected IFFT result. If a spike is detected in the correlation of the expected OFDM symbol and the received signal, the receiver learns that a neighboring node is going to broadcast in the channel.

Fig. 6.10a shows the correlation result for the channel in which a broadcast intention sequence is indeed transmitted. The x-axis is the temporal progression of collected samples and the yaxis is the correlation value normalized to the expected correlation value of the base broadcast intention sequence. The figure shows that a node can easily detect a broadcast intention sequence in a channel through the correlation spikes. The correlation spikes are clear even at a signal-tointerference-plus-noise ratio (SINR) of 0 dB. Note that Wi-Fi requires a minimum SINR of $4 \sim 5$ dB to decode the lowest rate packet [84]. Therefore, if the broadcast intention sequence cannot be detected in a channel, the channel should not be used for data transmission. For any channel that is available for data transmission, the SINR is expected to be high enough for the identification of a broadcast intention sequence in the channel through the partial spectrum correlation.

Two nodes may broadcast at the same time. If the receiver can detect the collision in the spectrum agreement phase, it can notify the transmitters to prevent collision in data transmission. One OFDM symbol consists of 128 complex numbers when a 128-point IFFT algorithm is used. All possible values are represented in 128 correlation values. When two nodes collide in a channel, Fig. 6.10b shows that two correlation spikes are observed in an arbitrary segment of 128 correlation values. This happens when the two transmitters selected two different broadcast intention sequences. As analyzed in Section 6.3.1, the sinc function formed by the correlation results is shifted by different amount. As a result, more than one spike is observed.

If two nodes choose the same broadcast intention sequence, the two spikes are aggregated as a higher spike. The peak value is much greater than that observed in a channel where only one node is transmitting. As long as transmitters do not collide in all channels, the receiver can detect the collision through the peak value difference. Moreover, if a peak value exceeds the maximum normalized value of 1.0, the receiver also learns that there is a collision. In any of the conditions, the node should respond with a collision notification sequence to to let transmitters back off.

If no broadcast intention sequence is transmitted in a channel, the receiver should not detect one. Fig. 6.10c shows that if no broadcast intention sequence is transmitted in a channel, there is no spike during correlation. Although the absolute magnitudes of correlation values may increase with the interference and noise, there is no abrupt change of correlation value. We can calculate the moving average during correlation and used it as a reference to identify real spikes.

After the sender decides how to arrange packet segments, it has to inform each receiver of channels in which the receiver should decode the packet. Different from the broadcast intention sequence, the sender encodes multiple unicast sequences in a channel. In this experiment, the sender



(c) No broadcast intention.

Figure 6.10: An example of partial spectrum correlation for broadcast intention at 0 dB.

encodes four different unicast sequences (e.g., uc_1 , uc_2 , uc_3 , and uc_4) in channel 2. Fig. 6.11a shows the correlation results when the receiver expects the unicast sequence uc_4 to be present in the channel. Because the sequence is indeed present in channel 2, high correlation spikes are observed. Although four unicast sequences are encoded in the channel, only one spike exists in any segment of 128 correlation values. Two nodes' unicast sequences do not cause high correlation peaks with each other. This is different from Fig. 6.10b where each broadcast intention sequence is the shifted version of the other.

The sequence uc_5 is not encoded in channel 2. Fig. 6.11b shows that there is no obvious correlation spikes even though the absolute magnitudes of correlation values are high. The moving average method is used to detect real spikes.

6.4.2 Simulation

To show the advantage of SFAB, we need more nodes to create the spectrum diversity. We conduct simulations in ns-2.35 to study how narrowband interference affects both the medium access opportunity at the senders and the packet decoding at the receivers.

6.4.2.1 Simulation Setup

Considering the current channel bonding in 802.11n, each node operates in a band of 40 MHz. The 40 MHz band is divided into 8 narrow channels of 5 MHz. Each narrowband device has a bandwidth of 5 MHz and can switch to any channel. The narrowband devices are randomly distributed around the wideband devices as interference sources.

With one antenna, the 40 MHz channel can provide a raw data rate of 135 Mbps using 64-QAM and 5/6 coding rate as defined in the 802.11n standard [41]. Suppose the raw data rate supported by each 5 MHz channel is about 16 Mbps. To create medium interference, we let each narrowband device generate packets at a rate of 8 Mbps.



300 400 Sample Number (b) The expected sequence is not encoded in the channel.

Figure 6.11: An example of partial spectrum correlation for spectrum notification at 0 dB.

6.4.2.2 Efficiency of Partial Spectrum Correlation

Although we target at a more complex problem of broadcast, the partial spectrum correlation is also useful for unicast in CRNs. In CRNs, the transmitter and the receiver need to agree on narrow channels that will be used for data transmission.

Because the transmitter does not know which channel is available at the receiver side, it may have to transmit a control packet in each narrow channel to make sure that the receiver is informed of the transmission. Sending a short control packet in each narrow channel is less efficient than the proposed partial spectrum correlation.

As defined in 802.11n standard [41], the PLCP preamble is 64 *us* and the SIGNAL symbol is 16 *us* when using OFDM in an independent 5 MHz channel. Each OFDM symbol contains 48 data subcarriers and lasts 16 *us*. Suppose at least three OFDM symbols are needed to transmit the control message with headers and cyclic redundancy check (CRC). The overhead incurred by the control packet is 128 *us*.

On the contrary, the partial spectrum correlation only incurs an overhead of two OFDM symbols. The partial spectrum correlation does not need the receiver to decode a packet. The transmitter simply performs 256 point IFFT on the 40 MHz channel so that each 5 MHz channel can map a unicast sequence of length 32. The time used to transmit the OFDM symbol with a cyclic prefix of itself is 256 * 2/40 = 12.8 us, which is one order of magnitude less.

We set 10 wideband devices to communicate with each other. Four narrowband devices are placed to occupy narrow channels randomly. When packets are transmitted at high speeds, the actual data transmission time is short. For example, the time used to transmit a 1500 byte packet at 1 Mbps is 12 *ms*, but the airtime is reduced to 20 *us* at 600 Mbps in 802.11n. The control overhead thus significantly impact the communication efficiency. We reduce the packet size to shorten the airtime of a transmission. Fig. 6.12 shows that the proposed partial spectrum correlation can help a CR device reduce the overhead introduced by dynamic spectrum access. When the packet is small or the data rate is high, the control overhead plays an important role in determining the efficiency.



Figure 6.12: The control overhead has a significant impact on the throughput in high speed transmissions.

improves the throughput by 3 times when the airtime of a transmission is short.

6.4.2.3 Broadcast

We model three different types of broadcast. The first one is the typical 802.11 device that uses a contiguous band of spectrum for transmission. The broadcast initiator has to defer its transmission to the time when all narrow channels are idle. The second type is to use current spectrum-agile designs that utilize all available narrow channels detected before transmission. The method increases the medium access opportunities of a wideband device. The third one is the proposed SFAB.

We deploy 10 wideband devices and each device may broadcast to the others. Fig. 6.13 shows the results of adding narrowband interference sources. In 802.11, the broadcast initiator broadcasts a packet if it detects that the entire 40 MHz wideband channel is idle. Because narrowband devices work independently in different narrow channels, the wideband devices have fewer medium access



Figure 6.13: Throughput comparison with different number of narrowband interference sources.

opportunities. In addition, some narrow channels may be available at the sender side but unavailable at some receivers. These receivers cannot decode the packet. The throughput drops quickly because of the nonuniform spectrum availability caused by narrowband interference.

When more narrowband devices are deployed, the broadcast initiator becomes harder to win medium access opportunities because two narrowband devices do not contend with each other if they are in different narrow channels. Fig. 6.13 shows that a spectrum-agile design can increase the medium access opportunity. However, it cannot address the spectrum disparity problem. When the sender broadcasts a packet, it may not be decoded at some receivers. SFAB collects information of receivers' spectrum availability, and arranges transmission based on the agreed spectrum availability. By allocating packet segments to the right spectrum fragments, SFAB maximizes the channel width used for broadcast and the throughput is significantly increased.



Figure 6.14: Throughput comparison with different data rates of narrowband interference sources.

The impact of narrowband interference on broadcast also depends on how active the narrowband devices are. Even with 4 narrowband devices, the throughput is high when they are not active as shown in Fig. 6.14. However, when the narrowband devices become active, wideband devices suffer from two major issues. First, the sender can hardly win medium access opportunities for the entire wideband channel because narrowband devices work in different narrow channels independently. Second, the receivers may not be able to decode the broadcasted packet in channels that are interfered by the narrowband devices. Current spectrum-agile designs can address the first issue but not the second one. SFAB uses partial spectrum correlation to achieve spectrum agreement between the transmitter and the receivers efficiently. With collected spectrum availability information, SFAB also divides a packet into segments and allocates them to multiple channels for parallel transmission.

6.5 Summary

In this chapter, we extend broadcast from a single channel to spectrum fragments. A spectrum fragment agile broadcast (SFAB) protocol is proposed to address the broadcast challenges imposed by CRNs. To address the spectrum diversity, an efficient spectrum agreement mechanism based on partial spectrum correlation is proposed. The spectrum agreement ensures that the broadcasted packet will not be corrupted at the receiver side. With collected spectrum availability information, SFAB divides a packet into segments and allocate them to different channels in a way that maximizes the channel width for broadcast. Experiments and simulations show that the proposed SFAB provides higher throughput than other broadcast schemes under active interference.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this chapter, we conclude the work presented in this dissertation and discuss a few promising future directions to extend this dissertation.

7.1 Conclusions

Fast growing demand for wireless broadband has drawn great research interest in dynamic spectrum access which is enabled by the emerge of cognitive radios. The new technology brings us new perspectives and new challenges in designing wireless communication protocols. In the dissertation, we identify several problems introduced by the flexible spectrum use. Since non-contiguous orthogonal frequency division multiplexing (NC-OFDM) is widely adopted in dynamic spectrum access, we propose several techniques that exploit the subcarriers in NC-OFDM to address the aforementioned problems.

When licensed frequency bands are open for opportunistic use, a primary issue is to ensure that the licensed users do not experience undue interference. A predictive method that infers the availability of spectrum holes can help reduce the interference. In the dissertation, OFDM is employed to measure a wide band of spectrum and subcarriers are grouped to subchannels based on similar spectrum use activities. In each subchannel, a spectrum occupancy prediction model based on partial periodic pattern mining (PPPM) is introduced to identify frequent spectrum occupancy patterns that are hidden in the spectrum use of a channel. Due to noise, sensing errors, and irregular user behaviors, the periodicity of a spectrum occupancy pattern may be partially observable. The mining is thus tailored for identification of real patterns that exhibit partial periodicity only. The mined frequent patterns are then used to predict whether or not the next slot will be available and for how long a channel will be available. Using real-world network activities, we demonstrate that the prediction performance is better than the traditional frequent pattern mining (FPM) and the fixed-order hidden Markov model (HMM). With the spectrum hole prediction, we show that unlicensed users can aggressively utilize spectrum holes without introducing significant interference to licensed users.

In addition, timely estimation of spectrum availability is also critical in dynamic spectrum access. To speed up the mining process, an index list structure along with an Apriori-like property and a backward-extension rule is introduced. The index list structure allows us to avoid scanning the entire database to count the occurrences of each pattern, the Apriori-like property reduces the number of candidate frequent patterns by identifying patterns that cannot yield longer frequent patterns, and the backward-extension rule allows us to avoid redundant mining on two branches that produce identical frequent patterns. Performance evaluation shows that the three designs make PPPM the fastest prediction model among FPM and HMM.

In the spectrum holes discovered above, we need to address the contentions between devices. We have investigated both single-channel contention and multichannel contention.

• For single-channel contention, we consider contention efficiency. The high physical layer data rates have resulted in very short transmission time of a typical data packet. The transmission time is expected to be further shortened as wireless networks move toward wider channels. The random backoff for collision avoidance arises as a major factor that causes the low efficiency in contention-based wireless communication. To reduce the medium access overhead, we propose a collision detection and bitwise arbitration (CDBA) mechanism that takes advantage of self-interference and subcarrier nulling to achieve collision detection in the frequency domain. Essentially, each node chooses a different subset of subcarriers to transmit arbitration preamble and performs spectrum analysis on received signal at the same time. If it detects signals on subcarriers that are deactivated by itself, it learns that there is a collision. As collisions are detectable, collision avoidance is no longer mandatory. All nodes can transmit immediately if the channel is detected to be idle. If a collision is detected, a bitwise arbitration mechanism is introduced to efficiently determine the winner between contending transmitters. The bitwise arbitration makes a node abort its transmission when

detecting a higher priority transmission. At the end of the arbitration phase, usually only one node survives to data transmission. Compared with widely adopted collision avoidance protocols, we show that CDBA significantly reduces the medium access overhead and the collision probability, making devices actually benefit from the high physical layer data rates introduced by advancements in wireless technologies. In addition, the fairness is improved because collisions are resolved before data transmission and the impact of radio capture effect is reduced. Further, the CDBA facilitates real time prioritized communication by making high priority messages get through the arbitration unimpeded.

• In single-channel contention, all devices use the same channel width. However, different applications have different requirements on channel width. When wideband devices contend with narrowband devices, wideband devices can hardly win medium access opportunities because they have to wait for the entire wide band of spectrum to be idle but narrowband devices can work independently in different narrow channels. To address the contention problem when heterogeneous radios coexist in a contention domain, we propose an adaptive channel bonding (ACB) protocol, in which a wideband device contends with narrowband devices in each narrow channel independently. In ACB, a wideband device is allowed to start a transmission with available narrow channels detected before transmission and gradually aggregates more narrow channels during transmission. Because a wideband device evaluates each narrow channel independently, a wideband device is able to contend with narrowband devices in each narrow channel fairly. To increase the chance to win some narrow channels, a wideband device contends in each narrow channel with a different priority. This is achieved by mapping different priority sequences to subcarriers of different narrow channels. In each narrow channel, the CDBA introduced above is employed to determine the winner based on the priority sequences of contending transmitters. After a transmitter wins some narrow channels, it encodes spectrum agreement sequences on subcarriers of each occupied narrow channel to inform the receiver of narrow channels from which the receiver should expect data. At the receiver side, correlation is used to detect spectrum agreement sequences. Experiments show that the correlation-based spectrum agreement is effective and more efficient than the control packet based spectrum agreement. Because the adaptive channel bonding fully exploits spectrum opportunities, it achieves the highest throughput in comparison with current static channel bonding and spectrum-agile designs.

After the contention problem is addressed, we study the impact of multipath interference on transmission. We observe that if a subcarrier is deactivated, it is less susceptible to multipath interference because no signal is transmitted. We thus employ subcarrier nulling to inform the receiver of subcarrier state change. The state change implies that a known sequence of bits should be inserted to the data stream. We propose a modulation scheme coding (MSC) method to find the representative sequence of bits for each modulation scheme so that the number of known bits is maximized. Because the subcarrier nulling represents more bits than that can be represented by a symbol in low-density modulation schemes, the transmission time is reduced. In both low-density and high-density modulation schemes, the known representative sequence of bits for each modulation scheme help improve decoding performance. Performance evaluation shows that MSC improves the throughput due to reduced transmission time and bit error rate.

In wireless networks, broadcast is a fundamental mechanism for route discovery, code update, disseminating control information and alarm to all nodes. Because devices may experience different narrowband interference at different locations, they may have different available spectrum fragments. The broadcast in dynamic spectrum access needs to consider the diversity of spectrum availability. We propose a spectrum fragment agile broadcast (SFAB) protocol that employs partial spectrum correlation to achieve fast spectrum agreement between the transmitter and the receivers. The partial spectrum correlation employs a common Fourier transform pair to help detect unique sequences encoded on subcarriers of each narrow channel. The spectrum agreement mechanism enables low-overhead exchange of spectrum availability information between the transmitter and the receivers. With collected spectrum availability information, SFAB divides a packet into small parts and allocate them to different narrow channels in a way that maximizes the aggregate channel width for broadcast. Experiments and simulations are conducted to show that SFAB provides

higher throughput than other broadcast schemes under active narrowband interference.

Dynamic spectrum access is envisioned to be the key to relax the exclusive licensing constraint on spectrum access. This dissertation contributes in designing efficient algorithms and protocols to support dynamic spectrum access. By exploiting subcarriers in multicarrier modulation, various designs have been presented in this dissertation with a goal to break barriers for implementing dynamic spectrum access. Spectrum prediction helps relieve concerns about interference introduced to licensed users. Spectrum contention protocols are prepared for severe contention introduced by radios' flexible change of frequency bands and unfair contentions between heterogeneous radios. As dynamic spectrum access enables wireless networks to move toward wider channels for higher speed, frequency diversity and spectrum disparity are also considered in this dissertation. While this dissertation in general builds a practical dynamic spectrum access system, it can definitely be extended.

7.2 Future Work

In spectrum prediction, we need to capture the regularity of licensed users' spectrum use activities. However, we are lack of knowledge about licensed users' communication strategies. If the measurement granularity is not appropriately set, it is hard to identify the real patterns of licensed users' spectrum use activities. For example, suppose a slot of 600 μ s is used in a time frame of 5 *ms*. If we set measurement granularity to 10 *ms*, we may always detect busy state. On the contrary, if we set measurement granularity to 10 μ s, we may discover many patterns that always give busy state or idle state. Useful patterns that indicate state change are buried by long periods of busy state or idle state. It is desirable to have the measurement automatically converge to the appropriate measurement granularity. This helps identify regularity of spectrum use activities.

This dissertation has studied the spectrum disparity problem between the transmitter and the receiver. It ensures that the spectrum fragments used for transmission are available at both the sender side and the receiver side. However, it does not guarantee that the transmission will not cause interference at neighboring nodes. Most studies in dynamic spectrum access only focus on

communication success between the transmitter and the receiver. It is a potential research area to study spectrum disparity in a larger scope. The spectrum agreement mechanism developed in our SFAB may be used to collect spectrum availability information from neighboring nodes. With the spectrum availability information, the transmitter can avoid interference to neighboring nodes.

The spectrum disparity problem also leads to a security question. Unlicensed users should vacate the channel upon licensed users' return. How do we identify and discourage malicious users that misuse spectrum resources? Without a protection for legitimate transmissions, the licensed users will soon become conservative about opening their licensed bands. The secure spectrum access of licensed users is a critical issue to ensure successful adoption of dynamic spectrum access.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] E. Magistretti, K. K. Chintalapudi, B. Radunovic, and R. Ramjee, "WiFi-Nano: reclaiming WiFi efficiency through 800 ns slots," in *Proc. MobiCom*, 2011, pp. 37–48.
- [2] "General survey of radio frequency bands (30 MHz to 3 GHz): Vienna, Virginia, September 1-5, 2009," http://www.sharedspectrum.com/papers/spectrum-reports/.
- [3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.
- [4] "FCC-10-174," http://www.fcc.gov/Daily_Releases/Daily_Business/2010/db0923/ FCC-10-174A1.pdf.
- [5] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: a large-scale spectrum measurement study," in *Proc. MobiCom*, 2009, pp. 13–24.
- [6] J. Han, G. Dong, and Y. Yin, "Efficient mining of partial periodic patterns in time series database," in *Proc. ICDE*, 1999, pp. 106–115.
- [7] I. A. Akbar and W. H. Tranter, "Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case," in *Proc. SoutheastCon*, 2007, pp. 196–201.
- [8] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng, "On the feasibility of effective opportunistic spectrum access," in *Proc. IMC*, 2010, pp. 151–164.
- [9] Z. Wen, T. Luo, W. Xiang, S. Majhi, and Y. Ma, "Autoregressive spectrum hole prediction model for cognitive radio systems," in *Proc. ICC Workshops*, 2008, pp. 154–157.
- [10] J. Su and W. Wu, "Wireless spectrum prediction model based on time series analysis method," in *Proc. CoRoNet*, 2009, pp. 61–66.
- [11] C.-J. Yu, Y.-Y. He, and T.-F. Quan, "Frequency spectrum prediction method based on EMD and SVR," in *Proc. ISDA*, 2008, pp. 39–44.
- [12] Z. Wang and S. Salous, "Spectrum occupancy statistics and time series models for cognitive radio," *J. of Signal Processing Systems*, vol. 62, no. 2, pp. 145–155, Feb. 2011.
- [13] C.-H. Park, S.-W. Kim, S.-M. Lim, and M.-S. Song, "HMM based channel status predictor for cognitive radio," in *Proc. APMC*, 2007, pp. 1–4.
- [14] V. K. Tumuluru, P. Wang, and D. Niyato, "Channel status prediction for cognitive radio networks," *Wireless Commun. and Mobile Comput.*, vol. 12, no. 10, pp. 862–874, Jul. 2010.
- [15] C. Ghosh, C. Cordeiro, D. P. Agrawal, and M. B. Rao, "Markov chain existence and hidden Markov models in spectrum sensing," in *Proc. PerCom*, 2009, pp. 1–6.
- [16] Z. Chen and R. C. Qiu, "Prediction of channel state for cognitive radio using higher-order hidden Markov model," in *Proc. SoutheastCon*, 2010, pp. 276–282.
- [17] C. Devanarayana and A. Alfa, "Predictive channel access in cognitive radio networks based on variable order markov models," in *Proc. GLOBECOM*, 2011, pp. 1–6.
- [18] C. Li and J. Wang, "Efficiently mining closed subsequences with gap constraints," in *Proc.* SDM, 2008, pp. 313–322.
- [19] B. Ding, D. Lo, J. Han, and S.-C. Khoo, "Efficient mining of closed repetitive gapped subsequences from a sequence database," in *Proc. ICDE*, 2009.
- [20] M. Zhang, B. Kao, D. W. Cheung, and K. Y. YIP, "Mining periodic patterns with gap requirement from sequences," ACM Trans. Knowledge Discovery from Data, vol. 1, no. 2, pp. 7:1–7:39, Aug. 2007.
- [21] X. Zhu and X. Wu, "Mining complex patterns across sequences with gap requirments," in *Proc. IJCAI*, 2007, pp. 2934–2940.
- [22] W. G. Aref, M. G. Elfeky, and A. K. Elmagarmid, "Incremental, online, and merge mining of partial periodic patterns in time-series databases," *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. 3, pp. 332–342, Mar 2004.
- [23] N. Chomsky, "Three models for the description of language," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 113–124, 1956.
- [24] M. A. Harrison, *Introduction to Formal Language Theory*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., 1978.
- [25] S. Geirhofer, L. Tong, and B. Sadler, "A measurement-based model for dynamic spectrum access in WLAN channels," in *Proc. MILCOM*, 2006, pp. 1–7.
- [26] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting WiFi white space for ZigBee performance assurance," in *Proc. ICNP*, 2010.
- [27] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. VLDB*, 1994, pp. 487–499.
- [28] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid, "Periodicity detection in time series databases," *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 7, pp. 875–887, Jul. 2005.
- [29] C. Berberidis, W. G. Aref, M. Atallah, I. Vlahavas, and A. K. Elmagarmid, "Multiple and partial periodicity mining in time series databases," in *Proc. ECAI*, 2002.
- [30] S. Huang, X. Liu, and Z. Ding, "Optimal transmission strategies for dynamic spectrum access in cognitive radio networks," *IEEE Trans. Mobile Computing*, vol. 8, no. 12, pp. 1636–1648, Dec. 2009.

- [31] "Anonymized packet traces and AP syslog," http://www.cs.umd.edu/projects/wifidelity/ sigcomm08_traces/.
- [32] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *Proc. MobiCom*, 2011, pp. 301–312.
- [33] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proc. MobiCom*, 2010, pp. 1–12.
- [34] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: interference cancellation for wireless LANs," in *Proc. MobiCom*, 2008, pp. 339–350.
- [35] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless LAN," in *Proc. SIGCOMM*, 2010, pp. 147–158.
- [36] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "No time to countdown: migrating backoff to the frequency domain," in *Proc. MobiCom*, 2011, pp. 241–252.
- [37] B. Radunovic, D. Gunawardena, P. Key, A. Proutiere, N. Singh, V. Balan, and G. Dejean, "Rethinking indoor wireless mesh design: Low power, low frequency, full-duplex," in *Proc.* 5th IEEE Workshop on Wireless Mesh Networks (WIMESH), June 2010, pp. 1–6.
- [38] D. W. Bliss, P. A. Parker, and A. R. Margetts, "Simultaneous transmission and reception for improvedwireless network performance," in *Proc. 14th IEEE/SP Workshop on Statistical Signal Processing (SSP)*, 2007, pp. 478–482.
- [39] I. Gheorma and G. Gopalakrishnan, "RF photonic techniques for same frequency simultaneous duplex antenna operation," *Photonics Technology Letters, IEEE*, vol. 19, no. 13, pp. 1014–1016, July 2007.
- [40] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 544–556, April 2012.
- [41] "IEEE 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," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, 2012.
- [42] S. S. Hong, J. Mehlman, and S. Katti, "Picasso: flexible RF and spectrum slicing," in *Proc.* SIGCOMM, 2012, pp. 37–48.
- [43] S. W. Smith, *Digital Signal Processing: A Practical Guide for Engineers and Scientist*. Newnes Books, 2003.
- [44] R. G. Lyons, Understanding Digital Signal Processing (3rd Edition). Prentice Hall, 2010.
- [45] S. W. Smith, *The Scientist & Engineer's Guide to Digital Signal Processing*. California Tech. Publ., 1997.

- [46] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51 83, jan. 1978.
- [47] National Instruments. Windowing: Optimizing FFTs Using Window Functions. [Online]. Available: http://www.ni.com/white-paper/4844/en
- [48] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC protocol implementations on software-defined radios," in *Proc. NSDI*, 2009, pp. 91–105.
- [49] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," Digital Equipment Corp, Tech. Rep. DEC-TR-301, 1984.
- [50] K. Chintalapudi, B. Radunovic, V. Balan, M. Buettener, S. Yerramalli, V. Navda, and R. Ramjee, "WiFi-NC: WiFi over narrow channels," in *Proc. NSDI*, 2012.
- [51] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to share: narrowbandfriendly wideband networks," in *Proc. SIGCOMM*, 2008, pp. 147–158.
- [52] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng, "Supporting demanding wireless applications with frequency-agile radios," in *Proc. NSDI*, 2010, pp. 5–5.
- [53] X. Zhang and K. Shin, "Adaptive subcarrier nulling: Enabling partial spectrum sharing in wireless LANs," in *Proc. ICNP*, 2011, pp. 311–320.
- [54] E. Chai, J. Lee, S.-J. Lee, R. Etkin, and K. G. Shin, "Building efficient spectrum-agile devices for dummies," in *Proc. MobiCom*, 2012, pp. 149–160.
- [55] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531–532, 1972.
- [56] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, pp. 1613–1621, 1997.
- [57] F. Tufvesson, O. Edfors, and M. Faulkner, "Time and frequency synchronization for OFDM using PN-sequence preambles," in *Vehicular Technology, IEEE Conference*, vol. 4, 1999.
- [58] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and MAC protocols," in *Proc. MobiCom*, 2009, pp. 193–204.
- [59] S. Singh, M. Shahbazi, K. Pelechrinis, K. Sundaresan, S. V. Krishnamurthy, and S. Addepalli, "A case for adaptive sub-carrier level power allocation in OFDMA networks," in *Proc. MobiHoc*, 2012, pp. 225–234.
- [60] A. Dammann, "Space-time-frequency diversity in the next generation of terrestrial digital video broadcasting," in *Multi-Carrier Systems and Solutions*, ser. Lecture Notes in Electrical Engineering. Springer Netherlands, 2009, vol. 41, pp. 101–110.
- [61] S. Banerjee, R. Jesme, and R. Sainati, "Investigation of spatial and frequency diversity for long range UHF RFID," in *Proc. Antennas and Propagation Society International Symposium*, 2008, pp. 1–4.

- [62] A. Bhartia, Y.-C. Chen, S. Rallapalli, and L. Qiu, "Harnessing frequency diversity in Wi-Fi networks," in *Proc. MobiCom*, 2011, pp. 253–264.
- [63] S. Jakubczak and D. Katabi, "A cross-layer design for scalable mobile video," in *Proc. Mo-biCom*, 2011, pp. 289–300.
- [64] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang, "ParCast: Soft video delivery in MIMO-OFDM WLANs," in *Proc. MobiCom*, 2012, pp. 233–244.
- [65] J. Huang, Y. Wang, and G. Xing, "LEAD: Leveraging protocol signatures for improving wireless link performance," in *Proc. MobiSys*, 2013, pp. 333–346.
- [66] P. Huang, X. Yang, and L. Xiao, "WiFi-BA: Choosing arbitration over backoff in high speed multicarrier wireless networks," in *Proc. INFOCOM*, 2013, pp. 1375–1383.
- [67] "Wireshark," http://www.wireshark.org/.
- [68] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Mar 1974.
- [69] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. Miller, "Maranello: Practical partial packet recovery for 802.11," in *Proc. NSDI*, 2010, pp. 14–14.
- [70] Y. Kondareddy and P. Agrawal, "Selective broadcasting in multi-hop cognitive radio networks," in *Proc. IEEE Sarnoff Symposium*, 2008, pp. 1–5.
- [71] C. J. L. Arachchige, S. Venkatesan, R. Chandrasekaran, and N. Mittal, "Minimal time broadcasting in cognitive radio networks," in *Proc. ICDCN*, 2011, pp. 364–375.
- [72] S. Ji, R. Beyah, and Z. Cai, "Minimum-latency broadcast scheduling for cognitive radio networks," in *Proc. SECON*, 2013.
- [73] Y. Song and J. Xie, "A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks without a common control channel," in *Proc. INFOCOM*, 2012, pp. 2273–2281.
- [74] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1867–1881, 2012.
- [75] Y. Song and J. Xie, "A distributed broadcast protocol in multi-hop cognitive radio ad hoc networks without a common control channel," in *Proc. INFOCOM*, 2012, pp. 2273–2281.
- [76] C. J. L. Arachchige, S. Venkatesan, R. Chandrasekaran, and N. Mittal, "Minimal time broadcasting in cognitive radio networks," in *Proc. ICDCN*, 2011, pp. 364–375.
- [77] S. Ji, R. Beyah, and Z. Cai, "Minimum-latency broadcast scheduling for cognitive radio networks," in *Proc. SECON*, 2013.

- [78] E. Magistretti, O. Gurewitz, and E. W. Knightly, "802.11ec: collision avoidance without control messages," in *Proc. MobiCom*, 2012, pp. 65–76.
- [79] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the RF smog: Making 802.11n robust to cross-technology interference," in *Proc. ACM SIGCOMM*, 2011, pp. 170–181.
- [80] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531–532, 1972.
- [81] R. M. Karp, "Reducibility among combinatorial problems." in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, New York, 1972, pp. 85–103.
- [82] D. Zuckerman, "NP-complete problems have a version that's hard to approximate," in *Proc.* of the Eighth Annual Structure in Complexity Theory Conference, 1993, pp. 305–312.
- [83] U. Feige, "A threshold of ln n for approximating set cover," J. ACM, vol. 45, no. 4, pp. 634–652, July 1998.
- [84] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *Proc. SIGCOMM*, 2013, pp. 375–386.