MAC/PHY Co-Design of CSMA Wireless Networks Using Software Radios

by

Xinyu Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2012

Doctoral Committee:

Professor Kang Geun Shin, Chair
Associate Professor Achilleas Anastasopoulos
Assistant Professor Prabal Dutta
Professor Brian Noble

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to many people who helped me during my graduate study.

First, I am deeply grateful to my advisor, Professor Shin, for his guidance and support throughout my Ph.D. study. Professor Shin created a perfect research environment where I can focus on research and pursue interesting projects. He imparts not only knowledge, but also methodologies for managing time, distilling innovative ideas, leading and mentoring other researchers, *etc.* He has also been a role-model for me in keeping a peaceful mind and maintaining an energetic and enthusiastic state even under pressure-filled situations. The spirits that I inherited from him will continue inspiring me throughout my career and life.

I would also like to thank Professors Brian Noble, Prabal Dutta and Achilleas Anastasopoulos for serving on my thesis committee. I thank them for spending time to review the thesis and provide valuable feedback.

I am thankful to colleagues in the Real-Time Computing Laboratory (RTCL), who have contributed to the development of this thesis. Eugene Chai has been my mentor in launching software radio based projects. Many ideas in my thesis benefited from stimulating discussions with him and the extensive comments from him. Alex Min taught me the basics in cognitive radio networks and pointed me to important problems in this area. Fangjian Jin, Caoxie Zhang, Kyu-Han Kim, Hyoil Kim, Ashwini Kumar, Hahn-Sang Kim and Jaehyuk Choi offered very detailed comments in my research projects. I am also grateful to Karen Hou, Xin Hu, Yuanyuan Zeng,

Matt Knysz, Katharine Chang, Zhigang Chen, Jian Wu, Antino Kim, Buyoung Yun and Jisoo Yang for making RTCL a nice and active research environment.

I would also like to thank my Master's thesis advisor, Professor Baochun Li of the University of Toronto, for helping and encouraging me during my M.S. study. He helped me grow from a novice to a mature researcher in wireless networking. Without his inspiration, I would not have walked so far along the academic career. During my Master's study at the University of Toronto, colleagues in the iQua group helped me a lot. Chen Feng offered insightful comments regarding research methodologies, especially the analytical approach. Yunfeng Lin pointed me to important research materials and potential directions. Chuan Wu offered many valuable comments on my research and continuously encouraged me during my graduate study. Di Niu, Henry Xu, Hui Wang, Jin Jin and Mea Wang have also been great friends and co-workers.

During my summer internships, Dr. Kun Tan (MSRA), Dr. Karthik Sundaresan and Dr. Amir Khojastepour (NEC Labs America) have offered valuable insights regarding research and helped shape my research style. I appreciate their patience and kind help.

In addition, I gratefully acknowledge the financial support from the National Science Foundation under grants 0519498 and 0721529.

Last but not least, I wish to express my gratitude to my family, whose unconditional support has been the most important power that made me persevere throughout the past years. To them I dedicate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

MAC/PHY Co-Design of CSMA Wireless Networks Using Software Radios

by

Xinyu Zhang

Chair: Kang G. Shin

In the past decade, CSMA-based protocols have spawned numerous network standards (e.g., the WiFi family), and played a key role in improving the ubiquity of wireless networks. However, the rapid evolution of CSMA brings unprecedented challenges, especially the coexistence of different network architectures and communications devices. Meanwhile, many intrinsic limitations of CSMA have been the main obstacle to the performance of its derivatives, such as ZigBee, WiFi, and mesh networks. Most of these problems are observed to root in the abstract interface of the CSMA MAC and PHY layers — the MAC simply abstracts the advancement of PHY technologies as a change of data rate. Hence, the benefits of new PHY technologies are either not fully exploited, or they even may harm the performance of existing network protocols due to poor interoperability.

In this dissertation, we show that a joint design of the MAC/PHY layers can achieve a substantially higher level of capacity, interoperability and energy efficiency than the weakly coupled MAC/PHY design in the current CSMA wireless networks. In the proposed MAC/PHY co-design, the PHY layer exposes more states and ca-

pabilities to the MAC, and the MAC performs intelligent adaptation to and control over the PHY layer. We leverage the reconfigurability of software radios to design smart signal processing algorithms that meet the challenge of making PHY capabilities usable by the MAC layer. With the approach of MAC/PHY co-design, we have revisited the primitive operations of CSMA (collision avoidance, carrier signaling, carrier sensing, spectrum access and transmitter cooperation), and overcome its limitations in relay and broadcast applications, coexistence of heterogeneous networks, energy efficiency, coexistence of different spectrum widths, and scalability for MIMO networks. We have validated the feasibility and performance of our design using extensive analysis, simulation and testbed implementation.

CHAPTER I

Introduction

1.1 Background

Since its introduction in the 1970s, carrier sensing multiple access (CSMA) has been widely adopted to arbitrate the channel access of competing radio devices. CSMA has experienced a boom especially thanks to its application to the WiFi standards (IEEE 802.11a/b/g/n/ac) for wireless LANs, which now form a multi-billion consumer market and continue growing. Besides, CSMA is being adopted by many emerging wireless architectures, such as the IEEE 802.15.4 wireless personal area networks [4], wireless sensor networks, mesh networks, cognitive radio networks (IEEE 802.22) and white-space networks (IEEE 802.11af).

The main reason for the wide adoption of CSMA lies its simplicity, distributed and asynchronous nature. By integrating several primitive functionalities, such as carrier sensing and backoff, CSMA keeps the collision between neighboring radio devices to a minimum level. Unlike other schemes (*e.g.*, TDMA and FDMA), CSMA does not require synchronization among nearby radio devices, which substantially simplifies the establishment of network topology and enables the support for node mobility. Asynchronous operation also relaxes the required clock accuracy, thus reducing the hardware cost. In addition, CSMA adopts a hierarchical network topology — access points (APs) are deployed to provide infrastructure support for mobile clients. Such

a topology makes a tradeoff between network reliability and service availability.

All the benefits of CSMA come from some basic MAC-layer primitives, which remain intact throughout the decades of evolution. Below we briefly describe these core primitives.

1.1.1 MAC layer primitives

1.1.1.1 Carrier sensing

Carrier sensing (or listen-before-talk) is the first feature that distinguishes CSMA from other MAC protocols. Before sending a frame, the radio must sense the channel and ensure it is idle for a certain period of time. This operation is also referred to as clear channel assessment (CCA) in existing standards (*e.g.*, 802.11 and 802.15.4). CCA may be realized in two forms: physical carrier sensing and virtual carrier sensing.

In physical carrier sensing, the transmitter assesses the channel status by comparing the energy level with a CCA threshold (*e.g.*, -81dBm in 802.11 [5]). The energy level is essentially the accumulated energy of multiple samples produced by the radio's ADC (analog to digital converter). In virtual carrier sensing, the transmitter attempts to decode incoming signals and parse the header portion of packets, which may contain control information, such as packet type and duration. Packet headers are usually sent with the lowest level of modulation, and have relatively higher SINR than the data portion. Essentially, virtual carrier sensing is the same as idle listening — the radio needs to continuously sense the channel, detect incoming packets, parse their headers, and then determine if the packet is intended for it (address filtering).

Virtual carrier sensing is useful when explicit channel reservation is needed. For example, 802.11 includes a TxOP primitive, which allows a transmitter to reserve the channel by broadcasting a packet that declares a busy period. A similar principle is applied in the RTS/CTS exchange between transmitter/receiver, which precedes the data packet, reserves channel, and prevents other hidden terminals from interrupting

the transmission.

However, virtual carrier sensing is only applicable for nodes within the same contention domain. In heterogeneous wireless networks (*e.g.*, nearby wireless LAN cells have different channel widths or coexist with alien devices like 802.15.4 ZigBee nodes), different links cannot parse each other's packets due to PHY-layer heterogeneity. Hence, physical carrier sensing based on energy detection becomes the only measure of CCA.

1.1.1.2 Carrier signaling

Carrier signaling is the primitive that a radio uses to declare a busy channel to all its neighbors. In early generations of CSMA [124, 57], a dedicated busy-tone packet is sent explicitly, and concurrently with the data packet, but through a separate control channel. Modern CSMA networks (*e.g.*, WiFi and ZigBee) have adopted an implicit carrier signaling scheme — The data packet itself is used for declaring a busy channel. Implicit carrier signaling substantially simplifies the radio hardware, but at the cost of sacrificing the merits of dedicated busy-tones. In particular, it fails in heterogeneous wireless networks where transmitters have different power levels — low-power transmitters' data packets cannot be heard by high-power transmitters who are far away but may still cause interference.

1.1.1.3 Collision avoidance

CSMA relies on a randomized backoff protocol to reduce the risk of collision. The backoff duration is determined by the window size. A transmitter randomly chooses a backoff window size, and counts down the window whenever the channel remains idle for one time slot. It starts transmission once the window reaches 0. Collision may still occur (though with a low probability) if two transmitters choose the same initial window size, but it is resolved by allowing the transmitters to reinitialize the

backoff window and resend the packets.

1.1.1.4 Spectrum access

CSMA wireless networks have mostly been using the 2.4GHz and 5GHz ISM spectrum. In each wireless LAN or PAN cell (consisting of one access point and multiple clients), to ensure seamless communication, all nodes must reside on the same radio spectrum, defined by a center frequency and spectrum width (bandwidth). Spectrum is allocated to a cell *a priori* and the MAC protocol needs not be aware of the actual spectrum in use. When multiple cells are co-located, their spectrum may partially overlap, and a variety of spectrum widths may coexist with each other. Ideally, a wide spectrum should provide higher capacity than a narrow one. However, as we will discuss in Chapter III, due to spectrum heterogeneity, the dumb access mechanism in traditional CSMA may result in the converse.

1.1.2 Transmitter cooperation

Transmitter cooperation is a primitive that schedules concurrent transmissions from multiple senders, so as to achieve diversity gain (*i.e.*, reducing packet loss rate) or multiplexing gain (*i.e.*, increasing the number of concurrent data streams). Theoretical work has been done to explore transmitter cooperation between distributed wireless transmitters [125]. In practice, transmitter cooperation in CSMA networks has only been realized in the form of MIMO (as in 802.11n) or Multi-User MIMO (MU-MIMO, as in 802.11ac) communications, where the transmitters are antennas co-located on the AP (Fig. 1.1). For such a MIMO architecture, the diversity or multiplexing gain is limited to each contention domain (*i.e.*, a WLAN cell). The gain is not scalable to multiple cells since different APs still need to contend for channel access independently. In other words, the CSMA-based MAC layer lacks a transmitter cooperation primitive that is specifically designed for large-scale multi-cell networks.

Figure 1.1: Transmitter cooperation in the CSMA-based WiFi networks.

1.1.3 PHY-layer evolution of CSMA

The PHY layer of CSMA wireless networks involves not just hardware design, but also communications and signal processing algorithms. The advances in these domains have continuously driven the evolution of the CSMA PHY layer.

Early generation of 802.11 networks adopted variants of DSSS (direct-sequence spread spectrum) communication schemes, but the new generations (*e.g.*, 802.11a/g/n/ac) have mostly adopted OFDM which claims higher spectrum utilization. Furthermore, the low-level modulation mechanisms improved from BPSK/QPSK to 16QAM and 256QAM, which dramatically increases the data rate.

Meanwhile, the radio hardware is becoming more heterogeneous. Although WiFi is the dominant wireless device used for mobile Internet access, other types of devices (*e.g.*, ZigBee) are gradually deployed to support alternative applications such as smart homes and industrial monitoring [83]. These devices may be deployed near WiFi networks and share the same ISM spectrum with them. However, different devices may have disparate PHY-layer characteristics, such as communications mechanism, transmission power (range), time resolution, *etc.*

The spectrum width used by radio devices witnessed a similar level of heterogeneity. For example, the WiFi spectrum increased from the standard 20MHz in early generations of 802.11 to the 40MHz in 802.11n, and 160MHz in 802.11ac, in order to support high-rate applications such as HD video streaming. The 802.11-2007 [5]

also introduced narrower spectrum usage (5MHz and 10MHz), which supports applications that require low bit-rate but high energy efficiency.

In addition, as mentioned in Sec. 1.1.2, the number of antennas increased from 1 in 802.11a/b/g to 4 in 802.11n/ac, evolving the links from SISO to MIMO and multi-user MIMO mode. Such PHY-layer advances result in a continuous growth of the wireless network capacity, which matches the growing demands from mobile network devices and applications.

1.2 Motivation

Over the past decade, CSMA spawned numerous network standards and incorporated many advanced communications technologies. Such evolution is accompanied by unprecedented challenges, especially the coexistence of different network architectures and communications devices. Meanwhile, many intrinsic problems of CSMA remain in its derivatives, such as ZigBee, WiFi, and mesh networks. In this dissertation, we have identified the following limitations of CSMA wireless networks.

Redundant collision avoidance. In multi-hop wireless networks such as 802.11s based mesh networks, neighboring transmitters often have to forward packets containing the same information. For example, in network-wide broadcast, each node that receives the broadcast message needs to continue to forward it to other neighboring nodes. But traditional CSMA does not discriminate the packets — neighboring transmitters need to transmit sequentially to avoid collision, even though they intend to forward the same information. As a result of the sporadic schedule from collision avoidance, network-wide delay-optimal broadcast for CSMA remains an open problem. However, from an information-theoretic perspective, when multiple transmitters attempt to send the same information, they should enhance rather than collide with each other. Such a conceptual idea has already been discussed in information theory [125] and can be realized using interference-cancellation-based PHY layer communi-

cation algorithms. Hence, a protocol that is aware of such PHY layer capabilities may enable concurrent transmission of neighboring forwarders, thereby reducing the cost of collision avoidance.

Coexistence of heterogeneous spectrum widths. Emerging WLAN standards have been incorporating a variety of channel widths ranging from 5MHz to 160MHz, in order to match the diverse traffic demands on different networks. Unfortunately, the current 802.11 MAC/PHY is not designed for the coexistence of variable-width channels. With extensive measurement (Chapter III), we find that overlapping narrow-band channels may block an entire wide-band channel, resulting in severe spectrum under-utilization and even starvation of WLANs on the wide-band. A similar peril exists when a WLAN partially overlaps its channel with multiple orthogonal WLANs.

Coexistence of heterogeneous networks. In current CSMA wireless networks, pectrum sharing among the same network of devices can be arbitrated by the MAC operations, but the coexistence between heterogeneous networks remains a challenge. The disparate power levels, asynchronous time slots, and incompatible PHY layers of heterogeneous networks severely degrade the effectiveness of traditional MAC. Our measurement study shows moderate to high WiFi traffic to severely impair coexisting ZigBee's performance (Chapter IV). These effects have also been observed in real-world deployment of ZigBee sensor networks [83]. Hence, it is imperative to refine CSMA to enable the coexistence of heterogeneous networks.

Energy waste in idle listening. WiFi interface is known to be a primary energy consumer in mobile devices, and idle listening (IL) is the dominant source of energy consumption in WiFi. Unfortunately, IL is useless from the PHY layer perspective, since no information is been sent or received during IL. Most existing protocols, such as the 802.11 power-saving mode (PSM), attempt to reduce the time spent in IL by sleep scheduling. However, through an extensive analysis of real-world

traffic, we found more than 60% of energy is consumed in IL, even with PSM enabled (Chapter V).

Limited scalability for MIMO networks. MIMO communications evolved from theory to practice, and became a landmark for the PHY layer of advanced CSMA networking standards (*e.g.*, IEEE 802.11n and 802.11ac). However, these standards limit the MIMO operation within each contention domain. Network-wide MIMO cooperation remains an open problem in practice, simply because MIMO cooperation requires stringent synchronization between distributed transmitters, which is against the decentralized and asynchronized nature of CSMA. Hence, the actual capacity of current MIMO networks is far from the theoretical limit [125].

Most of the above problems are caused by the way how the MAC layer interfaces with the PHY layer through abstraction. Although PHY-layer features are constantly evolving, the basic MAC operations remain intact, and they tend to abstract the PHY layer merely as a module that provides a certain data rate. The abstract interface enables easy maintenance of the MAC and PHY layers as developers can change either layer without extensive knowledge of the other. However, it misses many opportunities to improve the network performance and interoperability. As the PHY layer evolves, it may even become the bottleneck that prevents PHY layer advances from being translated into network-level performance improvement.

1.3 Research Objectives and Contributions

We propose co-design of MAC/PHY layers that synthesizes the basic MAC operations with novel PHY algorithms for CSMA wireless networks, in order to overcome the above limitations that prevent CSMA from achieving a higher level of capacity, interoperability and energy-efficiency. Instead of being abstracted as providing certain data-rate, the PHY layer can expose a richer set of states and capabilities (*e.g.*, the capabilities of resolving collision, changing clock rate and spectrum widths) to

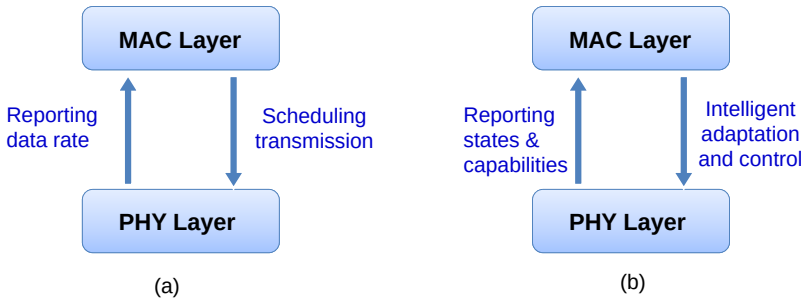

Figure 1.2: Interface between the CSMA MAC&PHY layers: (a) the traditional CSMA networks with an abstract interface; (b) MAC/PHY co-design which encourages richer interactions between MAC and PHY.

the MAC layer. Then, the MAC layer performs intelligent adaptation and control over these PHY layer capabilities (Fig. 1.2), thereby achieving much better network performance than the conventional CSMA networks.

With the co-design of MAC/PHY in mind, we revisit five primitive operations of CSMA, and overcome its limitations in capacity, interoperability and energy-efficiency (Table 1.1). A key challenge in realizing MAC/PHY co-design is how to make the PHY capabilities and states usable by the MAC layer. We have used software radios extensively to design smart signal processing algorithms, which are controllable through MAC-layer schemes. These signal processing algorithms require modifications to the PHY layers (radio firmware/hardware) and are not directly executable on the current wireless transceivers. However, the advent of high-performance software radios will eventually enable reconfigurable transceivers and the deployment of such algorithms. Further, we note that our design (Table 1.1) focuses on multiple basic CSMA operations, each targeting different network scenarios (relay network, heterogeneous networks, large-scale MIMO networks *etc.*), but they can be integrated into one reconfigurable radio platform. In effect, each design can be triggered by the built-in MAC layer mechanism (*e.g.*, the cognitive sensing in CSMA/CR, the temporal/frequency sensing in ASN) that identifies its application scenarios. In what follows, we summarize the rationale and contributions behind our design.

Chapter	System	Redesigning CSMA operation	PHY states and capability	MAC adaptation	Objective
II.	CSMA/CR	Collision avoidance	Resolving collision caused by packets carrying the same data	Cognitive sensing and scheduling of packets with the same identity	Delay-optimal broadcast and distributed asynchronous relaying
III.	ASN	Spectrum access	Fine-grained access to OFDM subbands via subcarrier nulling	Scheduling fine-grained spectrum access and adapting packet size	Coexistence of different WiFi generations (with heterogeneous spectrum widths)
IV.	CBT	Carrier signaling	WiFi/ZigBee's spectrum distribution; ZigBee's frequency flip capability	Scheduling frequency flip and busy-tone signaling	Coexistence of heterogeneous CSMA networks (ZigBee and WiFi)
V.	E-MiLi	Carrier sensing	Downclocking and sampling-rate invariant detection	Opportunistic downclocking; Minimum-cost address sharing	Energy-efficiency (reducing the dominating idle listening power)
VI.	NEMOx	transmitter cooperation	Network-level MIMO cooperation	Decentralized scheduling of distributed cooperating points	Scalable network MIMO cooperation

Table 1.1: Summary of the contributions and approaches in MAC/PHY co-design.

Redesigning the collision resolution mechanism, to enable delay-optimal broadcast and asynchronous cooperative relaying. We introduce a new MAC/PHY mechanism called CSMA with collision resolution (CSMA/CR) to overcome the inefficiency of CSMA in relay and broadcast applications. In CSMA/CR, a node that receives overlapping copies of the same packets (sent by different transmitters) can resolve the resulting collision using a PHY layer signal processing algorithm. The PHY layer exposes such a collision resolution capability to the MAC layer. The MAC layer then uses a cognitive sensing mechanism to identify and encourage collisions caused by neighboring senders holding the same outgoing packets.

CSMA/CR enables a collision-tolerant broadcast protocol called *Chorus*, which is proved to achieve asymptotically optimal delay performance, and exhibits high resilience to packet loss and node mobility in large-scale simulation experiments. To validate the feasibility of CSMA/CR, we prototyped and experimented with the collision resolution on a software radio platform.

In addition, CSMA/CR leads to the design of an asynchronous cooperative relaying protocol. Traditionally, cooperative communication requires nanosecond-level synchronization accuracy among distributed relays, which has been a major obstacle for its practical usage. Using CSMA/CR, the relays only need millisecond-level synchronization, but can still harvest the advantages from cooperation. Observing that the cooperation gain sacrifices the spatial reuse opportunity from competing flows, we establish a probabilistic and graph-theoretic model that quantifies this fundamental tradeoff, and identifies the range where the gain dominates.

Redesigning the spectrum access mechanism, to enable partial spectrum sharing in CSMA networks. We attribute the main reason of CSMA's failure in heterogeneous spectrum widths to an obsolete design choice: it deems an entire channel as an atomic spectrum block, and hence, a wideband may be blocked or even starved when it partially shares spectrum with narrowband channels. We solve

this problem with a new mechanism called *adaptive subcarrier nulling* (ASN), which enables finer-grained spectrum access in wireless LANs. ASN redesigns the packet detection and decoding algorithms in 802.11, so that a transmitter can use subchannels to send packets, and the receiver can receive a packet without prior knowledge of its spectrum usage. Such a salient PHY layer capability allows the MAC layer to opportunistically schedule transmission over a group of idle subchannels, and avoid collision with busy subchanels. The MAC layer further ensures fair access to shared subchannels by adapting the packet duration together with spectrum width. We implement a prototype of ASN using software radios and also validate its performance using large-scale trace-driven simulations. ASN represents another co-design of MAC/PHY to address a general problem that accompanies the evolution of CSMA networks.

Redesigning the carrier signaling mechanism, to enable the coexistence of heterogeneous CSMA networks. Observing the failure of CSMA in heterogeneous networks is due mainly to its implicit carrier signaling scheme, we propose a mechanism called cooperative busy tone (CBT) to enhance coexistence. The basic idea is to separate carrier signaling from data transmission — CBT employs a separate ZigBee node (called a signaler) to emit a busy-tone, thereby improving the visibility of ZigBee devices to WiFi. The key challenge of CBT lies in concurrently scheduling the busy-tone and data packet without causing interference between them. To overcome this challenge, we apply the principle of MAC/PHY co-design, allowing the PHY layer to expose ZigBee/WiFi's spectrum distribution and ZigBee's channel switching capability to the MAC layer. The MAC layer then schedules the busy-tone at appropriate time and frequency, in order to prevent mutual interference between the busy-tone signaler, ZigBee and WiFi transmitter. With a prototype implementation, CBT is shown to reduce collision rate by 40% to 90% compared to CSMA. It is further validated in a stochastic framework, which is the first model to analyze the

coexistence of different CSMA protocols.

Redesigning the carrier sensing mechanism, to reduce the dominant idle listening power. We propose E-MiLi to reduce the dominating IL power of CSMA by adaptively downclocking the radio. In E-MiLi, the PHY layer exposes the capability of reducing clock rate, and the MAC layer is responsible for determining when to downclock the radio without hurting the receiver's performance.

Downclocking has long been deemed as infeasible by network researchers, because it violates the Nyquist-Shannon sampling theorem and causes decoding failure for all packets. E-MiLi circumvents this fundamental challenge by separating packet detection from decoding. It incorporates a novel signal processing algorithm that ensures accurate packet detection and address filtering even when the receiver is significantly downclocked. With this smart signal processing algorithm, E-MiLi makes the PHY layer downclocking capability usable by the MAC layer. After detecting a packet, the receiver restores full clock rate and decodes the data following the sampling theorem. We prototype E-MiLi on software radios, and observed around 44% energy saving by running it over real-world WiFi traffic traces.

Redesigning the transmitter cooperation mechanism to achieve scalable network MIMO. To scale the MIMO advantage to large CSMA wirelss networks, we propose a new network architecture and protocol called NEMOx. NEMOx's PHY layer fully leverages the diversity/multiplexing gain of MIMO through cooperation between distributed antennas. The MAC layer maintains the CSMA-style channel contention, while controlling the grouping/cooperation of antennas and scheduling their transmission.

In NEMOx, the network is organized into multiple clusters, each consisting of one cluster head connected to multiple distributed antenna elements (referred to as *cooperating points*, or CPs) spanning a large area (covering multiple WLAN cells). The CPs are synchronized to the cluster head via RF cables, and thus many traditional

multi-user transmission and detection (e.g., MU-MIMO [122, 50] and interference alignment [27, 52, 84]) schemes can be applied to enable concurrent link transmissions. However, in between cells, contention still occurs and needs to be arbitrated by a distributed medium access control scheme. Within such an architecture, we show that a greedy approach where each AP always contends for the opportunity to enable all CPs may even perform worse than CSMA without link cooperation. We propose an opportunistic cooperation scheme that enables scalable MIMO cooperation. Further, we redesign the backoff and association mechanisms, to ensure fair channel access between the DAS cells. NEMOx marks the first step towards a practical DAS for CSMA wireless LANs, and a framework that synthesizes prior work on PHY-layer cooperation for network performance improvement.

1.4 Thesis Organization

The remainder of this dissertation is organized as follows. In Chapter II we introduce the proposed collision resolution mechanism and its application in broadcast and cooperative relaying for wireless mesh networks. In Chapter III we describe the adaptive subcarrier nulling scheme that redesigns the spectrum access and enables partial spectrum sharing in CSMA wireless networks. In Chapter IV we introduce cooperative busy-tone (CBT), an explicit carrier signaling mechanism that makes CSMA effective for heterogeneous networks. In Chapter V, we redesign the carrier sensing and idle listening mechanism, in order to boost the energy-efficiency of CSMA wireless networks. In Chapter VI, we propose NEMOx, a generalized framework to enable scalable MIMO cooperation for wireless LANs. Finally, Chapter VII summarizes the contribution of this dissertation and proposes future work.

CHAPTER II

Redesigning the Collision Resolution Mechanism

2.1 Introduction

Much of the recent work in multi-hop wireless mesh networks [10] has assumed an 802.11 based MAC/PHY layer. The 802.11 family of protocols [67] are built on the CSMA/CA scheduling mechanism, which senses the channel via energy detection, and performs exponential backoff upon transmission failure. Such a conservative scheduling protocol has demonstrated effectiveness for reducing collision in single-hop wireless LANs, when different clients request independent traffic. However, CSMA/CA ignores the existence of homogeneous traffic in two important communication primitives: i) broadcast, which delivers a packet (or a continuous stream of packets) from the source node to all other nodes in the network; and ii) cooperative relaying, which allows a relay to overhear the source's transmission, and then forward the data to the desired receiver in case the direct delivery attempt fails. In such cases, the same packet may be repeated by multiple transmitters. Ideally, transmissions of the same packet should complement, or at least do not interfere each other. However, the CSMA/CA mechanism is designed oblivous of such homogeneous traffic. This obliviousness is mainly due to the separation of concern in the early development of wireless MAC protocols and PHY hardware. However, with the advent of high-performance software radios, such as Sora [120], it becomes possible to directly program the MAC/PHY of

wireless protocols and make it application aware. In this report, we propose such a protocol called CSMA with collision resolution (CSMA/CR), and use it to boost the performance of broadcast and cooperative relaying protocols.

The key insight behind CSMA/CR is that *packets carrying the same data can be detected and decoded, even when they overlap at the receiver with comparable strength.* Via MAC layer cognitive sensing and scheduling, CSMA/CR encourages concurrent transmission of the same packets from different relays. It then uses PHY layer signal processing to resolve the resulting collisions. Based on CSMA/CR, we build an efficient broadcast protocol called *Chorus*, and a cooperative relay protocol called *DAC* (distributed asynchronous cooperation). The following scenarios illustrate the motivation behind this set of protocols.

2.1.1 Motivating Scenarios

2.1.1.1 CSMA/CR for efficient broadcast

Fig. 2.1(a) illustrates a typical scenario where CSMA/CA limits the broadcast efficiency. With CSMA/CA, at least three time slots are necessary to deliver one packet from source S to all other nodes. A and B cannot transmit concurrently, even if they have to forward the same packet. In a lossy network, suppose node D had already received the packet, while C and E await the retransmission from A and B, respectively. In an optimal scheduling protocol, A and B are allowed to transmit the packet concurrently, oblivious of the collision at D. However, this is not possible in CSMA/CA, as one of them will back off immediately upon sensing the other's activity.

In contrast, with Chorus (Fig. 2.1(b)), A and B can now transmit packets immediately and independently after receiving them from the source. Node D exploits collision resolution to decode the two collided packets from A and B. Therefore, only 2 time slots are required to deliver 1 packet over the entire network, due to the improved

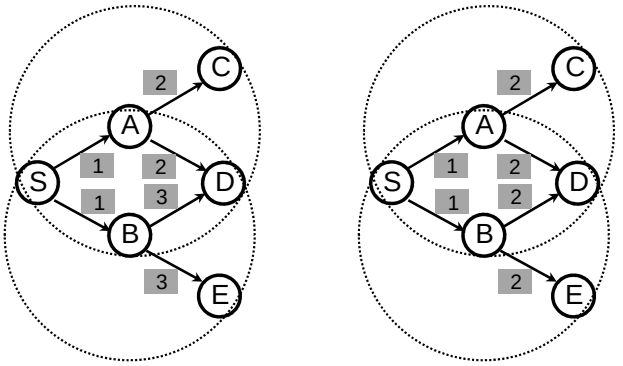

Figure 2.1: Broadcast with traditional CSMA/CA in 802.11, in comparison with CSMA/CR (CSMA with collision resolution). The shaded tags denote the order of transmissions.

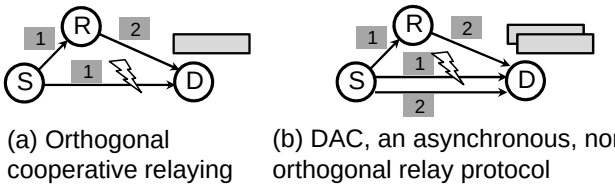

Figure 2.2: A contrast between traditional relaying and DAC.

spatial reuse. Moreover, when links are unreliable, the two decoded packets from A and B create *transmit diversity* for the common receiver D, without consuming any additional channel time.

2.1.1.2 CSMA/CR for cooperative relaying

It has been well-understood in information theory that relays' cooperation can improve the rate and reliability of wireless links [125]. A typical cooperative communication protocol allows a relay to overhear the source's transmission, and then forward the data to the desired receiver in case the direct delivery attempt fails.

Existing non-orthogonal relaying schemes [79, 17] allow the relay and source to transmit at the same time in the second stage. In these seminal information-theoretic approaches, perfect time synchronization among relays is assumed *a priori*. However, unlike point-to-point MIMO links, cooperative communication is asynchronous by its nature since there is no global clock shared by the relays. Practical cooperative

relay protocols have mostly adopted a non-orthogonal approach, *i.e.*, only allowing the relay to transmit at the second stage (Fig. 2.2(a)). However, this approach essentially degrades the cooperative relaying to two-hop routing, and thus its performance is incomparable to non-orthogonal schemes.

With the CSMA/CR based protocol, DAC, it becomes possible to circumvent the synchronization barrier in non-orthogonal relaying schemes. DAC allows two relays (or the source and one relay) to concurrently forward the same packet to the destination (Fig. 2.2(b)). Even if one of them fails, the other can still be decoded without incurring additional channel access time. Hence, DAC improves the link reliability by exploiting additional spatial diversity from co-located relays.

2.1.2 Design Principles

Both the spatial reuse and transmit diversity gain in CSMA/CR are realized via its collision resolution scheme. Unlike traditional transmit diversity schemes such as beamforming [94], CSMA/CR does not require symbol time synchronization nor instantaneous channel state information. In reality, it is infeasible to synchronize the independent transmitters (such as S and R in Fig. 2.2) at symbol level [22, 94]. The CSMA/CR PHY layer exploits the asynchrony between them to identify collision-free symbols in the overlapping packets. It then initiates an iterative decoding process that subtracts clean and known symbols from collided ones, and obtains estimations of unknown symbols. The decoding succeeds as long as one packet has sufficient SNR, hence realizing the diversity offered by multiple transmitters.

At the MAC layer, CSMA/CR extends the widely-used CSMA/CA and integrates the collision resolution PHY with it. A key idea in our design is to use *cognitive sensing* and *cut-through relaying* to maintain maximal compatibility with the 802.11-style mechanism. Specifically, the relays forward a packet immediately (without buffering it) upon overhearing or seeing a retransmission header from the original source node.

Hence, the relays make transparent contributions without disrupting the retransmission, carrier sensing and exponential backoff decisions made by the source.

The collision-resolution capability enables efficient broadcast in the Chorus protocol, without any topology or neighborhood information. It also enables the DAC protocol to improve existing routing protocols by adding a secondary relay to each existing relay, and allowing them to forward packets concurrently. The rationale behind both protocols is that CSMA/CR improves the transmit diversity and spatial reuse of wireless mesh networks via intelligent scheduling and signal processing.

2.1.3 Evaluation Approaches

To verify the feasibility of collision resolution, we design and implement the CSMA/CR PHY layer on the GNURadio/USRP software radio platform [2, 39]. The core components in our design include packet-offset identification, channel parameter estimation, and sample level signal modeling and cancellation, which are detailed in Sec. 3.3. Our experimentation on a small relay network show that DAC can indeed make a diversity gain for typical SNR ranges.

Due to the limitation of our software radio platform, we cannot directly implement the CSMA/CR MAC, and the broadcast/relaying protocols. Therefore, we develop an analytical model with closed-form characterization of CSMA/CR's achievable bit error rate (BER) and packet error rate (PER). We modify the ns-2 PHY with this new packet reception model, and implement the Chorus and DAC protocol based on it.

We compare Chorus with a typical CSMA/CA based protocol. In a large set of randomly-chosen topologies, Chorus shows several-fold performance improvement in latency and PDR. The performance gain is relatively insensitive to network size, source rate and link quality, and is observed in both single- and multi-source broadcast scenarios. These properties are especially valuable for information dissemination in

large-scale wireless networks. To understand the asymptotic performance of Chorus, we rigorously analyze its network-level performance in terms of latency and throughput. We show that Chorus can achieve $\Theta(r)$ latency (r is the network radius), which is asymptotically lower than existing practical schemes.

Our simulation experimentation also demonstrates that the DAC protocol can significantly improve the throughput and delay performance of existing loss-aware routing protocols. It thereby reveals the potential and practicality of non-orthogonal cooperation in wireless relay networks. In applying the DAC relaying protocol to multiple network flows, we identify an important tradeoff between the diversity gain provided by concurrent relays, and the multiplexing loss due to expanded interference region. Our analysis reveals that DAC improves network throughput when the link loss rate is below a certain threshold, which can be exactly profiled for simplified topologies. Therefore, DAC is best applicable to lossy wireless networks (such as unplanned mesh networks [20]), where it can enhance the network throughput by improving the reliability of bottleneck links with a low reception rate.

Both Chorus and DAC signify the importance of exploiting PHY-layer signal processing to improve application performance.

2.1.4 Summary of Contributions

The main contributions of this work can be summarized as follows.

- We design and implement a collision resolution based PHY layer and test it on an actual ratio platform. The BER and PER performances are characterized theoretically.

- We design a MAC protocol that allows for concurrent scheduling of homogeneous traffic via collision resolution.

- We propose Chorus, a CSMA/CR based broadcast protocol that has asymptot-

ically higher performance than the widely used CSMA/CA approach.

- We use CSMA/CR to circumvent the synchronization barrier in non-orthogonal cooperative relaying protocols. We design DAC, a new relaying scheme that incorporates CSMA/CR into existing routing protocols. Based on an asymptotic analysis in tractable network models, we profile the sufficient condition when DAC improves the performance of existing routing protocols.

2.1.5 Organization

Sec. 5.9 reviews related efforts in wireless broadcast, cooperative relaying, and software radio based protocol design. Sec. 3.3 describes the design and implementation of the CSMA/CR PHY, *i.e.*, the collision resolution module. Sec. 3.4 introduces the MAC layer cognitive sensing and scheduling schemes in CSMA/CR. Sec. 2.5 and Sec. 2.6 discuss the detailed design of the Chorus and DAC protocols, respectively. Sec. 4.4 analyzes the BER, PER, and network-level asymptotic performance CSMA/CR, Chorus, and DAC. Further simulation experiments are presented in Sec. 5.7 to validate their performance. Finally, Sec. 5.10 concludes the report and discusses our future work. For clarity, *detailed proofs for all the analytical results are included in the Appendix.*

2.2 Related Work

2.2.1 Broadcast in Multihop Wireless Mesh Networks

Efficient broadcast in multihop wireless networks has been studied extensively, from both theoretical and practical perspectives. From the theoretical perspective, it is well-known that scheduling a minimum latency broadcast is NP-hard, either in a general undirected graph [63] or in a unit disk graph (UDG) [48]. Without the minimum latency constraint, analytical solutions demonstrated the feasibility of

scheduling with time complexity $\Omega(r \log n)$ [32] in a distributed anonymous broadcast, and $r + O(\log r)$ [64] in centralized broadcast with known topology, where r and n denote the network radius and number of nodes. More recent work has improved the efficiency, and adopted more realistic models such as the interference graph [88].

Practical broadcast protocols have mostly adopted the 802.11 CSMA/CA and extended it to multi-hop networks. A main mechanism is to prune the topology, leaving only a backbone that covers the entire topology. The double-coverage broadcast [86], for example, reduces redundant transmissions by selecting nodes that cover more neighbors, while ensuring each node is covered at least twice, such that retransmission can be exploited to improve delivery ratio. The fundamental difference between Chorus and such existing protocols lies in its MAC layer scheduling protocol. With a joint design of CSMA/CR and network level broadcast, Chorus can achieve the $\Theta(r)$ latency bound, hence it has both theoretical and practical relevance.

2.2.2 Cooperative Relaying

Cooperative diversity was originally proposed in information theory to realize the capacity of MIMO systems. The distributed space-time code [79] for two-stage cooperative communications has been widely explored to improve the performance of relay networks (see [77] for a survey). One remarkable progress is attributed to Azarian *et al.* [17], who showed that non-orthogonal cooperation schemes can approximate the performance of centralized MIMO systems through multiple relays. However, these cooperative relay protocols assume perfect time synchronization among relay nodes. Recently, Wei [126] and Li *et al.* [82] reduced the synchronization constraint to sub-symbol level, but assumed known and controllable time offsets between relays. DAC's diversity gain is incomparable with such synchronized schemes, and it only allows for two concurrent relays. However, to our knowledge, it is the first non-orthogonal relaying protocol without any symbol-level timing constraint.

The implication of cooperative relaying for higher layers has been studied recently. Jakllari *et al.* [70] directly applied the synchronized space-time code to establish virtual MISO links for routing. Sundaresan *et al.* [74] showed that the more practical two-phase orthogonal relaying scheme (Fig. 2.2(a)), driven by the retransmission diversity from relays equipped with smart antennas, can make a remarkable throughput gain.

An alternative approach to exploiting diversity gain is the orthogonal *opportunistic relaying* [22], which selects the best among all relays that overheard the source's packet, based on *instantaneous* channel feedback. In Sec. 3.4, we show that DAC can serve as a complement to opportunistic relaying. By allowing two relays, it provides redundancy across independently faded packets, thus further improving the link reliability.

2.2.3 Software Radio Solutions to MAC Problems

The advent of high-performance software radios has been inspiring wireless protocols beyond the CSMA/CA paradigm. For instance, interference cancellation [58] can be used to resolve two collided packets with disparate strength. The main challenge in applying interference cancellation to multi-hop wireless networks is that the transmitters need delicate power control to ensure decodability. In CSMA/CR, even two packets with similar strength can be effectively decoded, because each sees the other as a complement, rather than interferer. If the RSS of one packet is significantly lower than the other, such that it cannot be detected, then CSMA/CR automatically resorts to the capture effect to decode the strong packet.

CSMA/CR is partly inspired by the ZigZag protocol [51], which exploits the signal processing capability of software radios to solve the hidden terminal problem in WLANs. ZigZag extracts symbols from collided packets by identifying repeated collisions of two hidden terminals. It treats each collided packet as a sum over two packets.

The two original packets are recovered from two known sums, similar to solving a linear system of equations. CSMA/CR' collision resolution PHY is similar to ZigZag, but aims to resolve packets from a single collision with *sample level estimation and cancellation*. CSMA/CR aims at improving broadcast and relaying efficiency in wireless mesh networks, where it exploits transmit diversity and spatial reuse, using MAC layer cognitive sensing, scheduling and network level relay selection.

The feasibility of allowing concurrent transmissions to create diversity has also been explored in communications. Concurrent cooperative communication [109], for example, allows co-located wireless nodes to transmit at the same time, thus forming a virtual antenna array that increases signal strength at the common receiver. Beamforming protocols [94] synchronize the transmitters, such that their signals can combine coherently at the receiver. These techniques require strict frequency, phase, and time synchronization at the symbol level, among distributed transmitters. Such fined-grained synchronization remains an open challenge [94], due to the limited time resolution at the wireless nodes, and the variation of the wireless channels.

2.3 Collision Resolution: The PHY Design

The core component of CSMA/CR PHY lies in the signal processing module at the receiver, which can decode two overlapping packets carrying the same data. In this section, we focus on the design and implementation of this customized receiver module.

2.3.1 An Overview of Iterative Collision Resolution

Suppose two relays transmit the same packet towards the destination. Due to the randomness introduced by the transmitters' higher-layer operations, the probability that the two versions of the packet being aligned perfectly is negligible. The receiver identifies the natural offset between these two packet copies by detecting a preamble

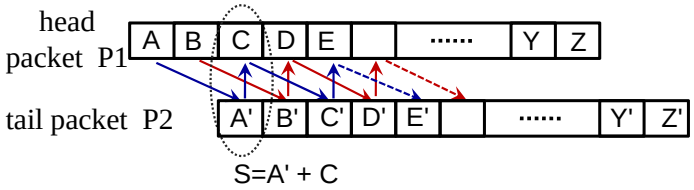

Figure 2.3: Iteratively decoding two collided packets carrying the same information, coming from two relays (or one source and one relay), respectively.

attached in their headers. It first decodes the clean symbols in the offset region, and then iteratively subtracts decoded symbols from the collided ones, thereby obtaining the desired symbols.

For instance, in Fig. 2.3, two packets (named *head packet* P1 and *tail packet* P2 respectively according to their arrival order) overlap at the receiver. We first decode the clean symbols A and B in P1. Symbol C is corrupted as it collides with A' in P2, resulting in a combined symbol S. To recover C, we note that symbols A' and A carry the same bit, but the analog forms are different due to the independent channel distortion. Therefore, we need to reconstruct an image of A' by emulating the channel distortion over the corresponding bit that is already known via A.

After reconstruction, we subtract the emulated A' from S, obtaining a decision symbol for C. Then, we normalize the decision symbol using the channel estimation for P1, and use a slicer to decide if the bit in C is 0 or 1. For BPSK, the slicer outputs 0 if the normalized decision symbol has a negative real part, and 1 otherwise. The decoded bit in C is then used to reconstruct C' and decode E. This process iterates until the end of the packet. The iteration for other collided symbols proceeds in a similar way.

2.3.2 Transceiver Design

The transmitter module in CSMA/CR (Fig. 2.4) is similar to legacy 802.11b, except that it adds a *CSMA/CR preamble* that assists packet detection. The transmitter maps a digital bit to a symbol according to a complex constellation ("1" and

"0" are mapped to 1 and -1, respectively). The symbol then passes through a root raised cosine (RRC) filter, which interpolates the symbol into I samples (we adopt a typical value $I = 8$) to alleviate inter-symbol interference. The RRC shaped symbol is the final output from the transmitter.

The receiver module is also illustrated in Fig. 2.4. In the normal case of decoding a single head packet, the receiver acts like a typical 802.11b receiver. Upon detecting a tail packet immersed in a head packet, the receiver identifies the exact start of the tail packet, rolls back to its first symbol, and starts the iterative cancellation algorithm. The receiver needs to replay the bit-to-samples transformation at the transmitter, as well as the channel distortion, when reconstructing a symbol in the tail packet. The channel distortion, including amplitude attenuation, phase shift, frequency offset, and timing offset, must be estimated and updated dynamically, since channel parameters vary during the decoding procedure, and the estimation error can accumulate, eventually corrupting the entire packet.

The main challenge in implementing CSMA/CR lies in identifying the exact offset between the two packets, and remodeling the symbols in the tail packet based on channel parameter estimation. Unlike interference cancellation [58], we must deal with the common case where collided packets have comparable RSS. Otherwise, the weak packet may be captured and offers no diversity gain. Unlike the symbol cancellation algorithm in ZigZag [51], the channel parameters must be estimated in a single collision. To obtain accurate estimation and reconstruction of the symbols, we extensively use *sample-level* correlation, remodeling, and cancellation, as discussed below.

2.3.3 Packet Detection and Offset Estimation

The original 802.11b PHY detects the start of a packet by identifying a sequence of known bits from the slicer output. In CSMA/CR, we need to detect the presence

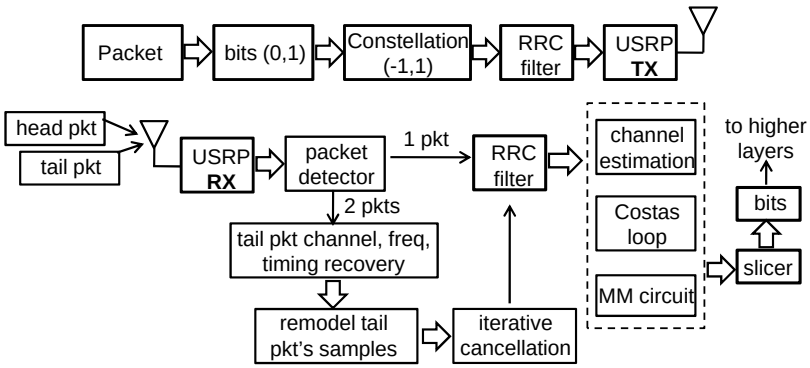

Figure 2.4: Flow-chart for CSMA/CR transmitter (upper) and receiver (lower).

of one or more packets before feeding the symbols into the slicer. This is achieved by using a combination of *energy* and *feature* detection.

Energy detection estimates a packet's arrival by locating a burst in the magnitude and phase of the received symbol. According to our experiment, a data symbol typically has at least 8dB SNR in order to be decoded error-free. Therefore, it is easy to identify the first symbol of the head packet. When the tail packet arrives and overlaps with the head packet, their corresponding complex samples add up. The magnitude and phase of the resulting symbol thus deviates from the previous symbols, which are relatively stable. CSMA/CR uses this deviation as a hint for packet collision.

Energy detection can provide a symbol-level offset estimation, while CSMA/CR necessitates sample-level estimation accuracy, since the overlapping symbols do not align perfectly. In addition, energy detection's false positive rate increases when ambient noise raises the RSS variation. Therefore, we combine it with feature detection to reduce false positives. Specifically, we correlate the raw decoded symbols with a 256-bit known preamble to confirm the packet arrival event. We use *differential correlation* (*i.e.*, correlating the phase difference of adjacent symbols with the known difference obtained from the preamble) in order to cancel out the transmitter/receiver frequency offset. The correlator outputs a peak whenever a packet arrives. The threshold configuration for peak detection is similar to [51]. Note that the correlation peak is 256

bits behind the first symbol, and therefore CSMA/CR maintains a circular buffer storing the latest 256 symbols and their samples, and rollbacks to the first symbol before cancellation.

The energy and feature detection confirms the packet arrival and indicates the symbol-level offset. The exact sample-level collision position is then identified by correlating the *samples* near the beginning of the tail packet with the known samples in the first 16 bits of the preamble (hence 128 known samples in total). The position where the maximum correlation magnitude occurs indicates the start of useful samples. To isolate channel distortion from transceiver distortion, the known samples are obtained offline from the output of a transmit filter.

2.3.4 Channel Estimation

We use the collision-free symbols in the beginning of the head packet to estimate its channel. The beginning of the tail packet is immersed in strong noise (*i.e.*, the signals in the head packet), and hence, a direct estimation is severely biased. Unlike prior signal cancellation algorithms [51, 58] that exploit signal capture or repeated collisions, we obtain coarse estimation of the tail packet by correlating and cancelling the known preamble, and then refine the estimation on-the-fly.

2.3.4.1 Amplitude and phase distortion

A coarse estimation of the channel can be obtained via sample level correlation. Suppose the known samples are $x(t), \forall t \in [1, K_s]$ ($K_s = 128$, as discussed above), then the received complex samples after channel distortion should be: $y(t) = Ax(t)e^{j\theta+j2\pi\Delta ft} + n(t)$, where $n(t)$ is the noise process; A and θ are the channel amplitude and phase distortion; Δf is the frequency offset between the transmitter and the receiver. After correlation, we get $Y = A\sum_{t=1}^{K_s}[x(t)e^{j\theta+j2\pi\Delta ft} + n(t)]x(t)$. The phase error due to frequency offset is typically on the order of 10^{-4} rad per sample, and

thus, its accumulating effect over the K_s samples is negligible. Further, the ambient noise plus the random samples from the head packet can partly cancel out, resulting in $\sum_{t=1}^{K_s} x^2(t) \gg \sum_{t=1}^{K_s} x(t)n(t)$. Therefore, we approximate the complex channel distortion as $C_d = Y(\sum_{t=1}^{K_s} x^2(t))^{-1}$.

2.3.4.2 Frequency offset estimation

We use the Costas loop [117] to estimate the residual frequency error in the received baseband signals, which is also the frequency offset between the transmitter and the receiver. Costas loop calculates the phase change between two adjacent symbols, and then updates the frequency error via first-order differentiation: $\delta f = \delta f + \omega \cdot (p(t+1) - p(t))$, where $p(t)$ is the symbol phase at time t, and ω is an update parameter, typically set on the order of 10^{-5}.

2.3.4.3 Timing recovery

Ideally, a receiver should align its sampling time with the transmitter to achieve maximum SNR. In practice, the sampling time may deviate from the peak position of the RRC-shaped sample envelop, reducing the effective SNR. A widely-adopted method to correct for sampling offset is the MM circuit [34], which uses a nonlinear hill-climbing algorithm to tune the received signals, such that the sample point is asymptotically aligned with the optimal sampling time.

Remarkably, the MM circuit works only when adjacent symbols have a comparable magnitude, which holds for single-packet decoding. For CSMA/CR, the collided symbols have large variations since they consist of symbols from different channels. Hence, we enable the MM circuit timing update only after the symbol cancellation. Further, we need to freeze the MM circuit, *i.e.*, fix its sampling step, whenever an energy burst is detected, indicating a potential collision. We re-enable it for each symbol in the head packet after the corresponding symbol in the tail packet is subtracted

out.

2.3.4.4 Transmitter distortion

Beside the channel distortion, the transmitter also pre-processes the signals using the RRC filter to combat multi-path fading. The RRC converts a symbol (1 or -1) into $I = 8$ samples as follows:

$$s_i(t) = x(t-1)F(\frac{3I}{2} + i) + x(t)F(\frac{I}{2} + i), i \in [0, \frac{I}{2})$$
$$s_i(t) = x(t)F(\frac{I}{2} + i) + x(t+1)F(i - \frac{I}{2}), i \in [\frac{I}{2}, I)$$

where $F(i)$ denotes the i-th filter coefficients. At the receiver side, this filtering process is replayed for the tail packet, observing that the digital bits $x(t)$ are already known from prior decoded bits in the head packet.

2.3.4.5 Correcting channel-estimation errors

Recall the initial correlation only provides coarse estimation of the channel gain in the tail packet. During the iterative cancellation procedure, we need to refine the estimation via a simple feedback algorithm. Specifically, we reconstruct an image of symbols in the head packet, and subtract these symbols, to get a refined estimation of symbols in the tail packet. We use the difference between this refined estimation and the original reconstructed image to calculate the channel estimation error, and then update the frequency and time offsets, in a similar manner to the above estimation for the head packet. Observing that the channel gain remains relatively stable for one packet, we use a moving average approach to update the channel amplitude and phase distortion for the tail packet.

One observation from our implementation is that the collision offset identification may also deviate from the exact collision position by one or two samples, especially when SNR is low. We exploit the MM circuit output to compensate for this error.

When the MM circuit outputs a sampling step larger than I, it indicates that the collision position is likely to be larger than initially estimated. Our algorithm then increases a credit value by Δt ($0 < \Delta t < 1$). When $\Delta t > 1$, we update the packet offset by 1. A symmetric update procedure is used when the sampling step is smaller than I.

2.3.5 Harvest Diversity with Packet Selection

Beside the iterative decoding in the forward direction, CSMA/CR can also work backward, starting from the clean symbols in the tail packet (symbol Y' and Z' in Fig. 2.3), until reaching its beginning, thus obtaining a different estimation of the packet. Since these two packets arrive at the receiver via two independent links, even if one fails in decoding, the other may still be correctly decoded. This is the basis of CSMA/CR's diversity gain, and will be rigorously justified in our analysis and experiments.

Note that the diversity gain comes at the expense of additional overhead, including the preamble and the extended reception time due to the packets' offset. However, the preamble length we use is only $K_b = 256$ bits, and the offset time can be easily confined within the duration of tens of bits, with state-of-the-art software radios [120]. In contrast, a typical data payload is around 1K Bytes. Therefore, the additional overhead of CSMA/CR is only on the order of 1%.

Also note that the channel estimation, sample remodeling and cancellation only involves linear-time operations. The correlation has $\Theta(n^2)$ complexity (n is the correlation length), but is only needed for around K_b symbols after the energy detection is triggered. In addition, the implementation of CSMA/CR is built on BPSK. However, the estimation, reconstruction and cancellation for higher-order modulation schemes, such as M-PSK (M=4, 8, 16, 64), can be realized in a similar way, except that the signal constellation is mapped to different complex vectors [51].

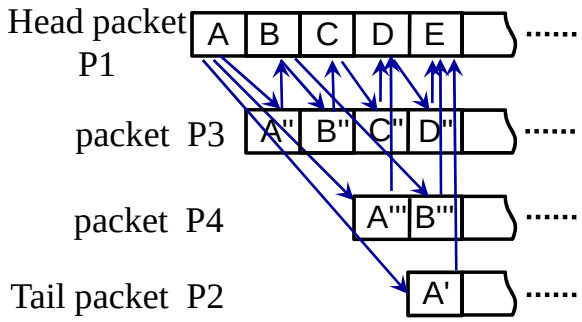

Figure 2.5: Collision resolution: the multi-packet collision case.

2.3.6 Multi-packet Collision Resolution

Since CSMA/CR allows concurrent transmissions, multiple versions of a packet can collide, especially when running broadcast and when the network has high density. The resolution of multi-packet collision is complicated by the fact that intermediate packets no longer have clean symbols at the beginning or end. Fig. 2.5 illustrates a typical scenario.

Denote the earliest and latest packets as *head packet* and *tail packet*, respectively. To decode the head packet, CSMA/CR proceeds in a way similar to the two-packet case, except that it needs to subtract multiple reconstructed symbols, including the one from the tail and those from the intermediate packets. Similarly, another version can be obtained by decoding the tail packet, but in reverse order, starting from its end backward to the beginning. To obtain additional versions from intermediate packets, the receiver performs simple hard decoding. It tracks the packet symbol-by-symbol, treating all others as noise. Intuitively, the results have reasonable confidence only when this packet has much higher strength than others. The achievable decoding confidence will be rigorously characterized in Sec. 4.4.

2.4 CSMA/CR: the MAC Design

We now introduce the MAC layer of CSMA/CR. We extend the 802.11-style CSMA, but integrate it with the *Collision Resolution* PHY. In designing CSMA/CR,

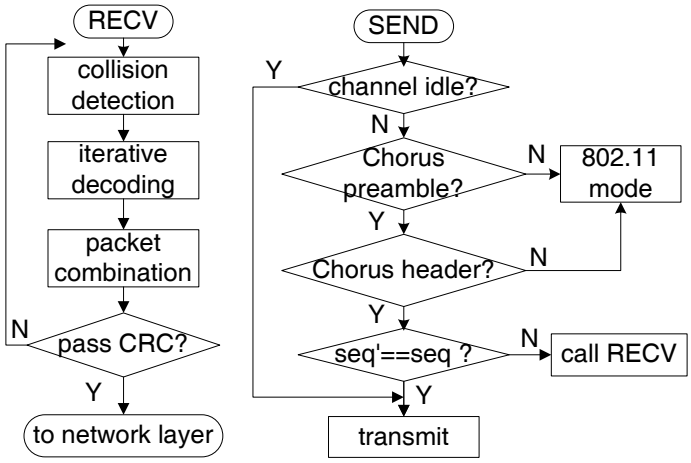

Figure 2.6: The MAC layer control flow in CSMA/CR. seq' denotes the sequence number of the packet on the air.

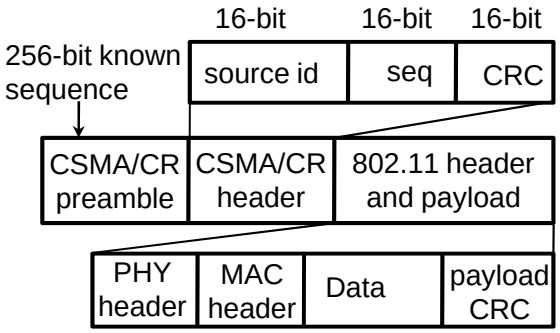

Figure 2.7: The packet format in CSMA/CR.

we aim at maintaining maximal compatibility with 802.11, adding the least overhead and modification to the original design.

2.4.1 MAC Layer Cognitive Sensing and Scheduling

CSMA/CR maintains the carrier sensing and backoff in the 802.11-based CSMA protocol, but adopts cognitive sensing that exploits the collision-resolution advantage, while avoiding unresolvable collisions. The principle of cognitive sensing is to decode the identity of the packet on the air, and accordingly, make the transmission decision. To this end, we first add a new header field into the 802.11 packet.

2.4.1.1 Packet format

Fig. 2.7 illustrates the modification to an 802.11 packet. First, a known *preamble* is attached to facilitate packet detection and offset identification (Sec. 2.3.3). Second, a *header* field is added, which informs the receiver of the packet's identity, including the session [1] ID and the packet's sequence number. A 16-bit CRC (Cyclic Redundancy Check) [117] is included in this header. In case of CRC failure, this packet is discarded as it conveys wrong identity information.

When the headers of two packets collide, CSMA/CR proceeds with the iterative decoding, assuming they have the same identity. After the decoding, it performs CRC over the header of each packet to ensure they are identical. If not, a decoding failure occurs, and both packets will be discarded. A decoding failure also happens when the CRC over the payload fails.

2.4.1.2 Cognitive Sensing and Scheduling

With the collision-resolution capability, each transmitter calls a SEND procedure to perform cognitive sensing, as shown in Fig. 2.6 Transmitters make scheduling decision following three rules:

R1. Forward a packet immediately if the channel is idle.

R2. If the channel is busy, and the packet in the air is exactly one of the packets in the transmit queue, then start transmitting the pending packet.

R3. If the channel is busy, but a preamble cannot be detected, or the header field of the packet on the air cannot be decoded, or a different packet is on the air, then start the backoff procedure according to the 802.11

R1 is typical of all CSMA protocols. R2 is unique to the CSMA/CR-based scheme. It enforces the principle behind collision resolution, *i.e.*, overlapping packets carrying

[1] A *session* is an end-to-end network flow that is identified by its source and destination ID. In a broadcast protocol, a session can be identified by its source ID.

the same data may not cause collisions. Instead, by collision resolution, these packets offer transmit diversity to the receiver. Therefore, a sender node, such as node B in Fig. 2.1, can transmit its pending packet if it has the same identify as the one on the air (*e.g.*, the one that A is transmitting). In contrast, CSMA/CA transmitters stall and back off whenever the channel is busy.

R3 ensures friendliness to alien traffic, and is relevant for multi-source broadcast and co-existence with CSMA/CA based unicast traffic. To prevent unresolvable collisions between different packets, a transmitter starts the normal 802.11 backoff if it senses that the channel is occupied by such alien traffic. To reduce interference to co-existing traffic, it also backoffs conservatively if the identity of the packet on the air cannot be decoded.

The advantages of cognitive sensing and scheduling come at the expense of additional overhead. In 802.11b, the sensing time slot is 50 μs (*i.e.*, the DIFS time [67]), equivalent to the channel time of 50 bits in the broadcast mode. In contrast, a CSMA/CR based protocol such as Chorus needs to sense over the entire preamble and the header (304 bits in total, as indicated in Fig. 2.7). However, this overhead is negligible compared to the typical packet length. We formalize the cost of the header overhead using both asymptotic analysis (Sec. 4.4) and simulation experiments (Sec. 5.7).

2.4.2 Discussion

The idea of allowing relay operation in the middle of source transmission has long been adopted by cut-through routing in wireline networks [49]. It has not been adopted in wireless networks, which typically operates on time-orthogonal mode, schedules transmission on a per-packet basis, and allows only one transmitter within the carrier sensing range. However, emerging high performance software radios makes it viable in wireless networks. For example, Sora [120] achieves a scheduling-granularity

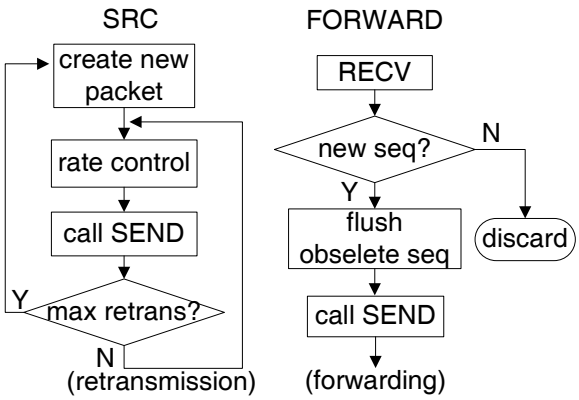

Figure 2.8: Control flow for scheduling network-wide broadcast.

comparable with the high-rate wireless standards (such as 802.11a) via programmable software and reconfigurable hardware.

For radio devices incapable of cut-through relaying, we adopt the following scheme built atop the 802.11 RTS/CTS mechanism. Before retransmission, the source sends an RTS packet, piggy-backing the retransmission bit and the packet's identity information in it. Upon overhearing this RTS and the subsequent CTS, both the source and the relay transmit the data packet. In current wireless transceivers, the decision making time is typically on the order of several microseconds [22], this randomness is sufficient to offer several bits' offset between the two transmissions, thus allowing for collision resolution at the PHY. For transceivers with higher time resolution, randomness can be introduced by allowing the source and relay to randomly backoff before starting the retransmission.

2.5 Chorus: Scheduling Network Wide Broadcast

We apply the above CSMA/CR MAC/PHY protocol to a simple collision tolerant broadcast protocol called Chorus. Broadcast in Chorus is anonymous and decentralized. The source and relays do not need any topology information or neighbor identity. Following the SRC procedure in Fig. 2.8, the source node composes a Chorus packet, and transmits it like a normal 802.11 broadcast packet. Each neighbor

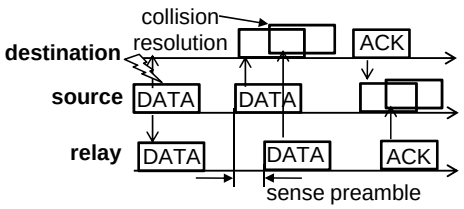

Figure 2.9: Cooperative relaying in DAC.

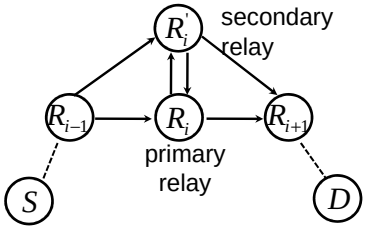

Figure 2.10: Improving an existing routing protocol using DAC.

who overhears this packet provides best-effort service by forwarding it *once*, following the FORWARD procedure. Receivers with overlapped packets perform collision resolution before continuing with the packet relaying. After each successful reception, a receiver flushes those pending packets with obsolete seq, in order to prevent unresolvable collisions between packets with different sequence numbers. Intuitively, multiple versions of a packet proceed in parallel like a wavefront, which stops at the network edge. In case of continuous broadcast, the source node can control its rate to prevent congestion, and perform retransmission to improve PDR. These further optimizations are up to the application and will not be used in our evaluation.

When multiple broadcast sessions are running concurrently, their packets are identified through the source-id field in the header part. Each relay maintains a transmit queue storing the packets to be forwarded. When the channel is idle, it directly transmits the head-of-line packet. Otherwise, it follows the MAC layer cognitive scheduling protocol, which maximizes the spatial reuse opportunity by scheduling the same packets, while avoiding collision with other broadcast sessions. Note that the co-existence with unicast traffic is a special case of multi-source broadcast. In effect, the latter case requires more conservative scheduling because of more severe interference, and therefore it will be used as a benchmark for validating Chorus' friendliness to alien traffic.

2.6 The DAC Cooperative Relaying Protocol

In this section, we introduce the CSMA/CR based cooperative relay protocol, *i.e.*, the DAC (distributed asynchronous relaying) protocol. A joint design of CSMA/CR and routing can provide optimal end-to-end delay performance. For simplicity and to emphasize the advantage of non-orthogonal cooperation, however, we adopt a simple and generic relay-selection approach in DAC that integrates CSMA/CR into existing routing protocols, given that the routes had already been selected. Specifically, we use the ETX routing [33] as a basis, and show how to improve its reliability and throughput using DAC relays.

2.6.1 Adapt CSMA/CR to Single-hop Non-orthogonal Relaying

Fig. 2.9 illustrates the basic operation of non-orthogonal cooperative relaying in a single-hop relay network (such as the relay network in Fig. 2.2). Suppose a direct source-destination link is already established by a routing protocol. The source makes a first attempt to transmit the data packet, which can be overheard by both the relay and the destination. If the packet reaches the destination, then CSMA/CR proceeds like CSMA/CA. Upon a failure, *i.e.*, the source receives no ACK from the destination, then it schedules a retransmission and sets a indicator bit in the header of the retransmitted packet. When the packet is emitted, the relay will forward the same packet it overheard, immediately after decoding the retransmission bit and the packet's identity (flow id, sequence number, and transmitter id), which are included in its header. This *cut-through relaying* introduces offset between the arrival time of the source's and relay's retransmission packets, and provides the necessary condition for collision resolution at the receiver.

Due to this asynchrony, the source still senses a busy channel immediately after completing the retransmission. It thereby extends the ACK timeout by the duration between current time and the end of this busy period, which is also the offset between

the source and relay's retransmissions. This procedure repeats until the source receives an ACK from the destination. To improve the reliability of ACK, the relay also schedules a cut-through relaying of the ACK packet, when it overhears the header of the ACK packet from the destination.

One remarkable point is that the relay facilitates the retransmission only when it asserts that the source be the only active transmitter within sensing range. This decision is made by looking into the NAV field in 802.11 MAC, which indicates activities in neighboring region, and by looking into the carrier sensing record right before the source's retransmission. If the relay senses a busy channel but cannot decode the identity of the transmitter, then it remains as a normal 802.11 transceiver.

2.6.2 A Generic Multi-Hop Relaying Scheme

The multihop cooperative relaying scheme in DAC is built upon an existing routing protocol, referred to as ETX routing [33]. Observing that real-world mesh networks tend to have a majority of links with intermediate quality [20], the ETX protocol adopts a loss-aware link metric, which is the expected number of transmissions needed for successfully delivering a packet on a link. This metric is used to find the shortest path for each data session (a source-destination pair).

Our basic idea is to optimize the ETX route on a per-hop basis. As shown in Fig. 2.10, suppose a *primary path* ($S \cdots R_{i-1} \rightarrow R_i \rightarrow R_{i+1} \cdots D$) consisting of *primary relays* has been established by ETX. For each primary relay R_i, we decide whether to add a *secondary relay* to it, and select the best secondary relay R_i', according to the potential performance gain in terms of reducing the delay from the previous hop R_{i-1} to the next hop R_{i+1}.

Before analyzing the potential gain, we first introduce the cooperation between the primary and secondary relays. Take the scenario in Fig. 2.10 as an example. In the normal mode, R_{i-1} makes a first attempt to forward a packet to R_i. Upon

successful reception, either R_i or R'_i or both of them can return an ACK. The DAC collision-resolution PHY ensures no ACK collision happens. From the perspective of R_{i-1}, it proceeds to the next packet as long as it can decode an ACK.

If only R_i receives the packet, then it schedules the forwarding following a normal DAC MAC, regarding R'_i as the relay. If both of them receive the packet, then R'_i will perform the cut-through relaying immediately after it senses R_i transmitting the packet it overheard. A primary relay piggybacks the session ID (represented by the source-destination of the path), sequence, and sender ID in the forwarded packet's header, so that it can be recognized in time by the secondary relay. An exception happens when only the secondary relay R'_i receives the packet. R'_i estimates the occurrence of such an event via the absence of R_i's ACK header that is intended for R_{i-1}. In this case, R'_i sends the ACK immediately, and then temporarily takes the position of R_i, serving as the primary forwarder, forming a typical 3-node local relay network together with R_i, following the DAC MAC. The control goes back to the primary relay R_i in the next successful packet transmission from R_{i-1} to R_i.

2.6.3 Relay Selection in DAC

The above protocol operations allow us to derive a model for analyzing the expected transmission delay, and selecting the optimal relay that incurs the minimum delay. Specifically, we model the progress of a packet as a Markov chain, driven by the transmission, cooperation and forwarding operations among primary and secondary relays. Following notations similar to those in Sec. 3.4, we have the following proposition.

Proposition II.1. *The expected delay in delivering a packet from R_{i-1} to R_{i+1} is:*

$$T = (1 - q_{i-1,i}q_{i-1,i'})^{-1}[ZD^{-1} + p_{i-1,i}q_{i-1,i'}T_{i'} + p_{i-1,i}p_{i-1,i'}T_{i,i'} + q_{i-1,i}p_{i-1,i'}T_i]$$

where $T_{i'} = \frac{Z}{D} \cdot \frac{1+(q_{i',i+1}p_{i,i'})(1-q_{i,i+1}q_{i',i+1})^{-1}}{1-q_{i',i+1}q_{i,i'}}, T_{i,i'} = \frac{Z}{D(1-q_{i,i+1}q_{i',i+1})}, T_i = \frac{Z}{D} \cdot \frac{1+(q_{i,i+1}p_{i',i})(1-q_{i,i+1}q_{i',i+1})^{-1}}{1-q_{i,i+1}q_{i',i}}$.
The best relay should have minimal delay T^* *among all secondary relay candidates.*

In the actual implementation of DAC, a relay R'_i is included in the candidate set of secondary relays only if it has a non-zero reception probability with R_{i-1}, R_i and R_{i+1}. Further, based on the above proposition, we can obtain a closed-form expression for the cooperation gain using DAC relaying in terms of throughput improvement: $g^* = \frac{D}{Z} \cdot (p_{i-1,i}^{-1} + p_{i,i+1}^{-1}) \cdot T^{*-1}$. We adopt a secondary relay only if the potential gain g^* is larger than a threshold T_D (set to 1.1 in our design).

To reduce the signaling overhead, we again used the mean link loss rate as a metric for selecting a fixed secondary relay, instead of adjusting the selection for each packet. As shown in existing measurement and routing design [20, 33], the mean link loss rate is relatively stable on an hourly basis, and it can be obtained from the delivery probability of data packets.

The above scheme based on secondary relay selection can be used to improve other routing protocols. For example, we can improve a traditional orthogonal relaying based routing protocol [74] by adding a secondary relay for the existing primary relay. Similar idea can be applied to assist opportunistic routing [21], in which two forwarders who overheard the same packet can be scheduled concurrently, following similar negotiation mechanism in ExOR [21]. The pros and cons of using a DAC based secondary relay will be further clarified in our analysis.

2.7 Asymptotic Performance Analysis

In this section, we analyze the performance CSMA/CR in terms of achivable SNR, BER (bit error rate) and PER (packet error rate), and the network level performance of Chorus and DAC, in terms of throughput and delay.

Unless noted otherwise, we use the following set of notations: L for the packet length, F the offset between two collided packets, D the data rate, W the signal

bandwidth, N the noise power, and δ^2 the noise variance. Multiple collided packets are indexed according to their arrival time, and γ_i denotes the SNR of packet i. We assume all links adopt the 1Mbps basic access mode using BPSK [67] (assuming $D = 1$Mbps, $W = 1$MHz).

2.7.1 Achievable SNR

We begin with an elementary scenario where two versions of a packet (denoted as P1 and P2) from different transmitters collide. This scenario is analogous to the two-user uplink channel in information theory [125], which adopts interference cancellation as the optimal decoder. However, CSMA/CR' application scenario is unique in that P1 and P2 carry the same data. Ideally, they should complement, or at least do not interfere with each other. This intuition is formalized in the following set of theorems.

Theorem II.2. *The achievable SNR of CSMA/CR in the two-packet collision case is* $\Lambda = \max\{\frac{P_1}{N}, \frac{P_2}{N}\}$. *When decoding* m *overlapped packets, the achievable SNR of collision resolution is* $\Lambda = \max\{\frac{P_1}{N}, \frac{P_i}{\sum_{j\neq i} P_j + N}, \frac{P_m}{N}\}$, $i \in \{2, \ldots, m-1\}$.

The above SNR bounds can be transformed to the BER bound that is directly related to the decoding performance [117]: $\text{BER} = Q(\sqrt{2\Lambda W D^{-1}}) = Q(\sqrt{2\Lambda})$, where the *Q-function* $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty e^{-\frac{x^2}{2}} \mathrm{d}x$. $Q(y) \to 0$ exponentially when $y < 1$ and $y \to -\infty$, which also holds for $y > 1$ and $y \to \infty$. This implies that BER decreases exponentially with the achievable SNR.

2.7.2 BER and PER in Collision Resolution

The above SNR and BER bounds are simplified in that they ignore the error propagation along sequentially-decoded symbols. The iterative collision resolution in CSMA/CR can cause error propagation, due to the correlation between consecutively decoded symbols. For example, in Fig. 2.3, if symbol A produces an erroneous bit,

then the error propagates to A', which affects subsequent symbols such as C. Fortunately, such error propagation stops if the actual bits of A' and C are the same. In this case, after subtracting the error image of A', we obtain a strengthened symbol indicating the correct bit of C. Error propagation also stops when symbol C has a much higher strength than A'. Based on these two intuitions, we prove:

Lemma II.3. *The error propagation probability in forward-direction decoding can be characterized as:*

$$P_s \approx Q(\sqrt{2\gamma_1} - 2\sqrt{2\gamma_2})$$

where γ_i denotes the SNR of packet i. The Q-function *is defined as:$Q(y) = (2\pi)^{-\frac{1}{2}} \int_y^{\infty} e^{-\frac{x^2}{2}} \mathrm{d}x$. A symmetric equation holds for backward direction decoding.*

Fig. 2.11 plots the probability that error propagation stops ($P_{bc} = 1 - P_s$) as a function of SNR. It can be seen that $0.5 \leq P_{bc} \leq 1$, and P_{bc} transits fast from 0.5 to 1 when $\gamma_1 \ll \gamma_2$. This means that the error stops propagation with probability larger tahn 0.5 in the common cases.

Based on Lemma II.3, we further prove that the probability that an error propagates along i bits decays exponentially as i increases, as reflected in the following result.

Lemma II.4. *Denote the packet length as L and packet offset as F, then the steady state error length probability can be characterized as:*

$$\pi_i = \pi_0 P_e P_s^{i-1}, \forall i \in (1, G],\ \pi_0 = (1 + P_e \cdot \frac{1 - P_s^G}{1 - P_s})^{-1}$$

where $P_e = Q(\sqrt{2\gamma W D^{-1}})$ is the BER of a non-collided packet with SNR γ, data rate D and signal bandwidth W. $G = \lfloor \frac{L}{F} \rfloor$.

With the above lemmas, we can bound the BER in DAC's iterative collision-resolution algorithm.

Theorem II.5. *Let P_e' be the BER in forward-direction decoding in DAC, and P_e be the BER of a single head packet without collision, then $P_e \leq P_e' < 2P_e$.*

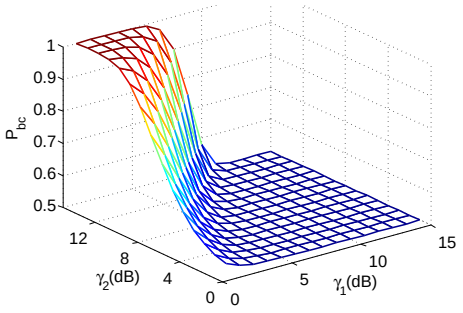

Figure 2.11: Head packet's P_{bc}: the probability that error stops propagating to the next bit.

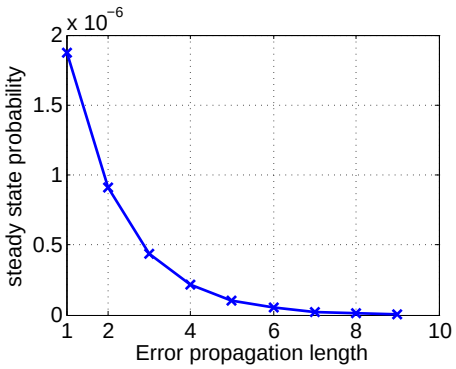

Figure 2.12: Steady state distribution of error length. $\gamma_1 = 10, \gamma_2 = 7$. $F = \frac{L}{64}$. Error length 0 is not shown.

Combining the bounds for P_s and P_e with Lemma II.4, we conclude that *while resolving a given collision, the error propagation probability decays exponentially with the error length* (also shown in Fig. 2.12). This is consistent with the empirical observation in [51]. The above reasoning can be straightforwardly extended to multi-packet collision resolution, where the probability that error stops propagating is also close to or larger than 0.5, because previous erroneous bit may strengthen the current bit with probability 0.5.

A more relevant metric is the packet error rate (PER), which will be used to characterize the gain of DAC over CSMA/CA based non-cooperative schemes. With respect to PER, we have:

Theorem II.6. *Let P_h and P_t denote the PER when the head and tail packets are decoded without collision, respectively, then the overall PER in bi-directional collision resolution is $P_v = P_h P_t$.*

Theorem II.6 implies that *by allowing two relays to transmit concurrently, PER can be reduced to the PER product of the two independent packets.* It seems counter-intuitive that error propagation does not affect the PER. The reasons for this are twofold. First, since the channel estimation for the tail packet is based on preamble correlation, the estimation error is negligible compared to the bit errors in the head packet caused by channel distortion. Second, we do not use any error correction code,

which is beneficial for single-packet decoding. A joint design of error correction and collision resolution may also guarantee better performance for DAC, and this is left as our future work.

2.7.3 Asymptotic Performance of Chorus Broadcast

We now analyze Chorus' network-level performance, including latency and throughput. Similar to existing asymptotic analysis [48, 32, 64], we assume perfect reception within the transmission range if no collision occurs. The network radius is r, *i.e.*, it spans r hops from the source to the receiver farthest away. Let h denote the size of Chorus preamble plus Chorus header, then we have the following asymptotic performance bound regarding broadcast latency and throughput.

Theorem II.7. *The worst-case latency and throughput of Chorus is* $\frac{r(L+h)}{D}$ *and* $\frac{LD}{3(L+h)}$, *respectively.*

From Theorem II.7, we see that the asymptotic latency of Chorus satisfies $\frac{rL}{D} \leq \Theta(r) \leq \frac{r(L+h)}{D}$. Under a unit disk graph model, Chorus' latency can be close to the trivial lower bound $\frac{rL}{D}$, since $h \ll L$. This is in sharp contrast with the $\Omega(r \log n)$ latency for anonymous broadcast using CSMA/CA [32].

Theorem II.7 also reveals that the maximum supportable source rate (or maximum throughput) of Chorus is insensitive to the network size. Since this is a worst-case bound, it can be used to control the source rate in continuous broadcast, in order to prevent the collision of different packets and avoid congestion.

2.7.4 Asymptotic Performance of DAC

Although DAC improves link reliability via concurrent cooperative relays, it comes at the cost of reducing the multiple access opportunity of competing network flows. This essentially reflects the tradeoff between *diversity gain* and *multiplexing gain* at a

network scale, and poses a question: does DAC increase or decrease the total network throughput when multiple flows co-exist?

For multihop networks with cooperative relays, the general capacity-scaling law is still an open problem, and existing work has characterized it for special topologies with a single flow [118]. The focus of our analysis here is on characterizing the condition when DAC can outperform non-cooperative routing protocols without calculating the exact capacity bound. We start from a simplified grid topology. Denote Φ_c and Φ_d as the achievable network throughput of a CSMA-based routing protocol, and the corresponding DAC-enhanced routing protocol, then:

Theorem II.8. *In a grid network with homogeneous link-reception probability p, $\Phi_d > \Phi_c$ when $p < 0.86$. The throughput gain $\frac{\Phi_d}{\Phi_c}$ decreases monotonically with p.*

Theorem II.8 can be extended to a more general case as follows:

Corollary II.9. *In an arbitrary network topology with homogeneous link-reception probability p, a sufficient condition for $\Phi_d > \Phi_c$ is $p < 0.64$.*

These analytical results imply that DAC is guaranteed to improve throughput only when the average link quality is sufficiently low. Remarkably, real-world mesh networks tend to have a majority of links with intermediate quality [33] because of channel attenuation, and because optimal rate adaptation schemes may prefer high data-rate links with low quality, than low data-rate links with full reception rate [20].

In a single 802.11 based wireless LAN, at any time, at most one transmitter can be active. Hence, the DAC relaying scheme achieves diversity gain without reducing the channel access opportunity of any transmitter, and it has higher throughput than CSMA, as long as the links have non-zero loss rates.

2.8 Experimental Evaluation

In this section, we present experimental justification for the feasibility and performance of the collision resolution based protocols. We have built a small software

radio network to validate the collision-resolution PHY. Based on insights from our experimental and analytical results, we implement the the CSMA/CR MAC layer, the Chorus broadcast protocol, and the DAC routing protocols in the ns-2 simulator, and evaluate its effectiveness in a large network.

2.8.1 The Collision-Resolution PHY

We design and prototype the CSMA/CR PHY based on the GNURadio/USRP platform [2]. USRP is a software radio transceiver that converts digital symbols into analog waves centered around a carrier frequency within the ISM band. It can also receive analog signals via its RF front-end, and down convert them into the baseband. The baseband digitized raw signals are sent to a general-purpose computer running the Python/C++ based collision resolution PHY modules built atop the GNURadio library.

The USRP does not yet support MAC operations requiring instantaneous response (*e.g.*, ACK, carrier sensing and cut-through relaying), because of the inefficient user-mode signal processing modules and its high communication latency with the computer. Therefore, we focus on the core components of the CSMA/CR PHY layer, *i.e.*, the collision-resolution modules. Our testbed environment consists of three USRP nodes, which is used to mimic the typical relay network in Fig. 2.2(b). The center frequency of USRPs is set to 2.4145GHz, located in between the 802.11 channel 1 and 2. The USRP transmitter's sampling rate is 128 MSamples/s and interpolation rate 32. With BPSK, each digital bit is mapped to one symbol, and each symbol consists of 8 samples after the RRC. Hence, the effective data rate is $\frac{128}{32\times 8} = 0.5$Mbps. Each packet has a 256B payload, which takes the same channel time as a 1KB packet in an actual 2Mbps-mode 802.11b network.

We use two USRPs as the source and relay and allow them to send packets with the same payload. In the common case of DAC relaying, the links of these two concurrent

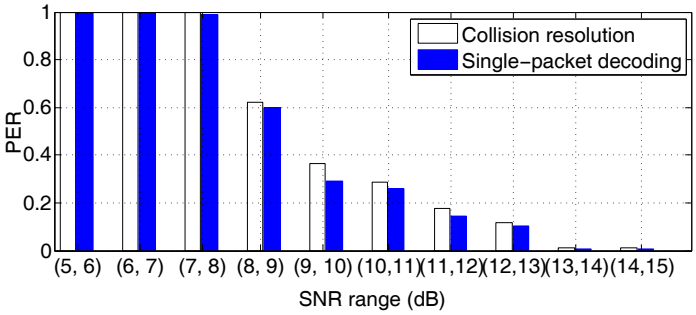

Figure 2.13: Comparison between collision resolution and single-packet decoding without collision.

transmitters have comparable strength. Otherwise, the PER reduction is negligible according to Theorem II.6 and we can just select a single best relay. In addition, the link with much higher SNR may capture the other, and collision resolution is no longer needed. Therefore, we make coarse adjustment on the SNR between each relay and the shared receiver by varying the transmit power and link distance, so the difference in mean SNR falls below 1dB. [2]

We evaluate the PER when using the CSMA/CR PHY to resolve two overlapping packets, and compare it with the decoding probability of a single non-collided packet. Due to channel variations, the SNR value cannot be *precisely* controlled. We thus log the decoded packets, group them according to the received SNR, and calculate the mean packet error rate (PER) for packets falling in the same SNR range (in 1dB unit). The resulting SNR-PER relation is plotted in Fig. 2.13, where each vertical bar represents 10^4 packets collected over four different time periods. We observed a transition of PER from 1 to 0 when SNR becomes larger than 8dB. CSMA/CR achieves similar PER to the single-packet decoding, which verifies our claim that *single-direction collision resolution does not increase PER, compared to single-packet decoding, and thus bi-directional collision resolution achieves the PER product of the head and tail packet* (Sec. 2.7.2). Notably, our analysis is developed based on a Gaussian channel model, but the result is consistent with the testbed experiments

[2]For SNR calculation, we note that the signal power is the square of the mean magnitude of non-collided known symbols. Noise power equals the statistical variance of these symbols [117].

which are carried out in an office environment with rich multipath fading. This is because the RRC filters partly cancel out the inter-symbol interference, rendering the noise approximately Gaussian.

2.8.2 Performance of Chorus Broadcast

We now evaluate the broadcast performance of Chorus. We implement the cognitive sensing and broadcast scheduling protocols based on the 802.11b module in ns-2. We adopt the collision-resolution module as the PHY-layer packet reception model. This module computes the SNR for a given collision pattern, following the analysis in Sec. 4.4. The resultant SNR is then compared with the SNR threshold to determine whether the reception succeeds. We do not consider error propagation since it has negligible effect on PER, as shown in our previous analysis and simulation. We only use the selective combination when multi-packet collision occurs.

We use a typical CSMA/CA-based protocol, *Double-Coverage Broadcast* (DCB) [86] as a performance benchmark. In order to reduce the latency caused by redundant transmissions, DCB prunes the network topology, such that only those nodes with the potential to deliver packets to many downstream receivers will be selected. It further improves PDR by ensuring that each receiver is covered at least twice by other selected forwarders. DCB has been compared with a number of other CSMA/CA-based broadcast protocols and demonstrated superior performance.

We have implemented DCB based on the ns-2 802.11b MAC, following the specification of Algorithm 5 in [86]. Since it requires a strict definition of neighborhood, DCB assumes a transmission range exists, within which all nodes receive packets from the transmitter with the same probability. To improve accuracy while satisfying this requirement, we use the following channel model. We define transmission range at a distance where reception succeeds with an *edge reception probability* ϵ. Within this range, the RSS follows the log-normal distribution [28], with mean 4 and std

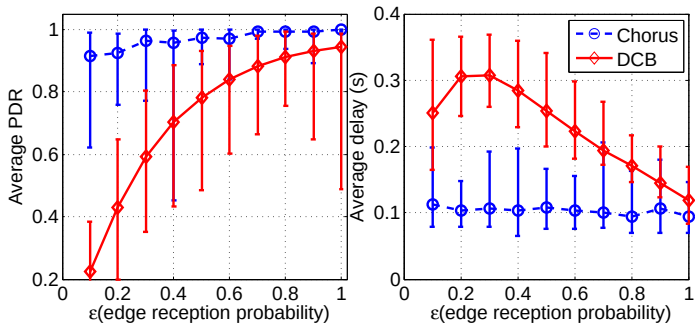

Figure 2.14: The impact of link quality (reflected by ϵ) on latency and PDR. The error bars indicate variation over 30 random topologies.

5 (dB). This channel model represents a middle ground between the UDG and the log-normal shadowing model. When ϵ is close to 1, it approaches the UDG model. As ϵ approaches 0, it is equivalent to a shadowing model. For a given topology, as ϵ decreases, the average link quality decreases. Similar to DCB, we assume a SNR threshold exists, above which packets cannot be received. Given the edge reception probability ϵ and noise power, the SNR threshold is calculated by inverting the log-normal function [28].

All experiments are repeated on 30 randomly-generated topologies with node degree ranging from 2 to 9. We measure PDR according to the fraction of nodes that successfully receive a packet, and latency the duration between its release and the last successful reception. Both the PDR and latency are averaged over 1000 packets for each topology, and evaluated with respect to: link quality (indicated by ϵ), network size, source rate and packet size. The typical settings are: source rate 1 pkt/s (packets/second), packet size 1KB, edge reception probability $\epsilon = 0.5$, network size (number of nodes) 100 with average node density 6. Unless noted otherwise, we isolate the effect of each factor by varying it while fixing others to the typical values.

Our experimental results on DCB are consistent with [86] at a high link quality, low source rate, small packet size and small network size. However, in the general case, DCB's performance degrades fast. In contrast, Chorus demonstrates significant advantages in all cases. We report the detailed experiments below.

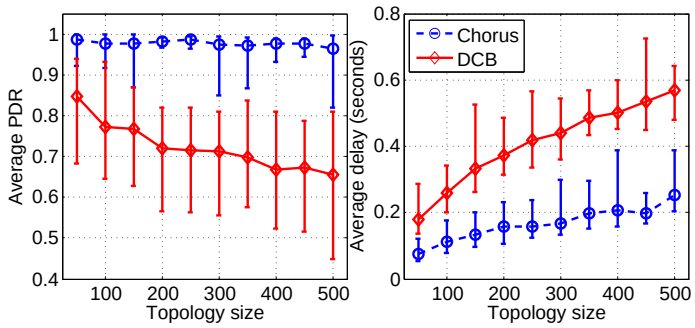

Figure 2.15: Scalability of the broadcast protocols as the topology size (number of nodes) grows.

2.8.2.1 Link quality

We vary the link quality by tuning the edge reception probability ϵ. A higher ϵ value implies a lower packet loss rate for average links in the network. As shown in Fig. 2.14, the PDR of both Chorus and DCB decreases with loss rate. However, Chorus is much less sensitive to the link condition, owing to the diversity provided by collision resolution. As ϵ changes, Chorus' latency remains around 0.1 second, while DCB's latency varies from 0.12 to 0.3. More importantly, Chorus keeps more than 90% PDR under all link conditions, while DCB's average PDR drops from 90% to 20% as ϵ decreases. Note that DCB's latency may drop as the link quality decreases. This is at the expense of severe packet losses as indicated by the decrease of PDR.

2.8.2.2 Network size

Sensitivity to network size indicates the scalability of the broadcast protocol. To quantify scalability of Chorus, we keep the average network density to 6 while increasing the total number of nodes in the network. The network radius grows accordingly. Fig. 2.15 plots the resulting latency and PDR. Chorus demonstrates negligible loss of PDR as the networks size grows. In addition, its latency is 75% lower than that of DCB. Consistent with the asymptotic analysis, its latency increases with the network size. However, the growth rate or sensitivity to network size is much lower than DCB.

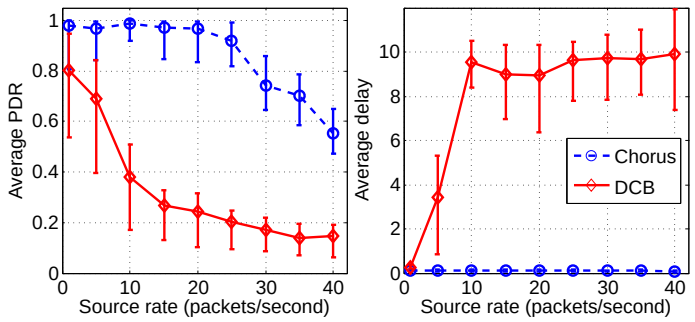

Figure 2.16: Sensitivity to source rate, which indicates the maximum supportable throughput of a broadcast protocol.

2.8.2.3 Source rate

It is well-known that in end-to-end unicast or broadcast, the throughput drops when the source rate is too high and the network becomes congested. Therefore, the maximum supportable source rate reflects the maximum throughput of a broadcast protocol. In Fig. 2.16, we vary the rate at which the source node generates broadcast packets, and track the resulting latency and PDR. Both Chorus and DCB's PDR decreases abruptly beyond certain margins, which roughly indicate their supportable throughput. We observe that the supportable throughput of Chorus is around 20 pkts/second, in contrast to 1 pkt/second in DCB. In addition, DCB' latency increases from 0.1 second to 10 seconds as the source rate increases from 1 to 40 pkts/second, while Chorus maintains around 0.1 second latency across this range.

2.8.2.4 Packet size

Fig. 2.17 shows how packet size affects the broadcast performance when coupled with variation of source rate. When source rate is low (1 pkt/s), the network is less congested, thus Chorus' spatial reuse advantage is less obvious. Owing to the diversity gain, however, it maintains a PDR higher than 95%, in contrast with 80% when running DCB. In addition, its latency is 60% lower than DCB for all packet sizes. When source rate is high (10 pkt/s), Chorus's PDR and latency remains the same. In contrast, DCB suffers from a sharp degradation of performance – its latency

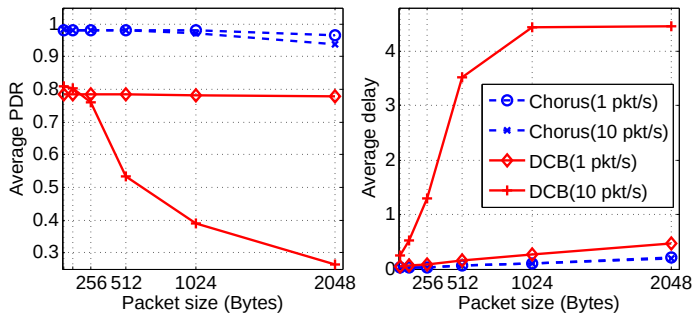

Figure 2.17: Impact of packet sizes, which range from 64 to 2048 bytes.

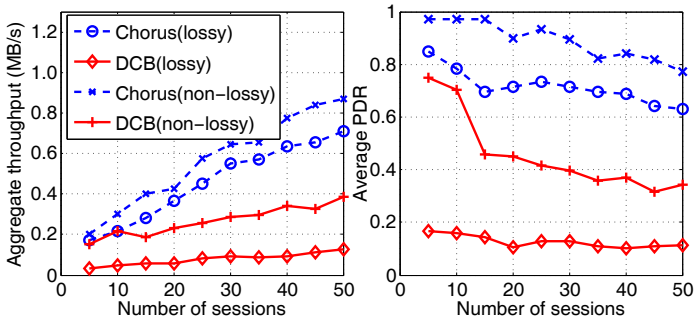

Figure 2.18: Total broadcast throughput and average PDR when multiple sources transmit different data, for lossy (edge reception probability $\epsilon = 0.1$, average link quality $q = 0.51$) and non-lossy ($\epsilon = 0.5, q = 0.83$) networks.

increases from 0.2 to 4 seconds as packet size grows from 64B to 1024B. Again, this is due to its limited supportable throughput. For larger packets, the source injects more data into the network per unit time, which causes congestion. In addition, the cost of losing one packet increases, resulting in higher latency and lower PDR.

As indicated in Sec. 4.4, the worst-case delay of Chorus is affected by its packet overhead. The experiment results in Fig. 2.17 show that Chorus is relatively insensitive to packet overhead, in contrast to the analysis. This is because the worst case in Fig. 2.25 rarely occurs in a random network, and the overhead is negligible compared with packet length.

2.8.2.5 Multiple broadcast sessions

We proceed to evaluate the case where multiple broadcast sessions co-exist, each corresponding to one randomly selected source node in a 50-node topology. We set $\epsilon =$

0.1 and $\epsilon = 0.5$ to represent a lossy and non-lossy network, respectively. The former case is close to a real world mesh network [20] in which most links have intermediate reception rate. We focus on two metrics: average PDR among all sessions, and broadcast throughput, which equals the total amount of data delivered to all nodes within unit time, summed over all the sessions. Fig. 2.18 plots these metrics as a function of traffic load (the number of sessions). In a lossy network, Chorus achieves 3x higher throughput than DCB, and maintains a PDR above 60%, which indicates the friendliness among different traffic. The performance gain over DCB is less in a non-lossy network, where the main benefit of Chorus comes from spatial reuse, rather than diversity gain. Also note that although throughput increases when the traffic load is high, the cost is lower PDR, implying that most traffic is confined to around the source nodes, especially for the DCB protocol.

2.8.3 Performance of DAC-Enhanced Routing

2.8.3.1 Experimental setup

In the asymptotic analysis, for tractability, we make simplifications including fixed transmission range and homogeneous loss probability. To evaluate more realistic scenarios, we implement the DAC-enhanced routing protocol (Sec. 2.6) in the ns-2 simulator. The primary path discovery is the same as the ETX routing (which is built atop existing ad-hoc routing protocols) [33]. The secondary relay-selection algorithm runs on each primary relay, which measures the quality of adjacent links, and exchanges link-quality information for those links connecting secondary relay candidates and their previous and next hops. The underlying CSMA/CR protocol is implemented by modifying the 802.11b MAC in ns-2. We add the DAC header and preamble to each packet, modify the carrier sensing and transmission timeout, so as to support the direct cut-through relaying, as discussed in Sec. 3.4 and Sec. 2.6.

The simulation runs in a mesh topology with 50 randomly-deployed nodes in a

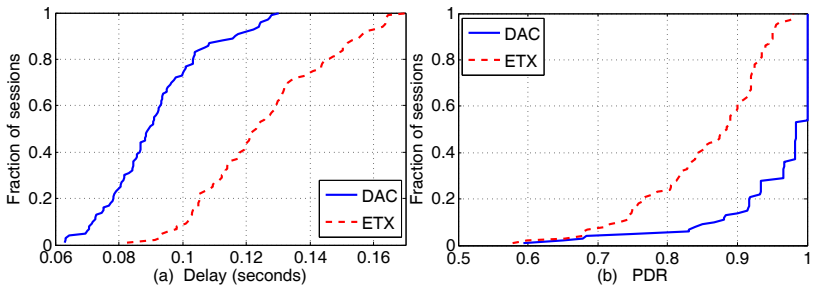

Figure 2.19: Distribution of delay and packet-delivery ratio (PDR) for single-unicast sessions.

1km×1km region. We use the log-normal shadowing model with pass-loss exponent 4.0 and shadowing deviation 5.0dB. We replace the ns-2 PHY packet reception model with the analytical model for DAC PHY in Theorem II.6, which has been verified in our experiments. The transmit power and reception threshold is configured such that the reception probability is 0.1 at 250m. Overall, this topology has an average link-quality 0.51 and median 0.47, consistent with the measurement from Roofnet [20] which indicates that most links have an intermediate quality.

2.8.3.2 Single-unicast scenario

We evaluate the performance of DAC in comparison with the original ETX routing for two scenarios: single-unicast and multiple-unicast. In the first case, a pair of source-destination nodes are randomly selected to start an end-to-end data session. Since no other competing flows co-exist, we are interested in the average end-to-end packet delay and reliability (indicated by packet-delivery ratio, PDR) for each session. This set of experiments essentially reveal the performance gain of DAC in an unsaturated network. We evaluate these two metrics over 100 sessions, with packet size 1KB and source rate 0.2Mbps.

The CDF plot in Fig. 2.19(a) reveals that DAC reduces end-to-end delay for most sessions. The average delay reduction is 27.3%. This improvement comes with much higher PDR, as shown in Fig. 2.19(b). Since DAC boosts the reception rate of low-quality links with concurrent transmissions from secondary relays, the PDR for a

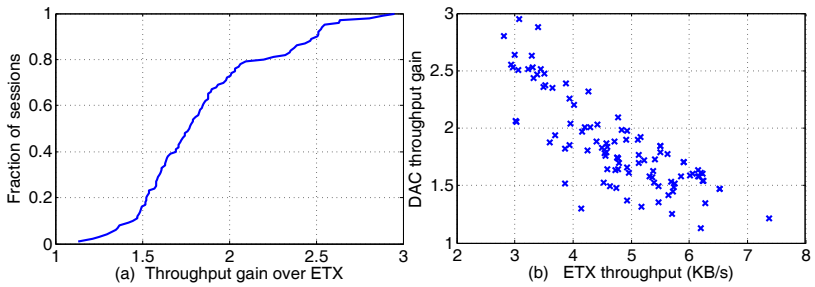

Figure 2.20: Throughput gain of DAC over ETX. (a) the CDF plot; (b) the scatter plot, each point corresponding to one session.

majority of sessions is increased to more than 90%.

We further evaluate the saturated throughput of DAC. We increase the source rate such that the source node's transmit queue remains backlogged. We use *throughput gain* as the metric, defined as the end-to-end throughput of DAC divided by that of ETX. The throughput gain distribution for 100 random sessions is shown in Fig. 2.20(a). It can be seen that DAC can achieve a 3x throughput improvement over ETX, with an average throughput gain 1.73. In a saturated network, throughput depends on the bottleneck link, *i.e.*, the link with the lowest quality along the selected path. Hence, DAC is most effective for paths with low-quality links. This can be seen from the scatter plot in Fig. 2.20(b). Obviously, DAC achieves higher throughput gain for those sessions where ETX has below-average throughput. These sessions tend to have links with high loss rate along their paths.

2.8.3.3 Multiple unicast sessions

We proceed to examine DAC's performance when multiple competing flows coexist, where the fundamental tradeoff between diversity and multiplexing gain becomes an important factor in determining the total network throughput, as discussed in Sec. 4.4. We evaluate the network throughput as a function of the traffic load. Specifically, we fix the source rate at 10Kbps and increase the total number of sessions. As illustrated in Fig. 2.21(a), the total network throughput increases with the number of sessions when traffic load is low. In such cases, DAC can have 2x im-

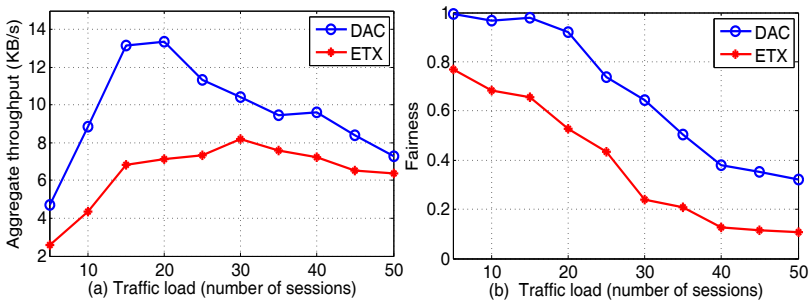

Figure 2.21: Total network throughput and fairness vs. traffic load.

provement over ETX routing. As the network becomes congested, the non-orthogonal cooperation may sacrifice the channel access time of other concurrent sessions, and therefore, the advantage is less obvious.

While DAC's higher throughput comes from the diversity gain, we need to ensure this advantage does not reduce the fairness among sessions. To evaluate fairness, we use the Jain's fairness index [69] as a metric. A fairness level of 1 indicates all sessions have the same throughput, whereas a close-to-zero fairness indicates some sessions achieve higher throughput by starving others. It can be seen from Fig. 2.21(b) that DAC always maintains a higher level of fairness. This is because it only rescues the bottleneck links on low-throughput paths (which is reflected in the threshold T_D in designing DAC routing). Overall, both the throughput and fairness are improved by exchanging the multiplexing opportunity of high-throughput sessions for the diversity gain in low-throughput sessions[3]

To make this intuition more concrete, we generate a mesh topology with a majority of high-quality links (the average link quality is 0.826). Fig. 2.22 shows the resulting network throughput and fairness. Although DAC still maintains a higher level of fairness, much less throughput gain is achieved. This is because ETX tends to select high-quality links whenever available, which are abundant in such a topology. For DAC, the opportunity of exploiting the diversity gain is scarce. Combined with the previous experimental results, this signifies the generality of the analysis in

[3]Note that the traffic load higher than 40 sessions is less relevant since the fairness level is low, and most sessions are starved.

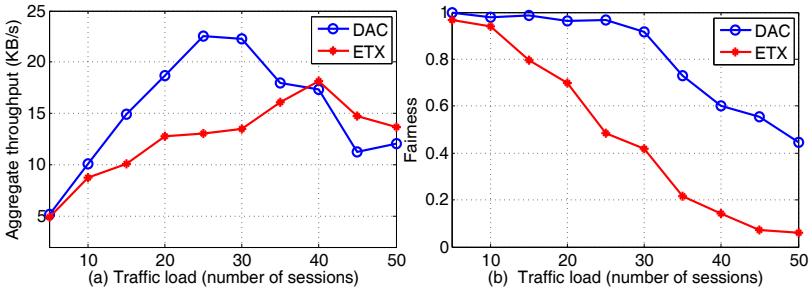

Figure 2.22: Total network throughput vs. traffic load in a network with a high reception rate.

Theorem II.8 , *i.e.*, as a non-orthogonal relaying scheme, DAC guarantees throughput gain for networks with intermediate link quality, such as the unplanned mesh network Roofnet [20].

2.9 Conclusion and Future Work

In this project, we provide theoretical and experimental results that demonstrate the feasibility and advantage of CSMA/CR, a collision-resolution based MAC/PHY scheme. The key idea behind CSMA/CR is that two partially-overlapping packets carrying the same information from different relays can be decoded independently by using an iterative collision-resolution algorithm at the PHY layer. By decoding multiple versions of a packet at once, CSMA/CR achieves transmit diversity and improves loss resilience without any retransmission. More importantly, with a collision-tolerant MAC, it significantly simplifies the CSMA scheduling and improves its spatial reuse.

Based on CSMA/CR, we design a simple collision tolerant broadcast protocol called Chorus. We prove that Chorus has an asymptotic latency bound of $\Theta(r)$ when using Chorus for broadcast, where r is the network radius. Our network-level experiments further show that Chorus outperforms a typical CSMA/CA-based broadcast protocol by a significant margin, in terms of latency, reliability, throughput, and scalability. These features make Chorus suitable especially for fast information dissemination in large-scale networks, such as wireless mesh networks.

We further design a cooperative relay protocol, DAC, which adopts a generic approach that incorporates CSMA/CR into existing routing protocols. Using network-level simulation in ns-2, we show that DAC can improve the network performance in terms of throughput, delay and fairness, especially for lossy wireless mesh networks. As non-orthogonal relaying has fundamental advantage over traditional orthogonal relays [17], DAC marks an effective step towards exploiting the potential of non-orthogonal cooperative communications.

2.10 Appendix

2.10.1 Proof for Proposition II.1

Proof. We model the propagation of a data packet as a Markov chain, as shown in Fig. 2.23. Each state denotes the current holder of the packet. State i represents the fact that R_i has received the packet but R_i' has not. State ii' denotes the fact that both R_i and R_i' have received the packet and the cut-through relaying starts. The expected transmission delay is essentially the first passage time from R_{i-1} to R_{i+1}, denoted as T_{i-1}. Similarly, the expected first passage time from state i, i', ii' to $i+1$ are denoted as $T_i, T_{i'}$ and $T_{ii'}$, respectively. The outcome of the first transmission attempt by R_{i-1} can be classified into three cases:

First, only the direct link $R_{i-1} \to R_i$ succeeds, which happens with probability $p_{i-1,i}(1-p_{i-1,i'})$, and it takes $\frac{Z}{D}$ time to finish this transmission.

Second, only $R_{i-1} \to R_i'$ succeeds, which happens with probability $(1-p_{i-1,i})p_{i-1,i'}$, and takes time $\frac{Z}{D}$.

Third, neither $R_{i-1} \to R_i$ nor $R_{i-1} \to R_i'$ succeeds, which happens with probability $1-(1-p_{i-1,i})(1-p_{i-1,i'})$, and wastes $\frac{Z}{D}$ time. Afterwards, the transmission starts again from R_{i-1}, and again taking T_{i-1} time in expectation.

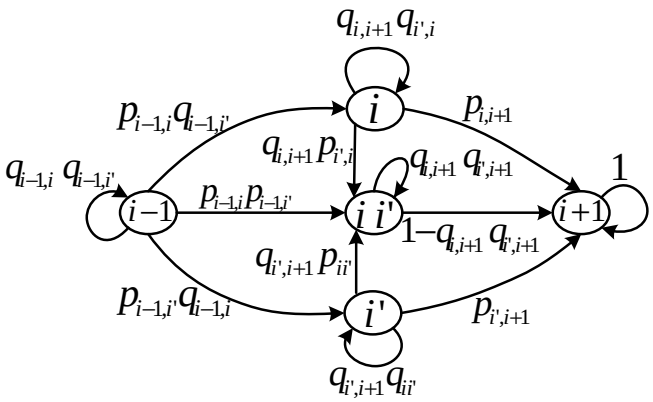

Figure 2.23: Modeling the packet propagation in the DAC primary-secondary relay-selection algorithm as a Markov chain.

Overall, the expected time for a packet to reach R_{i+1} from R_{i-1} is:

$$T_{i-1} = \frac{Z}{D} + p_{i-1,i}q_{i-1,i'}T_i + q_{i-1,i}p_{i-1,i'}T_{i'} + p_{i-1,i}p_{i-1,i'}T_{ii'} + q_{i-1,i}q_{i-1,i'}T_{i-1}.$$

When the packet is in state i, it may proceed with three possible outcomes. First, R_i succeeds in delivering it to R_{i+1} directly, which happens with probability $p_{i,i+1}$.

Second, the direct delivery fails, but $R_{i'}$ overhears the packet, and consequently the system evolves to state ii'. This happens with probability $q_{i,i+1}$.

If neither happens, then the system remains in state i and repeats the above trials.

Therefore, the expected transmission time from R_i to R_{i+1} is:

$$T_i = \frac{Z}{D}p_{i,i+1} + q_{i,i+1}q_{i'i}T_i + q_{i,i+1}p_{i'i}T_{ii'}. \tag{2.1}$$

Similarly, for state i', we have:

$$T_{i'} = \frac{Z}{D}p_{i,i+1} + q_{i',i+1}q_{ii'}T_i + q_{i',i+1}p_{ii'}T_{ii'}. \tag{2.2}$$

For state ii', the expected transmission time is the expectation of a geometric

random variable with mean:

$$T_{ii'} = \frac{1}{1 - q_{i,i+1}q_{i',i+1}} \tag{2.3}$$

and the joint PER when both $R_i \to R_{i+1}$ and $R'_i \to R_{i+1}$ transmit concurrently is based on Theorem II.6.

By solving the above equations, we can obtain a closed-form expression for T_{i-1}, thus completing the proof for Proposition II.1. □

2.10.2 Proof for Theorem II.2

Proof. The proof follows from CSMA/CR' iterative decoding. We represent symbols in the complex form. Suppose at time t, symbol $\tilde{s}_1(t) = a_1 e^{j\theta_1} x_1(t)$ in P1 collides with $\tilde{s}_2(t) = a_2 e^{j\theta_2} x_2(t)$ in P2. Let v denote the receiver noise, then the received symbol $\tilde{s}(t) = \tilde{s}_1(t) + \tilde{s}_2(t) + v$. If we decode P1 first (forward-direction decoding), then $x_2(t) = x_1(t-F)$. In addition, the channel amplitude a_2 and phase θ_2 can be estimated via correlation, which can achieve high accuracy and introduces negligible noise [51]. Therefore, we can obtain a decision symbol for $x_1(t)$ as: $\tilde{s}(t) - \tilde{s}_2(t) = a_1 e^{j\theta_1} x_1(t) + v$. The resulting SNR level is: $\frac{|a_1 e^{j\theta_1}|^2}{2\delta^2} = \frac{P_1}{N}$, which equals the SNR when $s_1(t)$ is decoded independently.

Similarly, if the clean symbols in P2 are decoded first (backward-direction decoding), then we can obtain $\frac{P_2}{N}$. Taking the maximum of these two yields $\Lambda = \max\{\frac{P_1}{N}, \frac{P_2}{N}\}$.

When m packets collide, the head and tail packets have clean symbols, and the achievable SNRs are $\frac{P_1}{N}$ and $\frac{P_m}{N}$, respectively, following a similar line of reasoning as above. Since Chorus performs hard decoding over intermediate packets, the achievable SNR for an intermediate packet is the same as treating other packets as noise, *i.e.*, $\frac{P_i}{\sum_{j \neq i} P_j + N}, \forall i \in \{2, \ldots, m-1\}$. The result follows directly after taking the maximum

SNR of all packets. □

2.10.3 Proof for Lemma II.3

Proof. BPSK symbols can be represented as real values subject to channel attenuation, since decoding only depends on the in-phase part of the received symbol. Back to the example in Fig. 2.3, suppose symbol C carries bit "0" (mapped to -1 in BPSK), and the channel attenuation over C is X_a, then symbol C is represented as $-X_c$. Suppose symbol A' carries bit "1" (mapped to 1 in BPSK) with channel attenuation $X_{a'}$, then the collided symbol $S = -X_c + X_{a'} + v$, where v is the additive white Gaussion noise. In this case, Chorus should subtract $X_{a'}$ from S. However, if the estimation of symbol A is incorrect, it will propagate to C via A'. Specifically, Chorus erroneously subtracts $-X_{a'}$, resulting in a decision value $Y_c = -X_c + 2X_{a'} + v$. Similarly, when A' carries bit "0" but Chorus estimates it as "1" via A, the resulting decision value is $Y_c' = -X_c - 2X_{a'} + v$. A symmetric argument applies to the case when symbol C carries bit "1". Therefore, the probability that the collision resolution outputs a correct bit is:

$$\begin{aligned} P_{bc} &= \frac{1}{2}P\{Y_c' < 0\} + \frac{1}{2}P\{Y_c < 0\} \\ &= \frac{1}{2}P\{w < 2X_{a'} + X_c\} + \frac{1}{2}P\{w < X_c - 2X_{a'}\} \end{aligned} \tag{2.4}$$

The first term in Eq. (2.4) can be bounded as:

$$\begin{aligned} \Gamma' &= P\{w < 2X_{a'} + X_c\} = 1 - P\{w \geq 2X_{a'} + X_c\} \\ &\geq 1 - \frac{\delta^2}{(2X_{a'} + X_c)^2} \ (\textit{Chebyshev Inequality}) \\ &= 1 - \frac{1}{(2\sqrt{2\gamma_2} + \sqrt{2\gamma_1})^2}. \end{aligned}$$

Both γ_1 and γ_2 are in normal scale, corresponding to practical log scale values ranging from 6dB and above [51]. Therefore, in the above equation, it is reasonable to assume

$\gamma_1 \gg 1, \gamma_2 \gg 1$. Consequently, $P\{w < 2X_{a'} + X_c\} \approx 1$.

For the second term in Eq. (2.4), a closed-form estimation can be obtained as:

$$\Gamma = P\{w < X_c - 2X_{a'}\} = 1 - \frac{1}{\delta\sqrt{2\pi}} \int_{X_c - 2X_{a'}}^{\infty} e^{-\frac{u^2}{2\delta^2}} \mathrm{d}u$$

$$1 - \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2\gamma_1} - 2\sqrt{2\gamma_2}}^{\infty} e^{-\frac{z^2}{2}} \mathrm{d}z \ (note : z = \frac{u}{\delta})$$

$$= 1 - Q(\sqrt{2\gamma_1} - 2\sqrt{2\gamma_2}).$$

Consequently, $P_s = 1 - \Gamma - \Gamma' \approx 1 - \Gamma' = Q(\sqrt{2\gamma_1} - 2\sqrt{2\gamma_2})$. The proof is thus completed. □

2.10.4 Proof for Lemma II.4

Proof. We set up a Markov chain model that relates error propagation to the SNR of each packet, and the offset between collided packets. Again, we start with the two-packet collision scenario in Fig. 2.3 and analyze the iterative decoding of the head packet P1. As shown in Fig. 2.24, we define *states* according to the error propagation length, *i.e.*, the number of consecutive errors in a run. The state transition can be classified into two cases: (i) the probability that an independent decoding error occurs (transition from state 0 to state 1), which equals the BER of clean symbols in P1 (denoted as P_e), and (ii) the probability P_{bc} that error propagation stops, *i.e.*, the next bit is correct even when the current bit is erroneous. The probability of continuing error propagation is $1 - P_{bc}$. The maximum error-propagation length starting from a clean symbol is $G = \lfloor \frac{L}{F} \rfloor$, since the distance between any two consecutively-decoded symbols equals F.

Obviously, this Markov chain is aperiodic and has a single recurrent class, and thus, the steady-state distribution exists. Let π_i be the steady-state probability of

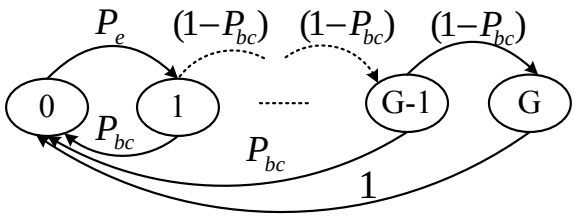

Figure 2.24: The error-propagation process as a Markov chain.

state i, then we have the following balance equations:

$$\begin{cases} \pi_1 = \pi_0 \cdot P_e \\ \pi_i = \pi_{i-1} \cdot (1 - P_{bc}), i = 2, 3, \cdots, G. \\ \sum_{i=0}^{G} \pi_i = 1. \end{cases}$$

Solving this set of equations, we can obtain π_0 and π_i as shown in Lemma II.4, thus completing the proof. □

2.10.5 Proof for Theorem II.5

Proof. In practice, since the two packets are from two different transmitters, the difference between γ_1 and γ_2 is larger than 1, even in dB scale. Given the exponential decaying of the $Q(\cdot)$ function, a practical estimation is $\Gamma \approx 1$ if $\gamma_1 \gg \gamma_2$ and $\Gamma \approx 0$ if $\gamma_1 \ll \gamma_2$.

Combining the analysis of the two terms in Lemma II.3, we have $0.5 \leq P_{bc} \leq 1$, and P_{bc} transits fast from 0.5 to 1 when $\gamma_1 \ll \gamma_2$.

Back to Lemma II.4, we have $\pi_0 \leq \frac{1}{1+P_e} \approx 1 - P_e$. π_0 approximates this upper-bound as $G \to 1$, *i.e.*, the offset between the two packets approaches the packet size. Furthermore, in the common case $G > 1$, we have:

$$\pi_0 \geq \frac{1}{1 + P_e \frac{1}{P_{bc}}} \geq \frac{1}{1 + 2P_e} \geq 1 - 2P_e. \tag{2.5}$$

Therefore, the bit error probability P_e' in iterative decoding is bounded as:

$$P_e \leq P'_e = 1 - \pi_0 \leq 2P_e \tag{2.6}$$

which completes the proof of Theorem II.5. □

2.10.6 Proof for Theorem II.6

Proof. We start with the forward-direction decoding, *i.e.*, decoding the head packet by subtracting the tail packet from it. We assume no error correction code is used, and therefore, the packet is corrupted once the first bit error occurs.

We represent symbols in the complex form. Suppose at time t, symbol $\tilde{s}_1(t) = a_1 e^{j\theta_1} x_1(t)$ in P1 collides with $\tilde{s}_2(t) = a_2 e^{j\theta_2} x_2(t)$ in P2. Let v denote the receiver noise, then the received symbol $\tilde{s}(t) = \tilde{s}_1(t) + \tilde{s}_2(t) + v$. If we decode P1 first (forward-direction decoding), then $x_2(t) = x_1(t - F)$. In addition, the channel amplitude a_2 and phase θ_2 can be estimated via correlation, which can achieve high accuracy and introduces negligible noise [51]. Therefore, we can obtain a decision symbol for $x_1(t)$ as: $\tilde{s}(t) - \tilde{s}_2(t) = a_1 e^{j\theta_1} x_1(t) + v$. The resulting SNR level is: $\gamma_1 = \frac{|a_1 e^{j\theta_1}|^2}{2\delta^2} = \frac{P_1}{N}$, which equals the SNR when $s_1(t)$ is decoded independently. With γ_1, we can get the BER for typical fading pattern and noise profile [117]. Suppose the relation between BER and SNR is $P_e = f(\gamma_1)$, then P_h equals the probability that the first L trials succeeds for a geometrical random variable with mean P_e, *i.e.*, $P_h = 1 - (1 - P_e)^L$, which is equivalent to the probability of a bit error event in the head packet when it is decoded alone.

Similarly, we can decode the tail packet with PER P_t. With a selective combination, the overall PER equals the probability that both forward and backward decoding fail, which is $P_h P_t$. □

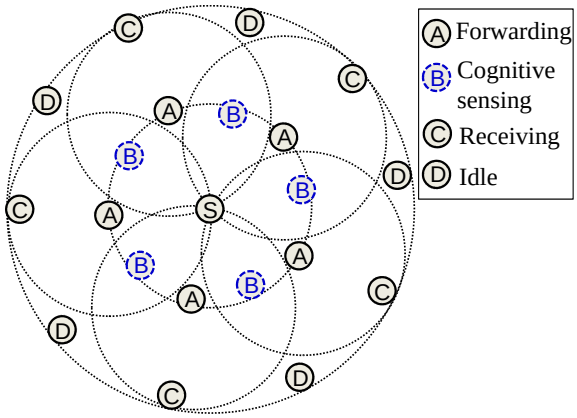

Figure 2.25: The worst-case latency scenario in Chorus broadcast.

2.10.7 Proof for Theorem II.7

Proof. The network can be divided into r rings centered around the source node. A trivial lower-bound on the latency is $r\frac{L}{D}$, *i.e.*, all nodes within the same ring transmit concurrently after the previous ring, and the packet is repeated exactly r times. However, this is only achievable when the cognitive sensing function is disabled. The worst-case scenario happens when cognitive sensing induces the longest delay between adjacent rings, as shown in Fig. 2.25. Specifically, at most a half of the nodes within each ring is transmitting while others within the same ring are transmitting. This induces latency equal to the duration of the Chorus preamble and header, which equals $\frac{h}{D}$. In addition, the latency can be repeated at most r times over the network, resulting in the worst-case latency $r\frac{L+h}{D}$.

In continuous broadcast, packets of different sequences must not collide as the collision cannot be resolved. To prevent such collisions, nodes within two hops cannot send different packets concurrently. Therefore, a new packet can be sent from the source only after the previous packets have propagated at least three hops away, which takes time $3\frac{L+h}{D}$. As a result, the amount of data transmitted within a unit time is: $\frac{L}{3\frac{L+h}{D}}$, which is equivalent to the broadcast throughput of Chorus. □

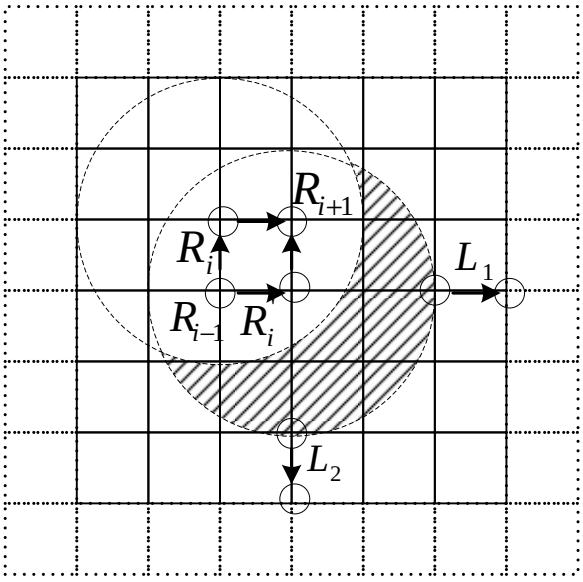

Figure 2.26: Grid topology with homogeneous link-reception probability.

2.10.8 Proof for Theorem II.8

Proof. Consider the grid network shown in Fig. 2.26. Let the edge length be e, and assume the transmission range equals e and interference range equals $2e$. Suppose $R_{i-1} \to R_i, R_i \to R_{i+1}$ are two consecutive links used along the path of a flow, which is selected by a CSMA-based routing protocol. Denote $I(R)$ as the interference region of node i. Following the secondary relay selection rules in Sec. 2.6, it is straightforward that the only secondary relay available is R_i'. Similar to the proof of Proposition II.1, we can derive a balance equation for the expected packet delay from R_{i-1} to R_{i+1}:

$$\begin{aligned} T_{i-1} &= 2pq \cdot \frac{1}{p} + \frac{p^2}{1-q^2} + q^2 T_{i-1} + 1 \\ T_{i-1} &= \frac{1+2q}{1-q^2} + \frac{p^2}{(1-q^2)^2} \end{aligned} \tag{2.7}$$

where $q = 1 - p$ and the packet transmission time is normalized to 1. In comparison, a CSMA-based routing entails $\frac{2}{p}$ average delay from R_{i-1} to R_{i+1}.

On the other hand, when R_i' is used by DAC, it transmits concurrently with R_i, expanding the interference region by $I(R_i) - I(R_i) \cap I(R_i')$ (the shaded region in

Fig. 2.26), compared with a CSMA-based orthogonal scheduling protocol. Within the expanded interference region, at most two transmitters can be scheduled (L_1 and L_2) at the same time without interfering with each other. However, on average, a perfect orthogonal scheduling protocol allocates $\frac{1}{5}$ fraction of time for each node in this grid (because an optimal schedule achieves the minimal coloring of the nodes, and the minimal coloring of a grid has chromatic number 5 [24]). Further, note that the two nodes' transmissions succeed only with probability p, and the secondary relay of DAC is used with probability p^2 and over $\frac{1}{1-q^2}$ transmission attempts towards R_{i+1}. Therefore, on average, the loss of multiplexing time is $\frac{2}{5} \times p \times \frac{p^2}{1-q^2} = \frac{2p^3}{5(1-q^2)}$.

DAC is guaranteed to reduce the network delay if the diversity gain dominates the multiplexing loss, *i.e.*,

$$f(p) = \frac{2}{p} - \frac{1+2q}{1-q^2} - \frac{p^2}{(1-q^2)^2} - \frac{2p^3}{5(1-q^2)} > 0. \tag{2.8}$$

We can numerically solve the equation $f(p) = 0$ and get its solution within $(0, 1)$, which equals 0.83. By taking the first-order derivative of $f(p)$, it can be easily seen that $\frac{\mathrm{d}f(p)}{\mathrm{d}p} > 0, \forall p \in (0, 1)$. Therefore, $f(p)$ is monotonically decreasing within $(0, 1)$. This establishes that the diversity gain of DAC always dominates its multiplexing loss when $p < 0.86$, thus completing the first part of the theorem.

For the case with orthogonal relays, the expected delay from R_{i-1} to R_{i+1} equals $(\frac{1}{1-q^2} + \frac{1}{p})$, since either R_i or R'_i can forward the packet to R_{i+1}. Therefore, the sufficient condition for guaranteeing DAC's gain is:

$$f_{or}(p) = (\frac{1}{1-q^2} + \frac{1}{p}) - \frac{1+2q}{1-q^2} - \frac{p^2}{(1-q^2)^2} - \frac{2p^3}{5(1-q^2)} > 0$$

from which we get an equivalent condition $p < 0.73$, thus completing the proof of Theorem II.8. □

2.10.9 Proof for Corollary II.9

Proof. In an arbitrary topology, DAC selects a secondary relay only if it is connected to the primary relay, the previous hop and the next hop. Therefore, the maximum interference expansion of DAC is $I(R_i) - I(R_i) \cap I(R_i') < A(R)$, where $A(R)$ is the area of a triangle with edge length equal to the interference range R. Further, the maximum independent set that can be packed into $I(R_i')$ is a regular hexagon with edge length R. Since $I(R_i) - A(R) < I(R_i) \cap I(R_i')$, at least two vertices of this hexagon fall in $I(R_i) \cap I(R_i')$. Therefore, the interference region expanded by the secondary relay affects at most 4 other vertices. Among the 4 vertices, at most two can transmit concurrently under a CSMA scheduler. Therefore, the average loss of multiplexing time is $2p \times \frac{p^2}{1-q^2} = \frac{2}{p(1-q^2)}$, and a sufficient condition for DAC to have performance gain is its diversity gain dominates multiplexing loss, *i.e.*,

$$f(p) = \frac{2}{p} - \frac{1+2q}{1-q^2} - \frac{p^2}{(1-q^2)^2} - \frac{2}{p(1-q^2)} > 0 \tag{2.9}$$

which yields $p < 0.64$ in $(0, 1)$. □

CHAPTER III

Redesigning the Spectrum Access Mechanism

3.1 Introduction

Most WiFi networks today operate with the default 20MHz bandwidth [5]. This bandwidth has been exhausted in the widely used 802.11g standard to provide up to 54Mbps data rate, but is becoming insufficient for throughput-demanding applications such as high-definition video streaming. The recently ratified 802.11n standard doubles the data rate using 40MHz channel width. The emerging 802.11ac [60] further enables Gbps wireless communications with 80MHz and 160MHz channels. On the other hand, narrow-band channels (5MHz and 10MHz) have also been incorporated in the recent 802.11 standard [5] to support WLANs with low throughput demands but high energy-efficiency requirements [30].

Although a variety of channel widths can be used, the spectrum is still a limited resource. For example, on the 2.4GHz ISM band used by 802.11b/g/n, the total spectrum width is only 83.5MHz. Hence, it is impractical to guarantee orthogonality between the channels used by every co-located WLAN, especially in the current high-density enterprise and public WiFi networks [9]. Thus, a WLAN often needs to share part or all of its spectrum with others. Most WiFi WLANs today reside on the three non-overlapping 20MHz channels 1, 6, and 11 specified by 802.11 [9], and thus, neighboring WLANs tend to be either orthogonal or sharing an entire channel. But as

channel widths become more heterogeneous, partial spectrum sparing is unavoidable.

The current 802.11 relies on CSMA/CA to coordinate transmitters on the same channel, but it is not inherently designed for partial sharing of the spectrum. An 802.11 transceiver treats an entire channel as a whole spectrum block to perform carrier sensing and packet transmission. It has to defer its transmission even if part of the spectrum is occupied (*e.g.*, by a WLAN that has a narrower bandwidth as shown in Fig. 6.1(a), or resides on a partially overlapped channel). We refer to this problem as *partial-channel blocking*. Partial-channel blocking causes severe under-utilization of non-overlapped spectrum, which should otherwise be able to provide a higher throughput due to less contention. A more critical problem occurs when multiple narrowbands coexist and overlap with a wideband channel. With the 802.11 MAC, the wideband will be able to transmit only if all the narrowbands are idle, resulting in highly unfair channel access opportunities and even starvation of the wideband WLAN.

In this project, we introduce a new mechanism called *Adaptive Subcarrier Nulling* (ASN), to enable partial spectrum sharing between WLANs. ASN builds on the OFDM PHY used by 802.11g/n and other emerging standards [65], in which a channel comprises many small spectrum units called *subcarriers*. ASN groups the subcarriers into several *subbands*, and allows neighboring WLANs to share and contend for access to each subband. When a shared subband is occupied by one WLAN, another WLAN can opportunistically *null* the corresponding subcarriers in that subband, and use those non-overlapping subbands to send packets. ASN performs this adaptation on a *per-packet* basis, so as to fully utilize the available spectrum whenever possible, and to ensure fair access to shared spectrum. With ASN, the partial-channel blocking problem can be naturally solved (Fig. 6.1(b)).

Subcarrier nulling can be realized straightforwardly in the 802.11 OFDM PHY: instead of sending information bits (1 or -1), the transmitter can simply feed 0's to

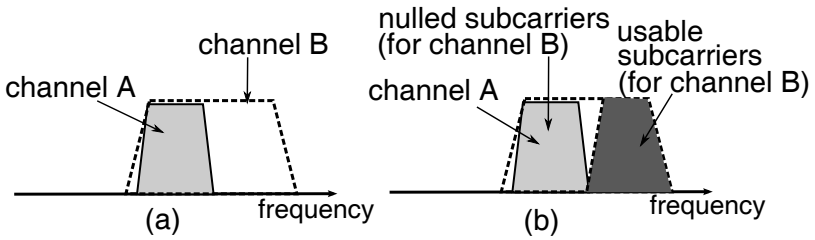

Figure 3.1: (a) Partial-channel blocking problem in wireless LANs. (b) Adaptive subcarrier nulling (ASN) nulls the shared busy subband (containing a number of subcarriers) and leverages the non-overlapping subbands to send data.

the subcarriers, resulting in zero power on the corresponding spectrum. However, it is nontrivial to ensure the receiver can correctly decode the remaining non-zero subcarriers. Since the transmitter decides on the set of subbands to be used for each packet, the receiver has no prior knowledge of the spectrum to be used by an incoming packet, yet it still needs to detect the packets, synchronize to them, and then decode the information bits.

ASN meets these challenges by redesigning the preamble structure, packet detection and decoding algorithms in 802.11. It uses correlation-based algorithms to detect a packet and identify the subbands used by it. It further adapts the pilot-based approach in 802.11 to estimate the channel, and then decodes the bits carried by each subcarrier. In addition, ASN combines the time-domain energy sensing with frequency-domain spectrum sensing, so that a transmitter can identify the spectrum currently in use by neighboring WLANs. Although similar PHY layer problems have been addressed in non-contiguous OFDM (NC-OFDM) communications systems [100, 101, 7, 42] (more details available in Sec. 5.9), ASN represents a complete 802.11 based NC-OFDM design that solves a network-level problem, *i.e.*, partial spectrum sharing for WLANs.

At the MAC layer, ASN retains the carrier sensing and backoff mechanism in 802.11, but makes the busy/idle decision based on the time/frequency domain spectrum sensing. ASN maintains a backoff counter for each subband, and allows decrementing the backoff counter if at least one subband is idle. This simple extension to

802.11 CSMA/CA (referred to as *ASN with direct access*, or ASN-DA) alleviates the partial channel blocking problem, but may cause certain transmitters to dominate a subband. Therefore, we propose an alternative protocol, *ASN with water filling access* (ASN-WF), which aligns the busy time of subbands by adapting the packet size, thereby balancing the access opportunities of different WLANs to shared subbands.

We have implemented an ASN prototype on the GNURadio/USRP platform. Our experimental results show that ASN can sense, synchronize, and decode partial spectrum, with a level of accuracy comparable to the legacy 802.11g that uses a full spectrum. We further use detailed simulation in ns-2 to evaluate ASN in multi-channel, multi-cell wireless LANs. Our experiments demonstrate that ASN significantly improves the throughput and fairness of spectrum sharing. In particular, when two WLANs of different widths coexist, it improves the total network throughput by up to 147.7%, by solving the partial-channel blocking problem. When multiple narrowband WLANs coexist with a wideband WLAN, ASN enables close-to-equal access to shared spectrum, providing an order of magnitude of throughput improvement for the wideband WLAN that tends to be starved by 802.11.

The remainder of this chapter is organized as follows. In Sec. 5.2, we experimentally study the problems caused by partial spectrum sharing and analyze the reasons behind them. In Sec. 3.3, we introduce the detailed design of ASN's channel sensing, detection, and decoding algorithms. Sec. 3.4 describes the two medium access protocols for ASN. Sec. 5.7 presents the implementation and evaluation of ASN. Sec. 5.9 discusses related work and finally, Sec. 5.10 concludes the chapter.

3.2 Motivation

The problem of partial spectrum sharing is akin to the well-explored effects of partially-overlapping channels in 802.11b WLANs [92]. In the 2.4GHz ISM band for 802.11b/g/n, 11 channels of 20MHz bandwidth each can be used, and adjacent

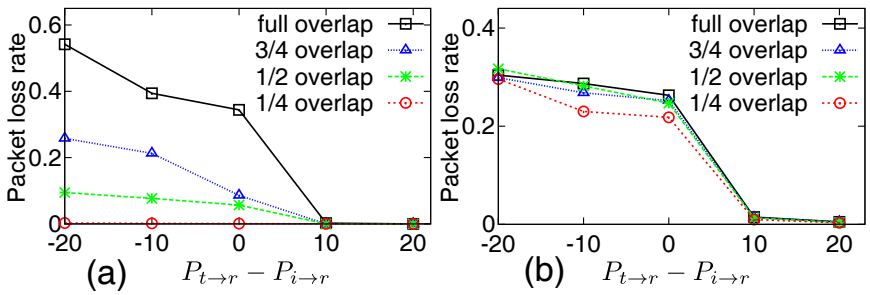

Figure 3.2: Effects of partial-channel interference for 802.11b and 802.11g.

channels' center frequencies are separated by 5MHz. Hence, neighboring WLANs may have $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$ or full overlap, if any. For 802.11b, interference from partially-overlapping channels is proportional to the amount of overlap, which may be much less than a full overlap, and thus, partially-overlapping channels can be simultaneously active in many cases [92]. However, does this apply to 802.11g, which builds on a distinct PHY layer? In this section, we answer this question with detailed experiments, and then discuss the advantages of ASN in OFDM WLANs.

3.2.1 Partially-Overlapping Channels for 802.11b and 802.11g

We measure the interference caused by partially-overlapping channels using a small testbed that consists of a transmitter (N_t), receiver (N_r) and interferer (N_i), which are laptops equipped with Atheros 5414 802.11b/g NIC, running on MadWiFi trunk-r4134. N_t and N_r use the same 20MHz channel, while N_i resides on a 20MHz channel that partially or fully overlaps with them, and its carrier sensing function is disabled. N_t continuously transmits ICMP Ping-broadcast packets to N_r at 100 pkts/second with packet size 1.4KB, while at the same time N_i emits Ping-broadcast packets with the same rate and size. We adjust the transmit power of N_t and N_i, thus varying the relative power received by N_r (denoted by $P_{t\to r}$ and $P_{i\to r}$) when N_t and N_i use the same channel.

Fig. 3.2(a) shows the packet loss rate of 802.11b (with 2Mbps data rate) subject to interference. When $P_{t\to r} - P_{i\to r} < 10$dB, interference may become detrimental to the

data transmission. However, different fractions of channel overlap between N_i and N_t lead to disparate loss rates. When $P_{t\to r}$ is 20dB lower than $P_{i\to r}$, a fully-overlapped channel suffers 54.7% loss, whereas a $\frac{1}{4}$-overlapped channel has nearly 0 loss. This result is consistent with existing measurements of 802.11b [92].

However, for 802.11g, packet loss rate is almost invariant to the channel overlap (Fig. 3.2(b)), *i.e.*, the effect of interference from a $\frac{1}{4}$-overlapped channel is comparable to that from a fully-overlapped channel. Therefore, existing approaches that exploit concurrent transmissions from partially-overlapping channels [92] are not applicable to 802.11g.

The distinct effects of partially-overlapping channels for 802.11b and 802.11g root in their PHY layers. The 802.11b PHY is based on DSSS (direct-sequence spread spectrum), which spreads one bit of information over an entire spectrum of 20MHz. Its SINR equals the total power of the non-interfered spectrum divided by that of the interfered part. For example, even when $P_{t\to r} = P_{i\to r}$, the resulting SINR is up to $10\log_{10}(4) = 6$dB when $\frac{1}{4}$ of N_i's spectrum overlaps with N_t. This SINR is enough to ensure close to 100% decoding probability at a low modulation level (*e.g.*, BPSK) [13].

In contrast, for the OFDM PHY used by 802.11g, a 20MHz channel is divided into 64 spectrum units (*i.e.*, *subcarriers*), each carrying one (or more) bits of information. An 802.11g packet comprises multiple *OFDM symbols* each occupying the 64 subcarriers and transmitted consecutively over time. When N_t and N_i's spectrum overlap by $\frac{1}{4}$ and $P_{t\to r} = P_{i\to r}$, $\frac{1}{4}$ of the subcarriers in each OFDM symbol will have an SINR of $10\log_{10}(1) = 0$dB, which are unlikely to be correctly decoded. Equivalently, $\frac{1}{4}$ of an interfered packet will be corrupted and is unlikely to be recovered. This is the reason why energy sensing is mandatory in 802.11g [5, Sec. 17.3.10.5] to sense and prevent interference from partially-overlapping channels.

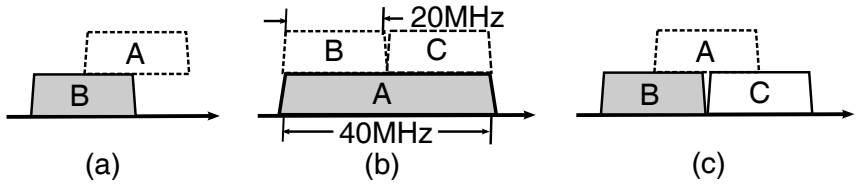

Figure 3.3: Heterogeneous channel width or partially-shared channels cause inefficient or unfair spectrum usage in 802.11.

3.2.2 Why ASN?

Given that partially-overlapping channels in OFDM WLANs cannot transmit concurrently, coexistence of multiple channels faces several critical challenges, which can be solved by ASN.

3.2.2.1 Partial-channel blocking

As discussed in Sec. 5.1, the partial-channel blocking problem occurs in the 802.11g WLAN when part of the channel is used by a co-located narrowband WLAN, and hence, the entire channel must suspend its transmission. In the example of Fig. 6.1, suppose WLAN A and B are 20MHz and 40MHz, respectively. Both transmit packets of the same size, but the transmission takes only 1 time slot for the 40MHz, and 2 for the 20MHz channel. Using 802.11, both WLANs have an equal chance to access the medium, resulting in mean spectrum utilization of $(20 \times 2 + 40 \times 1)\frac{1}{3} \approx 26.7$MHz. In contrast, ASN can activate the right half of channel B even when A is transmitting, thus maintaining 40MHz spectrum utilization at any time. With respect to individual spectrum usage, for legacy 802.11, the 20MHz channel would achieve $\frac{20\times2}{3} \approx 13.3$MHz, and the 40MHz channel achieves $\frac{40}{3} \approx 13.3$MHz — clearly, the 40MHz WLAN does not gain advantage when coexisting with a 20MHz one. With ASN, the 20MHz WLAN still has 13.3MHz channel utilization, but the 40MHz WLAN achieves $\frac{20+20+40}{3} \approx$ 26.7MHz, thereby doubling its throughput.

A similar scenario occurs when two channels partially overlap, as shown in Fig. 3.3(a). Since the legacy 802.11g can only activate one channel at a time, its spectrum usage

is only 20MHz. In contrast, by nulling the overlapping subcarriers and reusing the non-overlapping ones, ASN fully exploits the 30MHz spectrum, improving spectrum utilization by 50%.

3.2.2.2 Channel starvation

The CSMA mechanism in 802.11 may starve a wideband WLAN when it coexists with multiple narrowband WLANs. Fig. 3.3(b) illustrates the case when a 40MHz channel A partially overlaps with two orthogonal 20MHz channels B and C. With 802.11, A can transmit only if both B and C are idle, which occurs only when B and C finish their transmission approximately at the same time, and subsequently A wins the contention over both. Clearly, this is a rare case when B and C have backlogged traffic, so A will remain starved most of the time, although nominally it should have a higher throughput with larger bandwidth.

In general, such starvation effects occur whenever a WLAN partially shares spectrum with several other orthogonal WLANs (*e.g.*, the scenario in Fig. 3.3(c)). Using ASN, the vulnerable WLAN can opportunistically null the busy part of the spectrum, and access the idle part, thus preventing starvation. It might seem feasible to achieve the same result by directly reducing the channel width of A and relocate it to the idle part of the spectrum. In practice, however, the channel switching time is in the order of several packets' duration [105], and the channel status may have already changed after relocating the channel. In ASN, a transmitter fixes its center frequency and maximum bandwidth, and performs subcarrier nulling on a per-packet basis, thus it needs not switch the channel and wait for the radio to stablize.

3.2.2.3 Experimental validation

To validate the above motivating problems, we measure the throughput of three partially-overlapping WLANs running 802.11g (*i.e.*, the scenario in Fig. 3.3(c)). Each

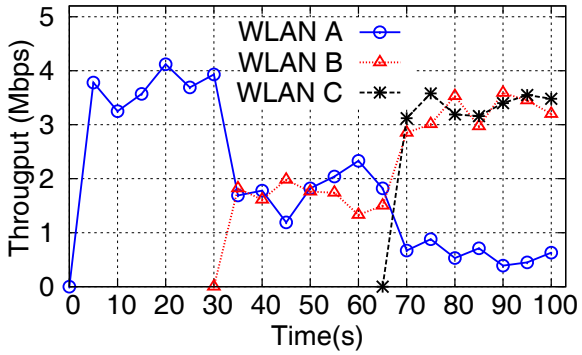

Figure 3.4: Throughput evolution of 3 partially-overlapping WLANs.

WLAN consists of an AP and a client, with saturated downlink transmission, 6Mbps data rate and 1KB packet size. WLAN A, B and C are activated at 0s, 30s, and 65s, respectively. Fig. 3.4 shows the resulting throughput over time. When A and B are activated, only one of them can transmit at any time. Although they occupy 30MHz channels in total, the total throughput is similar to that of a single 20MHz WLAN (equivalent to the scenario in Fig. 3.3(a)). After all WLANs are activated, A's throughput is only around 17% of the two competitors' (the scenario in Fig. 3.3(c)). The same starvation effect would occur for the scenario in Fig. 3.3(b), where the access opportunity of A remains the same as in Fig. 3.3(c). Clearly, 802.11 results in inefficient and unfair spectrum usage in the presence of partially-shared channels.

3.3 OFDM Subcarrier Nulling

A key challenge in realizing ASN is to ensure a node can sense partially-used channels, and can detect, synchronize, and decode a packet, without knowing in advance the spectrum used by the transmitter. In this section, we present the detailed design of ASN to address this challenge.

3.3.1 ASN: An Overview

ASN allows a node to adaptively use a subset of subcarriers within its channel bandwidth. Observing that the channel bandwidth and overlap between channels in

802.11 is a multiple of 5MHz, ASN manages the spectrum in the unit of 5MHz subband, each comprising a group of 16 subcarriers. During carrier sensing, a transmitter senses the subbands within its bandwidth separately, and runs a CSMA/CA-like medium access protocol (Sec. 3.4) to schedule the transmission. The receiver uses a self-correlation algorithm to detect packets, and runs a cross-correlation with known sequence patterns to determine the subbands used by the transmitter and achieve synchronization. It then estimates the channel coefficients and decodes all subcarriers carrying information bits. In what follows, we detail each step throughout this process. Without loss of generality, we assume the maximum bandwidth used by the transceivers is 20MHz.

3.3.2 Sensing Subbands

An ASN-enabled transmitter needs to promptly identify the subbands currently in use. This is achieved by combining time and frequency domain energy sensing. Fig. 3.5 illustrates a typical procedure of subband sensing.

The time domain sensing is akin to the built-in carrier sensing primitive in 802.11g. It calculates the energy level via a moving average of the digital signals (*i.e.*, the sequence of discretized, complex samples provided by the radio's analog-to-digital converter) within a short period, and declares a busy channel if the output exceeds the CCA (clear-channel assessment) threshold. The window size of the moving average is set to half of the length of an 802.11 preamble, to ensure a packet can be sensed promptly.

Time-domain sensing alone can sense a busy channel, but does not discriminate subbands. ASN needs to further analyze the frequency domain of the signals. Specifically, it calculates the power-spectrum density (PSD) of the recent N samples using FFT (N is called the FFT size). To ensure sufficient resolution, N needs to be larger than the number of subcarriers used by the entire channel ($N = 64$ for a 20MHz

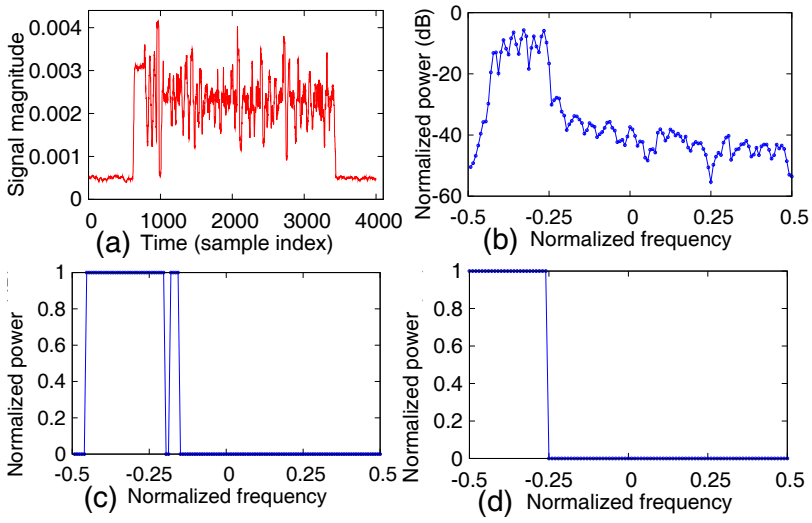

Figure 3.5: Subband sensing in ASN (the transmitter's channel has $\frac{1}{4}$ overlap with the carrier sensing node): (a) receiving time domain samples and perform time domain energy detection (b) analyzing the PSD of samples (FFT size is 256) (c) regularizing the PSD (d) matching with an ideal overlapping pattern.

channel).

Based on the PSD, ASN analyzes the power distribution and compares it with all possible channel-overlapping patterns. Intuitively, if the power is uniformly distributed over the entire spectrum, then the signals on the air come from a fully-overlapped channel; otherwise, only a fraction of the channel is occupied. The exact fraction of channel in use is hard to calculate, because different subcarriers may exhibit different power levels due to frequency-selective fading, and the imperfect hardware filter (used to confine the radio's bandwidth) smears the boundary of the PSD curve. Fortunately, in 802.11g, the minimum separation between adjacent channels is 5MHz. Hence, for a 20MHz channel, for example, the overlapping pattern is one or a combination of only 4 possible overlapping cases: $\frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and full overlap. Based on this observation, ASN first regularizes the PSD into a rectangular curve, compares it with all possible overlapping patterns, and then selects the one with maximum matching (Fig. 3.5(c) and (d)). The PSD regularization is equivalent to thresholding points on the PSD curve with the frequency domain CCA threshold, which equals the

time domain energy sensing threshold (-62dBm in 802.11g [5]) normalized by channel bandwidth.

Note that the complexity of time-domain sensing is the same as the RSSI calculation in typical communications systems, which is linear with respect to the number of incoming samples. Since frequency sensing is performed only after a sequence of signals pass the time-domain sensing, it takes constant time no matter how many samples come. The constant actually depends on the number of packets that cause the time-domain sensing to return "busy".

3.3.3 Packet Detection and Synchronization

In ASN, a receiver must be able to detect a packet and synchronize to it, without prior knowledge of the spectrum usage. Energy sensing alone is insufficient for packet synchronization. ASN meets this challenge by redesigning the preamble structure of 802.11g.

3.3.3.1 Preamble structure in 802.11

The original 802.11g preamble (also referred to as *STF*) lasts $8\mu s$ and occupies all 64 subcarriers. From the frequency perspective, it comprises a random complex sequence spreading over every 4 subcarriers. Other subcarriers are set to 0. Owing to the duality between frequency-domain discretization and time-domain periodicity, the time domain of STF is a periodic signal that repeats every $\frac{64}{4} = 16$ complex samples [71]. The receiver performs *self-correlation* between the latest 16 samples and previous 16 samples, which has an outstanding output only if two consecutive sequences of samples match (*i.e.*, an STF appears), and the corresponding output is comparable to the signal's energy level [71]. After detecting the STF, the receiver further performs *cross-correlation* between the received STF samples and the original samples in the STF. An outstanding peak appears only when the received samples

align with the known STF, and the peak position is used as a synchronization point marking the start of the packet.

3.3.3.2 Preamble structure in ASN

When subcarrier nulling is enabled, the random sequence in STF becomes shorter and vulnerable to noise. For example, when 48 subcarriers are nulled and the remaining 16 subcarriers are used for packet transmission, only 4 non-zero subcarriers remain in STF, which is insufficient for generating outstanding correlation output. Therefore, we modify the 802.11g preamble as follows.

First, we spread a non-zero random sequence over every 2 subcarriers in the STF, resulting in a time-domain sequence of period 32. Consequently, the cross-correlation peak results from correlation with a random sequence that has twice the length compared with the 802.11 sequence. For example, even when only a single subband (16 subcarriers) is used, 8 non-zero subcarriers are used to carry the random sequence, and therefore the STF becomes more resilient to noise.

Second, we assign different random sequences for different channel widths. For $\frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ and full channel width (corresponding to 1 to 4 subbands), each of them has a unique random sequence for STF. The receiver can easily identify the fraction of channel used by the transmitter by correlating the detected STF with all possible random sequences. The one that outputs peaks with the highest magnitude corresponds to the sequence used by the transmitter, and the peaks are used as synchronization points.

Fig. 3.6 illustrates the packet detector's output when a packet occupying one subband is received. The experiments runs on our prototype of ASN on the GNU-Radio/USRP platform (more details in Sec. 3.5.1.1). It can be seen that the self-correlation output is close to the energy level only at the preamble part; and is much smaller otherwise. Hence, it is used as a baseline for detecting the STF. In addition,

when the length of the cross-correlation sequence mismatches the number of subbands used by the incoming packet, the output peaks have a much lower magnitude than those when the correct sequence is used.

3.3.4 Decoding Bits from Subbands

To decode a packet in 802.11g, the receiver first estimates the channel coefficients (including magnitude attenuation and phase distortion) of each subcarrier, and the frequency offset between transmitter and receiver, using an additional preamble following the STF, called *long-training field* (LTF). LTF comprises two duplicated versions of a random sequence (consisting of 1 and -1) of length 64 carried by the 64 subcarriers [5]. In ASN, when part of the channel is used, the random sequence is truncated accordingly (*i.e.*, the nulled subcarriers carry 0). To obtain the channel coefficients and frequency offset, the receiver performs self-correlation between the two truncated random sequences and normalizes it by the magnitude, similar to an 802.11 channel estimator [71]. To decode the bits, the receiver first performs IFFT over each 64 samples within an OFDM symbol, to obtain the complex samples corresponding to each subcarrier, and then normalizes the samples with the subcarrier' channel coefficient. The normalized complex number is then mapped to the closest constellation point to obtain the digital information bits (for BPSK modulated bits, the constellation points lie at 1 and -1).

Due to temporal variation, the channel coefficients must be continuously updated when decoding the OFDM symbols. Moreover, the frequency offset estimation must be continuously refined, because even small errors in the initial LTF-based estimation may accumulate and result in decoding failure near the end of the packet. ASN updates the channel estimation using a *pilot* scheme similar to 802.11g. Specifically, among all the non-zero subcarriers in one OFDM symbol, several subcarriers (*i.e.* pilots) always send known bits. The phase drift between pilot subcarriers is used

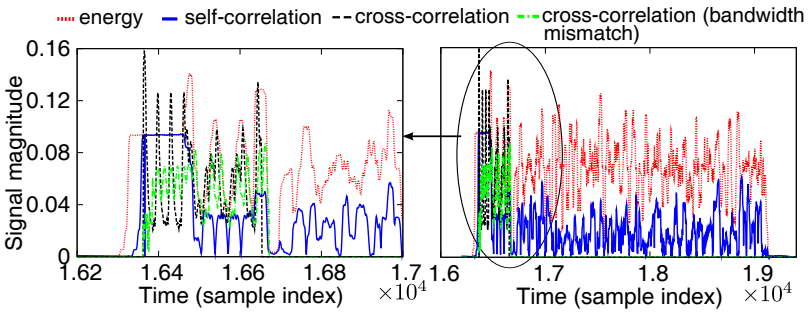

Figure 3.6: Detector's output when a packet arrives.

to update the frequency offset and channel coefficients [71]. When a partial-channel is used, ASN only uses 2 pilot subcarriers (due to reduced number of subcarriers available) instead of the 4-pilot scheme in 802.11 [5].

3.3.5 Managing Adjacent Channel Interference

Although different subbands are orthogonal, their PSD is imperfect and may leak power and cause interference to adjacent subbands used by other WLANs, referred to as adjacent channel interference (ACI). To alleviate ACI, the 802.11g OFDM PHY specified a guardband for each 20MHz channel. Among the 64 subcarriers (each is 312.5KHz), 6 are dedicated as guardband for the left border of the channel, and 5 for the right border [5, 128]. This guardband configuration is over-provisioned for most network topologies and under-utilizes spectrum [128].

ASN employs fixed, but narrower guardbands. For a single subband, 1 subcarrier is used on the left boarder and 2 on the right boarder as guardband. Hence, two adjacent subbands are separated by 3 subcarriers, which is sufficient for most network topologies [128]. When all subbands are aggregated (*i.e.*, an entire channel is used), ASN restores the guardband size used by 802.11g.

In practice, harmful interference may still occur when links are closely located, even with a conservative guardband size as in 802.11. A larger guardband size may reduce such hazards, but at the cost of lower data rate. An optimal guardband configuration scheme would set the guardband according to the network topology

and intensity of interference between links [128]. We leave such schemes as our future work.

3.4 ASN-Aware Medium Access

When multiple WLANs partially overlap, a MAC protocol is necessary to arbitrate their contention for use of the shared subbands. We propose two MAC protocols, ASN-DA and ASN-WF, to achieve this objective.

3.4.1 ASN with Direct Access (ASN-DA)

The ASN-DA protocol adopts a CSMA/CA algorithm similar to the legacy 802.11, but manages sensing, backoff and transmission for each subband. When some subbands are busy, it opportunistically nulls subcarriers in those subbands, aggregates the remaining subbands, performs backoff and sends packets through them. Fig. 3.7(a) illustrates a typical process of ASN-DA when two orthogonal 20MHz WLANs share the channel with a 40MHz WLAN (*i.e.*, the scenario in Fig. 3.3(b)).

Whenever a packet is queued, the transmitter first calls the PHY layer for time-frequency domain CCA. It freezes the backoff counter if the entire 20MHz channel is sensed busy. Otherwise, if at least one subband is idle, it generates a common backoff period for those idle subbands, using the binary exponential backoff algorithm in 802.11 [5]. Then, these subbands start decrementing the backoff counter for each idle time slot (specified to $9\mu s$ in 802.11g).

When aggregating multiple idle subbands, ASN-DA must take into account the heterogeneity in their channel status, including the backoff-counter's status and back-off window size. During the count-down process, a subband may be acquired by other WLANs, and its backoff counter must be frozen. Therefore, the initial idle subbands may end up with a different firing time. ASN-DA sends the queued packet through the set of subbands that first fire (*i.e.*, their backoff counters decrement to 0, and they

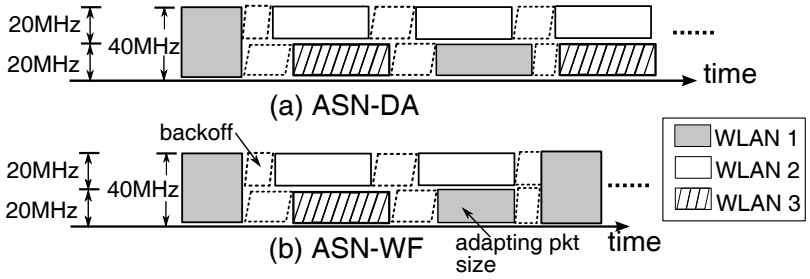

Figure 3.7: ASN-Aware MAC protocols. WLAN A uses a 40MHz channel, and the other two are using 20MHz.

remain idle for a DIFS period [5]). Meanwhile, other subbands will be frozen. Similar to 802.11, backoff windows of used subbands grows exponentially upon transmission failure. Therefore, not all subbands have the same backoff window size at any time. When aggregating subbands, ASN-DA generates the backoff counter based on the average of their backoff window size.

A drawback of ASN-DA is that it may lead to unfair access to shared subbands. For example, in Fig. 6.1, when the channel A acquires the shared subbands, it may constantly hold the subbands, while channel B can only access the remaining non-overlapping subbands. This problem may be alleviated by the post-backoff mechanism in 802.11 (*i.e.*, transmitters need to back off after successfully completing a transmission [5]), which may eventually grant the opportunity for channel B to acquire the subbands. But such opportunities are rare when both WLANs have backlogged traffic. Therefore, we design an alternative protocol, ASN-WF, to address this problem.

3.4.2 ASN with Water-Filling Access (ASN-WF)

The basic idea behind ASN-WF is to adapt the size of each packet, so that its duration (including the ACK) aligns with the earliest timestamp that another subband is expected to become idle (a typical procedure shown in Fig. 3.7(b)). From the time-domain perspective, ASN-WF attempts to "fill" the current idle subband, while maximizing the opportunity to aggregate with other subbands. To this end,

ASN ensures multiple WLANs can have the opportunities to start from scratch and contend for the entire set of subbands within its channel bandwidth, thus preventing the case where a certain subband is exclusively occupied by one WLAN and achieving better fairness.

ASN-WF determines the busy duration of each subband based on the network allocation vector (NAV) provided by transmitters occupying the subband. The NAV is embedded in the header of a signaling packet preceding the actual data. It piggybacks the duration (number of time slots) that a packet plus ACK will occupy the subband. Since different subbands may be shared with different WLANs, the transmitter embeds the NAV into each subband that it uses to inform all those WLANs.

After completing each transmission, a transmitter usually has more idle subbands available to contend for. However, it still needs to start the normal CCA and backoff procedure for all idle subbands, in oder to prevent unfair occupation. ASN-WF uses the same algorithm as in ASN-DA to increment/decrement the backoff window size.

3.5 Implementation and Evaluation

In this section, we first validate the feasibility and performance of ASN on the GNURadio/USRP software platform. Since this platform does not yet support MAC-level functionalities due to its large response time, we use detailed simulation with ns-2 to evaluate the MAC-layer performance of ASN in multi-cell WLANs.

3.5.1 Performance of Subcarrier Nulling

3.5.1.1 Implementation and experiment setup

We implement ASN's PHY-layer functionalities on GNURadio and test it on USRP. USRP is a software radio transceiver that converts digital symbols into analog waves carried by a center frequency within the ISM band. It can also receive ana-

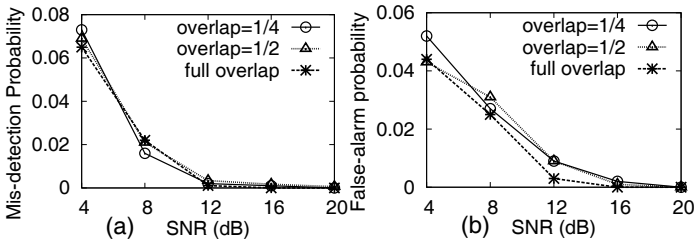

Figure 3.8: Accuracy of carrier sensing for packets from partially-overlapping channels.

log signals via its RF front-end, and down-convert them into baseband signals, *i.e.*, discretized complex samples. The baseband signals are sent to a general-purpose computer running the ASN packet processing modules built atop the GNURadio library.

Our implementation is based on the 802.11g specification [5], but removes certain modules that are used to strengthen robustness to bit-errors, such as the interleaver and error-correction code. The transmitter module first maps digital information bits (0 and 1) to complex BPSK signals, and then modulates the BPSK signals into OFDM symbols. For each OFDM symbol, the pilot subcarriers are inserted according to the number of subbands to be used, following the description in Sec. 3.3.4. The preambles (STF and LTF) are designed offline and prepended to each packet. At the receiver side, the time-frequency domain carrier sensing function and packet detector are running continuously. Once an STF preamble is detected, the receiver identifies the subbands in use and synchronizes to the packet. Then, the channel estimator and decoder follow immediately to decode all the OFDM symbols.

We run the carrier sensing, packet detection, and decoding algorithms of ASN on USRP2 radios equipped with the XCVR2450 daughterboard. We set the maximum bandwidth of the USRP2 transceivers to 20MHz, and vary their center frequencies to create the overlapping patterns consistent with 802.11g channels. As a PHY layer prototype, we run a single pair of transmitter and receiver to demonstrate the feasibility of ASN.

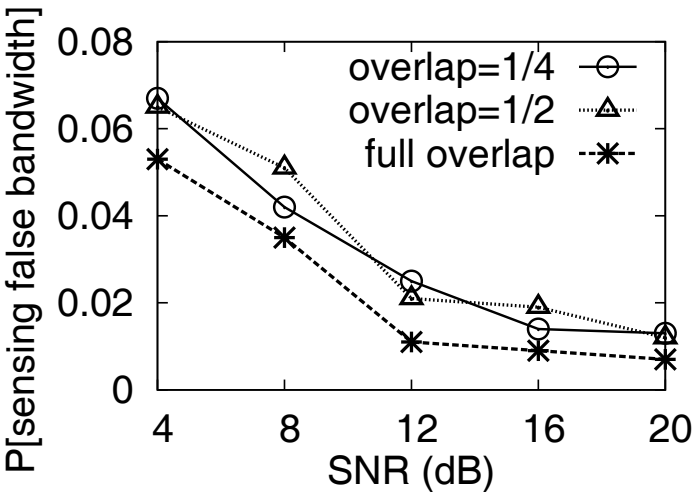

Figure 3.9: Accuracy of sensing the fraction of overlapping spectrum.

3.5.1.2 Carrier sensing of subbands

To test the subband sensing capability of ASN, we adjust the transmit power and distance between the transceivers, thereby creating various levels of signal strength. Since the USRP2 radio does not have a direct mapping between the quantized signal magnitude and absolute power level (in dBm), we measure the relative signal strength (*i.e.*, SNR) instead. The SNR is estimated as $SNR = \frac{E_s - E_N}{E_N}$, where E_s is the average energy level of incoming samples when a packet is present, and E_N the noise floor, both smoothed using a moving average with the window size equal to half of the STF length. In 802.11 [5], packets must be accurately sensed by the energy detector when the signal strength is above -81dBm [5], while the noise floor (which is also the receiver sensitivity) of typical WiFi NIC is -96dBm. Thus, ASN must be able to accurately sense a packet if its signal strength is 15dB above the noise floor.

In the experiments, the transmitter sends 10^6 packets with a constant inter-arrival time, bit-rate of 12Mbps and packet size 512 bytes. We use the mis-detection probability (P_m) and false-alarm probability (P_f) as the performance metrics. P_m is calculated by the fraction of timestamps where a packet is expected to arrive but fails to be sensed within the STF preamble duration; and vice versa for P_f.

Fig. 3.8 plots the resulting P_m and P_f under various levels of SNR and channel

overlapping. When SNR is around 4dB, the CCA may miss packets or trigger false alarms with a relatively high probability (around 0.06). As SNR increases, both P_m and P_f decrease sharply. Above 12dB, both metrics approach 0. In addition, under the same level of SNR, the CCA performance remains almost the same for different levels of channel overlapping. It should be noted that the signal from a partially-overlapped channel is weaker than that from a fully-overlapped one. For example, for a $\frac{1}{4}$-overlapped channel (*i.e.*, overlapped by 1 subband), ASN must be able to detect its packets even though the SNR is $10\log_{10}(4) \approx 6$dB lower than a packet from a fully-overlapped channel.

We further evaluate the accuracy of ASN's frequency domain CCA, *i.e.*, sensing the width of spectrum being used by an overlapped channel. The results in Fig. 3.9 show that the sensing error decreases with SNR, and approaches 1% when SNR is above 15dB. In addition, channels with a wider overlap are easier to be identified, since more matching points in the regularized PSD curve are available (Sec. 3.3.2).

3.5.1.3 Detecting packets

To evaluate the accuracy of detecting a packet intended for the receiver, we configure the transmitter and the receiver to the same center frequency and maximum channel width. Under this setting, the transmitter may still send packets through a fraction of the channel. We denote B_r as the actual bandwidth that the transmitter uses relative to the channel bandwidth. Without loss of generality, we evaluate three cases: $B_r = \frac{1}{4}, \frac{1}{2}$ and 1.

Fig. 3.10 shows the resulting P_m and P_f. We observe a similar trend as in the subband sensing experiments when SNR varies. However, both P_m and P_f are lower compared to pure energy sensing in Fig. 3.8, especially under low SNR. This is because the packet detector uses self-correlation and cross-correlation to enhance resilience to noise, thus achieving higher accuracy.

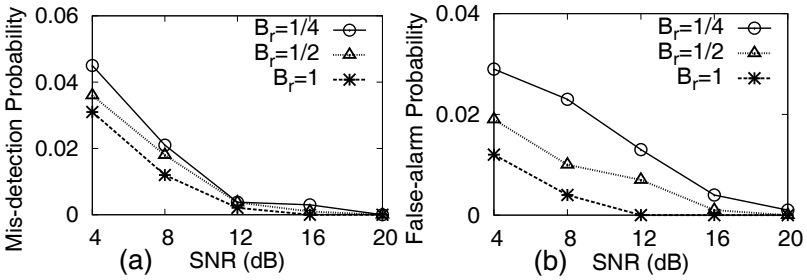

Figure 3.10: Accuracy of detecting packets intended for the receiver.

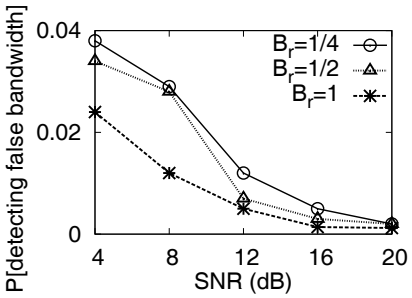

Figure 3.11: Accuracy of detecting the bandwidth used by the transmitter.

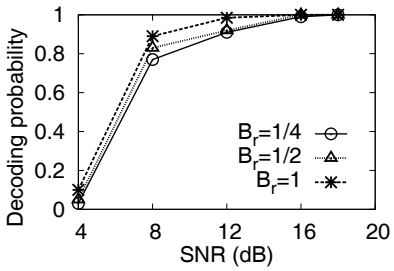

Figure 3.12: Decoding probability of a packet.

We make an additional observation that a lower B_r may lead to lower detection performance, especially when SNR is low. A narrower bandwidth has fewer number of non-zero subcarriers in the STF preamble, corresponding to a shorter sequence for correlation-based detection, and are thus more susceptible to noise. Nevertheless, ASN can easily satisfy the requirement of accurate detection with above 15dB SNR, even if $B_r = \frac{1}{4}$.

Besides, ASN has to identify the packet's bandwidth (*i.e.*, the subbands in use). Recall the packet detector uses cross-correlation with known STF preambles to achieve this, and the accuracy is expected to be higher than a pure energy detector. This is justified in our experiment results in Fig. 3.11. Compared to the energy detector (Fig. 3.9), the detection error is typically reduced by 25% under low SNR, thus ensuring the correct channel width be fed to the decoder to recover the packet.

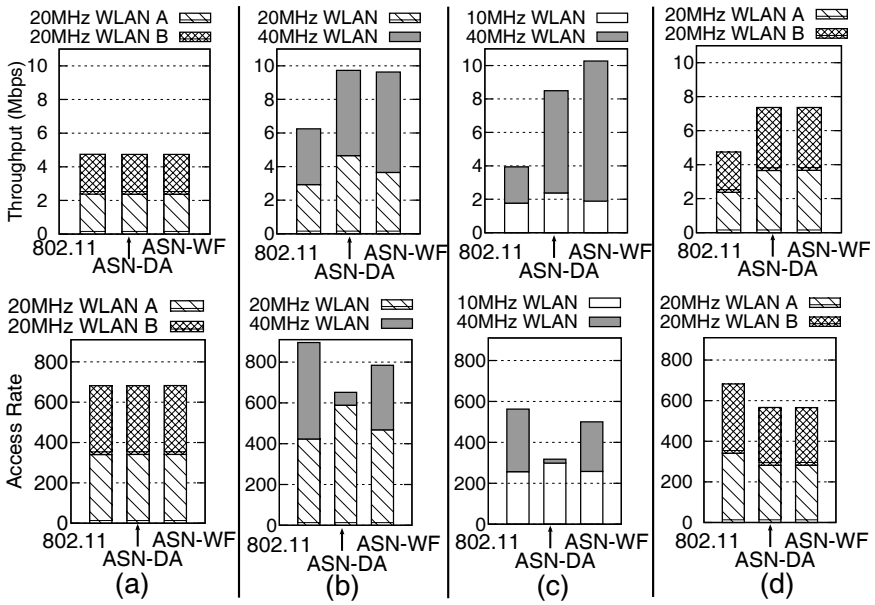

Figure 3.13: Throughput and fairness when two WLANs share spectrum. (a) two 20MHz WLANs with full overlap. (b) a 20MHz WLAN overlap with a 40MHz WLAN (*i.e.*, the scenario in Fig. 6.1). (c) a 10MHz WLAN overlapping with a 40MHz WLAN. (d) two 20MHz WLANs overlapping by 10MHz (*i.e.*, the scenario in Fig. 3.3(a)).

3.5.1.4 Decoding packets

After detecting a packet's preamble, the subsequent OFDM symbols can be decoded to recover the information bits. The decoding probability depends on the SNR level as well as on the accuracy of channel estimation. Our experimental results in Fig. 3.12(a) show that the decoding probability is close to 100% when SNR is above 12dB.[1] Below that SNR level, decoding rate drops sharply. Notably, the decoding performance of packets that partially use the channel is comparable to those on an entire channel, though slightly lower at low SNR due to fewer pilots used for channel estimation. It should be mentioned that error-correction codes may significantly boost the decoding performance, but are not used in our prototype implementation.

[1]We measure the SNR of decoded packets via SNR=$\frac{E_b}{N_0}$, where E_b is the energy-per-bit, equivalently the average magnitude of decoded complex symbols. N_0 is the noise energy-per-bit, equivalently the variance of the magnitude. This SNR metric accounts for the noise introduced by the decoder's channel estimator.

3.5.2 Network Performance

The above experiments on USRP/GNURadio have shown the feasibility of ASN packet detector and decoder, and justified that it can achieve comparable performance with an 802.11 transceiver with full bandwidth under practical SNR settings. Due to the high latency of the user-space signal processing modules of USRP/GNURadio, we were unable to integrate the PHY directly with a MAC and evaluate it in a large network. Hence, we implement ASN in ns-2.34, and use the PHY-layer results to drive the network-level simulation. We modify the PHY parameters in ns-2 and ensure they are consistent with the default values in 802.11g. The original ns-2 uses a binary interference model that declares collision whenever two packets (partially) overlap. We implement an SINR based interference module that accumulates the power level of all interfering packets, and declares a collision only if the SINR is below the decoding threshold (6dB for BPSK [13]). The collision model takes into account the possible partial overlap between packets from different channels. We have also incorporated the features of the ASN PHY: A transmitter can sense subbands separately and send packets even when certain subbands are occupied, and a receiver can detect and decode packets in each subband. The ASN-DA and ASN-WF MAC protocols are implemented on top of this modified PHY layer.

We compare three protocols: the legacy 802.11 MAC, ASN-DA and ASN-WF, using two performance metrics: throughput and *access rate*, *i.e.*, the number of transmission attempts (after CCA and backoff) per second on a shared subband. Access rate is used to study the fairness among different WLANs to access the shared spectrum. Ideally, all contenders should have the same rate of access to a shared subband, and thus, a WLAN with wider channel should achieve higher throughput.

3.5.2.1 Two WLANs partially sharing spectrum

We start with the case where two co-located WLANs are sharing part of the spectrum, each including one AP and two clients, both having saturated downlink traffic[2] running constant-bit-rate UDP file transfer. The packet size is fixed to 1KB. The data rate of 20MHz WLAN is fixed to 6Mbps, and that of 40MHz, 10MHz, 5MHz WLANs fixed to 12Mbps, 3Mbps, 1.5Mbps, respectively, the basic rate defined in 802.11g and 802.11n [5]. We assume the ACI is at least 6dB lower than the received signal strength for each receiver, such that ASN can be fully exploited without causing collision. This can be easily satisfied since the ACI is more than 20dB lower than the signal power even with a guardband size of 2 subcarriers [5, 128]. All our experiments run for 1000 seconds in simulation time, and the results are averaged over 10 repetitions with different random seeds. Fig. 3.13 shows the experimental results.

When the same channel width of 20MHz is used and the two WLANs' channels fully overlap (Fig. 3.13(a)), the legacy 802.11 results in an equal share of throughput and access rate for both. ASN naturally downgrades to the legacy 802.11, and achieves the same level of performance.

The aggregate network throughput increases when a 20MHz WLAN shares its channel with a 40MHz channel (Fig. 3.13(b)), since the 40MHz WLAN takes much less time to send a acket. However, the 40MHz WLAN has almost the same throughput as the 20MHz one, although its data rate is twice as high. This is consistent with the motivating example in Fig. 6.1(a) — 802.11 results in an identical access rate of both WLANs to the shared 20MHz band, but at the same time, the 40MHz WLAN treats its entire channel as a single band, and accesses the non-overlapping 20MHz band at the same rate as the shared one, causing severe under-utilization of spectrum.

[2]Although we only simulate downlink traffic, the direction of traffic does not affect the performance gain of ASN. As long as multiple links coexist and partially share spectrum, the unfairness and inefficiency of CSMA do occur and ASN becomes beneficial.

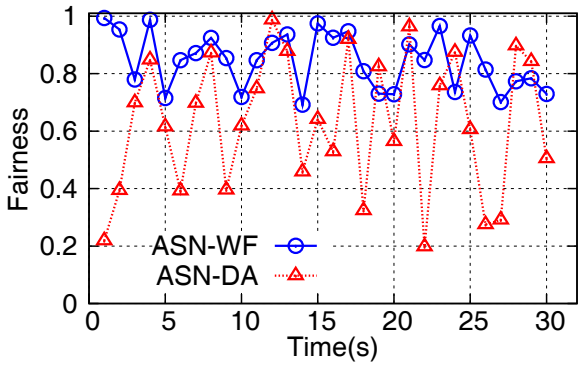

Figure 3.14: Short-term fairness, with respect to access rate to the shared spectrum.

By contrast, with ASN-DA, both halves of the 40MHz channel can be opportunistically exploited at any time. Compared with the legacy 802.11, ASN-DA increases the throughput of the 20MHz(40MHz) WLAN by 58.7% (53.1%), and total network throughput by 55.7%. The downside of ASN-DA is that the 20MHz WLAN gains unfairly a high rate of access to the shared spectrum, leaving the 40MHz WLAN to only exploit the other 20MHz most of the time. This effect is mitigated by ASN-WF. Compared to 802.11, ASN-WF maintains similar throughput for the 20MHz WLAN, but increases throughput of the 40MHz WLAN by 80.1%. It achieves this by allowing fair access to the shared spectrum, while granting the non-overlapping spectrum exclusively to the 40MHz WLAN. In this sense, ASN-WF realizes the intuition that wider channels should gain higher throughput.

When the width of the narrowband WLAN reduces from 20MHz to 10MHz (Fig. 3.13(c)), the total network throughput decreases when running 802.11, although more non-overlapping spectrum is available. ASN-DA shows similar trends of throughput and access rate as in the case of 20/40MHz spectrum sharing. Compared to 802.11, it improves throughput by 34.4% and 181.7% for the 10MHz and 40MHz WLAN, respectively, and the total throughput by 115.5%. ASN-WF achieves the same throughput as 802.11 for the 10MHz WLAN, but improves that of the 40MHz WLAN by 286.5%, and the total throughput by 147.7%. In summary, the spectrum underutilization of 802.11 gets severer as the ratio of the shared spectrum to the channel bandwidth

decreases, and ASN becomes more important in such cases.

Fig. 3.13(d) plots the experimental results for the case where two 20MHz channels overlap with each other by 10MHz (*i.e.*, the scenario in Fig. 3.3(a)). For the legacy 802.11, the same throughput is achieved for the case where two 20MHz channels fully overlap with each other (Fig. 3.13(a)). ASN-DA and ASN-WF can fully exploit non-overlapping spectrum, increasing the throughput of both WLANs by 54.4%. Both protocols lead to an equal access rate to the shared spectrum.

The above evaluation focuses on the long-term access rate to shared spectrum. In the short-term, however, ASN-DA may result in dominant access to certain subbands. We investigate this effect for the case of 20/40MHz WLAN coexistence (*i.e.*, the scenario in Fig. 6.1). We define short-term fairness by the ratio between the minimum and maximum access rate (to the shared 20MHz spectrum) of the two WLANs averaged over a short period (*e.g.*. 1 second). Fig. 3.14 shows the temporal variation of short-term fairness. The fairness level of ASN-DA ranges from 0.2 to 0.98 and exhibits a significant variation. In contrast, ASN-WF maintains much more stable fairness, ranging from 0.67 to 0.99. This justifies the effectiveness of the water-filling approach in ASN-WF for arbitrating fair access to shared spectrum.

3.5.2.2 Multiple WLANs sharing subbands

When multiple WLANs of different channel widths coexist, the one partially overlapping with multiple other WLANs may be starved (Sec. 5.2). In this section, we justify the effectiveness of ASN as a countermeasure. Without loss of generality, we first explore the case where a single 40MHz WLAN overlap with several other orthogonal narrowband channels (*i.e.*, the scenario in Fig. 3.3(b)). The network traffic settings are the same as above.

Fig. 3.15(a) shows that, when the 40MHz WLAN coexists with two orthogonal 20MHz WLANs running 802.11, its throughput approaches 0, while the 20MHz

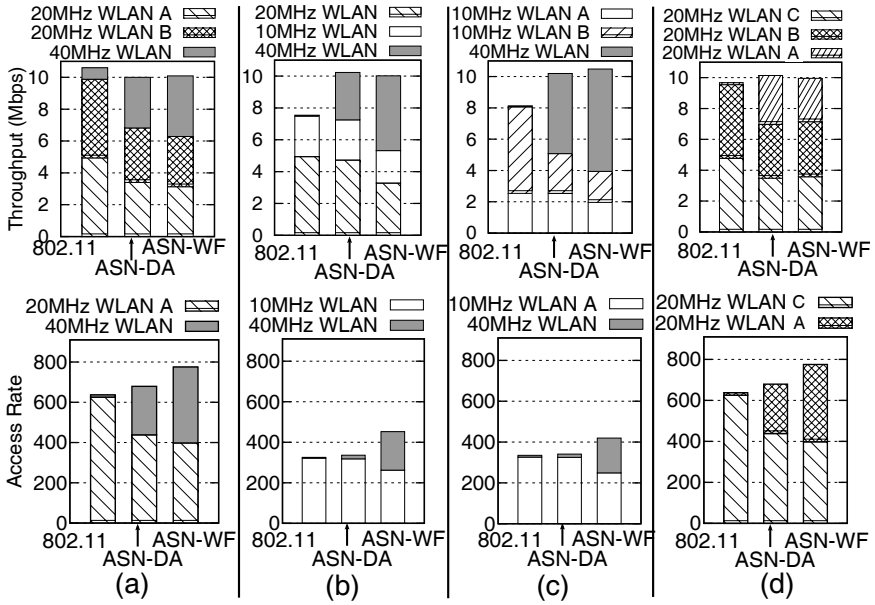

Figure 3.15: Throughput and fairness when multiple WLANs of different channel width coexist.

WLANs have similar throuhgput to the case without any contenders. This is because the 40MHz WLAN can hardly find any slots when both 20MHz contenders are idle. Accordingly, its access rate to a shared 20MHz band is close to 0, far below the equal sharing objective.

Using ASN-DA, the 40MHz WLAN can opportunistically transmit over one of the 20MHz subbands, thus achieving a similar level of throughput as the two 20MHz WLANs. However, its access rate to each 20MHz band is only around $\frac{1}{2}$ compared to that of the 20MHz narrowband WLAN, resulting in low fairness. Again, ASN-WF alleviates this problem and enables close-to-equal access to the shared spectrum.

When the two narrowband WLANs reduce their channel width (*i.e.*, the case for 20/10/40MHz and 10/10/40MHz coexistence in Fig. 3.15(b) and Fig. 3.15(c)), the 40MHz channel remains starved when running 802.11. In contrast, by enabling access to partially-shared spectrum, ASN improves the throughput by an order of magnitude. Owing to efficient usage of the non-overlapping spectrum, the total network throughput is also increased by around 29%.

Fig. 3.15(d) shows a case where three 20MHz channels partially overlap with each

other (*i.e.*, the scenario in Fig. 3.3(c)). Consistent with our measurement (Sec. 5.2), for legacy 802.11, the WLAN that shares spectrum with the other two orthogonal channels is starved. With ASN, all the WLANs achieve a similar level of throughput and access rate to the shared spectrum. Therefore, ASN is both necessary and effective whenever partial spectrum sharing occurs.

3.6 Related Work

Besides the problem of partially-overlapping channels [92] that we discussed in Sec. 5.2, researchers have explored other related problems and proposed their solutions.

Fine-grained channel access. FICA [121] reduces the MAC-layer overhead of high-rate WLANs by splitting a channel into multiple subchannels, and allowing contention for subchannels. It uses a frequency-domain backoff algorithm distinct from the traditional CSMA, and thus cannot coexist directly with current 802.11 WLANs. Moreover, FICA requires tight synchronization (with accuracy below 0.8μs) between all nodes that contend for spectrum. Similar approaches have been proposed to extend the OFDM-based multiple access scheme in WiMax to WiFi WLANs [54]. In contrast, ASN retains the distributed, asynchronous CSMA/CA mechanism, and can be deployed directly and coexist with current 802.11 WLANs. It targets the spectrum under-utilization problem that occurs when 802.11 WLANs partially share spectrum with each other.

Channel width adaptation. The proposal of variable-width channels in recent IEEE standards, such as 802.11-2007 [5], 802.11n [68], and 802.11ac [65], has generated interests in adaptively changing channel width. Chandra *et al.* explored the benefits of adapting channel width to balance the tradeoff between throughput and energy-efficiency [30]. Subsequent efforts [93] proposed to assign spectrum of different widths

to WLANs according to their traffic load, similarly to the notion of traffic-aware channel assignment [105]. With diverse channel widths, the partial channel sharing problem becomes inevitable, and hence, ASN can be used to further enhance such protocols.

Narrowband-wideband coexistence. In [103], a MAC/PHY mechanism called SWIFT is proposed to enable the coexistence between OFDM-based ultra-wideband (UWB) system and the WiFi WLANs that have a relatively narrower bandwidth. SWIFT allows UWB radios to identify the busy channels and then null them to prevent interference to WiFi. However, it identifies busy spectrum by poking the WiFi devices with a jamming tone and observing their backoff reaction. It enables UWB to achieve long-term coexistence with WiFi by evacuating the spectrum where WiFi resides on. ASN adopts OFDM subcarrier nulling similar to SWIFT, but is able to perform such adaptation on a per-packet basis, via a non-intrusive way of spectrum sensing. Using ASN, a WiFi WLAN can access spectrum with short-term fairness even if it is shared with other WLANs.

An alternative approach, Remap [81], is proposed to facilitate the coexistence between partially-overlapping 802.11 channels. Remap resolves the collision due to concurrent access to shared spectrum, by shuffling the OFDM subcarriers and harvesting diversity from repeated collisions. It can be combined with ASN as a means of collision resolution, since the CSMA/CA in ASN alone does not guarantee collision-free transmissions.

Subband nulling for OFDM networks. Subband nulling has also been used for different purposes. For example, MPAP [59] enables WiFi and ZigBee APs to operate on the same radio platform by nulling certain WiFi subcarriers and allocating them to ZigBee. In the context of OFDM cellular networks, there have also been proposals to null subbands that experience deep fading [73], or cause severe interference to adjacent cells [75], and reallocate the power to usable subbands. In contrast, ASN nulls

subbands that are already occupied by existing WLAN cells, and uses the remaining subbands to transmit data to improve the fairness and efficiency of multi-cell WLANs.

Non-contiguous OFDM (NC-OFDM) for cognitive radio networks (CRN). The PHY-layer challenges of ASN resemble those in non-contiguous OFDM (NC-OFDM), a key enabling technology for CRN where available spectrum tends to be scattered over a wide range. Poston *et al.* [100] demonstrated the feasibility of NC-OFDM using a software radio based prototype, which was implemented by directly nulling the subcarriers of an OFDM communications system. Qu *et al.* [101] proposed two decision-theoretic algorithms for detecting active OFDM subbands occupied by primary users in CRN. The detection algorithms rely heavily on *a posteriori* probability of each subband's being active, which must be obtained via extensive training and is more suitable for static networks. In [7], a packet synchronization mechanism for NC-OFDM is proposed, which leverages a cyclic pattern of OFDM symbols, and is suitable for CRNs with unknown preambles. In [42], another PHY-layer challenge, *i.e.*, detecting which subcarriers are occupied, is addressed by modifying the random sequence in the 802.11g preamble. ASN's subband detection algorithm is based on a similar rationale, but becomes much simpler by leveraging the specific channel overlapping patterns in 802.11.

In summary, algorithms have been proposed to solve various PHY-layer problems in NC-OFDM communications. ASN's PHY layer can be considered as a specific NC-OFDM, but it represents a complete 802.11-based system design that includes subband sensing, detection, synchronization, and decoding. More importantly, it uses such a PHY layer to solve network-level problems, *i.e.*, partial channel blocking and wideband starvation which, to the best of our knowledge, have not been discussed elsewhere.

3.7 Conclusion

In this work, we investigated the inefficiency and unfairness of 802.11 in coordinating partial spectrum sharing between WLANs, which occurs due to partially-overlapping channels or coexistence of heterogeneous channel widths. We proposed an innovative solution, ASN, that opportunistically splits the channel into subbands, nulls busy subbands, aggregates idle subbands, and transmits packets through them. We designed a set of OFDM packet processing algorithms that enable an ASN receiver to sense, detect, and decode the packets without prior knowledge of the subbands to be used by the transmitter. We also proposed two ASN-aware MAC protocols that are 802.11-compatible, but enable efficient and fair access to partially-shared spectrum in wireless LANs. Our design was validated with implementation and experimentation on the GNURadio/USRP platform and the ns-2 simulator. As future work, we plan to extend ASN to facilitate the spectrum sharing in the whitespace networks where spectrum tends to be fragmented and partial spectrum sharing becomes unavoidable.

CHAPTER IV

Redesigning the Carrier Signaling Mechanism

4.1 Introduction

Spectrum scarcity is known to be a main obstacle to the scaling of wireless network capacity. Spectrum sharing has been advocated as a key remedy for this problem, especially after the successful deployment of WLAN and WPAN devices on an unlicensed band. However, severe performance degradation has been observed when heterogeneous devices share the same frequency band (*e.g.*, WiFi & Bluetooth [56], WiFi & ZigBee [99], WiFi & WiMax [133]). Such a coexistence problem is rooted at their mutual interference due to the lack of coordination. Although most systems incorporate interference avoidance mechanisms, such as listen-before-talk, they are designed to resolve the collision between the same type of networks. These built-in mechanisms become less effective for heterogeneous MAC/PHY protocols/standards, which adopt asynchronous time slots, different scheduling modes (*e.g.*, TDMA vs. CSMA), disparate transmission/interference ranges, and incompatible communication mechanisms. The problem is likely to persist and exacerbate in future, especially within the recently opened-up TV white-space [41] for unlicensed users.

We address a key question related to this trend: *how should heterogeneous wireless MAC/PHY protocols coexist to share spectrum?* We will focus on two such protocols, WiFi (IEEE 802.11) and ZigBee (IEEE 802.15.4), that share the 2.4GHz ISM band.

WiFi is typically deployed for pervasive Internet access or medium-scale WLANs, whereas ZigBee targets monitoring and control applications for home, hospital, or enterprise environments [61]. The conflicting coexistence between them has been observed in existing measurement studies [99, 55], and their underlying cause is representative of many other coexisting networks. In particular, ZigBee packets are transmitted with 20dB lower power than WiFi packets, and tend to be invisible to, and often interrupted by, WiFi transmitters. Even when it can be sensed by WiFi, a ZigBee transceiver has a 16× longer response time, and is often preempted by WiFi, when it switches from sensing to transmission, or transmission to reception mode. Besides, ZigBee allows for TDMA mode, which operates without carrier sensing, and may arbitrarily collide with an ongoing WiFi transmission. Therefore, by resolving the coexistence between ZigBee and WiFi, one could naturally extend the solution to other heterogeneous networks facing similar problems.

To meet this goal, we propose a new paradigm, called *Cooperative Busy Tone* (CBT), that enhances the mutual observability between ZigBee and WiFi, thereby improving their coexistence. CBT builds atop the legacy ZigBee MAC, but allows the clients to cooperatively strengthen their visibility to WiFi. Unlike the traditional CSMA that relies on a data packet as an implicit busy tone, CBT designates a separate node (either a ZigBee client closer to the WiFi transmitter, or a dedicated high-power ZigBee transceiver) as a *signaler* that emits the busy tone. The busy tone harbingers the actual data transmission, and continues throughout the DATA-ACK transmission, so as to prevent WiFi preemption.

An immediate challenge to CBT is: "how to prevent the busy signal from interfering with the data packet?" We introduce an innovative *frequency flip* mechanism that temporarily re-locates the signaler to an orthogonal ZigBee band, but still ensures that the busy tone is perceived by the WiFi transmitter.

There is an additional concern: "how much performance improvement will CBT

bring to ZigBee, and what is the cost to WiFi?" We develop an analytical framework that quantifies the network performance. Our analysis reveals that the legacy ZigBee MAC suffers a 11–23% collision rate even when WiFi leaves the channel unused for 80% of time, and suffers an up to 79% collision rate when WiFi becomes saturated. With CBT, the collision rate can be reduced to below 5% under medium to low WiFi interference, and to below 20% under saturated WiFi traffic. The performance can be improved further by tuning the design parameters, such as the start time and duration of the busy tone. Our analysis also shows that for typical low duty-cycle applications, CBT introduces negligible performance degradation to WiFi, as compared to the legacy ZigBee.

The above analytical results are validated via detailed simulation of CBT in ns-2. We have also prototyped CBT based on TinyOS and the GNURadio library [2]. Our experiments on the MicaZ motes and USRP2 [39] software radio platform further corroborate the feasibility and effectiveness of CBT.

The remainder of this chapter is organized as follows. Sec. 5.9 reviews existing studies on the coexistence of heterogeneous wireless networks. Sec. 4.3 introduces the key components in CBT. Sec. 4.4 establishes a theoretical framework to analyze the performance of the ZigBee-WiFi network, with and without CBT. Sec. 5.7 validates CBT with ns-2 simulation and real experiments. Finally, Sec. 5.10 concludes the chapter.

4.2 Related Work

Coexistence has long been a problem for protocols operating on the ISM band. Industrial associations, such as the ZigBee Alliance [110], demonstrated that ZigBee can coexist well with WiFi in home networks. However, their experiments were conducted under light WiFi traffic conditions. Many empirical studies revealed severe collision when ZigBee coexists with medium to high WiFi traffic [55, 99].

The IEEE 802.15.2 [3] proposed an adaptive frequency hopping (AFH) mechanism to smooth the coexistence among incompatible MAC/PHY protocols, such as Bluetooth/ZigBee and WiFi. However, AFH is ineffective at WiFi hotspots where the entire 2.4GHz spectrum is congested by multiple WLAN cells configured to orthogonal channels. AFH also incurs substantial overhead to a ZigBee WPAN, as the network coordinator needs to scan the entire 16 channels and re-establish connections with clients. This problem becomes more pronounced in a dynamic network with mobile WiFi nodes and bursty interference.

Alternatively, coexistence can be arbitrated in space by adjusting the transmit power and carrier sensing threshold. Gummadi *et al.* [55] proposed a policy framework that assigns such parameters to coexisting networks, so as to minimize mutual interference. This framework requires an arbitrator that can communicate with different network devices. It is only applicable to static networks, as any node movement would require the arbitrator to re-initiate a spectrum survey and re-allocate the parameters.

Another approach, called WISE [62], aims to enhance coexistence in the temporal domain. WISE harnesses the white spaces between WiFi transmissions, and opportunistically schedules ZigBee traffic therein. However, WISE needs to suspend ZigBee transmissions during each WiFi burst. It is unsuitable for TDMA mode, and for delay-sensitive applications.

To the best of our knowledge, CBT is the first attempt that allows ZigBee to coexist and even contend with WiFi in frequency, spatial and temporal domains. Our key observation is that a sufficient idle channel time exists and can be exploited by ZigBee, but the WiFi's unawareness of ZigBee causes severe collisions. By enhancing the visibility of ZigBee to WiFi while preserving the carrier-sensing-based spectrum etiquette, CBT can substantially improve channel utilization without compromising WiFi performance.

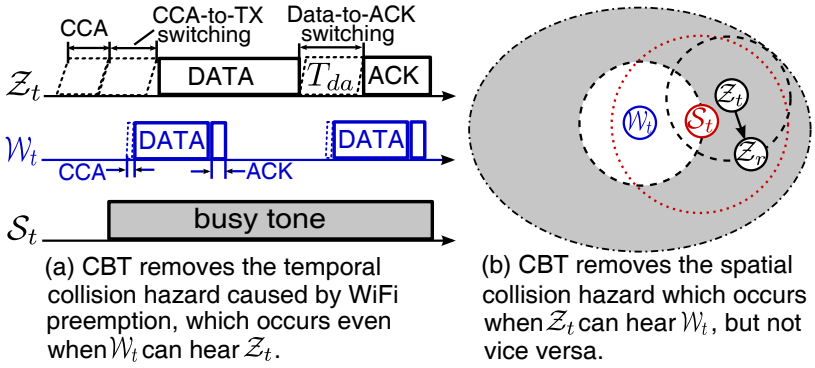

Figure 4.1: Principles behind CBT. $\mathcal{Z}_t$, $\mathcal{Z}_r$, $\mathcal{S}_t$, and $\mathcal{W}_t$ are the ZigBee transmitter, receiver, signaler, and WiFi transmitter, respectively.

This work is also the first that establishes a comprehensive analytical framework to quantify the performance of coexisting ZigBee and WiFi networks. Our analysis is inspired by the pioneering efforts on renewal process models for 802.11 and 802.15.4 MAC protocols [78, 116]. The key challenge lies in modeling the disparate MAC-layer operations. Using reasonable simplifications, our analysis can accurately capture different performance metrics, such as collision probability and throughput. The results are also used to balance the cost and effectiveness of CBT.

4.3 Cooperative Busy Tone (CBT)

In this section, we present the key principles and components of CBT. CBT is built atop the ZigBee MAC/PHY, but adopts an innovative way of signaling a busy channel to WiFi. It employs a separate ZigBee node (signaler) to emit a busy tone concurrently with the desired data transmission, thereby eliminating the following collision hazards induced by MAC/PHY heterogeneity.

Temporal collision hazards Due to their disparate time resolutions, ZigBee transmissions may be easily preempted by WiFi transmissions. ZigBee takes 128 μs to perform CCA (clear channel assessment), and an additional 192 μs to switch from the CCA to transmission mode, and even longer from receiving a packet to sending

the ACK [4]. In contrast, WiFi nodes take only 28 μs for CCA and an average of 72 μs for a backoff (with the default backoff window size in 802.11a/g/n) [5]. Therefore, a WiFi node may finish the entire backoff process and start transmission within the switching time of ZigBee, thus causing collision (Fig. 4.1(a)). CBT reduces such temporal collision hazards by allowing the signaler to emit a busy tone, which is long enough to cover the data packet, the switching time and the ACK packet. It starts the busy tone before the actual data transmission and carrier sensing, in order to "reserve" the channel and prevent WiFi preemption.

Spatial collision hazards Due to their disparate power levels (-25 to 0dBm for ZigBee vs. 15 to 20dBm for WiFi), ZigBee signals may not be effectively sensed by WiFi. As illustrated in Fig. 4.1(b), there exists a "gray region" where ZigBee can hear WiFi, but WiFi is oblivious of ZigBee and may thus interrupt it arbitrarily. To combat such spatial collision hazards, CBT allows the ZigBee node close to WiFi interferers (or a dedicated high-power ZigBee node such as XBee [37]) to work as the signaler, by transmitting a busy tone synchronously, thus notifying WiFi to suspend its transmission.

An immediate challenge to the above principles is: how to prevent the signaler from interfering with the transmitter, and how to synchronize the busy-tone and data transmission? We resolve these challenges using a *frequency flip* scheme and a *busy tone scheduler*.

4.3.1 Frequency Flip

The frequency flip exploits the inherent spectrum heterogeneity between ZigBee and WiFi. On the 2.4GHz ISM band, each WiFi channel occupies 22MHz, and overlaps with 4 orthogonal ZigBee channels. When running the frequency flip, the signaler hops to an adjacent channel before starting the busy tone, and hops back to

the original channel immediately after the busy tone is transmitted. This way, CBT ensures the busy tone is orthogonal to the data packet, but still overlaps with the WiFi channel and can cause it to defer transmission.

Frequency flip incurs overhead to the signaler due to channel switching. However, the switching time is limited to 192 μs in ZigBee [4], and can be overlapped with the CCA-to-TX switching time (Fig. 4.1(a)). CBT assumes WiFi will defer when the ambient signal level is above its CCA threshold. This is a mandatory operation for 802.11a/g/n [5, Sec. 17.3.10.5]. However, CBT may become ineffective when it coexists with 802.11b, which can be configured to defer only for valid WiFi signals [5, Sec. 18.4.8.4].

4.3.2 Busy Tone Scheduler

In a ZigBee WPAN, a unique *coordinator* schedules a mixture of TDMA and CSMA slots periodically. Each scheduling period (called a *superframe*) starts with a beacon, followed by a number of CSMA slots and TDMA slots and then an inactive period.

CBT maintains the legacy scheduling protocol, but requires the signaler to dispatch the busy tone at an appropriate time, such that: *i)* it reduces the WiFi preemptions of ongoing or forthcoming ZigBee transmissions and *ii)* it minimizes the potential influence on WiFi performance. The busy tone scheduler is designed to address this tradeoff. It allows both the TDMA and CSMA mode of ZigBee to coexist with WiFi.

4.3.2.1 TDMA scheduler

Fig. 4.2(a) illustrates the procedure to send a TDMA packet in CBT. CBT maintains the original TDMA slot allocation mechanism in ZigBee, but ensures the start time of each slot is conveyed to the signaler as well as the target clients, through

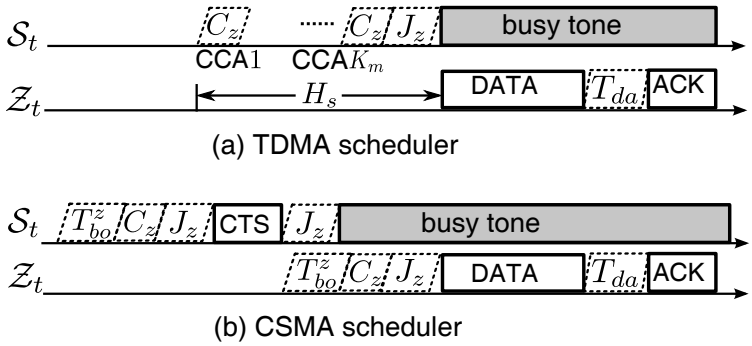

Figure 4.2: CBT scheduler. T^z_{bo} denotes the backoff time; $C_z(128\mu s)$ is the CCA duration; $J_z(192\mu s)$ the CCA-to-tx switching time (or channel switching time); T_{da} the data-to-ACK switching time.

persistent transmissions from the coordinator. Before each slot, the signaler performs CCA (for at most K_m times) in order to avoid interfering with WiFi. At the first idle CCA, it runs the frequency flip, switches to the adjacent channel, and starts the busy tone immediately. The busy tone lasts from the first idle CCA to the end of the TDMA slot. In this way, both the data and ACK packets can be protected from WiFi interruption.

A key parameter in the TDMA scheduler is the *harbinger time* H_s, defined according to how early the signaler starts the first CCA ($H_s = K_m C_z + J_z$, following Fig. 4.2(a)). If H_s is too long, the busy tone may occupy an unduly amount of channel time, thus reducing the channel utilization. If H_s is too short, the signaler may not be able to identify an idle slot before the scheduled transmission. Thus, it often has to abort the busy tone, degrading the effectiveness of CBT. In Sec. 4.4.2, we balance this tradeoff using a model-driven approach that relates H_s to network performance.

4.3.2.2 CSMA scheduler

Fig. 4.2(b) shows CBT's operations in CSMA mode. Each CSMA transmission is initiated by the signaler, which performs CCA and backoff just as a normal client. Upon detection of an idle CCA, the signaler broadcasts a notification message (referred to as *CTS*) to the clients, switches to the adjacent channel and starts emitting

the busy tone, as specified by the frequency flip. After receiving the CTS, the clients will contend for the channel access, following the same specification as in legacy ZigBee. However, if a client fails to acquire the channel, it needs to wait for the next CTS.

In designing the CSMA scheduler, we assume the signaler is able to obtain a rough estimate of the CSMA traffic demand in each superframe. This can be achieved by allowing the clients to report to the coordinator the number of pending packets in the current superframe. The coordinator then conveys the aggregated amount of unsatisfied traffic demand to the signaler, who then adjusts the number of CTS attempts in the next superframe.

Note that the CTS preceding the actual CSMA data induces extra overhead. However, the CTS has equal length with an ACK packet, which contains only 5 bytes payload (11 bytes in total if the PHY layer preamble is included), much smaller than a typical data packet size. To further reduce the overhead, we adopt a *busy tone aggregation* scheme that allows $G > 1$ packets to be sent following the CTS, *i.e.*, each client can participate in G channel access contention upon receiving a CTS.

Since a data packet must follow the CTS from the signaler, there does not exist a harbinger time as in TDMA mode. However, the signaler needs to determine the busy tone duration, so that it covers the data and ACK with high probability, even after the random channel access among the contenders. Undoubtedly, setting the busy tone duration to the maximum backoff window plus the data and ACK duration would ensure full coverage. However, it may also waste channel time since the winning contender's backoff duration is random, and likely to be smaller than the maximum. In Sec. 4.4.3, we derive the busy tone duration that probabilistically makes this tradeoff.

It should be noted that CBT cannot completely remove the temporal collision hazards, because the signaler has the same time-resolution as a normal ZigBee node,

and may be preempted by WiFi before sending the busy-tone. However, CBT can significantly reduce the collision hazards by augmenting the signaler's CCA capability in the ZigBee's TDMA mode, and by combining the signaler's CCA with the normal CCA/backoff in the ZigBee's CSMA mode. Its potential benefits will become clear as shown in our analysis and experimentation below.

4.4 Performance Analysis and Optimization

In this section, we establish a theoretical framework to analyze the performance of CBT in comparison with the legacy ZigBee. Our analysis pinpoints the key design parameters that affect the effectiveness of CBT in improving channel utilization while causing minimal interference to WiFi.

4.4.1 Network Model

We consider a ZigBee WPAN co-located with a WiFi WLAN, both sharing the same spectrum, and adopting the energy sensing based CCA as a spectrum etiquette. We mainly focus on the case with unsaturated WiFi and ZigBee links. As we will clarify, the saturated WiFi traffic results in almost zero throughput for the legacy ZigBee, and is less relevant for coexistence analysis. We assume the packet arrival follows a Poisson distribution. With unsaturated traffic, the aggregated traffic pattern is still approximately Poisson [35]. Hence, it is reasonable to deem the aggregated traffic as coexisting transmissions between one ZigBee and WiFi link. The Poisson assumption here is used for analytical tractability. The rationale behind parameter optimization (*e.g.*, the busy tone duration) does not depend on the traffic pattern.

We introduce the following notations beside those in Fig. 4.1 and Fig. 4.2. τ_w and τ_{wa} denote $\mathcal{W}_t$'s data and ACK packet duration, respectively. T_w and λ_w denote the data packets' mean inter-arrival time and arrival rate ($T_w = \lambda_w^{-1}$). T_{bo}^w is the backoff duration, uniformly distributed between 0 and the backoff window (which may grow

from CW^w_{min} to CW^w_{max}). After a backoff, WiFi must ensure the channel is idle for $DIFS(28\mu s)$ before transmission. β_w denotes the duration from backoff until an ACK when channel is idle. τ_z, τ_{za}, T_z, λ_z, T^z_{bo} are the corresponding parameters for $\mathcal{Z}_t$. Further, we denote γ_z as ZigBee's data/ACK duration (including the switching time between them, *i.e.*, T_{da}), and thus $\gamma_z = \tau_z + T_{da} + \tau_{za}$. U_z is ZigBee's slot duration ($U_z = 320\mu s$ [4]) and R_z the retransmission limit of a packet (default to 3 [4]).

We use the $\subset$ notation to denote the observability between transmitters. We assume $\mathcal{S}_t$ is a high-power, ZigBee compatible node (*e.g.*, [37]), and $\mathcal{S}_t \subset \mathcal{W}_t$, *i.e.*, the $\mathcal{S}_t$'s busy-tone can be sensed by $\mathcal{W}_t$. Moreover, we assume the CTS packet from $\mathcal{S}_t$ will capture WiFi's packet even when collision occurs. Since $\mathcal{Z}_t$ has around 20dBm lower power than $\mathcal{W}_t$, we assume the common case where collision affects $\mathcal{Z}_t$'s packets, but not $\mathcal{W}_t$'s. These assumptions will be further justified in Sec. 4.5.1.3.

Our analysis incorporates both the TDMA and CSMA mode, for both the legacy ZigBee and CBT, considering both $\mathcal{Z}_t \not\subset \mathcal{W}_t$ and $\mathcal{Z}_t \subset \mathcal{W}_t$. The primary method is to derive the collision probability in each case, and then relate it to the network's performance metric. We first analyze the *temporal collision probability* (Sec. 4.4.2 and Sec. 4.4.3), *i.e.*, probability that packets from both networks overlap with each other, thus causing collision. Later in Sec. 4.4.4 we analyze the *spatial collision probability*, probability that overlapped packets (from *randomly* located transmitters) fail to be decoded, taking into account the capture effect. We focus on each network's normalized throughput as the performance metric, denoted as Γ_z (Γ_w), which is essentially the ratio between the data packet duration and the average packet service time (including CCA, backoff, ACK, and retransmissions).

4.4.2 ZigBee's TDMA Coexistence with WiFi

4.4.2.1 Collision probability

As $\mathcal{Z}_t$ usually runs in low duty-cycle mode ($T_z \gg T_w$), we can tag an arbitrary packet from $\mathcal{Z}_t$, and observe the collision with $\mathcal{W}_t$. For simplicity, we introduce the concept of *vulnerable period*. A $\mathcal{W}_t$ packet arrival within the vulnerable period will overlap with the tagged packet from $\mathcal{Z}_t$, resulting in collision. Let v be the duration of vulnerable period, then the collision probability becomes:

$$1 - \mathbb{P}[\text{no WiFi packet arrival in } v] = 1 - e^{-v\lambda_w}. \tag{4.1}$$

The following analysis derives the collision probability by analyzing v, depending on whether $\mathcal{W}_t$ can sense $\mathcal{Z}_t$ and CBT is adopted or not.

Case 1: Legacy ZigBee, $\mathcal{Z}_t \not\subset \mathcal{W}_t$. Collision occurs whenever $\mathcal{W}_t$ starts or ends its transmission within $\mathcal{Z}_t$'s packet duration. Hence, the vulnerable period is $\beta_w + \tau_z$ for $\mathcal{Z}_t$'s data packet, and $\beta_w + \tau_{za}$ for ACK packet. β_w involves a random variable T^w_{bo}, but for tractability, we approximate it with the mean $\overline{T}^w_{bo}$. When unsaturated, WiFi's backoff window tends to stay in CW^w_{min}. Hence, $\overline{T}^w_{bo} \approx \frac{1}{2} CW^w_{min}$. The data/ACK collision probability (denoted as P^d_1 and P^a_1, respectively) is readily obtained by following Eq. (4.1):

$$P^d_1 = 1 - e^{-\lambda_w(\beta_w + \tau_z)} \tag{4.2}$$

$$P^a_1 = 1 - e^{-\lambda_w(\beta_w + \tau_{za})} \tag{4.3}$$

Case 2: Legacy ZigBee, $\mathcal{Z}_t \subset \mathcal{W}_t$. In this case, $\mathcal{W}_t$ defers transmission if it senses $\mathcal{Z}_t$'s packets. Hence, collision occurs only if $\mathcal{Z}_t$'s packet starts while $\mathcal{W}_t$ is transmitting. Equivalently, at least one packet of $\mathcal{W}_t$ arrives within a vulnerable period β_w before $\mathcal{Z}_t$ starts.

In addition, $\mathcal{Z}_t$'s ACK packet is corrupted under two conditions: *i)* $\mathcal{W}_t$ starts its backoff and CCA within the data-to-ACK switching time T_{da} of $\mathcal{Z}_t$, corresponding to a vulnerable period $T_{da} - (\overline{T}_{bo}^{w} + DIFS)$. In slotted CSMA mode, T_{da} ranges from J_z to $U_z + J_z$, depending on the data packet size [4]. $(\overline{T}_{bo}^{w} + DIFS)$ is $\mathcal{W}_t$'s mean backoff plus defer time preceding each data packet. *ii)* The tail part of $\mathcal{W}_t$'s packet overlaps $\mathcal{Z}_t$'s ACK, corresponding to a vulnerable period β_w. Overall, the vulnerable period for the ACK packet is the minimum for condition *i* and *ii*. Again, following Eq. (4.1), the data/ACK collision probability for case 2 is:

$$P_2^d = 1 - e^{-\lambda_w \beta_w} \tag{4.4}$$

$$P_2^a = 1 - e^{-\lambda_w \cdot \min\{T_{da} - (\overline{T}_{bo}^{w} + DIFS), \beta_w\}} \tag{4.5}$$

It is easy to see that Case 2 has a shorter vulnerable period for both data and ACK packets, and thus $P_2^d < P_1^d$ and $P_2^a < P_1^a$, which agrees with the intuition that collision is reduced when $\mathcal{W}_t$ can sense $\mathcal{Z}_t$.

Case 3: CBT is adopted. When running CBT, a key problem is how the signaler's harbinger time H_s (Sec. 4.3.2.1) affects CBT's performance. The follow analysis derives the relation between H_s and the failure rate of busy tone.

Proposition IV.1. *When the harbinger time $H_s = K_m C_z + J_z$, the probability that CBT fails to send the busy tone is: $P_{b0} = P_b(1 - P_{i|b})^{K_m - 1}$, where P_b and $P_{i|b}$ are defined in Eq.* (4.6) *and* (4.10).

Proof. The probability that the first CCA returns busy, denoted as P_b, can be approximated by the fraction of time that the channel is occupied by $\mathcal{W}_t$, *i.e.*,

$$P_b = \gamma_w T_w^{-1} = \lambda_w \gamma_w, \tag{4.6}$$

where $\gamma_w \triangleq \beta_w - \overline{T}_{bo}^{w} - DIFS$ is equivalent to the total airtime of $\mathcal{W}_t$'s packet. Subsequent CCA events are complicated because they are correlated throughout $\mathcal{W}_t$'s

airtime. Let $P_{i|b}$ be the probability that a subsequent CCA returns idle conditioned on the fact that the previous one is busy. This event occurs if the previous CCA falls in the tail of one WiFi packet, and the second CCA falls in the gap to the next WiFi packets. The gap, denoted as I_w, is a random variable that depends on $\mathcal{W}_t$'s backoff and inter-arrival time, which needs to be examined first.

Since $\mathcal{Z}_t$ runs in low duty-cycle, it is reasonable to assume the busy/idle state created by $\mathcal{W}_t$ is independent of $\mathcal{Z}_t$. If additional WiFi packets arrive during the service time of one packet (with probability P_q), then consecutive transmissions are separated solely by the backoff and defer times. It follows immediately that $I_w \sim U(DIFS, D_m)$, where D_m is the maximum backoff plus defer time preceding a transmission ($D_m = CW^w_{min} + DIFS$). Otherwise, I_w equals the inter-arrival time minus the duration of the previous transmission. In summary, we have:

$$\mathbb{P}[I_w \leq t] = \begin{cases} 1 - e^{-\lambda_w(\tau_w + t)}, & t > D_m \\ \frac{P_q t}{D_m - DIFS}, & DIFS < t \leq D_m \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

We proceed to derive P_q, the backlog probability. The evolution of $\mathcal{W}_t$'s state can be modelled as an $M/D/1$ queue, with arrival rate λ_w and service time β_w. Let P_{qi} be the probability that i packets are held by $\mathcal{W}_t$ (one in service and others queued), then from well-established results in queuing theory [26], we have: $P_{q0} = 1 - \lambda_w \beta_w$, $P_{q1} = (e^{\lambda_w \beta_w} - 1)P_{q0}$, and $P_q = 1 - P_{q0} - P_{q1}$.

Back to the conditional probability $P_{i|b}$, we have:

$$P_{i|b} = \int_0^{C_z} \beta_w^{-1}(1 - \mathbb{P}[I_w \leq t])\mathrm{d}t \quad (\text{Note:} C_z < D_m) \tag{4.8}$$

$$= \int_0^{C_z} \beta_w^{-1}(1 - P_q(C_z - t)(D_m - DIFS)^{-1})\mathrm{d}t \tag{4.9}$$

$$= C_z\beta_w^{-1} - 0.5P_qC_z^2(D_m - DIFS)^{-1}\beta_w^{-1} \tag{4.10}$$

$$\approx C_z\beta_w^{-1} - 0.5P_qC_z^2D_m^{-1}\beta_w^{-1} \tag{4.11}$$

Consequently, the probability of aborting busy tone after K_m CCA attempts is: $P_{b0} = P_b(1 - P_{i|b})^{K_m - 1}$. □

Conditioned on the fact that CCA succeeds and the busy tone is sent, $\mathcal{W}_t$ may still preempt in the switching time of the signaler (Fig. 4.2(a)), with probability P_x:

$$P_x = \int_0^{C_z} C_z^{-1}\mathbb{P}[t \leq I_w \leq t + J_z]\mathrm{d}t \tag{4.12}$$

$$= C_z^{-1} \int_0^{C_z} P_q(D_m - t)D_m^{-1}\mathrm{d}t \approx \frac{1}{2}P_qC_z^{-1}D_m \tag{4.13}$$

Note that WiFi preemption results in collision only if the preemption time is after the K_s-th CCA, where $K_s = K_m - \lfloor(\beta_w - J_z)C_z^{-1}\rfloor$ is the time beyond which the end of $\mathcal{W}_t$'s packet always overlaps with $\mathcal{Z}_t$'s.

Summarizing the above analysis, the collision probability for data packets when using CBT is:

$$P_3^d = P_{b0}\,\mathbb{P}[\text{collision}|\text{CCA fail}]$$
$$+ \mathbb{P}[\text{CCA succeeds at } k, k > K_s]P_x \tag{4.14}$$

$$= P_bP_{b|b}^{K_m - 1}P_i^d + (1 - P_bP_{b|b}^{K_s})P_x \tag{4.15}$$

where $P_{b|b} = 1 - P_{i|b}$. The ACK packet can only be corrupted if CCA fails, exposing

the ACK to WiFi collision, just as the previous two cases. Hence,

$$P_3^a = P_{b0}P_i^a = P_b P_{b|b}^{K_m-1} P_i^a, i \in \{1,2\} \tag{4.16}$$

4.4.2.2 Network performance

Based on the above analysis, we can derive the throughput of $\mathcal{Z}_t$ for all 3 cases using a renewal model. We model the transmission attempts by $\mathcal{Z}_t$ as a renewal reward process. Each *renewal interval* is the service time of a packet, which starts with a transmission attempt, and ends with a successful ACK, or with a transmission failure if the retry limit R_z is exceeded. The *reward* is the amount of time in transmitting data without collision. Hence, the mean reward equals $[1-(1-P_{si})^{R_z}]\tau_z$, where P_{si} is the probability that both data and ACK are successfully delivered in case i (thus $P_{si} = (1-P_i^d)(1-P_i^a)$). The resulting throughput of $\mathcal{Z}_t$ equals the reward rate Γ_{zi}. Let T_{zi}^{sv} be the service time of a packet in case i, then:

$$\Gamma_{zi} = [1-(1-P_{si})^{R_z}]\tau_z(\overline{T}_{zi}^{sv})^{-1}, i \in \{1,2,3\} \tag{4.17}$$

The mean service time $\overline{T}_{zi}^{sv}$ depends on P_{si}, and the duration of each transmission attempt, denoted as T_i. For legacy ZigBee, we have $T_1 = T_2 = \gamma_z$. For CBT, from Fig. 4.2(a), we obtain $T_3 = U_z K_m + J_z + \gamma_z$. Further, note in all the cases, the first transmission attempt occurs with probability 1, and the second occurs only if the first fails. Following this reasoning, the expected service time:

$$\overline{T}_{zi}^{sv} = \mathbb{E}[T_{zi}^{sv}] = \sum_{k=0}^{R_z-1} T_i(1-P_{si})^k \tag{4.18}$$

Next, consider the performance of WiFi. In case 1 ($\mathcal{Z}_t \not\subset \mathcal{W}_t$), $\mathcal{W}_t$ is unaffected by $\mathcal{Z}_t$, thus its mean service time $\overline{T}_{w1}^{sv} = \beta_w$. In case 2 ($\mathcal{Z}_t \subset \mathcal{W}_t$), $\mathcal{W}_t$'s arrival overlaps $\mathcal{Z}_t$'s data or ACK packet, with probability $1 - e^{-\lambda_z(\tau_z+\tau_{za})} \approx \lambda_z(\tau_z + \tau_{za})$. Since $\mathcal{W}_t$

freezes its backoff upon a busy CCA, the corresponding service time equals β_w plus the extra time that it has to wait until $\mathcal{Z}_t$ finishes transmission. Otherwise, its service time remains β_w. Therefore, the mean service time is:

$$\begin{aligned}\overline{T}_{w2}^{sv} &= \beta_w[1-\lambda_z(\tau_z+\tau_{za})] \\ &\quad + \lambda_z\tau_z(\frac{\tau_z}{2}+\beta_w) + \lambda_z\tau_{za}(\frac{\tau_{za}}{2}+\beta_w)\end{aligned} \tag{4.19}$$

When CBT is used (case 3), its CCA succeeds at the i-th attempt with probability $(1-P_b)$ for $i=1$, and $P_bP_{b|b}^{i-2}P_{b|i}$ otherwise. Since each CCA attempt takes one slot, the busy tone duration is $\tau_b = U_z(K_m-i)+J_z+\tau_z+T_{da}+\tau_{za}$. The mean duration of a transmission attempt in CBT is thus:

$$\begin{aligned}\Phi_b =& (1-P_b)(C_z(K_m-1)+\tau_b) \\ &+ \textstyle\sum_{i=2}^{K_m} P_bP_{b|b}^{i-2}P_{i|b}(C_z(K_m-i)+\tau_b) \\ &+ P_bP_{b|b}^{K_m-1}(\frac{\tau_z}{\tau_z+\tau_{za}}\tau_z + \frac{\tau_{za}}{\tau_z+\tau_{za}}\tau_{za})\end{aligned} \tag{4.20}$$

and the WiFi service time can be approximated as:

$$\overline{T}_{w3}^{sv} = (1-\lambda_z\Phi_b)\tau_w + \lambda_z\Phi_b(\frac{\Phi_b}{2}+\tau_w) \tag{4.21}$$

Following a similar renewal model for $\mathcal{Z}_t$, the throughput of $\mathcal{W}_t$ for the 3 cases is:

$$\Gamma_{wi} = \tau_w(\overline{T}_{wi}^{sv})^{-1}, i \in \{1,2,3\} \tag{4.22}$$

4.4.3 ZigBee's CSMA Coexistence with WiFi

4.4.3.1 ZigBee's legacy CSMA mode (Case 4)

We model ZigBee's CSMA mode using a Markov chain shown in Fig. 5.8, where BS_k denotes the k-th backoff and CCA attempt. Similar to the TDMA analysis, we

make a key simplification that decouples the channel status from $\mathcal{Z}_t$'s transmission, *i.e.*, assuming the busy/idle state of WiFi is unaffected by $\mathcal{Z}_t$. This assumption will be removed when analyzing the effects of $\mathcal{Z}_t$ on $\mathcal{W}_t$.

In CSMA mode, $\mathcal{Z}_t$ must perform backoff and ensure two consecutive slots (CCA1 and CCA2) are idle before transmission, and abort the transmission if sensing a busy channel even after K attempts. Straightforwardly, the steady state probability that $\mathcal{Z}_t$ senses an idle channel in CCA1 is: $P_i = 1 - P_b$ (see Eq. (4.6) for P_b). Conditioned on the event that CCA1 is idle, CCA2 returns idle if no packets for $\mathcal{W}_t$ arrive between them, which has a probability: $P_{i|i} = e^{-\lambda_w U_z}$. Let P_{tx} denote the transmission probability after a backoff and CCA attempt, then:

$$P_{tx} = P_i P_{i|i} = (1 - P_b)e^{-\lambda_w U_z} \tag{4.23}$$

The duration of the k-th backoff attempt is uniformly distributed between 0 and [4]:

$$B_k = \min\{2^{minBE+k-1}, 2^{maxBE}\} \cdot U_z, k \in [1, K] \tag{4.24}$$

where *minBE* and *maxBE* are the minimum and maximum backoff exponent, default to 3 and 5, respectively. Note that CCA1 must be performed, but CCA2 is needed only if CCA1 returns idle. The mean time spent in a CCA attempt is thus: $U_z + P_i U_z = (1 + P_i)U_z$.

Given the above components, we can derive $\mathcal{Z}_t$'s service time. From the Markov chain model, the mean first passage time to data transmission is:

$$F_{TX} = \frac{b_1 + (1 - P_{tx})b_2 + \cdots + (1 - P_{tx}^{K-1})b_K}{1 - (1 - P_{tx})^K} \tag{4.25}$$

where $b_k = 0.5B_k + (1 + P_i)U_z$ is the expected duration of the k-th backoff plus

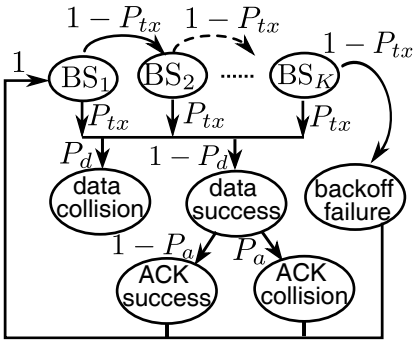

Figure 4.3: Markov chain model for analyzing ZigBee's throughput when it coexists with WiFi.

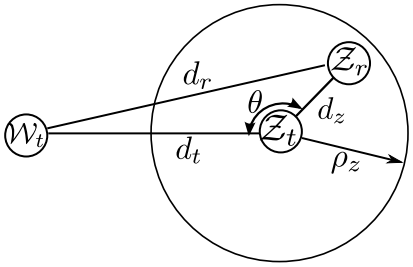

Figure 4.4: Analyzing the spatial collision probability under random link locations.

CCA attempt (BS_k). Intuitively, the numerator of Eq. (4.25) is the mean duration of all the attempts until data transmission or backoff failure. The number of attempts follows a geometrical distribution with success probability $1-(1-P_{tx})^K$. Consider $\mathcal{Z}_t \subset \mathcal{W}_t$, and let P_{s4} be the probability that both data and ACK are transmitted without collision, then the expected number of trials until transmission succeeds or retry failure is:

$$1+(1-P_{s4})+\cdots+(1-P_{s4})^{R_z-1}=\sum_{k=0}^{R_z-1}(1-P_{s4})^k$$

Each transmission takes F_{TX} plus the data and ACK (or timeout) duration γ_z. Thus the mean service time:

$$\overline{T}_{z4}^{sv}=(F_{TX}+\gamma_z)\sum_{k=0}^{R_z-1}(1-P_{s4})^k \tag{4.26}$$

Note that WiFi may preempt the CCA-to-TX and data-to-ACK switching time, with probability P_d and P_a, respectively (Fig. 5.8). Following similar analysis of the vulnerable period in the TDMA case 2, we obtain $P_d=1-e^{\lambda_w J_z}$, and $P_a=P_2^a$, from which P_{s4} can be derived. In case when $\mathcal{Z}_t \not\subset \mathcal{W}_t$, the success probability is equivalent to P_{s1}, and mean service time can be obtained in the same way as Eq. (4.26).

4.4.3.2 CBT's CSMA mode (Case 5)

The key problem in this case is to determine the busy tone duration T_b, which should cover the entire data transmission without wasting extra channel time. The following analytical results state that this can be realized with probabilistic guarantee.

Proposition IV.2. *For CBT with N clients and a single busy tone, with duration $T_b = \frac{kB_1}{N} + \gamma_z$, the busy tone covers the data and ACK packets with probability of at least $1 - e^{-k}$. For CBT with G aggregated busy tones, with duration $T_b^{'} = \frac{kGB_1}{N+1} + \gamma_z$, the busy tone covers all the packets with probability of at least $1 - \frac{1}{k}$.*

Proof. Let t_i be the backoff counter set by node i, $t_i \sim U(0, B_1)$. The probability that the busy tone covers the data/ACK duration of the winning contender is:

$$\mathbb{P}\Big[\min_i t_i + \gamma_z \leq T_b\Big] = \mathbb{P}\Big[\min_i t_i \leq T_b - \gamma_z\Big] \tag{4.27}$$

$$= 1 - (1 - \frac{T_b - \gamma_z}{B_1})^N \quad (\text{Let } T_b = \frac{kB_1}{N} + \gamma_z) \tag{4.28}$$

$$= 1 - (1 - \frac{k}{N})^N \geq 1 - e^{-k} \tag{4.29}$$

For the case with busy tone aggregation, we first derive the expectation of the minimum backoff counter among all the clients, denoted as M.

$$\mathbb{E}[M] = \int_0^{B_1} [1 - F_M(x)]\mathrm{d}x \qquad (\textit{cf.}\ [47, \text{Sec. } 1.3.5]) \tag{4.30}$$

$$= \int_0^{B_1} (1 - \frac{x}{B_1})^N \mathrm{d}x = \frac{B_1}{N+1} \tag{4.31}$$

Further, note that:

$$\mathbb{P}\Big[\gamma_z + \sum_{i=1}^{G} b_i \geq T_b^{'}\Big] \tag{4.32}$$

$$\leq \frac{\mathbb{E}\Big[\sum_{i=1}^{G} b_i\Big]}{T_b^{'} - \gamma_z} \quad \text{(Markov inequality)} \tag{4.33}$$

$$= \frac{G \cdot B_1}{N+1} \cdot \frac{1}{T_b^{'} - \gamma_z} \quad (\text{Let } T_b^{'} = \frac{kG \cdot B_1}{N+1} + \gamma_z) \tag{4.34}$$

$$= k^{-1} \tag{4.35}$$

Hence $\mathbb{P}[\sum_{i=1}^{G} b_i \leq T_b^{'}] \geq 1 - \frac{1}{k}$. □

Note that the above result is independent of the WiFi arrival time distribution. It implies that the busy tone needs extra number of slots (denoted as K_b) to compensate for the random backoff, *i.e.*, $T_b = K_b U_z + \gamma_z$. If the busy tone covers the backoff duration B_1 plus the data/ACK duration: $T_b^* \triangleq G(K_b^* U_z + \gamma_z), K_b^* = 2^{minBE}$, then collision rate can be reduced to 0.

When $G = 1$ and $T_b < T_b^*$, the end of the ACK packet may still be exposed to WiFi collision. Suppose $\mathcal{Z}_t \subset \mathcal{W}_t$, then this occurs when the data-to-ACK switching time is beyond the protection of the busy tone. Let $u = \tau_z + T_{da} - \overline{T}_{bo}^{w}$, then the ACK collision probability can be derived similarly to the TDMA mode:

$$P_5^a = \mathbb{P}\Big[\min_i b_i + u \geq T_b\Big](1 - e^{-\lambda_w(\min_i b_i + u)}) \tag{4.36}$$

$$\approx (1 - B_1^{-1}(T_b - u))^N (1 - e^{-\lambda_w(\frac{B_1}{N+1} + u)}) \tag{4.37}$$

Given T_b, we can further derive the service time, denoted as T_{z5}^{sv}. Note that the signaler's backoff and CCA operations are similar to the legacy case (Fig. 5.8), except that its CCA attempt always takes one slot U_z, and the transmission probability $P_{tx} = P_i$ and success probability of each transmission attempt $P_{s5} = 1 - P_5^d$. Therefore, the service time starts from the first backoff stage, and ends after the busy tone or

the ACK packet, whichever lasts longer. Denote T_{m5} as the expected duration of a transmission attempt after backoff and CCA succeed, then $T_{m5} = \max\{\frac{B_1}{N+1} + \gamma_z, T_b\}$. Similar to Eq. (4.25), the mean service time of a packet:

$$\overline{T}_{z5}^{sv} = (F_{TX} + T_{m5}) \sum_{k=0}^{R_z - 1} (1 - P_{s5})^k \tag{4.38}$$

Given the mean service time and collision probability in CSMA, the throughput readily follows from Eq. (4.17).

For the WiFi transmitter $\mathcal{W}_t$, the service time depends on the $\mathcal{Z}_t$'s traffic load. Consider the service time of each packet from $\mathcal{Z}_t$ as a renewal interval with length L_{zi}. Note that $\mathcal{W}_t$ tends to have higher priority than $\mathcal{Z}_t$, and thus its load determines the length of the L_{zi}: $L_{zi} = \min\{T_z, T_{zi}^{sv}\}, i \in \{4, 5\}$. In case 4, the mean number of transmission attempts by $\mathcal{Z}_t$ in its renewal interval is

$$N_{d4} = \sum_{r=1}^{R_z} \left[(1 - (1 - P_{tx})^K)(1 - P_{si})\right]^{r-1} \tag{4.39}$$

which is also the mean number of times that $\mathcal{W}_t$ defers its transmission. For case 5, since one busy tone for $\mathcal{Z}_t$ is sent in each renewal interval, $N_{d5} = 1$. For both cases, $\mathcal{W}_t$'s packets are disrupted with probability $P_{ri} = \frac{N_{di} T_w}{L_{zi}}, i \in \{4, 5\}$, resulting in mean service delay:

$$\overline{T}_{w4}^{sv} = (1 - P_{r4})\beta_w + P_{r4}(\tau_z + \beta_w) \tag{4.40}$$

$$\overline{T}_{w5}^{sv} = (1 - P_{r5})\beta_w + P_{r5}(T_{m5} - T_w + \beta_w) \tag{4.41}$$

The throughput can be derived similarly to the TDMA case (Eq. (4.22)).

4.4.4 Spatial Collision Probability

The above analysis focused on the collision probability between co-located WiFi and ZigBee. In practice, collision does not necessarily cause packet loss. Due to

the capture effect, the desired packet can still be decoded if its power is sufficiently higher than the interfering packet. Such opportunities depend on the relative location of WiFi and ZigBee links.

Consider a randomly located ZigBee link and WiFi interferer, as shown in Fig. 4.4. For simplicity, we denote $d_t = d(\mathcal{W}_t, \mathcal{Z}_t), d_z = d(\mathcal{Z}_t, \mathcal{Z}_r), d_r = d(\mathcal{W}_t, \mathcal{Z}_r)$, where the function $d(\cdot, \cdot)$ represents the distance between two nodes. We assume $\theta \sim U(0, 2\pi)$, and $d_z \sim U(0, \rho_z)$, where ρ_z is the maximum distance between $\mathcal{Z}_t$ and $\mathcal{Z}_r$. Since ZigBee and WiFi have similar receiver sensitivity (around -86dBm [5, 4]), we assume $\mathcal{W}_t$ and $\mathcal{Z}_t$ have the same carrier sensing threshold. We use a generic pathloss model, where the ratio between received power and transmit power is $cd_z^{-\alpha}$. The constant c depends on the propagation characteristics (*e.g.*, free-space or two-ray ground model), and α is the environment dependent pathloss exponent. Denote C_a as the capture threshold, Λ_z and Λ_w as the transmit power of $\mathcal{Z}_t$ and $\mathcal{W}_t$, respectively, then $\mathcal{Z}_r$ fails to decode the packet if:

$$\frac{\Lambda_z c d_z^{-\alpha}}{\Lambda_w c d_r^{-\alpha}} \leq C_a, \text{or equivalently, } \frac{d_r^2}{d_z^2} \leq \left(C_a \frac{\Lambda_w}{\Lambda_z}\right)^{\frac{2}{\alpha}} \tag{4.42}$$

The key to our analysis is to derive the probability that the two link distances satisfy the above collision condition, which we name as *spatial collision probability*. Following the cosine rule $d_r^2 = d_t^2 + d_z^2 - 2\cos\theta d_t d_z$, the spatial collision probability becomes:

$$\frac{d_r^2}{d_z^2} = \frac{d_t^2}{d_z^2} - 2\cos\theta \frac{d_t}{d_z} \leq \left(C_a \frac{\Lambda_w}{\Lambda_z}\right)^{\frac{2}{\alpha}} - 1 \triangleq c_1 \tag{4.43}$$

which can be transformed into:

$$(d_z + \frac{d_t \cos\theta}{c_1})^2 \geq \frac{d_t^2}{c_1} + \frac{d_t^2 \cos^2\theta}{c_1^2} = \frac{d_t^2}{c_1^2}(c_1 + \cos^2\theta) \approx \frac{d_t^2}{c_1}$$

The geometrical meaning of the above equation, combined with $d_z \sim U(0, \rho_z)$

(*i.e.*, $d_z^2 \leq \rho_z^2$), is the exclusion region of two circular areas. To further simplify, observe that $d_z \gg \frac{d_t \cos\theta}{c_1}$, and the random variable $\cos\theta$ has mean 0, hence the above two inequalities can be approximated as: $\rho_z^2 \geq d_z^2 \geq \frac{d_t^2}{c_1}$, which reduces the original circular intersection problem into a 1-dimensional, line-segment intersection problem (for a given d_t). Consequently, the exclusion region:

$$I_e = \max\{0, 1 - d_t c_1^{-\frac{1}{2}} \rho_z^{-1}\} \tag{4.44}$$

which is exactly the spatial collision probability. The above approximation will be verified in Sec. 4.5.1.3.

4.5 Experimental Results

This section compares the qualitative prediction of the analytical model with detailed simulation in ns-2. We find them matching well across a broad range of experiments. We further explore the design parameters of CBT using the analytical model, and demonstrate its performance in a real experimental testbed.

4.5.1 Simulation experiments

We have implemented the CBT protocol based on the ZigBee (CSMA) module in ns-2 (version 2.33). Following the IEEE 802.15.4 standard, we also developed a TDMA module for ZigBee. The main components of CBT, *i.e.*, the frequency flip and busy tone scheduler, are implemented on top of the TDMA/CSMA modules. The PHY-layer parameters are set to their default values. The WiFi module is configured consistently with the 802.11g standard. Our experiments first focus on the case where the two networks are in close proximity and can sense each other (Sec. 4.5.1.1, Sec. 4.5.1.2), and later explore the effects of link locations (Sec. 4.5.1.3, Sec. 4.5.1.4).

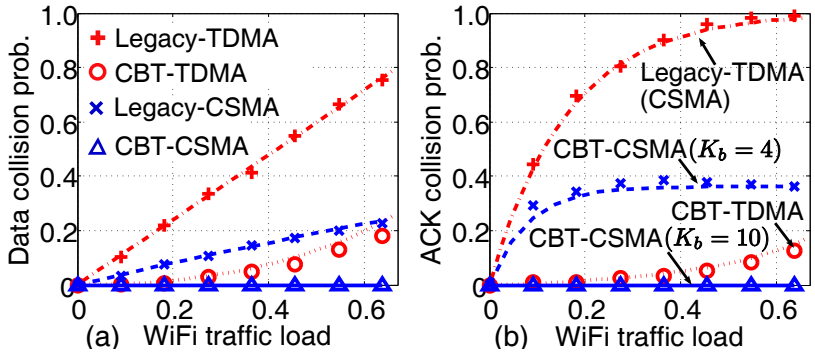

Figure 4.5: Collision probability of data and ACK packets. Markers and lines represent simulation and analytical results, respectively.

4.5.1.1 Temporal collision probability

We vary the *traffic load* of one network and observe its impacts on the other, where:

$$\text{Traffic load} = \frac{\text{packet size} \times \text{packet arrival rate}}{\text{PHY layer bit-rate}} \tag{4.45}$$

Throughout the experiments, ZigBee uses a fixed bit-rate of 250Kbps [4], packet size 63 bytes, and arrival rate 8 packets/second. WiFi bit-rate is set to 18Mbps, and packet size to 1KB. The corresponding saturation traffic load is around 0.67. Fig. 4.5(a) shows probability that ZigBee's data packets collides with WiFi packets, under varying WiFi traffic load. We see a close match between analysis and simulation, with a deviation of less than 1% for most experiments. In TDMA mode, coexisting WiFi traffic is detrimental to legacy ZigBee — the collision probability grows from 0 to above 0.79 as WiFi load increases from 0 to 0.67 (corresponding to the saturation load). With $K_m = 8$ (*i.e.*, harbinger time $H_s = 8C_z + J_z$), CBT reduces the collision probability to below 0.05 for medium to low WiFi traffic, and below 0.2 even when WiFi is saturated.

Measurement of WiFi hotspots observed traffic load of around 0.6 at peak hours [104]. Even under this level of interference, the data and ACK collision probability in CBT is 0.20 and 0.16, in contrast with 0.71 and 0.97 in legacy ZigBee, more than

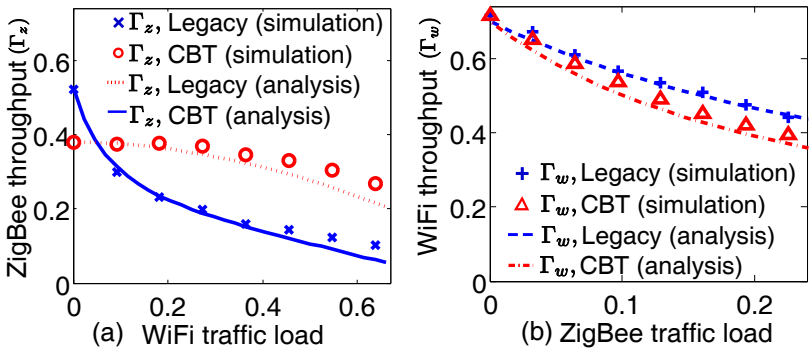

Figure 4.6: Throughput performance in TDMA mode.

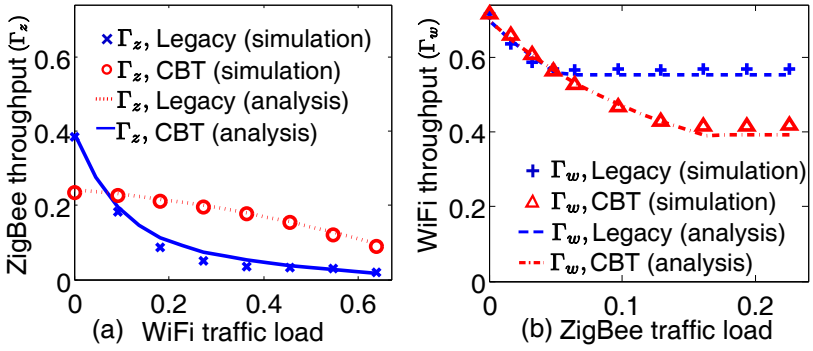

Figure 4.7: Throughput performance in CSMA mode.

72% reduction for both types of packet. With a low collision probability, packet loss can be easily recovered via retransmissions.

In CSMA mode, the CCA and backoff mechanism in legacy ZigBee alleviates the collision of data packets, but becomes ineffective when WiFi is heavily-loaded due to the preemption problem (Sec. 4.3). By preventing WiFi's preemption in ZigBee's CCA-to-TX switching period, CBT reduces the data collision probability to 0.

In addition, for legacy ZigBee, the ACK collision probability grows from 0 to 0.97 for both TDMA and CSMA mode (Fig. 4.5(b), only TDMA mode is plotted for clarity), implying a substantial number of redundant retransmissions due to ACK losses. For CBT, the probability is consistently below 0.2 in TDMA mode. In CSMA mode, when $K_b = 10$, the ACK collision probability is reduced to 0. As K_b is reduced to 4, the collision probability increases up to 0.4.

4.5.1.2 Throughput performance

Fig. 4.6 shows the throughput (Γ_z) of ZigBee in TDMA mode. Γ_z decreases as WiFi increases its load, because severe collision can cost extra service time, and lower the efficiency of channel utilization. When WiFi becomes saturated, ZigBee throughput is reduced to 0.09, 82% lower than the case without coexisting traffic. With CBT, throughput can be boosted by 47% to 112% when WiFi load is above 0.18. Notably, CBT may have lower throughput than ZigBee under mild interference, due to its CCA overhead. This implies CBT's benefit outweighs its overhead beyond a "sweet spot", which can be obtained by numerically solving for λ_w from the equation $\Gamma_{z2} = \Gamma_{z3}$ (see Eq. (4.17)). To optimize performance, the signaler can monitor the busy/idle status of WiFi and trigger CBT only when λ_w is beyond this point. We leave such optimization as future work.

Intuitively, the extra channel time taken by CBT will degrade WiFi's performance. However, the figure shows such effect is negligible when its traffic load is below 5%, and is comparable to legacy ZigBee even under heavy traffic. This is because CBT saves the retransmissions caused by collision, thereby counter-balancing the extra overhead. In practical ZigBee applications, the duty-cycle is typically below 1%, and up to 10% at its maximum [66]. Hence, in practice, CBT does not cause any additional performance degradation to WiFi performance.

Fig. 4.7 plots the throughput performance in CSMA mode. CBT (with $K_b = 10$ and without busy tone aggregation) consistently achieves a throughput of above 0.15, whereas the legacy ZigBee's throughput quickly drops below 0.03 as WiFi becomes saturated. CBT's advantage comes with the overhead from CTS and busy tone. Similar to the TDMA case, a "sweet spot" of WiFi load can be derived, beyond which the overhead is quickly outweighed by the saving in retransmission time, resulting in several folds performance improvement over the legacy ZigBee. The overhead can be further reduced by reducing K_b and enabling busy tone aggregation.

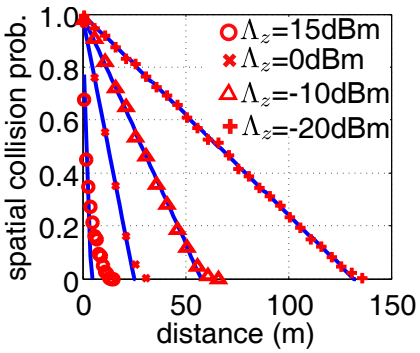

Figure 4.8: Spatial collision probability ($\Lambda_w = 15$dBm).

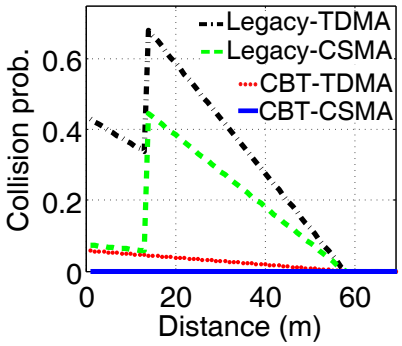

Figure 4.9: Spatial-temporal collision probability.

In addition, although both CBT and legacy reduce the throughput of WiFi, the reduction quickly converges to a certain level corresponding to the saturation throughput of ZigBee under the current WiFi traffic load. This again justifies the modeling assumption that WiFi has higher priority than ZigBee when both are running CSMA. The achievable throughput of ZigBee is determined by the load of coexisting WiFi traffic, but not vice versa. In addition, note that heavily loaded CBT degrades WiFi performance, but under practical low duty-cycle applications (load $< 5\%$), the degradation is negligible compared with the legacy ZigBee.

4.5.1.3 Spatial collision probability

We use Monte Carlo simulation to verify the analysis in Sec. 4.4.4. We first fix the locations of $\mathcal{W}_t$ and $\mathcal{Z}_t$, and then randomly uniformly generate the location of $\mathcal{Z}_r$ within the maximum range ρ_z (set to 6m) from $\mathcal{Z}_t$. Fig. 4.8(a) shows the spatial collision probability I_e for different d_t, each obtained from 10^4 samples. Consistent with the analytical results (solid lines), I_e decreases almost linearly with d_t. Beyond a certain threshold (as indicated in Eq. (4.44)), the ratio between the received power from $\mathcal{Z}_t$ and that from $\mathcal{W}_t$ is larger than the capture threshold (set to 10dB) with probability 1, and thus I_e becomes 0.

In addition, I_e increases as ZigBee's transmit power decreases from 0dBm to -20dBm, implying that a low-power mode suffers more from collisions, and may not

save energy due to the potential retransmission overhead. Note that the 15dBm transmit power is only practical for DC-powered ZigBee node (*e.g.*, the XBee module [37]), which can be used as the signaler. As this power level is comparable to WiFi, spatial collision occurs only when the two networks are extremely close. Thus, it is reasonable to assume the CTS packet from the signaler is unaffected by WiFi in common cases. In addition, note that a larger ρ_z results in lower mean SINR for $\mathcal{Z}_r$ and even larger I_e. We omit the detailed experiments as the effect is similar to the decrease in transmit power.

4.5.1.4 Joint spatial-temporal effects

Combining the analysis for temporal and spatial factors, we can analyze how ZigBee is affected by a WiFi transmitter at an arbitrary location. We fix the WiFi traffic load to a medium value 0.36, the ZigBee's transmit power to -10dBm, and then calculate the probability of packet loss, *i.e.*, both spatial and temporal collision occurs. As can be seen from the results (Fig. 4.9), the joint collision probability is bisected for legacy ZigBee, according to whether $\mathcal{W}_t$ and $\mathcal{Z}_t$ can sense each other. When $\mathcal{Z}_t \subset \mathcal{W}_t$, the joint collision probability is relatively low (below 41% and 17% for TDMA/CSMA), implying a low packet loss rate. However, when $\mathcal{Z}_t \not\subset \mathcal{W}_t$, the collision rate increases dramatically (to above 43%) for both TDMA and CSMA. Using the signaler, CBT extends the range where $\mathcal{Z}_t$ is visible to $\mathcal{W}_t$ and prevents WiFi preemption in the time domain. These two advantages together bring the collision probability to below 7%.

4.5.1.5 Exploring the design space

Having validated the accuracy of the analysis, we now employ it to flexibly navigate the impact of design parameters on the performance vs. cost tradeoff in CBT. Fig. 4.10 shows the throughput in TDMA mode for different harbinger time (deter-

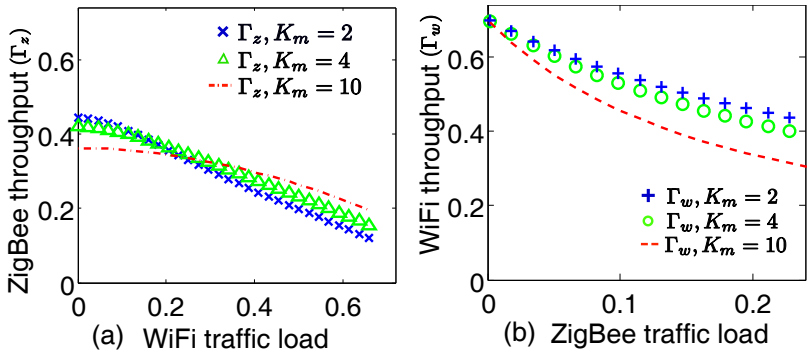

Figure 4.10: Impact of harbinger time in CBT TDMA mode.

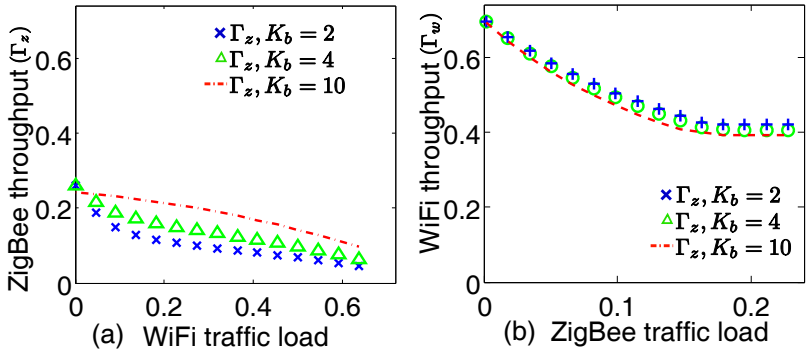

Figure 4.11: Impact of busy-tone duration in CBT CSMA mode.

mined by K_m). A small K_m induces less overhead, and gains higher throughput for ZigBee when the WiFi load is small (below 0.2), but becomes less effective under medium to high WiFi load. In addition, under low duty-cycle ZigBee traffic (below 0.05), WiFi throughput is virtually unaffected by K_m.

In CSMA mode, the key parameter is the busy tone duration (determined by K_b). As shown in Fig. 4.11, a large K_b ($K_b = 10$) gains 2× higher throughput for ZigBee than a small K_b ($K_b = 2$), only at the cost of minor throughput degradation for WiFi (less than 6%), for a broad range of traffic load values. Therefore, a large busy tone duration is always preferred when CBT runs in CSMA mode.

4.5.2 Testbed experiments

We have developed a preliminary version of CBT (TDMA mode) on TinyOS 2.0 and GNURadio 3.2.2. We implement a TDMA scheduling module based on openzb [1], a TinyOS branch for IEEE 802.15.4. The ZigBee hardware that we use, the MicaZ

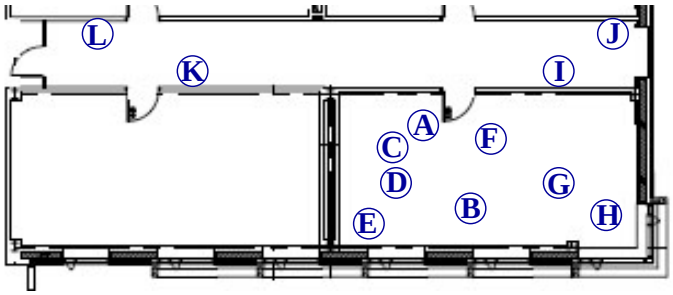

Figure 4.12: Location of ZigBee and WiFi links in the testbed. A→B is the WiFi link. All others are ZigBee nodes.

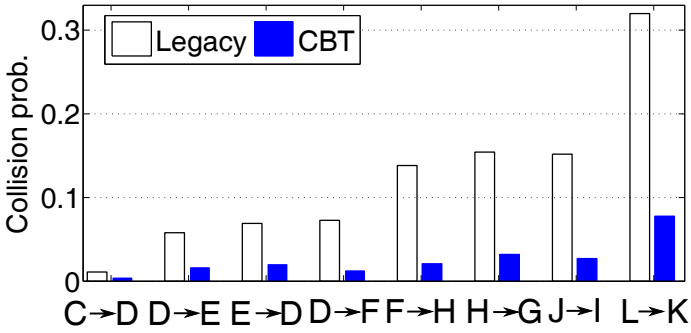

Figure 4.13: Collision rate for different ZigBee link locations.

mote, has a maximum power of 0dBm and packet size limit of 127 bytes, precluding a direct implementation of the signaler's functionalities. Therefore, we implement the signaler based on the 802.15.4 PHY in GNURadio, and run it on the USRP2 software radio [39], which does not have such limitations and can communicate seamlessly with MicaZ. The coordinator and clients are running on micaZ motes.

Due to the inefficient user-space signal processing, the GNURadio/USRP2 platform cannot perform carrier sensing in real time, and cannot be synchronized to follow the TDMA schedule set by the coordinator. Therefore, we disable the carrier sensing, fix the harbinger time to $H_s = J_z$, and allow the coordinator to send a 5-byte notification message to the USRP2 immediately before the harbinger time arrives. The message is sent without carrier sensing and may still be lost due to collision with WiFi (though with a lower probability compared with a larger data packet). Thus, the performance is expected to be lower than a full-fledged implementation on ZigBee-compatible hardware, such as the XBee module [37]. To alleviate the loss,

two back-to-back notification messages are used in the actual implementation. Such a patch costs additional channel time, and should affect WiFi more than a legitimate CBT.

We deploy the ZigBee coordinator, clients and signaler in an office environment, co-located with a pair of WiFi nodes. Fig. 6.10 shows a map of the node locations. The relative distances between ZigBee and WiFi links satisfy: $1m < d_z < 3m$, $0.5m < d_t < 7m$. The distance between WiFi and ZigBee transmitters (d_t)is limited to $7m$ because ZigBee link distances (d_z) are typically short, and ZigBee signal tends to capture the WiFi interference when d_t is large. Each ZigBee link consists of one coordinator and one client (the transmitter is randomly selected between them), and the signaler is placed near the coordinator. The ZigBee link sends TDMA packets at a duty cycle of 8 packets/second. The WiFi link is running constant-bit-rate UDP traffic, with traffic load 0.22, packet size and bit rate settings following Sec. 4.5.1.1.

Fig. 4.13 plots the collision rate between ZigBee and WiFi, which is measured by the one-way packet loss-rate of ZigBee. The combined effects of spatial and temporal collision result in diverse collision rate for different links. In general, when the ZigBee transmitter is close to the WiFi transmitter, the collision rate is lower due to ZigBee's better visibility to WiFi (*e.g.*, for ZigBee link C→D). When the ZigBee transmitter moves far away from the WiFi, it can no longer trigger the deferring and backoff, but may still be exposed to the interference from WiFi, thus causing severe collision (*e.g.*, for ZigBee link L→K). Overall, the collision rate is above 12% for half of the legacy ZigBee links. In contrast, when running CBT, collision rate for all the links is reduced to below 8%, and the reduction is above 60% for all links. In summary, the topological effects on collision rate and the effectiveness of CBT is consistent with the trend predicted by the analysis in Fig. 4.9. The actual value of collision rate does not match because of the simplified propagation model used in analysis (Sec. 4.4.4).

To examine the impact of ZigBee on WiFi, we create a *worst case* scenario for

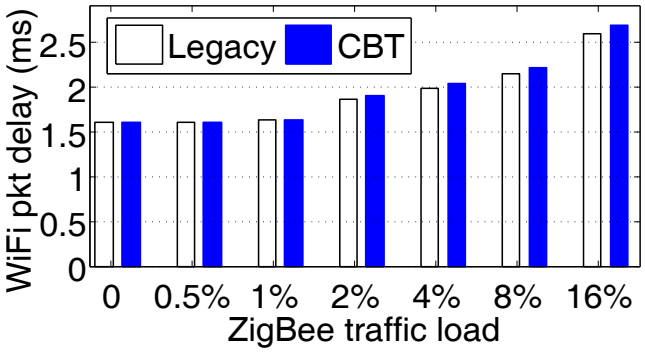

Figure 4.14: Impact of ZigBee on WiFi.

WiFi where the USRP2 sends legacy ZigBee packets and busy tones without carrier sensing. Fig. 4.14 plots the WiFi packet delay as a function of ZigBee load. The delay performance remains virtually unaffected for lightly loaded ZigBee traffic, *e.g.*, below 2%. More importantly, the extra overhead in CBT does not degrade the WiFi performance significantly compared to the legacy ZigBee, even though the benefit of CBT in saving retransmissions is unaccounted for here. Since ZigBee targets low duty-cycle applications, the experiment essentially shows that neither the legacy ZigBee nor CBT affect WiFi in the common cases.

4.6 Conclusion

In this chapter, we proposed CBT, a cooperative paradigm enabling low-power, low-rate ZigBee WPANs to coexist with a high-end WiFi WLAN. CBT maintains the spectrum etiquette based on energy detection, but overcomes its limitations in heterogeneous networks by separating the busy tone signaling from data transmission. We establish an analytical framework that quantitatively compares CBT with the legacy ZigBee, and allows flexible exploration of its design parameters. The analysis, combined with detailed simulation and prototype implementation, demonstrates several-fold performance improvement (in terms of collision probability and throughput), with negligible cost to the WiFi side. In future, we plan to extend CBT to a general framework allowing coexistence of heterogeneous MAC/PHY protocols in the

recently-opened TV band white-spaces.

CHAPTER V

Redesigning the Carrier Sensing and Idle Listening Mechanism

5.1 Introduction

Continuing advances of physical-layer technologies have enabled WiFi to support high data-rates at low cost and hence become widely deployed in networking infrastructures and mobile devices, such as laptops, smartphones, and tablet PCs. Despite its high performance and inexpensive availability, the energy-efficiency of WiFi remains a challenging problem. For instance, WiFi accounts for more than 10% of the energy consumption in current laptops [12]. It may also raise a smartphone's power consumption 14 times even without packet transmissions [8].

WiFi's energy-inefficiency comes from its intrinsic CSMA mechanism—the radio must perform *idle listening* (IL) continuously, in order to detect unpredictably arriving packets or assess a clear channel. The energy consumption of IL, unfortunately, is comparable to that of active transmission/reception [8, 31]. Even worse, WiFi clients tend to spend a large fraction of time in IL, due to MAC-level contention and network-level delay [85]. Therefore, minimizing the IL's energy consumption is crucial to WiFi's energy-efficiency.

A natural way to reduce the IL's energy cost is sleep scheduling. In WiFi's power-

saving mode (PSM) and its variants [5, 12, 106, 85], clients can sleep adaptively, and wake up only when they intend to transmit, or expect to receive packets. The AP buffers downlink packets and transmits only after the client wakes up. PSM essentially shapes the traffic by aggregating downlink packets, thereby reducing the receiver's wait time caused by the network-level latency. However, it cannot reduce the IL time associated with carrier sensing and contention. Through an extensive trace-based analysis of real WiFi networks (Sec. 5.3), we have found that IL still dominates the clients' energy consumption even with PSM enabled: it accounts for more than 80% of energy consumption for clients in a busy network and 60% in a relatively idle network.

Since the IL time cannot be reduced any further due to WiFi's CSMA, we exploit an additional dimension—reducing IL power consumption—in order to minimize its energy cost. Ideally, if the exact idle period is known, the radio could be powered off or put to sleep during IL, and wake up and process packets *on demand*. However, due to the distributed nature of CSMA, the idle time between packets varies widely and unpredictably. Under-estimation of an idle interval will waste energy, while an over-estimation causes the radio to drop all incoming packets during the sleep.

So, one may raise an important question: "is it possible to put the radio in a *subconscious* mode, where it consumes little power and can still respond to incoming packets promptly?" We answer this question by proposing Energy-Miminizing idle Listening (`E-MiLi`) that reduces the clock-rate of the radio during its IL period. The power consumption of digital devices is known to be proportional to their voltage-square and clock-rate [43, 36]. Theoretically, by reducing clock-rate alone, `E-MiLi` reduces the IL's power consumption linearly.

It is, however, nontrivial to ensure that packets can be received at a lower clock-rate than required. To decode a packet, the receiver's sampling clock-rate needs to be at least twice the bandwidth of the transmitted signal, following the Nyquist's

Theorem. WiFi radios have already been optimized under this theorem by matching the receiver's clock-rate with the Nyquist rate.

`E-MiLi` meets this challenge via a novel approach called *Sampling Rate Invariant Detection* (SRID). SRID separates the detection from the decoding of a packet. It adds a special preamble to each 802.11 packet, and incorporates a linear-time algorithm that can accurately detect the preamble even if the receiver's clock-rate is much lower than the transmitter's. SRID embeds the destination address into the preamble, so that a receiver may only respond to packets destined for it. Upon detecting this special preamble, the receiver immediately switches to the full clock-rate and then recovers the packet with a legacy 802.11 decoder.

`E-MiLi` allows SRID to be integrated into existing MAC or sleeping-scheduling protocols, using a simple *Opportunistic Downclocking* (ODoc) scheme. ODoc enables fine-grained, packet-level power management by adding a downclocked IL mode into the radio's state machine. ODoc exploits the burstiness and correlation structure of real traffic to assess the potential benefit of downclocking, and then downclocks the radio only if it is unlikely to incur significant overhead.

We have implemented an `E-MiLi` prototype on the GNURadio/USRP platform [39]. Our experimental evaluation shows that `E-MiLi` can detect packets with close to 100% accuracy even if the radio operates at $\frac{1}{16}$ of the normal clock-rate. Within a normal SNR range ($> 8dB$), `E-MiLi` performs comparably to a legacy 802.11 detector. Furthermore, from real traffic traces, we find that for the majority of clients, the overall energy saving with `E-MiLi` is close to that in pure IL mode with the maximum downclocking factor. According to our measurements, this corresponds to 47.5% for a typical WiFi card with a downclocking factor of 4, and 36.3% for a software radio with a downclocking factor of 8. Further, our packet-level simulation results show that `E-MiLi` reduces energy consumption consistently across different traffic patterns, without any noticeable performance degradation.

In summary, this chapter makes the following contributions.

- Exploration of the feasibility and cost of fine-grained control of radio clock-rate to improve energy-efficiency.
- Design of SRID, a novel packet detection algorithm that makes it possible to detect packets even if the receivers are down-clocked significantly.
- Introduction of ODoc, a generic approach to integrating SRID with existing MAC- and sleep-scheduling protocols.
- Implementation of `E-MiLi` on a software radio platform and validation of its performance with real traces and synthetic traffic.

The remainder of this chapter is organized as follows. Sec. 5.2 analyzes the energy cost of IL in WiFi networks and describes the motivation behind `E-MiLi`. Sec. 5.3 presents a measurement study of the relation between energy-consumption and clock-rate in WiFi and software radio devices. Following an overview of `E-MiLi` (Sec. 5.4), Secs. 5.5 and 5.6 present the detailed design of SRID and ODoc, respectively. Sec. 5.7 evaluates `E-MiLi`. Sec. 5.9 reviews related work and Sec. 5.10 concludes the chapter.

5.2 Why E-MiLi?

In this section, we motivate `E-MiLi` by showing a large fraction of time and energy spent in IL for real-world WiFi users. We also briefly discuss the reasons for the high power-consumption of IL by anatomizing a typical radio.

5.2.1 Cost of Idle Listening

We acquired packet-level WiFi traces from publicly available datasets: SIGCOMM'08 [111] and PDX-Powell [97]. The former was collected from a WLAN used for a conference session that has a peak (average) of 31 (7) clients. The latter was collected from

a public hotspot at a university bookstore, with a peak (average) of 7 (3) clients. We built a simulator that can parse the traces and compute each client's sojourn time in different states, including:

- *TX&RX*: the client is transmitting or receiving a packet.
- *Sleep*: the client is put to sleep. A client sets the power-management field in its packet header to 1 if it intends to sleep after the current frame transmission and ACK [5].
- *Idle listening* (IL): a state other than the above two. This includes sensing the channel, waiting for incoming packets, receiving packets not addressed to it, *etc.*. We exclude the SIFS time, which is a short interval (9–20μs [5]) between two immediate packets (*e.g.*, in between data/ACK). We also consider a client disconnected if it does not transmit/receive any unicast packets for 5 minutes or longer.

Fig. 5.1(a) plots the normalized fraction of time spent in the three modes, distributed among all the clients in the SIGCOMM'08 trace. More than 90% of clients enable power management and judiciously put their radios to sleep. However, clients spend most of the time in IL, rather than sleeping: the median IL time is 0.87, and is above 0.6 for more than 80% of clients. One may guess the reason for this to be the excessive contention in this busy network. However, even in the PDX-Powell trace (Fig. 5.1(b)), the IL time exceeds 0.52 for more than 70% of clients. In contrast, the actual TX&RX time is below 0.1 for more than 90% of clients in both networks. Since WiFi's PSM cannot eliminate MAC-layer contention and queueing delays [106], the IL still dominates the TX&RX time by a significant margin.

We further analyze the energy cost of IL. Since information on the actual type of clients' WiFi cards is unavailable, we assume that their energy profile follows that of a typical Atheros card [15, Sec. 10.1.5] (TX: 127mW, RX: 223.2mW, IL: 219.6mW,

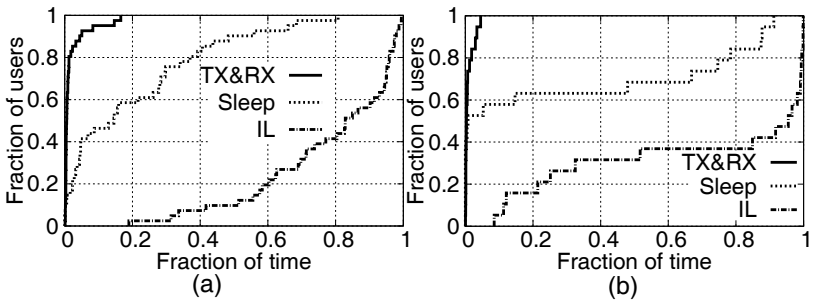

Figure 5.1: CDF of the fraction of time spent in different modes for (a) SIGCOMM'08 trace and (b) PDX-Powell trace.

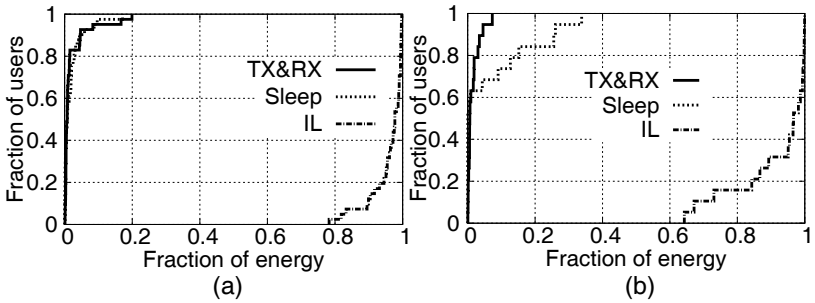

Figure 5.2: CDF of the fraction of energy spent in different modes for (a) SIGCOMM'08 trace and (b) PDX-Powell trace.

Sleep: 10.8mW). Although their absolute power consumption differs, many widely used WiFi cards have consistent relative power consumption among different states [16]. Fig. 5.2(a) shows that in a busy network, for more than 92% of clients, 90% of energy is spent in IL, *i.e.*, IL costs 9 times more energy than TX&RX for most clients. Moreover, although the sleep time is substantial, the sleep power is negligible, whereas the IL power is comparable to the TX/RX power, so the majority of cost is still with IL. For a network with less contention (Fig. 5.2(b)), IL costs less, yet still accounts for more than 73% of energy cost for 90% of clients. Note that the sleep energy may exceed the TX&RX energy, due to the significant amount of sleep time.

The above evaluation reveals that IL accounts for the majority of a WiFi radio's energy cost, and optimizing the IL time alone using PSM is not enough. If the IL power can be reduced, it will clearly improve the energy-efficiency of PSM-like sleep scheduling protocols. In addition, for real-time applications, the constant active mode (CAM) of WiFi is preferable, since PSM may incur an excessive delay and degrade

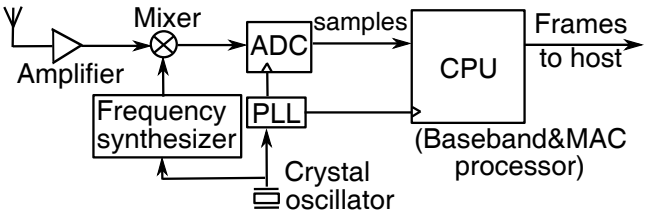

Figure 5.3: Architecture of a WiFi receiver.

the QoS [8]. By reducing IL power, even CAM can achieve high energy-efficiency.

5.2.2 Why Is Idle Listening So Costly?

Intuitively, a radio should consume less power when it is not actively decoding or transmitting packets, but the IL power of commodity WiFi and other carrier-sensing wireless (*e.g.*, ZigBee) devices is comparable to their TX&RX power [15, 8, 123]. In what follows, we briefly discuss the reason for this by anatomizing the radio hardware.

Fig. 5.3 illustrates the architecture of a typical WiFi receiver (based on an Atheros 802.11 chip [130]). An incoming signal is first passed through the RF and analog circuit, amplified and converted from RF (*e.g.*, 2.4GHz) to the baseband by a mixer. The analog baseband signal is sampled by an Analog-to-Digital Converter (ADC), and the resulting discrete samples are passed to the CPU (baseband and MAC processor), which decodes the signal and recovers the original bits in the data frame. The entire radio is driven by a 40MHz crystal oscillator, which feeds two paths. The first is the frequency synthesizer that generates the center frequency used for the RF and analog mixer. The other is the Phase-Locked-Loop (PLL) that generates the clocking signal for the digital circuit: the sampling clock for the ADC, as well as the main clock for the CPU.

Existing studies have shown the ADC and CPU to be the most power-hungry components of a receiver. In the Atheros 5001X chipset, for example, they account for 55.3% of the entire receiver power budget [91, Table. 5]. ADC and CPU power consumptions are also similar (1.04:1 [19]). During IL, both the analog circuits and

the ADC operate at full workload as in the receiving mode. Moreover, the decoding load of the CPU is alleviated, but it cannot be put into sleep—it needs to operate at full clock-rate in order to perform carrier sensing and packet detection. This is the reason why IL power consumption is comparable to that of receiving packets.

A similar line of reasoning applies to other wireless transceivers such as software radios. In software radios, the ADC feeds the discrete samples to an FPGA, which may further decimate (downsample) the samples and then send them to a general processor that serves as the baseband CPU. The similarity in hardware components implies that software radios are likely to suffer from the same problem with IL. Considering the trend of software radios getting gradually integrated into mobile platforms to reduce the area cost [38], it is imperative to incorporate a mechanism to reduce its IL power.

5.3 IL Power vs. Clock Rate

We propose to reduce the IL power by slowing down the clock that drives the digital circuitry in a radio. Modern digital circuits dissipate power when switching between logic levels, and their power consumption follows $P \propto V_{dd}^2 f$, where V_{dd} is the supply voltage and f the clock-rate [36, 43]. Hence, a linear power reduction can be achieved by reducing clock-rate. In practice, due to the analog peripherals, the actual reduction is less than ideal. For example, in the ADC used by an Atheros WiFi chip [119], halving the sampling clock-rate results in a 31.4% power reduction. Here, using detailed measurements, we verify the actual effects of reducing the clock-rate for both WiFi NIC and the USRP software radio.

5.3.1 WiFi radio

According to IEEE 802.11-2007 [5], the OFDM-based PHY supports 2 down-clocked operations with 10MHz (half-clocked) and 5MHz (quarter-clocked) sampling-

rate, in addition to the default full-clocked 20MHz operation. We test these two modes on the LinkSys WPC55AG NIC (version 1.3, Atheros 5414 chipset), with a development version of Madwifi (trunk-r4132), which supports 8 half-clocked and 18 quarter-clocked channels at the 5GHz band. The downclocked modes can be enabled by activating the "USA with $\frac{1}{2}$ and $\frac{1}{4}$ width channels" regulatory domain on the NIC.

As to measurement of the WiFi's power consumption, our approach is similar to that in [16]. We attach the NIC to a laptop (Dell 5410) powered with an external AC adapter, and use a passive current probe (HP1146A) and voltage probe (HP1160) together with a 1Gsps oscilloscope (Agilent 54815A) to measure the power draw. The actual power consumption is the difference between the measured power level in different radio modes and the base level with the NIC removed. During the measurement, we tune the WiFi to a channel unused by ambient networks. The IL power is measured when the NIC is activated but not transmitting/receiving packets. The TX/RX power is measured when the WiFi is sending/receiving one-way ping-broadcast packets at the maximum rate (100 packets per second). The different clock modes are configured to use the same bit rate (6Mbps) and packet size (1KB). Table 1 shows the measurement results.

It can be seen that the power consumption decreases monotonically with clock-rate. In particular, compared to a full-clocked radio, the IL power is reduced by 36% and 47.5% for half-clocked and quarter-clocked mode, respectively. The absolute reduction is found different from that reported in an existing measurement study [31]. We guess this discrepancy results from the use of a different WiFi card (Atheros 5212) in their experiment. As validated in [31], different NICs have very different power profiles at different clock-rates. To confirm that the power consumption vs. clock-rate relation is not limited to the WiFi radio, we have also conducted experiments with the USRP software radio.

	rate = 1	rate = 1/2	rate = 1/4
Idle	1.22	0.78	0.64
RX	1.66	1.44	0.98
TX	1.71	1.46	1.21

Table 5.1: Mean power consumption (in W) of WiFi under different clock-rates.

	rate=1	rate=1/2	rate=1/4	rate=1/8	rate=1/16
IL	10.27	7.96	7.07	6.54	5.88
TX	6.36	5.69	5.18	4.70	4.47

Table 5.2: Mean power consumption (in W) of USRP under different clock-rates.

5.3.2 Software radio

The original USRP is driven by an internal 64MHz clock, which is used by both the ADC and FPGA. We enabled the external clocking feature by resoldering the main clock circuit, following the instructions in [39]. We use the USRP E100 [39] as an external clock source, which has a programmable clock generator (AD9522) that produces reference clocks below 64MHz[1].

We mounted an XCVR2450 daughter board on the USRP, which was then connected to the PC host (a Dell E5410 laptop). The IL mode runs the standard 802.11a/g carrier sensing and packet detection algorithm (see Sec. 5.7 for the details of our implementation). The TX mode sends a continuous stream of samples prepended with 802.11 preambles. Since a complete 802.11 decoding module is unavailable, we only measure the IL and TX power. We measure the USRP power directly with the oscilloscope and current/voltage probes, and then add the power consumption of the external clock [11], which is 0.55W and does not vary with clock-rates. Note that the normal clock-rate of USRP is 64MHz, whereas the maximum signal bandwidth sent to the PC is 4MHz since the FPGA downsamples (decimates) the signals. While reducing the clock-rate, we ensure the signal bandwidth is decreased by the same ratio by adjusting the decimation rate.

[1]The USRP E100 cannot be tuned to signals below 32MHz. So, we used a signal generator to produce clock signals below 32MHz, with the same configuration as those produced by the E100.

Table 2 shows the measurement results. Similar to a WiFi radio, the USRP power consumption decreases monotonically with clock-rate. A power reduction of 22.5% (36.3%) is achieved for a downclocking factor of 2 (8). We found that at a 4MHz clock-rate (a downclocking factor of 16), the USRP can no longer be tuned to the 2.4GHz center frequency, but the ADC can still be tuned correctly to 4MHz sampling rate, and power consumption decreases further.

Since the PC host consumes a negligible amount of power when processing the 4MHz signal, we have omitted its power consumption in Table 2. Future mobile software radio systems may incorporate dedicated processors to process the baseband signals. By reducing the processors' clock-rate in parallel with the ADC and FPGA, the entire software radio platform can achieve higher energy-efficiency.

5.4 An Overview of E-MiLi

`E-MiLi` controls the radio clock-rate on a fine-grained, per-packet basis, in order to reduce the energy consumption of IL. It opportunistically downclocks the radio during IL, and then restores it to full clock-rate before transmitting or after detecting a packet. Fig. 5.4 illustrates the flow of core operations when `E-MiLi` receives and transmits packets.

`E-MiLi` prepends to each 802.11 packet an additional preamble, called *M-preamble*. During its IL period, a downclocked receiver continuously senses the channel and looks for the M-preamble, using the sampling rate invariant detection (SRID) algorithm. Upon detecting an M-preamble, the receiver immediately switches back to full clock-rate, and calls the legacy 802.11 decoder to recover the packet. The receiver leverages an implicit, PHY-layer addressing mechanism in SRID to filter the M-preamble intended for other nodes, and hence prevents unnecessary switching of clock-rate.

A TX operations follow the legacy 802.11 MAC, except that the carrier sensing is done by SRID. If the radio is downclocked during carrier sensing and backoff, it needs

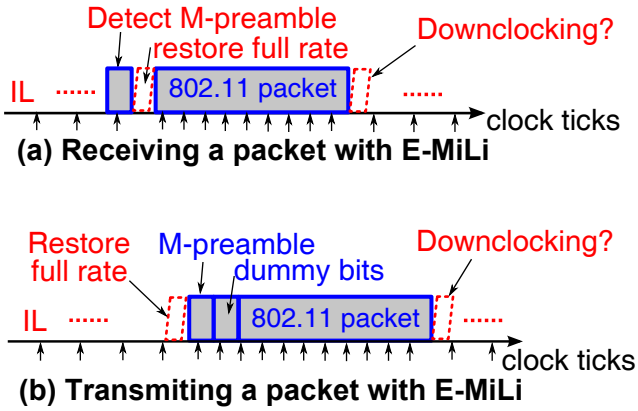

Figure 5.4: Idle listening and RX/TX operations in E-MiLi.

to restore full clock-rate before the actual transmission. The exact restoration time is scheduled by another component of `E-MiLi`, called *Opportunistic Downclocking* (ODoc).

After completing an RX or TX operation, the radio cannot downclock greedily. As we will verify experimentally in Sec. 5.6, switching clock-rate takes 9.5 to 151 μs for a typical WiFi radio. During the switching, the clock is unstable, and packets cannot be detected even with SRID. To reduce the risk of packet loss, `E-MiLi` employs ODoc again to make a downclocking decision using a simple outage-prediction algorithm, which estimates if a packet is likely to arrive during the clock-rate switching.

In addition, after sending the M-preamble, a transmitter cannot wait silently during the receiver's switching period; it may otherwise lose the medium access and be preempted by other transmitters. To compensate for the switching gap, the transmitter inserts a sequence of dummy bits between the M-preamble and the 802.11 packet. The dummy bits cover the maximum switching period so that the channel is occupied continuously. Note that the transmitter always sends the M-preamble, dummy bits, and 802.11 packets at the full clock-rate. It need not know the current clock-rate of the receiver.

When multiple clients coexist, `E-MiLi` assigns a broadcast address as well as multiple unicast addresses, each with a unique feature. This feature is embedded in the

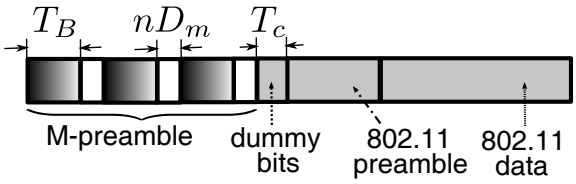

Figure 5.5: M-preamble construction and integration with an 802.11 packet.

M-preamble and detectable only by the intended receiver. To reduce the overhead of M-preamble, `E-MiLi` incorporates an optimization framework that allows multiple clients to share addresses at minimum cost.

In summary, `E-MiLi` always runs at full clock-rate to transmit or decode packets, but downclocks the radio during IL to detect implicitly-addressed packets, whenever possible. Next, we detail the design of components in `E-MiLi`.

5.5 Sample Rate Invariant Detection

To realize `E-MiLi`, its packet-detection algorithm must overcome the following challenges: (i) it must be resilient to the change of sampling clock-rate; (ii) it must be able to decode the address information directly at low sampling rates; and (iii) due to unpredictable channel condition and node mobility, its decision rule should not be tuned at runtime, and hence must be resilient against the variation of SNR. We propose SRID to meet these challenges via a joint design of preamble construction and detection.

5.5.1 Construction of the M-preamble

`E-MiLi` constructs the M-preamble to facilitate robust, sampling-rate invariant packet detection, while implicitly delivering the address information. An M-preamble comprises $C (C \geq 2)$ duplicated versions of a pseudo-random sequence, as shown in Fig. 5.5 (where $C = 3$).

Within the M-preamble duration, the channel remains relatively stable, and there-

fore the duplicated sequences sent by the transmitter maintain strong similarity at the receiver. Hence, a receiver can exploit the strong *self-correlation* between the C consecutive sequences to detect the M-preamble. More importantly, since radios sample signals at a constant rate, the receiver would obtain C similar sequences even if it down-samples the M-preamble.

To enhance resilience to noise, the random sequence in M-preamble must have a strong self-correlation property—it should produce the best correlation output only when correlating with itself. The Gold sequence [40] satisfies this requirement. It outputs a peak magnitude only for perfectly aligned self-correlation, and correlating with any shifted version of itself results in a low, bounded magnitude. For a Gold sequence of length $L = 2^l - 1$ (l is an integer), the ratio between the magnitude of self-correlation peak and the secondary peak is at least $2^{\frac{l-1}{2}}$. The original Gold sequence is binary [40]. To make it amenable for WiFi transceivers, we construct a *complex Gold sequence* (CGS), in which the real and imaginary parts are shifted versions of the same Gold sequence generated by the standard approach [40].

In addition, we use the *length* of the CGS to implicitly convey address information. An address is an integer number n, and corresponds to a CGS of length $(T_B + nD_m)$, where D_m is the maximum downclocking factor of the radio hardware. T_B is the minimum length of the CGS used for the preamble, also referred to as *base length*. To detect its own address (*e.g.*, n), at each sampling point t, the client simply self-correlates the latest T_B samples with the previous T_B samples offset by nD_m. When the client is downclocked by a factor of D, it scales down the base length to $T_B D^{-1}$ and offset to $nD_m D^{-1}$ accordingly. The nD_m value ensures that different addresses are offset by at least 1 sample, even if the CGS is downsampled by the maximum factor D_m.

One challenge related to the Gold sequence is that it only allows length of $L = 2^l - 1$. Hence, not all of the $(T_B + nD_m)$ samples can be exactly matched to a whole

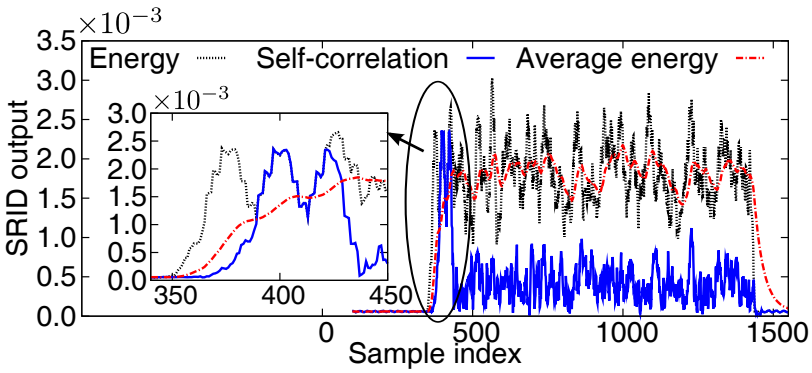

Figure 5.6: Detecting M-preamble using SRID (clock-rate=1/4).

Gold sequence. We solve this problem by first generating a long CGS, and then assign the sub-sequence of length $(T_B + nD_m)$ to the n-th address.

Clearly, to meet its design objectives, an ideal random sequence for M-preamble should have strong self-correlation even after it is *downsampled* and *truncated* (since we only use T_B of the $T_B + nD_m$ samples to perform self-correlation). We conjecture there does not exist such a sequence unless the sequence length is very large and the downsampling factor is small. We leave the theoretical investigation of this problem as our future work. In this project, we will empirically verify that the CGS with a reasonable length suffices to achieve high detection accuracy in practical SNR ranges.

5.5.2 Detection of the Preamble

We formally derive the detection algorithm in SRID by modeling how the receiver down-samples the M-preamble and identifies it via self-correlation.

Let $T = C(T_B + nD_m)$ be the total length of the M-preamble (Fig. 5.5), and $x(t), t \in [0, T)$, the transmitted samples corresponding to the M-preamble. For a full-clocked receiver, the received signals are:

$$y_o(t) = e^{2\pi\Delta f t} h(t)x(t) + n(t), t \in [0, T). \tag{5.1}$$

where $n(t)$ is the noise, $h(t)$ the channel attenuation (a complex scalar representing

amplitude and phase distortion), and Δf the frequency offset between the transmitter and the receiver. When a receiver operates at the clock-rate of $\frac{1}{D}$ (*i.e.*, with a downclocking factor of D), the received signals become:

$$z(k) = e^{2\pi\Delta f t} h(t)x(t) + n(t), t = kD, 0 \leq k < \lfloor \frac{T}{D} \rfloor.$$

Here D must be an integer divisor of the base length T_B of the CGS, *i.e.*, $\lfloor \frac{T_B}{D} \rfloor = \frac{T_B}{D} \triangleq T_1$. To detect M-preamble, at each sampling point k, the receiver with address n performs self-correlation between the latest T_1 samples and the previous T_1 samples offset by nD_mD^{-1}, resulting in:

$$R(k) = \sum_{i=k}^{k+T_1-1} z(i)z^*(i - T_1 - nD_mD^{-1}) \tag{5.2}$$

$$\approx \sum_{i=k}^{k+T_1-1} e^{2\pi\Delta f iD} h(iD)x(iD)\big[e^{2\pi\Delta f(iD-T_B-nD_m)}$$

$$h(iD - T_B - nD_m)x(iD - T_B - nD_m)\big]^* \tag{5.3}$$

$$\approx e^{T_B+nD_m}|h(kD)|^2 \sum_{i=k}^{k+T_1-1} |x(iD)|^2 \tag{5.4}$$

where $(\cdot)^*$ denotes the complex conjugate operator.

Eq. (5.3) is derived based on the fact that the signal level is usually much higher than the noise. Eq. (5.4) is based on the fact that (i) the random sequence $x(t)$ preserves similarity with its predecessor sequence, even though it is downsampled; and (ii) the channel remains relatively stable over its *coherence time*, which is much longer than the preamble duration. To see this, we note that the coherence time can be gauged as $T_o = \frac{\lambda}{\sqrt{2}\pi v}$, where λ and v denote the wavelength of the signal and the relative speed between the transmitter and the receiver [29]. At a walking speed of 1m/s, T_o equals 28.8 milliseconds, whereas the M-preamble duration lasts for tens of microseconds (see Sec. 5.5.3.1).

Meanwhile, the energy level of T_1 samples is calculated as:

$$E(k) = \sum_{i=k}^{k+T_1-1} |z(i)|^2 \approx |h(kD)|^2 \sum_{i=k}^{k+T_1-1} |x(iD)|^2. \tag{5.5}$$

From Eqs. (5.4) and (5.5), we get $|R(t)| \approx E(t)$. By contrast, if no M-preamble presents or an M-preamble with a different address a is transmitted, then the self-correlation yields:

$$|R(k)| \approx |h(kD)|^2 \Big| \sum_{i=k}^{k+T_1-1} x(iD)x(iD - T_B - aD_m)^* \Big| \approx 0$$

This is because the sequence $x(iD), i \in [k, k+T_1-1]$ is a truncated CGS and has strong correlation only with itself.

Fig. 5.6 shows a snapshot of $|R(t)|$ and $E(t)$ when receiving a packet prepended with M-preamble. $|R(t)|$ aligns almost perfectly with $E(t)$ in an M-preamble, even though the receiver is downclocked. In contrast, $|R(t)|$ differs from $E(t)$ significantly if noise or uncorrelated signals are present.

Based on the above findings, SRID uses the following basic decision rule to determine the presence of an M-preamble:

$$H < |R(k)| \cdot [E(k)]^{-1} < H^{-1} \tag{5.6}$$

where H is a threshold such that $H \lessapprox 1$. This decision rule has several key advantages. First, it normalizes the self-correlation with the energy level, so H need not be changed according to the signal strength. We will show experimentally (Sec. 5.7) that a fixed value of $H = 0.9$ is robust across a wide range of SNR. Second, it does not require estimation of the channel parameters or calibration of the frequency offset, and hence can be used in dynamic WLANs with user churn and mobility.

For further enhancement of resilience to noise, note that the decision rule (5.6)

Algorithm 1 Detecting the M-preamble using SRID.

1. **Input:** new sample $z(k+T_1-1)$ at sampling point $k+T_1-1$
2. **Output:** packet detection decision at sampling point k
3. /*Update energy level of past T_1 samples*/
4. $E(k) \leftarrow E(k-1) + |z(k+T_1-1)|^2 - |z(k-1)|^2$
5. /*Update average energy level*/
6. $E_a(k) \leftarrow T_1^{-1}E(k) + (1-T_1^{-1})E_a(k-1)$
7. /*Update self-correlation with predecessor sequence*/
8. $R(k) \leftarrow R(k-1) + z(k+T_1-1)z(k-nD_mD^{-1}-1)^*$
9. $\qquad -z(k-1)z(k-1-T_1-nD_mD^{-1})^*$
10. /*Apply SNR squelch and self-correlation decision*/
11. **if** $10\log_{10}\frac{E_a(k)}{E_a(k-TD^{-1})} > H_s$ && $H < \frac{|R(k)|}{E(k)} < H^{-1}$
12. $\quad$ **then** decisionQ $\leftarrow$ push 1
13. **else** decisionQ $\leftarrow$ push 0
14. **fi**
15. **if** sum(decisionQ) $> H_1 \cdot \frac{(C-1)(T_B+nD_m)}{D}$
16. $\quad$ **then return** 1
17. **fi**
18. **return** 0

is likely to be satisfied at all the sampling points from the second to the C-th CGS (Fig. 5.5). There are $\frac{(C-1)(T_B+nD_m)}{D} \triangleq T_2$ such points at a downclocking factor D, which can offer high diversity in a noisy or fading environment. To exploit this advantage, at each sampling point k, SRID stores the decision for the past T_2 samples in a FIFO queue, and then apply the following enhanced rule: for $k-T_2 < i \leq k$, the number of sampling points satisfying Eq. (5.6) $\geq H_1T_2$, where H_1 is a tolerance threshold and $H_1 \in (0,1]$.

In addition, during idle periods (*i.e.*, when no signal is present), both the self-correlation and the energy level may be close to 0 and close to each other, and hence the decision rule (5.6) may be falsely triggered. To prevent such false alarms, we added an *SNR squelch*, which maintains a moving average of incoming signals' energy level, with the window size equal to T_1:

$$E_a(k) = T_1^{-1}E(k) + (1-T_1^{-1})E_a(k-1) \tag{5.7}$$

The SNR squelch passes a sampling point to the self-correlator only if its SNR exceeds a threshold H_s, which corresponds to the minimum detectable SNR (set to $4dB$ for SRID). Since an idle period (noise floor) usually precedes the M-preamble (with length TD^{-1}) due to the MAC-layer contention, the SNR level can be estimated as:

$$SNR = 10 \log_{10} \frac{E_a(t)}{E_a(t-T)} \tag{5.8}$$

Algorithm 1 summarizes the detection of M-preamble in SRID. For each timestamp (sampling point), both the self-correlation in Eq. (5.2) and the energy level in Eq. (5.5) can be computed by a single-step operation, which updates the metrics with an incoming signal and subtracts the obsolete signal. Hence, the algorithm has linear complexity with respect to the number of samples, and is well suited for implementation on an actual baseband signal processor.

5.5.3 Address Allocation

5.5.3.1 Minimum-cost address sharing

Since M-preamble uses sequence length to convey address information, the addressing overhead increases linearly with network size. For a network with N nodes, the M-preamble has a maximum length of $C(T_B + ND_m)$. In our implementation, the base length $T_B = 64$, and CGS repetition $C = 3$. For a medium-sized network, say $N = 5$, and a maximum downclocking factor $D_m = 4$, the entire M-preamble would have a length of 252. When transmitted at a 20MHz sampling rate, the M-preamble only takes $\frac{252}{2\times10^7}s = 12.6\mu s$ channel time, which is comparable to the $16\mu s$ overhead of the 802.11a/g preamble [5]. However, for a large network, *e.g.*, $N = 50$, the M-preamble overhead increases to 69.6 μs, which may be overly large, especially for short packets.

To reduce the addressing overhead, `E-MiLi` allows multiple clients to share a limited number of addresses. Address sharing, however, introduces side effects: clients may unnecessarily trigger each other, thus incurring extra energy consumption. `E-MiLi` makes a tradeoff by carefully allocating addresses according to clients' relative channel usage, *i.e.*, the ratio of each client's TX&RX time to the total TX&RX time of the WLAN. The intuition behind this is that a client that transmits/receives packets more frequently should share his address with a fewer number of other clients, so as to minimize the cost of sharing.

We formalize this intuition with an optimization framework. Given the number of clients N, and the maximum address K_m, we seek the optimal address allocation that minimizes the overhead of `E-MiLi`, as follows:

$$\min \quad \sum_{k=1}^{K_m} L_k \left[\left(\sum_{i=1}^{N} p_i u_{ik} \right) \sum_{i=1}^{N} u_{ik} \right] \tag{5.9}$$

$$s.\ t. \quad \sum_{k=1}^{K_m} u_{ik} = 1, \quad \forall i \in [1, N]. \tag{5.10}$$

$$u_{ik} \in \{0, 1\}, \quad \forall i \in [1, N], \forall k \in [1, K_m] \tag{5.11}$$

where L_k is the overhead when the address k is used. p_i is client i's relative channel usage, and u_{ik} a binary variable indicating whether or not client i uses address k. Intuitively, the objective function (5.9) represents the sum of the overhead of each address, weighted by sum of the channel usages of all clients sharing that address and further multiplied by the number of such clients. The multiplication is necessary because a packet with address k triggers all clients with address k. Eq. (5.10) enforces the constraint that each client uses only one address.

This optimization problem is a non-linear integer program, which is NP-hard in general. In our actual implementation, we approximate the solution by relaxing the integer constraint (5.11) to $0 \leq u_{ik} \leq 1$, solving the resulting quadratic optimization

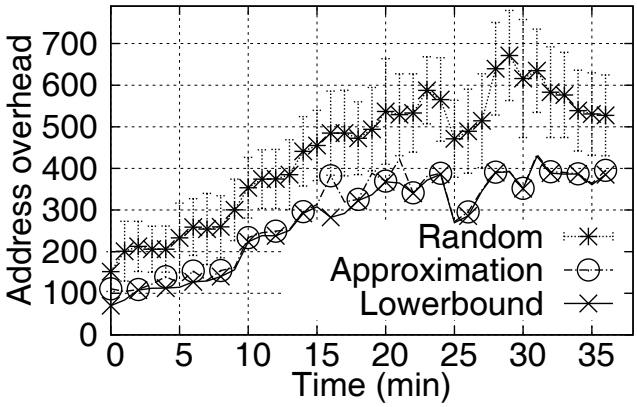

Figure 5.7: Performance of address sharing algorithms.

program, and then rounding the resulting u_{ik} back to its integer value. To implement the address sharing algorithm, the AP needs to periodically (*e.g.*, every 1 minute) compute the relative channel usage p_i, and then broadcast the new allocation to all clients.

To test the effectiveness of the approximation, we run the address sharing algorithm on the SIGCOMM'08 trace (assuming $K_m = 5$ and $L_k = kD_m$) and plot the total address overhead of `E-MiLi` in Fig. 5.7. We observe that the integer-rounding-based solution closely approximates the lower-bound enforced by the quadratic optimization over $0 \leq u_{ik} \leq 1$. On average, the approximate solution exceeds the lower bound by only 1.8%. Fig. 5.7 also shows the mean overhead of an algorithm that randomly assigns an address for each client (error bar shows standard deviation over 20 runs). We observe that the approximation algorithm can save more than 50% of overhead over the random allocation.

5.5.3.2 The broadcast address

In addition to the address designed for each node, `E-MiLi` assigns a broadcast address known to the AP and all clients. It corresponds to an M-preamble with address $n = 0$. Therefore, each node needs to maintain a self-correlator with offset $nD_m = 0$, in addition to the one with its own address.

For the carrier sensing purpose, a node also needs to identify the existence of packets from other transmitters. Similar to the original 802.11, SRID can perform both energy sensing and preamble detection. The former is achieved by following Eq. (5.7). When downclocked by a factor of D, a node can only sense D^{-1} of the energy compared with a full-clocked receiver. Hence, it reduces the energy detection threshold to D^{-1} of the original. When preamble-based carrier sensing is necessary, it can be realized by prepending an additional broadcast preamble. When this *first preamble* is detected, the node determines the channel to be busy, and continues to track the energy level of the entire packet. However, it will restore full clock-rate only when it detects a *second preamble*, which is either addressed to it or is another broadcast preamble.

`E-MiLi` can coexist with 802.11a/g clients even in the preamble detection mode. The 802.11a/g [5] employs self-correlation to detect a short preamble, which corresponds to a random sequence in the frequency domain, and a periodic sequence (period 16, with 10 repetitions) in the time domain. It can be considered as a subset of SRID, with base length $T_B = 16$, sequence repetition $C = 10$, node address 0 and no downclocking, and thus can be easily detected by `E-MiLi` clients. On the other hand, by replacing the first preamble with an 802.11 preamble, `E-MiLi` nodes can be detected by legacy 802.11 as well.

5.6 Opportunistic Downclocking

We now present the ODoc module, which schedules the downclocking to balance its overhead and maintain compatibility with existing MAC and sleep scheduling protocols. We start by inspecting the overhead in switching clock-rates.

5.6.1 Delay in Switching Clock-Rates

When switching to a new clock-rate, the radio needs to be stablized before transmitting/receiving signals. Since the frequency synthesizer and analog circuit's center frequency remain the same, the time cost mainly comes from stablizing the digital PLL (driving the ADC and CPU). This is only several microseconds in state-of-the-art WiFi radios. For example, in MAXIM 2831 [90], the PLL takes less than $8\mu s$ to stablize itself, and the ADC and CPU needs only 1.5 μs to reset, so the total switching time is below 9.5 μs.

We have also measured the switching delay of the Atheros 5414 NIC. We modified the ath5k driver that can directly access the hardware register and reset the clock-rate. After changing the clock-rate register, we repeatedly check a *baseband testing function* until it returns 1 (a conventional way of verifying if the ADC and baseband processor have become ready to receive packets in ath5k), and then record the duration of this procedure.

According to our experimental results, switching between clock-rate 1 and $\frac{1}{4}$ takes 139 μs to 151 μs, whereas switching between 1 and $\frac{1}{2}$ takes 120 μs to 128 μs. We note that this is a conservative estimation of the actual switching delay. To switch to a new rate, the Atheros NIC needs to reset not just the PLL, but also all registers for the OFDM decoding and MAC blocks in the CPU, so that the entire receiver chain can run a valid 802.11 mode. In contrast, `E-MiLi` only needs to reset the PLL, while keeping the registers in the CPU intact. In addition, the latency induced by the baseband testing function and its interface to the PC host is unknown, but is included in the switching delay in our measurement.

We will henceforth use the $9.5\mu s$ switching delay for the MAXIM 2831 chip as a lower bound, and use the measurement result for Atheros 5414 as an upper bound, although the ODoc module is not restricted to these bounds.

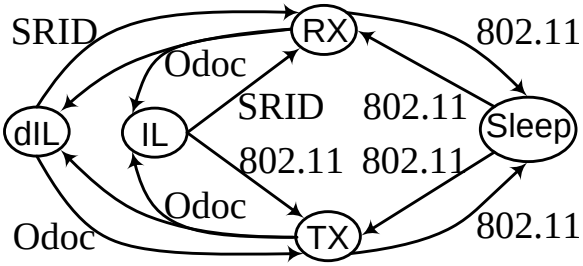

Figure 5.8: Radio state-transition when integrating E-MiLi with 802.11.

5.6.2 Scheduling of Downclocking

5.6.2.1 Control flow

`E-MiLi` interacts with the WiFi MAC/PHY using a simple interface. On the one hand, WiFi calls `E-MiLi` (the SRID module) to assess the channel availability. On the other hand, `E-MiLi` obtains the radio's state machine from the WiFi MAC and the sleep scheduler. Whenever the radio transits to IL, `E-MiLi` calls its ODoc module to determine whether and when to switch clock-rate.

Fig. 5.8 illustrates the state machine of `E-MiLi`. In downclocked IL (dIL) mode, the radio runs SRID continuously, and switches to the full-clocked RX mode immediately upon detection of an M-preamble. When there are packets to be transmitted, carrier sensing is performed by SRID, but the MAC schedule strictly follows the 802.11 CSMA/CA algorithm. ODoc continuously queries the 802.11 backoff counter, and reverts the radio to full clock-rate when the countdown value of the backoff counter is less than $T_c + SIFS$, where T_c is the maximum switching delay, and $SIFS$ is the short inter-frame space defined in 802.11 [5]. ODoc mandates the radio to perform carrier sensing within this $SIFS$ interval after switching to full-clock rate, in order to ensure the channel remains idle after switching. Otherwise, it needs to continue carrier sensing and backoff according to 802.11.

The state-transitions TX↔Sleep and RX↔Sleep are managed by 802.11 or other sleep-scheduling protocols. Whenever a TX or RX completes and the radio is not put to sleep, ODoc decides whether to switch to dIL or the normal IL mode. It makes

this decision using an outage prediction scheme, as detailed next.

5.6.2.2 Outage prediction

ODoc's outage prediction mechanism decides if the next packet is likely to arrive before the radio is stablized to a new clock-rate (referred to as an *outage* event). It first checks if there will be a deterministic operation, *i.e.*, an immediate response of the previous operation. For example, CTS, DATA, and ACK packets are all deterministic operations to follow an RTS. Such packets are separated only by an *SIFS*, which is usually shorter than or comparable to the switching time, so the radio must remain at full rate in between.

When a series of deterministic operations end, ODoc checks if an outage occurred recently. It maintains a binary history for each non-deterministic packet arrival, with "1" representing that the inter-packet interval is shorter than T_c, and "0" otherwise. It asserts that an outage is likely to occur and remains at full clock-rate, if the recent history contains a "1". The key intuition lies in the burstiness of WiFi traffic—a short interval implies an ongoing transmission of certain data, and is likely to continue multiple short intervals until the transmission completes.

An important parameter in ODoc is the size of history. A large history size may predict an outage when it does not occur, thus missing an opportunity of saving energy by downclocking. On the other hand, a small history size results in frequent mis-detection of packets arriving within T_c. Fortunately, *a mis-detection causes only one more retransmission*, because a missed packet will be detected in its next retransmission, when the receiver has already been stablized. Therefore, a small history size is always preferred when energy-efficiency is of high priority. As will be clarified in our experimental study, a history size of between 1 and 10 is sufficient to balance the tradeoff between false-prediction and mis-detection.

5.7 Evaluation

In this section, we present a detailed experimental evaluation of `E-MiLi`. Our experiments center around two questions: (1) How accurate can `E-MiLi` detect packets in a real wireless environment, and with different downclocking rates? (2) How much of energy can `E-MiLi` save for real-world WiFi devices and at what cost?

To answer these questions, we have implemented `E-MiLi` on software radios and network-level simulators as follows.

- We have implemented the SRID algorithm, including the M-preamble construction and detection, on the GNURadio platform and verify it on a USRP testbed. As a performance benchmark, we have also implemented the 802.11 OFDM preamble encoding/detection algorithm (Sec. 5.5.3.2).

- `E-MiLi`'s energy-efficiency depends on the relative time of IL, which, in turn, depends on network delay and contention, and hence, we leverage real WiFi traces again to evaluate the energy-efficiency of `E-MiLi`. We implemented the ODoc framework and address allocation algorithm by extending the trace-based simulator (Sec. 5.3), and then integrating results from the SRID experiments.

- We have also implemented ODoc in ns-2.34, which can be used to verify the performance of `E-MiLi` with synthetic traffic patterns (*e.g.*, HTTP and FTP) independently.

5.7.1 Packet-Detection Performance

We test the detection performance of SRID under different SNR levels and downclocking factors. The SNR is estimated as $SNR = \frac{E_s - E_N}{E_N}$, where E_s is the average energy level of incoming samples when a packet is present, and E_N is the noise floor, both smoothed using a moving average with the window size equal to the length of the M-preamble. Note that this SNR value over-estimates the actual SNR experienced

by the *decoder*, since the decoding modules will raise the noise level by around 3.5 dB [15]. Given that 802.11 needs at least $9.7dB$ SNR to decode packets [19], SRID must be able to detect packets accurately above $9.7dB$ SNR.

We set the base length of SRID's CGS to $T_B = 64$, and maximum downclocking factor $D_m = 16$. We fix the self-correlation threshold $H = 0.9$, and the tolerance threshold $H_1 = 0.6$ (Sec. 5.5). We will show that these thresholds are robust across different experiment settings.

5.7.1.1 Single link

We first test SRID on a single link consisting of two USRP nodes within Line-of-Sight (LOS). We downclock the receiver by different factors, and vary the link's SNR by adjusting the transmit power and link length/distance. Since the USRP fails to work when the external clock is downclocked to $\frac{1}{16}$, we scale its FPGA decimation rate by 16, which is equivalent to downsampling the signals by a factor of 16. Under each SNR/clock-rate setting, the transmitter sends 10^6 packets at full clock-rate with constant inter-arrival time. The mis-detection probability (P_m) is calculated by the fraction of timestamps where a packet is expected to arrive but fails to be detected, and vice versa, for the false-alarm probability (P_f).

Fig. 5.9 plots P_m and P_f as a function of a link's time-averaged SNR (rounded to integer values). P_m drops sharply as SNR increases, and approaches 0 as SNR grows above $8dB$. It tends to be higher under a high downclocking factor, mainly because fewer sampling points are available that satisfy the decision rule (5.6) and thus, SRID is more susceptible to noise. When SNR= $4dB$ and $D = 16$, P_m grows up to 6%. Under practical SNR ranges (above $9.7dB$), however, P_m is consistently below 1% for all the clock-rates. In addition, SRID shows a comparable detection performance with 802.11. In fact, it may have lower P_m when the down-clocking factor D is below 16. This is because SRID uses a longer self-correlation sequence than 802.11 (64 vs. 16),

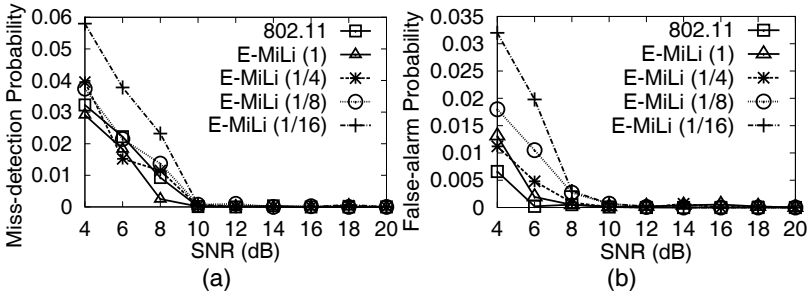

Figure 5.9: SRID performance for a single link.

which increases its robustness to noise. The false-alarm probability P_f in Fig. 5.9(b) shows a trend similar to P_m.

Recall SRID uses nD_m, the spacing between repetitive CGS to convey address n. A natural question is: how large can n be to ensure a high detection accuracy? Fig. 5.10 plots the detection performance as n increases. For a stationary link, both P_m and P_f remain relatively stable. This is because even for the address $n = 100$, two self-correlation sequences are separated by 1600 samples, corresponding to 400 μs at the 4MHz signal bandwidth of USRP, which is well below the channel's coherence time. For a mobile client (created by moving the USRP receiver around the transmitter at walking speed), the detection performance is only slightly affected by the address length, since the low mobility causes SNR variations, but does not change the coherence time significantly.

5.7.1.2 Testbed

We proceed to evaluate SRID on a testbed consisting of 9 USRP2 nodes (1 AP and 8 clients) deployed in a laboratory environment with metal/wood shelves and glass walls. Fig. 6.10 shows a map of the node locations. Node D is moving between point D and E at walking speed, and all others are stationary. This testbed enables the evaluation of SRID in a real wireless environment subject to effects of multipath fading, mobility, and NLOS obstruction. More importantly, it allows testing the false-alarm rate due to cross-correlation between different node addresses.

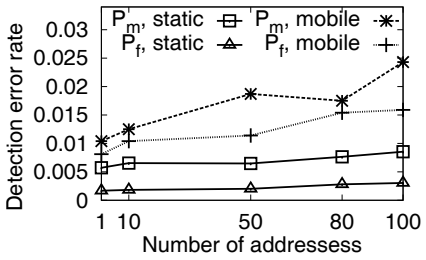

Figure 5.10: Detection performance vs. the number of unique addresses.

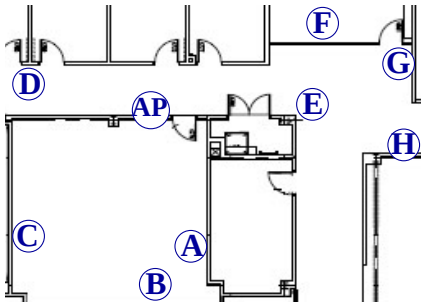

Figure 5.11: Network topology for evaluating SRID in a testbed.

Due to the limited number of external clocks, we create the effect of downclocking by changing the USRP2's decimation rate, so that the receiver's sampling rate becomes 1 to $\frac{1}{16}$ of the transmitter's. We allow the AP to send 10^6 packets to each client in sequence. Fig. 5.12(a) shows that, depending on node locations, P_m varies greatly. In general, nodes farther away (*e.g.*, H) or obstructed by walls (*e.g.*, F) from the AP has higher P_m. The mobile node D may have higher P_m than a node farther from the AP but is stationary (*e.g.*, node E). Consistent with the single link experiment, the downclocking factor 4 results in comparable P_m with 802.11.

Fig. 5.12(b) shows the false-alarm probability due to cross-correlation, *i.e.*, the probability that a client detects packets addressed to others. The relative P_f for different clients shows a similar trend as P_m, depending on the location and mobility. Unlike the single link case, the P_f tends to be larger than P_m, because the cross-correlation between sequences has stronger effects on P_f than pure noise. Remarkably, even for the worst link and with $D = 16$, P_f is below 0.04, implying negligible energy cost due to false triggering. We note that for 802.11, the address field must be decoded from the packet, so P_f here is not meaningful for it.

From the above experiments, we observe that SRID has close to 100% detection accuracy (and is comparable to 802.11) under practical SNR ranges and with downclocking rate up to 16. Hence, it can be used to realize `E-MiLi` in practical wireless networks.

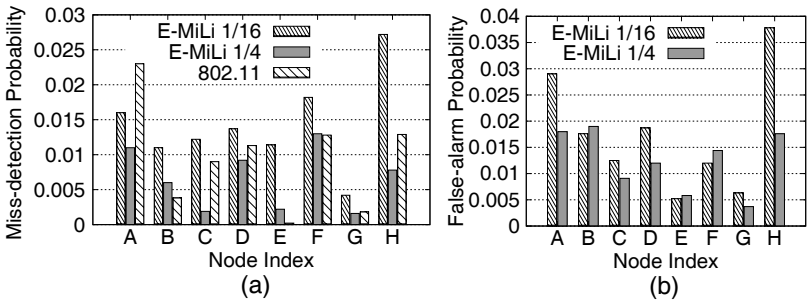

Figure 5.12: SRID performance in a USRP testbed.

5.7.2 Improving WiFi Energy-Efficiency

5.7.2.1 Real WiFi traffic

We now evaluate `E-MiLi`'s energy-efficiency through trace-based simulation. We obtain WiFi and USRP power-consumption statistics from actual measurements (Sec. 5.3). We use the $151\mu s$ switching time of the Atheros AR5414 NIC as the worst-case estimate of switching delay, assuming the power consumption during clock switching is the same as in full-clocked mode. As we will clarify, an outage due to the switching delay occurs with a less than 4.2% probability, so we assume an outage event does not affect the WiFi traces except causing one retransmission. In addition, we adopt the P_m and P_f values at $8dB$ as a conservative estimation of the packet loss or false alarm caused by SRID. Unless mentioned otherwise, 15 addresses are allocated and shared among all clients, and a history size of 5 is used in ODoc.

Energy savings. Fig. 5.13(a) illustrates the energy-saving of `E-MiLi`, assuming clients are using WiFi devices with a maximum downclocking factor of 4. For a large network (SIGCOMM'08 traces [111]), the energy saving ranges from 41% to 47.3%. Its CDF is densely concentrated—for around 92% of clients, the energy saving ranges between 44% and 47.2%, which is close to the 47.5% energy-saving when a client remains in downclocked IL mode (Sec. 5.3). In a small network (PDX-Powell traces [97]) with less contention, IL induces less energy cost, so the energy-saving ratio of `E-MiLi` is relatively low. However, since IL time still dominates, the median

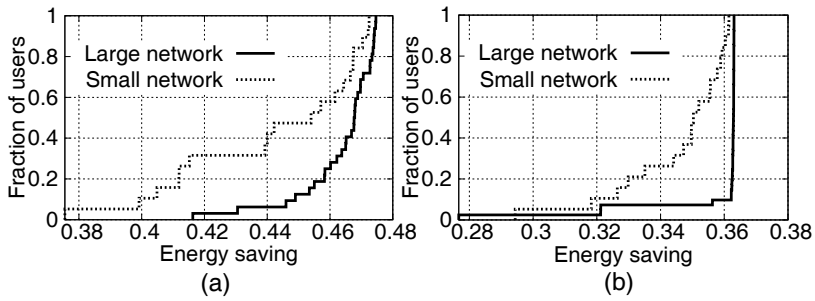

Figure 5.13: Energy saving ratio for (a) WiFi, maximum downclocking factor of 4; (b) USRP, maximum downclocking factor of 8.

saving remains around 44%, and minimum 37.2%. Fig. 5.13(b) plots the results assuming clients' power consumption is the same as the USRP device with a maximum downclocking factor of 8. Again, the energy-saving is concentrated near 36.3%, the saving in pure IL mode (Sec. 5.3).

These experiments reveal that `E-MiLi` can explore the majority of IL intervals to perform downclocking. Its energy-saving ratio can be roughly estimated as $\eta = \eta_c P_{IL}$, where η_c is the energy-savings ratio in pure IL mode using the maximum downclocking factor, and P_{IL} the percentage of idle listening energy during a radio's lifetime. Since P_{IL} is close to 1 for most clients, η is close to η_c.

Overhead of `E-MiLi` and effect of ODoc. The overhead of `E-MiLi` comes from misdetection (and retransmission) due to a packet arriving in between the switching time. Such events can be alleviated by ODoc's history-based outage prediction mechanism. In this experiment, we evaluate the cost of such outage and the effectiveness of ODoc in alleviating it. Fig. 5.14(a) shows that when history size equals 1, 4.2% packets may need to be retransmitted for some clients. With a history size of 10, retransmission is reduced to below 0.8% for 90% of clients. A further increase of the history size to 100 shows only a marginal improvement. On the other hand, Fig. 5.14(b) shows a small history size results in higher energy-efficiency, implying that the energy savings from aggressive downclocking dwarfs the small waste due to retransmissions. Hence, a small history size is preferable for ODoc if energy-efficiency is of high priority.

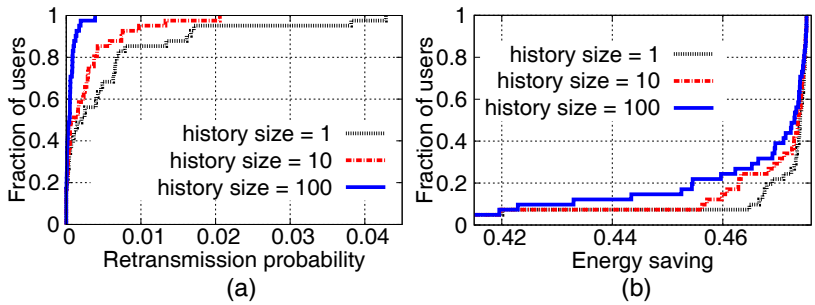

Figure 5.14: Effects of history size (SIGCOMM'08 trace).

5.7.2.2 Synthetic traffic patterns

To further understand `E-MiLi`'s benefits and cost under controllable network conditions, we implement and test it in ns-2.34. We compare performance of the legacy WiFi (including both CAM and PSM), and `E-MiLi`-enhanced WiFi (referred to as CAM+E-MiLi and PSM+E-MiLi). We modified the PHY/MAC parameters of ns-2 to be consistent with that in 802.11g, and fix the data rate to 6Mbps. We implement the ODoc based on 802.11, and configure it in a similar manner to the trace-driven simulator. The PSM module builds on the 802.11 PSM extension to ns-2 [46], and the power consumption statistics follow our measurement of AR5414 (Sec. 5.3). We evaluate two applications: Web browsing and FTP, which have different performance constraints.

Web browsing. We simulate a web browsing application using the PackMIME http traffic generator in ns-2, which provides realistic stochastic models of HTTP flows. The network consists of one HTTP server connecting to a WLAN AP via an ADSL2 link, with 1.5Mbps (0.5Mbps) downlink (uplink) bandwidth and exponentially distributed delay with mean 15 ms. The AP serves one HTTP client (with mean page request interval of 30s) and multiple background clients. Similar to [106], we study the effect of background traffic by running fixed-rate (200Kbps, 512-byte packet size) UDP file transfer between the AP and the background clients.

Fig. 5.15(a) shows the energy usage of a 5-minute web-browsing session. PSM

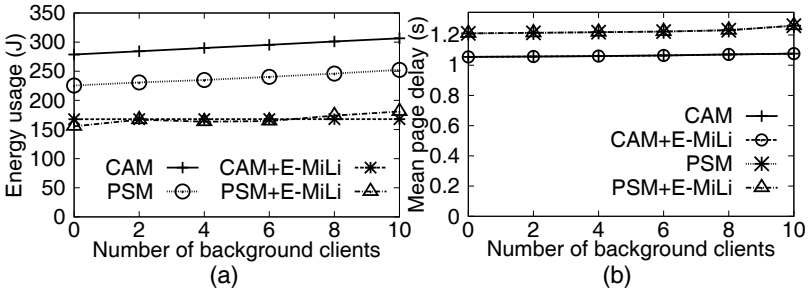

Figure 5.15: Performance of a 5-minute web browsing session.

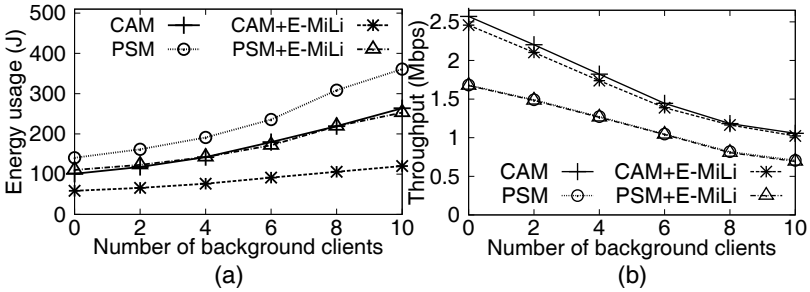

Figure 5.16: Performance when downloading a 20MB file using FTP.

shows around 18% energy saving over CAM. CAM+E-MiLi saves 39.8% of energy over CAM without background traffic, and 47.1% when the number of background clients grows to 10. Since PSM optimizes the sleep schedule of clients, the ratio of IL time is less, compared to CAM, and thus PSM+E-MiLi achieves less energy saving (33% to 37.1%) than CAM+E-MiLi. Also, note that `E-MiLi` is relatively insensitive to background traffic, as it can enforce address filtering even at low clock-rate.

Fig. 5.15(b) plots the average per-page delay during the web-browsing session. Clearly, `E-MiLi` incurs a negligible delay when integrated into legacy WiFi. Although the M-preamble and clock switching costs channel time, it is much shorter than the network and contention delay. Notably, PSM incurs a longer delay than CAM due to its sleep scheduling mechanism, and CAM+E-MiLi has a shorter delay, yet higher energy-efficiency than PSM. We expect an even better energy-delay tradeoff to be achieved by jointly designing the PSM sleep scheduling algorithm and `E-MiLi`. We leave such an optimization as our future work.

FTP. We proceed to evaluate `E-MiLi` using the FTP traffic generator in ns-2, as-

suming a client downloads a 20MB file (with packet size 1KB) directly from the AP. Compared to the fixed-duration web-browsing, the FTP's energy usage is more sensitive to the background traffic (Fig. 5.16(a)), because the downloading duration is prolonged by MAC-layer contention. PSM is found to consume 36.8% to 39.4% more energy than CAM, due to the fact that it may result in higher energy-per-bit than CAM [12]. In addition, although `E-MiLi` achieves a similar level of energy saving as in the Web browsing, it may degrade the FTP throughput by up to 4.4% in the absence of background traffic (Fig. 5.16(b)). This is due mainly to its overhead, *i.e.*, the switching delay, the extra channel time of the M-preamble, and the imperfect detector and outage predictor that incur MAC-layer retransmissions. Moreover, note that we assume no end-to-end delay and the throughput depends only on MAC contention, which zooms in the overhead from `E-MiLi`.

One caveat to `E-MiLi` is that the overhead of the M-preamble and the switching delay are fixed, whereas the channel time for transmission of useful data decreases as the data rate increases. The overhead of `E-MiLi` will thus be amplified at a high data rate. We illustrate this effect by varying the PHY-layer data rate for a file transfer (using FTP) with the number of contending clients fixed at 6. Fig. 5.17 shows that as the data rate increases, CAM+E-MiLi causes CAM more throughput degradation, and the amount of energy saving decreases due to the longer time in transferring the data. When the data rate reaches 54Mbps, CAM+E-MiLi degrades the throughput of CAM by 17.6% (the actual throughput is much lower than 54Mbps due to inherent overhead and collision induced by TCP when running over 802.11 [53]), while saving 23.1% of energy. However, when taking advantage of the short switching delay of recent WiFi chipset (*e.g.*, 9.5 μs in MAXIM 2831), the throughput degradation is negligible, and the energy saving ratio is consistently around 40% for all data rates. In addition, `E-MiLi` sees no throughput degradation when integrated with PSM, and the resulting energy saving is kept around 30%.

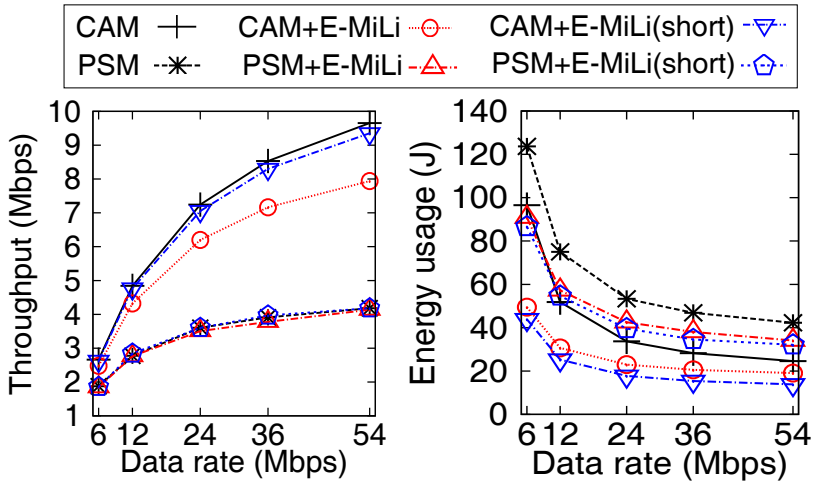

Figure 5.17: FTP performance when data rate varies. 'E-MiLi(short)' denotes E-MiLi with a short switching time (9.5 μs).

It should be noted that the effect of fixed preamble overhead is an inherent problem of high data-rate 802.11 protocols, and can be resolved by standard solutions such as the packet aggregation in 802.11n. Further, the effects of overhead of E-MiLi becomes less severe in a busy network, where contention is high and the channel time consumed by preamble and switching overhead becomes negligible compared to the contention delay. In addition, throughput is a critical metric only for rate-intensive applications like FTP. Mobile wireless devices are more likely to be dominated by elastic traffic such as VoIP and HTTP. Such traffic patterns tend to incur a significant amount of idle listening time which overwhelms the channel time consumed by `E-MiLi`'s preamble and switching delay. As already exemplified in our Web browsing experiments, they can make substantial energy saving by using `E-MiLi`.

5.8 Discussion

Scalability to MIMO radios. The overhead of `E-MiLi` is fixed even if the NIC were equipped with a MIMO transceiver. The overhead of `E-MiLi` mainly comes from the preamble and the clock switching delay. For MIMO systems such as 802.11n, all the RF chains of a receiver detect a single preamble embedded in each packet, and

then uses different preambles for channel estimation. Similarly, when using `E-MiLi`, they can share the same M-preamble for packet detection. In addition, the clock switching delay depends on the PLL settling time of each RF chain. Modern MIMO transceivers may either allow the RF chains to share the same PLL [107], or equip each RF chain with a separate PLL [76]. In the former case, the switching delay is fixed and shared among all RF chains. In the latter case, the settling time of all RF chains is similar and can overlap with each other.

In summary, neither the preamble overhead nor the switching delay increases with the number of MIMO RF chains. Therefore, `E-MiLi` works for modern MIMO NICs without introducing any extra overhead compared with the case of SISO NICs.

Enabling virtual carrier sensing.[2] An `E-MiLi` receiver employs SRID to detect packets intended for itself, and is able to carrier-sense other packets via energy detection. However, energy sensing alone may not be enough to address a pathological case, *i.e.*, the hidden terminal problem. In IEEE 802.11, virtual carrier sensing is an optional solution, which requires an RTS/CTS handshake before the actual data transmission. The RTS/CTS packet piggy-backs a duration of the forthcoming data packet. Neighboring transmitters overhear the RTS/CTS and extend the channel's busy time by the corresponding duration.

In `E-MiLi`, to enable virtual carrier sensing, the RTS/CTS may need to trigger all neighboring transmitters to restore to full clock-rate and decode the duration field, which incurs substantial energy cost. One possible solution to this problem is to embed the packet duration information after the broadcast preamble, and modulate such information in a redundant manner. For example, if the maximum downclocking rate is 4, then each bit of the packet duration information will be repeated 4 times by the transmitter, so that even if the receiver is downclocked by 4, it can still decode the information. We leave the detailed implementation and evaluation of such an

[2]This issue was brought up by Sunghyun Choi during the ACM MobiCom 2011, Las Vegas, NV, U.S.A.

approach for future work.

Association process. When `E-MiLi` coexists with legacy WiFi, the AP needs to discriminate them and prepend the M-preamble only for packets destined for `E-MiLi`-capable clients. The discrimination should be initialized during the association process, when a newly-joining `E-MiLi` client notifies the AP about its capability, and subsequently the AP runs the address allocation algorithm to assign an address to it (and possibly reassign addresses to existing `E-MiLi` clients using the address allocation algorithm).

5.9 Related Work

Energy-efficient protocols for WiFi. Energy-efficiency has long been a paramount concern for portable WiFi devices. Many MAC-level scheduling protocols have been proposed to reduce the energy wasted by IL. For example, NAPman [106] carefully isolates PSM clients' traffic using an energy-aware fair scheduler, so as to reduce unnecessary IL caused by background traffic. SleepWell [89] further isolates the traffic from different WLAN cells, by scheduling their wakeup time in a distributed TDMA manner. μPM [85] adopts a more fine-grained scheduler that aggressively puts clients to sleep even in between short packet intervals. `E-MiLi` can be integrated with these and other MAC-level energy-saving solutions, by adding the downclocked IL mode into their state machine (Sec. 5.6.2). `E-MiLi` can also work in CAM, thus overcoming the excessive delay typically seen in PSM-style protocols.

An alternative way of reducing the cost of IL is to wake up the receiver *on demand*. The wake-on-wireless scheme [115] augments a secondary low-power radio for packet detection, and triggers the primary receiver only when a new packet arrives. `E-MiLi` also adopts the philosophy of on-demand packet processing. Its energy saving may be less than wake-on-wireless, because it needs to keep the analog circuit active in IL. Its advantage is that no extra radio is required. In fact, it only requires a change of

firmware to support the construction and detection of M-preamble, and adjustment of clock-rate. `E-MiLi` can also be used with wake-on-wireless to optimize the power consumption of the secondary radio.

Low-power listening in sensor networks. In sensor networks, a popular MAC-layer energy saving mechanism is low-power listening (LPL), which is used by S-MAC [129], B-MAC [98] and many derivatives. Since sensor networks typically run low-rate, small duty-cycle applications, LPL shifts more power consumption to the transmitter side, thus reducing the time spent in idle listening. Specifically, a receiver periodically wakes up to detect packets from the transmitter, and the transmitter uses a long preamble that spans that period to ensure detectability. Similar to the WiFi's PSM, LPL is a sleep scheduling mechanism that reduces the IL time, and can be enhanced by integrating with `E-MiLi`. For example, since `E-MiLi` reduces IL power, it can shorten the receiver's wakeup period, thereby shortening the transmitter's preamble length and lowering its power consumption.

Packet detection. The general idea of correlation-based packet detection is not new. As mentioned in Sec. 5.5.3.2, the 802.11 OFDM PHY incorporates a preamble that allows self-correlation-based detection. Its variants have also been used in other software-radio implementations [72]. In `E-MiLi`, we have designed a new preamble mechanism that preserves the self-correlation property even when it is downsampled. Cross-correlation-based packet detection (*i.e.*, correlating the incoming signal with a known sequence) is an alternative way of detecting packets [113, 112], but cannot detect downsampled signals and is more susceptible to the frequency offset.

Dynamic voltage-frequency scaling (DVFS). DVFS is a mature technology used in microprocessor design [43]. It exploits the variance in processor load, lowering the voltage and clock-rate when few tasks are pending, and raising it when the processor is heavily loaded. It has also been proposed for Gigabit wireline links [114],

and for audio signal processing [36]. The key idea is to observe the peak frequency of the incoming workload, and then limit the processor's clock-rate to that level.

DVFS has not been used for improving the energy-efficiency for wireless radios, due mainly to a well-known paradox: the radio should be activated only after detecting a packet, but to detect the packet, the radio must always be active at its full sampling rate. We overcome this paradox by separating packet detection and decoding, and performing both at different rates. Our approach is partly inspired by the experiments by Chandra *et al.* [31], who found WiFi NIC's power consumption to scale linearly with the sampling bandwidth, and proposed the SampleWidth algorithm to adjust the bandwidth according to the traffic load. SampleWidth uses the same clock-rate for detection and decoding, and can only adjust clock-rate at a coarse-grained level, because the transmitter and the receiver must agree on the same clock-rate before packet transmissions.

5.10 Conclusion

We have presented `E-MiLi`, a novel mechanism for reducing the energy cost of idle listening (IL) that dominates the energy consumption in WiFi networks. Our goal was to exercise fine-grained IL power control by adjusting clock-rate without compromising packet-detection capability. We met this goal by devising a sampling-rate invariant packet detector, which enables a downclocked radio to detect packets with accuracy comparable to that of a full-clocked radio. We have also introduced an opportunistic downclocking scheme to balance the overhead in changing clock-rate and minimize its negative influence on network performance. Our experimental evaluation and trace-based simulation confirm the feasibility and effectiveness of `E-MiLi` in real WiFi networks with different traffic patterns.

`E-MiLi` has wider implications for wireless design than what we have explored in this chapter. Its simple MAC/PHY interface facilitates its integration with other car-

rier sensing based wireless networks, such as ZigBee sensor networks. In addition, we only explored the benefits of downclocking in `E-MiLi` due to hardware limitation. By changing the voltage along with clock-rate, additional energy savings can be achieved. This is a matter of our future inquiry.

CHAPTER VI

Redesigning the MIMO Cooperation Mechanism

6.1 Introduction

The notion of addressing interference in wireless networks has changed considerably in the past several years with the advent of MIMO. Traditional single-antenna communication protocols (*e.g.*, 802.11a/b/g) allow only one data stream to be sent in each contention domain (*e.g.*, a WLAN cell), and treat all other streams as interference (Fig. 6.1(a)). Recent MIMO technology like 802.11n enables multiple data streams to be sent concurrently between a pair of transmitter and receiver. More advanced protocols like multi-user MIMO (MU-MIMO) are being standardized (in IEEE 802.11ac [6]) that allow a multi-antenna access point (AP) to send different data streams in parallel to multiple clients.

Theoretically, MIMO allows the number of concurrent data streams to scale linearly with the minimum of the number of transmit and receive antennas [125]. Unfortunately, in current MIMO and MU-MIMO standards, the scalability is limited within each WLAN cell that contains multiple clients and one AP mounted with several co-located antennas. The APs in different WLAN cells operate independently and must contend for channel access. Network MIMO (netMIMO) [50], also referred to as distributed MU-MIMO, has the potential to eliminate inter-cell interference by allowing APs to tightly synchronize and share data packets. An ideal netMIMO

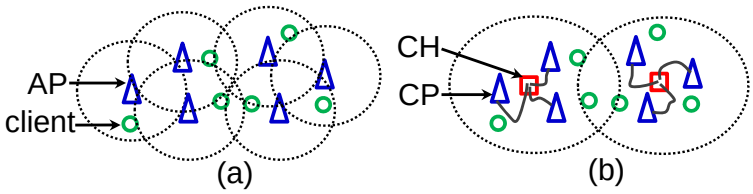

Figure 6.1: Architecture: (a) 802.11 WLAN; (b) NEMOx.

system can scale the downlink capacity linearly with the number of transmitters. Several works have looked into the design of netMIMO by combining distributed APs into one giant-MIMO [50, 52, 84]. They showcased the feasibility of netMIMO in small scale topologies, where APs fully synchronize at carrier-signal level using a GPS clock and share information through a wireline backhaul. Given such schemes, an obvious and important question is: *are these netMIMO gains scalable as the network size increases?* While a straight-forward approach would be to directly extend these solutions to larger networks, this would not be feasible (scalable) in terms of synchronization overhead. Also, the need for sharing data packets and channel state in realtime will easily exhaust the backhaul capacity even with a fiber connection [50].

Our Proposal: We believe the eventual means of leveraging netMIMO gains is to adopt a hierarchical architecture for coordinating the distributed transmitters. We realize this vision by proposing *NEMOx*, a practical system for scaling the capacity of large-scale WLANs. NEMOx decomposes the network into cooperating *clusters* (Figure 6.1(b)). Each cluster consists of one central cluster head (CH) that controls a set of distributed transmitters (referred to as *cooperating points* (CPs)).

For scalability, NEMOx allows each cluster to realize netMIMO independently, eliminating the need for synchronization and information exchange across clusters. With such a communication model that is extremely conducive for practical realization, one might wonder if direct application of existing netMIMO schemes within each cluster would achieve scalability. The answer is not optimistic, since the lack of coordination between clusters and the resulting interference may nullify the net-

MIMO gain within each cluster, thereby degrading the performance to worse than one without netMIMO. Hence, *addressing the scalability problem is fundamental to the realization of netMIMO gains (even within clusters) in any wireless network of practical size.*

NEMOx addresses the challenges associated with scalability through a joint MAC-PHY solution, whose components reinforce each other: (i) efficient channel access to leverage spatial reuse between clusters through interference avoidance - allows for throughput to scale with the number of clusters without hurting netMIMO performance within each cluster; and (ii) optimizing the netMIMO strategy within each cluster - allows for throughput to scale with the number of CPs in each cluster, while promoting opportunities for reuse across clusters.

(1) *Channel access mechanism across clusters.* In NEMOx, we advocate to maintain maximal compatibility with 802.11ac MAC, and adopt a CSMA-based framework for interference avoidance between clusters. The nature of operation in netMIMO, however, poses unique challenges in meeting this goal: the sensing and transmission units (*i.e.*, CPs - analagous to APs in WLANs; Fig. 6.1) collectively span a large area. This results in CPs within a cluster sensing disparate, asynchronized channel status, whereas their transmissions need to be synchronized to realize the netMIMO gain. Further, the lack of coordination and resulting interference between CPs across clusters can hurt not only reuse but also netMIMO performance within clusters. NEMOx accounts for these challenges and forward-engineers a CSMA-based mechanism with provable throughput optimality and fairness properties. To the best of our knowledge, this is the first decentralized medium access mechanism for network MIMO and could be of independent interest as well.

(2) *Communication strategy within clusters.* NEMOx incorporates two novel optimizations to maximize the netMIMO gain from CP cooperation within each cluster: (i) *Joint power control and client selection*: While the netMIMO scheme can be

devised by a migration of the 802.11ac precoding algorithm [14], we show through experiments that this could lead to a sub-optimal performance when the CPs are distributed. We propose to mitigate the effect by jointly optimizing the precoding scheme, the set of clients to serve, and power allocation from CPs to clients. (ii) *Intelligent CP suppression*: We observe that the cooperation gain in one cluster may come at the cost of spatial reuse in neighboring clusters. We balance this tradeoff via intelligent CP suppression, whereby each CP locally evaluates the netMIMO gain it contributes to its cluster, and refrains from transmission if it infers a higher potential for reuse in a neighboring cluster.

Implementation: We have prototyped NEMOx's PHY layer optimization on a small-scale testbed of two clusters using WARP radios. netMIMO within each cluster is realized through a distributed antenna system (DAS), whereby CPs (remote antennas) are wired to the CH through low-cost RF cables. Sharing a central processing unit at the CH, the distributed CPs achieve carrier-level synchronization without an external clock, and real-time data sharing without a backhaul network, thereby making it possible to realize netMIMO transmission within each cluster. Through detailed experiments, we find that NEMOx's components allow netMIMO gains to scale with the number of CPs within each cluster and yield around 7dB SINR improvement over existing schemes. Detailed simulation-based experiments over larger topologies reveal the effectiveness of its joint MAC-PHY solution in sustaining scalability with respect to number of clusters and yielding more than 3 folds gain over current wireless LAN architecture and existing netMIMO schemes. NEMOx confines its main operations to the CH and requires no modification to client receivers. It preserves the signal processing modules in 802.11ac hardware, thus making deployment feasible by simply upgrading the CH's driver and distributing its transmission points (antennas) using RF cables.

Contributions: Our contributions are multi-fold.

(a) Proposed a scalable, hierarchical architecture for netMIMO in wireless networks that realizes netMIMO efficiently within small clusters and employs asynchronous random access to enable spatial reuse across clusters.
(b) Forward-engineered a decentralized, provably optimal and fair random access MAC that efficiently addresses interference between neighboring netMIMO clusters.
(c) Optimized the netMIMO scheme that accounts for joint power control and client selection within each cluster, while reinforcing spatial reuse across clusters.
(d) Built a working prototype of NEMOx based on WARP that verifies the effectiveness of NEMOx in small topologies; complemented with detailed packet-level simulation for large-scale networks.

6.2 Preliminaries

Network Model. NetMIMO requires the cooperating transmitters to be perfectly synchronized (*w.r.t.* packet transmission time, sampling clock-rate, center frequency and phase). This is prohibitively difficult for large wireless networks with distributed transmitters. Hence, in reality, cooperation can only be realized in small localized neighborhoods with several transmitters [108].

While the distributed transmitters within each neighborhood can be networked through an ethernet backhaul, data sharing and synchronization would be hard to realize even for moderately sized neighborhood (5–10 transmitters) [50]. A more practical approach is to realize netMIMO through a distributed antenna system (DAS). With a single transmission source (CH) driving the distributed transmision antennas (CPs) at the physical layer, tight synchronization is achieved by way of sharing the same clock. All the netMIMO processing is run by the CH, with data sharing coming for free. DAS is becoming a popular radio architecture and a cost-effective way of deploying next-generation small cells [44, 45].

NEMOx adopts a hierarchical DAS-based architecture (Fig. 6.1(b)): each NEMOx

cluster is realized as a DAS with one CH and multiple distributed transmitters (referred to as cooperating points, or CPs). NEMOx limits the tight PHY-layer synchronization and cooperation to CPs within each cluster, and uses a CSMA-based MAC to arbitrate the channel sharing of CPs between neighboring clusters. Given a network of N clusters, each with m CPs (and m clients), allowing for performance to scale optimally with the network size (*i.e.*, the total number of CPs Nm) would require synchronization among all Nm CPs and incur a data sharing and channel estimation overhead of $O(N^2m^2)$. NEMOx's communication model allows for a scalable realization of netMIMO gains. Its data sharing (coming for free with channel estimation overhead ($O(Nm^2)$) and synchronization is restricted to clusters of m CPs each.

Note that the over-arching goal of NEMOx goes beyond the hardware realization of netMIMO within each cluster. Hence, NEMOx's contributions are equally applicable to other netMIMO realizations as well (and not specific to DAS-based). Further, while we consider single antenna clients and the CP to be a single remote antenna in our exposition, we discuss extensions to multi-antenna CPs and clients later (Section 6.8).

Communication Scheme. Similar to existing multi-user MIMO communications schemes like 802.11ac, NEMOx exploits the netMIMO gain for *downlink* transmissions, albeit over a spatially distributed network. NEMOx can exploit the *multiplexing gain* within each cluster: it allows multiple data streams to be sent concurrently, each targeting a different client. From the PHY layer perspective, this is realized using a precoding algorithm called zero-forcing beamforming (ZFBF) [125], which sends a linear combination of the data streams through each CP, such that unwanted streams (interference) cancel each other at each individual client, leaving only the desired one. The coefficients of the linear combiner form a two-dimensional matrix called *precoding matrix*, with columns representing data streams (or clients) and rows representing the transmitting CPs. By designing the precoding matrix properly,

ZFBF can also harvest the MIMO *diversity gain*: multiple CPs can send the same streams of data, and the signals are coherently combined at a receiver, to provide a logarithmic increase in capacity.

NEMOx adopts ZFBF mainly because of its simplicity (linear precoding/decoding complexity) and compatibility with existing protocols (*e.g.*, 802.11ac). It can also incorporate other communications schemes (*e.g.*, dirty-paper coding [125] and interference alignment [52]) as intra-cluster communication schemes, while preserving network-level scalability using its MAC.

There are two further aspects to NEMOx's network architecture and communication model that need careful consideration.

How to cluster the transmitters? A cluster refers to a set of cooperating transmitters (CPs). Note that the interference experienced by a link can be eliminated by allowing the interfering transmitter to cooperate. However, the resulting cooperation gain is directly proportional to the strength of interference. Hence, a cluster (cooperating set) can be defined with respect to every transmitting link based on the interference perceived (*dynamic cluster*). On the other hand, a cluster can be defined by statically grouping a set of neighboring transmitters without being link specific (*static cluster*).

While dynamic clusters accurately account for interference at every link, since the cooperating set varies from one link to another, this results in incomplete (and variable) interference information across neighboring links and hence conflicting precoding vectors. This makes it hard to design a netMIMO scheme with scalable network performance. Furthermore, since backhaul connectivity is needed between all transmitters in the cluster, dynamic clusters make it hard to add and remove links from the network from the perspective of deployment. Due to the overlapping nature of clusters across neighboring links, synchronization would also be needed across the entire network.

On the other hand, static clusters represent a disjoint set of clusters in the network that are not link-specific, making it scalable from a deployment perspective. Further, with statically defined cooperating sets, each transmitter in the cluster has access to all the information relevant to the cluster to deliver scalable netMIMO gains. The drawback of this is that links closer to the edge of the cluster will receive interference from neighboring clusters. The interference across clusters can, however, be addressed through the design of an asynchronous MAC mechanism for the clusters. This would also result in synchronization being needed only within each cluster.

What is the mode of communication? With asynchronous nature of channel access across clusters being inevitable for scalability, it is likely that different CPs in the same cluster will experience different levels of interference. Hence, employing all the CPs jointly for netMIMO may not be feasible. However, it is possible for some intially unavailable CPs to later become available during the execution of netMIMO. It is possible for such CPs to asynchronously join the netMIMO operation (*asynchornous netMIMO*) or wait for the next opportunity (*synchronous netMIMO*).

An approach based on interference alignment and cancellation (IAC) [84] allows for asynchronous transmitter cooperation, *i.e.*, additional streams can be initiated incrementally even if some streams have already been transmitted. Asynchronous NeMO may better leverage the available transmission opportunities in a single cluster. However, with multiple clusters, even if such transmission opportunities are not used in one cluster, they can be exploited by another neighboring cluster. Hence, it is not clear if asynchronous NeMO within clusters will contribute to a better performance (network-wide) compared to synchronous NeMO. Further, with asynchronous NeMO, data sharing cannot be leveraged effectively and hence its performance within a cluster will be bounded by the number of antennas on a link (and not transmitters) unlike in synchronous NeMO. In addition, synchronous NeMO is also a simpler and more practical from an implementation perspective.

Based on the pros and cons of the various aspects, *NEMOx employs a static clustering of the network, with synchronous netMIMO being realized through a DAS within each cluster, and interference across clusters handled through an asynchronous MAC.*

6.3 Design Guidelines

In this section, we highlight important tradeoffs that affect the efficiency of netMIMO, thereby motivating the design elements in NEMOx. We develop a C++ simulator that schedules interfering links in a randomized (but centralized), round-based manner across clusters, but allows concurrent scheduling of multiple links (CP→client) within the same cluster. We use an empirical propagation model recommended in 802.15 for 2.4GHz indoor environment [66]. With -81dBm carrier sensing threshold, -96dBm noise floor, and 20dBm transmission power (typical parameters for WiFi), the equivalent carrier sensing range is 155m and transmission range 80m under 10dB SINR threshold. The topology is generated such that clusters partially overlap to ensure full coverage of a rectangular area, the size of which varies with the number of clusters.

Mode of Cooperation. Unlike conventional omni-directional antenna networks, NEMOx needs to manage the cooperation between geographically distributed transmitters (*i.e.*, the CPs) within each cluster, and allow them to collectively contend with CPs in neighboring clusters. Given the decentralized and asynchronous operation across clusters in practice, different CPs in a cluster may experience inconsistent channel states or interference patterns at a given time. Unfortunately, these CPs can realize netMIMO operation only when all of their transmission attempts are synchronized (*i.e.*, each senses an idle channel and acquires it at the same time).

Emphasizing Cooperation: A straightforward approach is to wait for the opportunity when all CPs in a cluster are available (referred to as *FullCoop*). While such

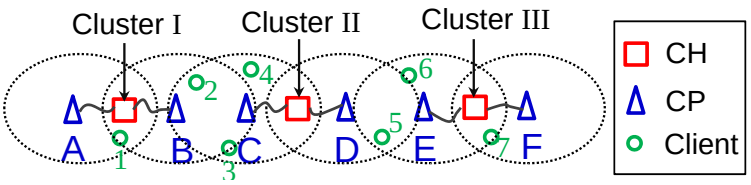

Figure 6.2: An example topology in NEMOx with 3 clusters each containing 2 CPs. Dotted circle represents a CP's interference range.

an approach would provide maximum netMIMO gain in each cluster, the strict binding of CPs can severely hurt the spatial reuse across clusters, especially when clusters span CPs with large spatial separation and hence weakly-coupled channel states. FullCoop essentially represents a straightforward extension of existing netMIMO schemes [102, 52, 84] to large networks with multiple contention domains.

Fig. 6.2 illustrates a typical operation of FullCoop. The CPs in cluster II contend with those from two independent clusters I and III. When CP C in cluster II becomes available, it must wait for D to be available as well, so that both can transmit concurrently to maximize the netMIMO cooperation gain. However, by the time D acquires the channel from neighboring CP E in cluster III, C may no longer be available. Hence, by restricting C and D to cooperate all the time, FullCoop loses the spatial reuse between them and may end up starving cluster II.

Emphasizing Spatial Reuse: Being a key factor impacting network performance, it is also feasible to consider an alternate scheme that emphasizes reuse but without CP cooperation. For example, in Fig. 6.2, all the CPs can independently contend for channel access using CSMA. However, this essentially gives up the netMIMO gain and would reduce NEMOx to the current CSMA WLAN architecture (referred to as *MultiAP*).

Therefore, while arbitrating channel contention between clusters, the medium access mechanism in NEMOx must *strike a balance between cooperation and spatial reuse for CPs within the same cluster*. This in turn can be realized by opportunistically grouping available CPs to perform netMIMO transmission (referred to as *OppCoop*).

For example, in Fig. 6.2, while E and F perform netMIMO, D becomes unavailable, but C can proceed to utilize its channel access opportunity (to client 4). The goal of our NEMOx MAC is to realize such opportunistic cooperation (OppCoop), albeit in a *decentralized* manner with provable performance.

Assuming an oracle is available to realize the above three schemes (FullCoop, MultiAP, OppCoop) in a centralized manner, we have used the C++ simulator to compare their scalability performance. Fig. 6.3(a) shows the network throughput as a function of density (m, the number of CP/clients within each cluster). We observe that the capacity of FullCoop and OppCoop increases with network density, due to more concurrency (multiplexing opportunities) brought by netMIMO (higher η_2). FullCoop, a straightforward application of netMIMO, has much lower performance than OppCoop. This is because it greedily waits for the chance to synchronize all CPs, and loses the spatial reuse opportunity (lower η_1) that occurs when CPs' channel status differ from each other. On the other hand, MultiAP's capacity increases initially due to higher spatial-reuse (higher η_1) created by a denser topology. However, the gain soon saturates when interference between links dominates. Thus, OppCoop strikes the best balance between multiplexing within each cluster (η_2) and reuse across clusters (η_1). Fig. 6.3(b) shows the scalability of different schemes with the number of clusters (N). Due to its better spatial reuse, OppCoop achieves much higher scalability than FullCoop. Its network throughput increases almost linearly with the number of clusters, with its per-cluster capacity being much higher than MultiAP due to multiplexing gain.

Inference 1: Due to the localized nature of interference, each CP perceives different channel status, so a fine balance needs to be struck between cooperation and spatial reuse. An intelligent medium access protocol is needed to create opportunistic cooperation for netMIMO, instead of naive synchronization of all CPs.

Power Allocation and Client Selection. A critical component in ZFBF-

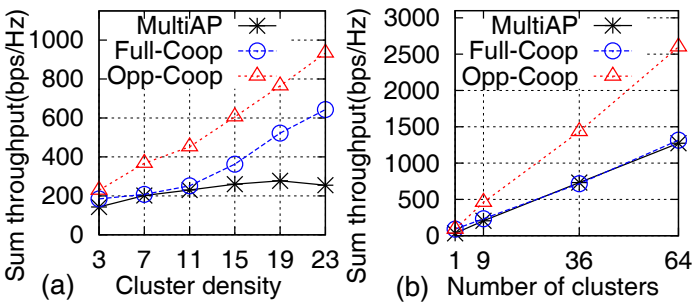

Figure 6.3: Opportunistic netMIMO: (a) N=9, (b) m=10

based netMIMO is the precoding algorithm, which determines the data streams to be sent and the transmit power allocated to each of them. Existing netMIMO schemes consider an equal power scheme, whereby power allocation to data streams is decoupled from precoding – the precoding matrix is obtained directly by inverting the channel matrix (*i.e.*, the matrix representing channel gains from CPs to clients). While this is shown to be optimal in practice [14] for co-located antennas (with comparable channel gains to a given client), it could suffer in netMIMO, where the CPs are spatially distributed, with disparate channel gains to clients, thereby resulting in *unbalanced* topologies. Further, capacity is severely degraded when the channel gain matrix exhibits a large variation between elements.[125, Ch.7.2]. Our simulation shows that random client selection and equal-power allocation perform an order-of-magnitude less than the optimum. Hence, optimizing netMIMO's performance requires not only the design of precoding matrix but also power allocation from each CP to client, and the set of clients to be served on each transmission attempt.

In Fig. 6.4, we use the simulator to compare the equal-power allocation scheme with a joint power-allocation and precoding scheme (see Sec. 6.5 for details). We also augment the schemes with an optimal client-selection procedure that searches for all possible sets of clients to serve and selects the one resulting in the highest capacity. With randomized client selection, equal-power performs an order-of-magnitude less than the optimum. However, with the optimal client-selection, it can achieve close-to-

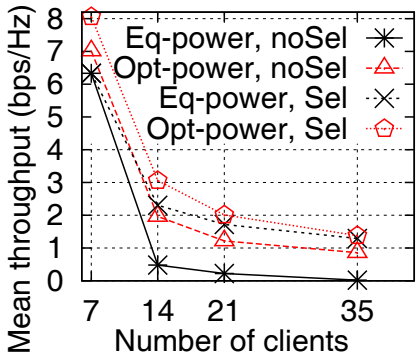

Figure 6.4: Power control; m=7.

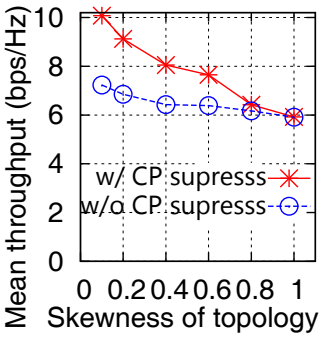

Figure 6.5: Cooperation vs. reuse.

optimal performance, especially when the client population is large. This is because intelligent client selection helps regularize the channel matrix, thereby diminishing the potential for power control on streams. However, for a smaller client density, where client-selection cannot help, a joint power control and precoding scheme is critical to alleviate the imbalanced topology and hence deliver significant gains.

Inference 2: Integrating client-selection with the design of netMIMO is necessary to reinforce the gains. In addition, joint precoding and power allocation across CPs is needed to reduce the variance of the channel gains and maximize performance under unbalanced topologies.

Diversity Gain vs. Spatial Reuse. When the number of clients (or the number of active data streams) is smaller than that of CPs in a cluster, multiple CPs may beamform the same data to the same client (*i.e.*, harvest diversity gain). While such cooperation only increases the cluster capacity logarithmically [125], it may linearly decrease the spatial reuse (multiplexing) opportunity of neighboring clusters, where additional streams may be served. Hence, the total network capacity may be reduced if all CPs in each cluster are used greedily towards cooperation. An intelligent netMIMO scheme should opportunistically *suppress* certain CPs, depending on their contribution to the netMIMO gain and the opportunity for reuse in neighboring clusters. For example, in Fig. 6.2, suppose both clusters I and III have one data stream each to clients 1 and 7 respectively, while cluster II has two data streams, one each

to clients 3 and 5. A greedy approach may allow CPs A and B to send the same data stream to client 1 and increase the downlink capacity logorithmically (similarly for CPs E and F). This would prevent cluster II from transmitting, thereby reducing its throughput to 0. Instead, the optimal medium access scheme should suppress B and E and enable transmissions from A, C, D and F, to clients 1, 3, 5 and 7 respectively, thereby allowing all 4 data streams to be sent concurrently.

Fig. 6.5 evaluates the importance of such a scheme with 2 clusters each with 10 CPs in our C++ network simulator. One of them contains 10 clients, while the other contains $10e$ clients, where e is the *skewness* of topology and $0 < e \leq 1$. Clearly, CP suppression has a large impact on the total network capacity—the larger the imbalance, the higher the opportunity for reuse and correspondingly the benefits of CP suppression.

Inference 3: Cooperation within a cluster can come at the cost of spatial reuse in neighboring clusters. Hence, the contribution of every CP towards netMIMO gain must be evaluated in determining its participation.

The above guidelines inspire us to incorporate key components at both MAC and PHY in NEMOx to achieve scalable netMIMO: *(i)* decentralized channel-access mechanism arbitrates CPs' access to the wireless medium through interference avoidance across clusters and enables netMIMO through opportunistic CP cooperation within each cluster; and *(ii)* optimized netMIMO scheme is applied on the set of CPs selected by the MAC in each cluster - precoding matrix is jointly designed with power allocation and client selection, and CP suppression strikes a balance between intra-cluster netMIMO gain and inter-cluster spatial reuse.

6.4 Efficient Medium Access for NEMOx

6.4.1 Overview

A MAC scheme by definition, optimizes a performance objective (throughput) subject to some notion of fairness among contending entities. The real challenge lies in designing MAC (especially distributed) schemes with provable optimality. While several efficient MAC schemes have been proposed for single-antenna (omni) networks [95, 80], this has not been addressed for netMIMO, which is an important, challenging problem in its own right.

CSMA as Basis: In NEMOx, we desire no coordination or synchronization between clusters for scalability. Hence, the channel-access mechanism needs to be asynchronous and decentralized in nature. CSMA provides a light-weight approach to achieving this goal, by enabling clusters to infer and react to interference via avoidance, while allowing for spatial reuse. The resolution of contention (collision) in CSMA mechanisms are traditionally addressed through two approaches: *persistence* (as in [95, 80] *etc.*.) or *backoff* (as in 802.11), where access to the medium is controlled by adapting a contention probability and backoff window, respectively. Backoff based MAC protocols cannot provide performance and fairness guarantees due to the lack of a closed-form throughput characterization. Most efficient MAC protocol designs in literature [95, 80] are based on persistence, which is also the approach in NEMOx.

Modeling Channel Access for netMIMO: As discussed in Sec. 6.3, NEMOx's medium access mechanism faces unique challenges due to the distributed nature of the sensing and transmission units (*i.e.*, the CPs). While allowing for opportunistic grouping of CPs for netMIMO (see inference on opportunistic cooperation from Sec. 6.3) is one way to approach the challenge, the question still remains as to *how to form a cooperating group?* A simple approach is to allow all CPs in a cluster to contend (with each other and with CPs in other clusters) independently and asynchronously. If multiple CPs in the cluster win channel access, they form a group and

send data streams through netMIMO. However, since different CPs may be contending with different clusters, it is hard to find the opportunities where multiple CPs win at the same time, thereby limiting the benefits of netMIMO.

NEMOx solves the problem using a model-driven approach: it first defines the potential groups (referred to as *cooperating sets* (CSets)) in each cluster, and then derives a MAC that allows the groups to contend for channel access. In the simplest form, the CSets in a cluster are equivalent to all its subsets of CPs (CSet determination in Sec. 6.4.4). For example, in Fig. 6.2, the CSets of cluster I include $\{A\}, \{B\}$ and $\{A, B\}$. The CSets contend for channel access with each other, and with those in neighboring clusters to avoid interference. The winning set in each cluster allows its member CPs to perform netMIMO transmission synchronously. Channel-access of each CSet is managed by a distributed *network utility maximization* (NUM) [80] framework, which runs locally at each cluster head (CH) and optimizes throughput efficiency (implicitly leverages reuse across clusters), while ensuring proportional fair access across CPs in the network.

6.4.2 MAC Design via Optimization

Unlike in existing MAC [95, 80], where fairness can be defined with respect to a link (Tx-Rx pair), such a notion does not exist in netMIMO, where a client's rate depends on the set of CPs that jointly serve it. Hence, we define fairness of access with respect to each CP. We handle fairness with respect to clients during the netMIMO operation once a CSet obtains channel access (Sec. 6.5.1). We associate a utility function $U(r_i)$ with each CP, c_i, where r_i is the rate received by it and $U(\cdot)$ is a non-negative, concave, differentiable function. The choice of the utility function determines the nature of fairness model achieved. The MAC design can be forward-engineered through the

following optimization.

$$\text{NCA:Maximize} \sum_{C_m \in \mathcal{C}} \sum_{i \in C_m} \alpha_m U(r_i)$$
$$\sum_{j:S_j \in M_k} q_j \leq 1; \quad \forall M_k \in \mathcal{M}$$
$$\text{where } r_i = \sum_{j:i \in S_j} (1 - P_j) q_j$$

where $\mathcal{C}$ denotes the set of clusters; α_m is the weight used for prioritizing cluster C_m; q_j represents the channel-access probability for each CSet S_j whose collision (contention loss or packet collision from sets in same or other clusters respectively) probability is P_j. We capture the interference relation in the network by defining maximal cliques (M_k) with respect to CSets, where a CSet is to collide with another if at least one of the joint transmissions in the other set suffers (SINR below a certain threshold). Note that one of the implicit clique constraints is that $\sum_{j:S_j \in \mathcal{S}_m} q_j \leq 1$, *i.e.*, only one set be chosen in each cluster, where $\mathcal{S}_m$ captures all the CSets in cluster m. The constraint to the optimization problem is that the net access probability in each maximal clique must be bounded by one to avoid collision. The resulting rate of each CP would be a function of all the CSets it belongs to as well as their collision probability. The output of this optimization problem directly provides the channel-access probability for each CSet.

Two noteworthy aspects of the above optimization are: (i) With utility being defined *w.r.t.* each CP, the above optimization implicitly favors CSets with more CPs and hence higher netMIMO gain. (ii) While the optimization involves knowledge of the maximal cliques and collision probabilities, we now show the problem can be solved in a decentralized manner without the need to compute these parameters directly.

6.4.3 Decentralizing Channel Access

Observing that NCA is a concave optimization problem, we can obtain its Lagrangian as:

$$L = \sum_{C_m \in \mathcal{C}} \sum_{i \in C_m} \alpha_m U(r_i) - \sum_j \sum_{k:S_j \in M_k} \beta_k q_j.$$

We can write β_k as $\beta_k = \beta p_k$, where p_k is the collision probability in maximal clique M_k. Since a CSet can belong to multiple maximal cliques, we have $P_j = 1 - \Pi_{k:S_j \in M_k}(1 - p_k)$. If the collision in each maximal clique is kept small, then we can approximate $P_j \approx \sum_{k:S_j \in M_k} p_k$. Thus, P_j can be approximated by its CSet by just keeping track of the net packet loss experienced (over transmitted packets) without having knowledge of its maximal cliques or loss probabilities in each of them. Now, substituting for β_k and applying the approximation, we have

$$L = \sum_{m:C_m \in \mathcal{C}} L_m$$
$$\text{where, } L_m = \sum_{i \in C_m} \alpha_m U(r_i) - \beta \sum_{j:(i \in S_j)\&(i \in C_m)} P_j q_j.$$

A rigorous analysis [80, 95] showed that the aggregate utility for a system of equations of the above form is maximized when the individual components maximize their own utility. While the component corresponds to a single link in omni-directional communication, this corresponds to a cluster in netMIMO, thereby requiring for joint adaptation of CSets within each cluster for optimality. Applying the KKT conditions for optimality with respect to each CSet, we obtain

$$\frac{dL_m}{dq_j} = \sum_{i \in S_j} \alpha_m U'(r_i)(1 - P_j) - \beta P_j = 0, \ \forall S_j \in \mathcal{S}_m \tag{6.1}$$

To achieve a proportional fairness objective, we employ logarithmic function of the rate for our utility functions ($U(r_i) = \log(r_i)$) [95]. Thus, at the optimum, for

every CSet we have

$$\frac{dL_m}{dq_j} = \sum_{i \in S_j} \frac{\alpha_m(1-P_j)}{r(i)} - \beta P_j = 0$$
$$\frac{dL_m}{dq_j} = \alpha_m(1-P_j) - \beta P_j \left(\frac{1}{\sum_{i \in S_j}(1/r_i)} \right) = 0.$$

While α_m can be used to prioritize different clusters, for an equal bias, we set $\alpha_m = \alpha$. Given that P_j can be locally inferred by each CSet S_j, the above optimality condition can be used to adapt its access probability in a completely decentralized manner as follows.

$$q_j \leftarrow q_j + \alpha - P_j \left(\beta (\textstyle\sum_{i \in S_j}(1/r_i))^{-1} + \alpha \right) \tag{6.2}$$

Such an adaptation has the following key properties.

(i) Joint adaptation within cluster: In NEMOx, the access adaptation for a CSet is coupled with the rate of the constituent CPs, which, in turn, is a function of all the CSets that each CP belongs to—in essence, the access adaptation of a CSet must be performed jointly with other sets in the cluster to achieve optimality.

(ii) Convergence: Though coupled, the adaptation mechanism for each CSet follows a gradient approach based on the KKT conditions. Further, since the utility function $U(r_i) = \log(r_i) = \log(\sum_{j:i \in S_j}(1-P_j)q_j)$ and the resulting Lagrangian are concave with respect to each q_j, there exists a unique maximum, to which the individual gradient-based adaptations converge.

(iii) Practical realization through DAS: An important aspect in realizing optimality through the above adaptation is the joint adaptation of access parameters, which requires the loss, access and rate parameters to be shared and coordinated among CSets within each cluster in real-time. This is hard to realize if distributed transmitters are coordinated on an ethernet backhaul to build netMIMO, but is a natural operation for the DAS architecture in NEMOx, as the MAC operations of all CSets within a cluster is delegated to the CH.

6.4.4 MAC Operations in Each Cluster

Adaptation: The CH runs the above medium access algorithm for each of its CSets, which can be in the Idle, Contend or Acquire states (Fig. 6.6). When all CPs in a CSet S_j sense an idle channel for a short duration (the DIFS defined in 802.11 [6]), S_j enters the idle state, and makes a contention attempt with probability q_j. Specifically, since $\sum_{j:S_j \in \mathcal{S}_m} q_j \leq 1$ (only 1 active CSet) in each cluster, the CH picks a local winning set with probability q_j, and enables netMIMO transmission among constituent CPs. Thus, intra-cluster contention is resolved prior to transmission, while inter-cluster contention is solved through collision and adaptation. The transmission operation can also be preceded by a random backoff window (0,B), where B is a constant window size, to reduce collisions (concurrent transmission attempts). If there is a contention loss (either intra- or inter-cluster with probability $P_j q_j$), the access probability is decremented by $\frac{1}{q_j}(\beta(\sum_{i \in S_j}(1/r_i))^{-1} + \alpha)$, and is incremented by α on a successful transmission. While there are multiple ways to realize Eq. (6.2) by distributing the adaptation between states, we realize it as in Fig. 6.6, where we found the increment of q_j in idle mode to contribute to faster convergence.

Pruning: In each cluster, the number of CSets increases exponentially with the number of CPs. However, not all cooperating sets contribute equally to the cluster's capacity. NEMOx discriminates them by defining a *dominance* relation. A CSet S_i is said to dominate S_j within the same cluster if S_j's CPs is a subset of S_i's, and S_i's interfering CPs in neighboring clusters is a subset of S_j's. In this case, S_i has similar access probability and causes similar interference to competing sets with S_j, but it has a higher multiplexing gain by enabling more concurrent transmissions. Hence, NEMOx eliminates the dominated set S_j to achieve a higher efficiency. Each CP learns the set of interfering CPs in neighboring clusters through carrier sensing, and the CH prunes dominated CSets accordingly. For example, for cluster I in Fig. 6.2, the CSet $\{A, B\}$ dominates $\{B\}$ because they are interfered by the same set of CPs

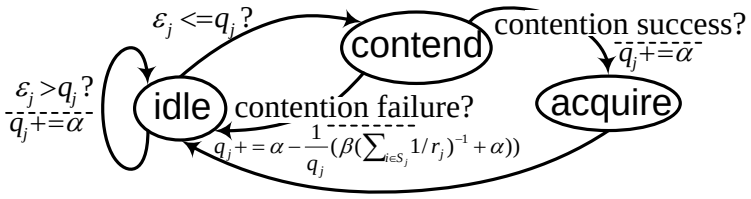

Figure 6.6: Adaptation at each S_j; ϵ_j is a r.v. $\in [0, 1]$.

(*i.e.*, the CP C) in other clusters, and $\{B\}$ is only a subset of $\{A, B\}$. $\{A, B\}$ does not dominate $\{A\}$ because the former's set of interfering CPs is not a subset of the latter's (which is a null set).

6.5 Optimized netMIMO within Clusters

Optimizing the netMIMO scheme within each cluster involves two components: joint precoding, power control and client selection for optimizing netMIMO gain; and CP suppression for creating reuse opportunities across clusters.

The first component is encountered after a CSet wins the channel access, and its constituent CPs need to select and serve a set of clients. This problem involves several coupled factors: *(i)* The number of clients must not exceed the number of CPs in the CSet; *(ii)* A client's achievable rate depends on not just its own channel gain, but also peers served by the same CSet; *(iii)* The power budget of each CP needs to be allocated such that sufficient "cancellation power" is used to eliminate the inter-stream interference, and the remaining power ensures efficient and fair rate allocation to clients.

6.5.1 Joint precoding and power allocation

MU-MIMO downlink transmission in traditional Co-located Antenna Systems (CAS) (*e.g.*, 802.11ac) uses zero-forcing beamforming (ZFBF), *i.e.*, using channel matrix inverse directly as the precoding matrix: $\boldsymbol{v} = \boldsymbol{h}'(\boldsymbol{h}\boldsymbol{h}')^{-1}$. Suppose h_{ik} are elements of the channel matrix $\boldsymbol{h}$, denoting the complex channel gain from CP k to

client i. Let v_{ki} be elements of the precoding matrix $\boldsymbol{v}$, and denote the precoding weight of k-th CP for i-th client (stream). Since $\boldsymbol{hv} = \boldsymbol{I}$ (unity matrix), each client only receives the data intended for itself, while other streams are nullified after precoding and channel distortion. As we observed in Sec. 6.3, a direct application of such a ZFBF scheme to netMIMO may severely degrade its performance. Hence, NEMOx adopts a rigorous formulation for *jointly* designing the precoding matrix and power allocation (for a given client selection), to maximize the aggregate rate, while ensuring proportional fairness among clients:

$$\text{JPP:} \quad \max \quad \sum_{i=1}^{|D|} w_i \log(1 + \frac{P_i}{N_0}) \tag{6.3}$$

$$\text{s.t.} \quad P_i = |\sum_{k=1}^{|S|} h_{ik} v_{ki}|^2, \forall i \in D \tag{6.4}$$

$$\sum_{i=1}^{|D|} |v_{ki}|^2 \leq P, \forall k \in S \tag{6.5}$$

$$\sum_{k=1}^{|S|} h_{jk} v_{ki} = 0, i \in D, j \neq i \tag{6.6}$$

where D and S are the set of clients and CPs, respectively. Eq. (6.5) is the per-antenna power constraint. Eq. (6.6) represents the precoding constraint, *i.e.*, precoded symbols intended for client i should cancel each other when arriving at client j after channel distortion ($i \neq j$). From Eq. (6.6), we see that when the estimated channel matrix for a client j becomes outdated (*e.g.*, due to movement), the interference from other streams may become non-trivial, but the change only affects client j. Therefore, the channel estimation can be on-demand — the CPs may request a client to feedback its channel only if it has high packet loss rate.

In the above formulation, w_i is the weight allocated to each client, and can be adjusted to achieve a certain long-term fairness objective. For *proportional fairness*, we can configure w_i to be the inverse of the time-averaged throughput of client i (denoted as R_i) [125], *i.e.*, for each slot of transmission attempt t, suppose client i achieves a rate of R_{it}, then we update the throughput and weight with a smoothing

factor γ $(0 < \gamma < 1)$ as:

$$R_i \leftarrow \gamma R_{it} + (1-\gamma)R_i; \text{ and } \quad w_i \leftarrow R_i^{-1} \tag{6.7}$$

The objective function of JPP has been proven to be non-convex [127] *w.r.t.* the real and imaginary components of v_{ki}, due to the norm operator in Eq. (6.4). Fortunately, by phase-shifting the vector $v_{ik}, \forall k \in S$ appropriately, we can restrict $\Im(\sum_{k=1}^{|S|} h_{ik} v_{ik}) = 0$, while both constraints (6.5) and (6.6) are invariant to the phase-shift.

To satisfy the additional constraint $\Im(\sum_{k=1}^{|S|} h_{ik} v_{ik}) = 0$, we decompose the variable v_{ki} into real and imaginary part, denoted as r_{ki} and e_{ki}, respectively. Then we reformulate the problem as:

$$\max \quad \sum_{i=1}^{|D|} w_i \log(1 + \frac{P_i}{N_0}) \tag{6.8}$$

$$\text{s.t.} \quad P_i = \left(\sum_{k=1}^{|S|} \Re(h_{ik}) r_{ki} - \Im(h_{ik}) e_{ki} \right)^2 \tag{6.9}$$

$$\sum_{k=1}^{|S|} \Re(h_{ik}) e_{ki} + \Im(h_{ik}) r_{ki} = 0, \forall i \in D \tag{6.10}$$

$$\sum_{i=1}^{|D|} r_{ki}^2 + e_{ki}^2 \leq P, \forall k \in S \tag{6.11}$$

$$\sum_{k=1}^{|S|} \Re(h_{jk}) r_{ki} - \Im(h_{jk}) e_{ki} = 0, \forall i \in D, j \neq i \tag{6.12}$$

$$\sum_{k=1}^{|S|} \Re(h_{jk}) e_{ki} + \Im(h_{jk}) r_{ki} = 0, \forall i \in D, j \neq i \tag{6.13}$$

The resulting problem then becomes convex (since its objective function is concave with respect to the variables r_{ki} and e_{ki} and the constraints are convex), and can be easily solved using standard convex optimization techniques.

6.5.2 Incorporating Client selection

When the number of clients $|D|$ is larger than the number of CPs $|S|$, to solve JPP, we need to first select a set of clients (maximum size $|S|$). This in turn must be done appropriately to achieve the desired efficiency and fairness objective. A straightforward approach is to augment an indicator variable to the JPP to select the optimal set of clients. However, this makes the problem intractable due to the integer and non-linear constraints. So, we propose an alternate, iterative approach for integrating client selection with the precoding and power allocation framework.

Step 1: Solve JPP when all available CPs are beamforming to client i. Pick the client i^* yielding highest utility.

Step 2: Solve JPP by grouping an un-selected client j with existing set of selected clients. Add client j^* yielding highest utility.

Step 3: Repeat Step (2) until current utility is lower than in previous round, or if there are no clients to be selected.

In a cluster, the clients and CPs tend to spread over a large area, and small-scale fading (which causes small magnitude and phase variation) has little impact on the netMIMO capacity. Therefore, NEMOx runs client selection using the latest channel gains, and only estimates the channel from CPs to the selected set of clients, so as to reduce the channel estimation overhead.

6.5.3 Balancing Performance with Complexity

The client selection becomes computationally inefficient when the client population is large, as it involves multiple rounds of non-linear optimization to solve (6.3). This is where we leverage our inference from Sec. 6.3, where we observe that the simple equal-power allocation degrades performance in an unbalanced topology but may have close-to-optimal performance otherwise. We thus characterize the topology imbalance using a ρ-factor. For a given set of CPs, the ρ-factor is the fairness index

[69] with respect to the number of times a CP is *preferred* in the set. Each client ranks the CPs according to the channel magnitude it sees from them, and a CP is *preferred* once if it is the top-rank of one of the clients. Intuitively, the ρ-factor is close to 0 when all clients are concentrated near one CP or when there are fewer number of clients than CPs, and close to 1 when they are evenly distributed among CPs (which is more likely to happen with a large client population). Thus, NEMOx adopts a *hybrid power allocation scheme*, using the ρ-factor as a decision variable – when ρ is above a threshold ρ_0 (we use an empirical value of 0.5), NEMOx employs the simple equal power allocation scheme during client selection, while restricting the optimized power allocation scheme to highly imbalanced topologies. This allows NEMOx to reduce complexity significantly without sacrificing performance appreciably.

6.5.4 CP Suppression

The second component of NEMOx's intra-cluster optimization aims to strike a balance between cooperation gain within a cluster and spatial reuse between clusters. Building on insights from Sec. 6.3, the CH in each cluster decides to opportunistically suppress those CPs that: *(i)* contribute mainly to the diversity gain in the cluster; *(ii)* experience sufficient contention with adjacent clusters and *(iii)* cause interference to neighboring clusters' clients with downlink packets.

Step 1: The CH runs a simple stable matching algorithm. It assigns to each client (with downlink traffic demand) a ranked list of "preferred CPs", based on the channel gain matrix between the CPs and the client, and to each CP a ranked list of "preferred clients". It then runs the classical stable marriage problem between the CPs and clients. CPs not in the matched set are considered to contribute little to the multiplexing gain, and will be suppressed when spatial reuse can be exploited.

Step 2: Each CP i locally computes a *contention factor* (F_i) that reflects the intensity of contention with other CPs in adjacent clusters. This is in turn defined as $F_i =$

$\sum_{j:i\in S_j}(P_j - P'_j)$. For each CSet S_j, P_j and P'_j denote the sum collision probability and contention loss probability due to other sets in the same cluster respectively, both of which can be locally measured. Hence, F_i captures the desired inter-cluster collision probability experienced by CP i. A large F_i value indicates higher contention, and hence a higher potential for spatial reuse to be effectively exploited between clusters.
Step 3: A CP i estimates whether clients in neighboring clusters have traffic demand, by overhearing their CTS packets (Sec. 6.5.5). If so, and if it is unmatched in Step 1, and F_i is larger than a threshold ϕ_i (we use an empirical value of 0.8), then CP i would refrain from leveraging diversity gain, and leave the opportunity for clients in neighboring clusters to exploit spatial reuse instead.

6.5.5 Channel Estimation, Reservation and ACK

To obtain the channel matrix from CPs in a CSet to selected clients, NEMOx uses the 802.11ac [6] channel estimation scheme, but integrates it with an RTS/CTS exchange. Before the transmission attempt, all CPs within a CSet synchronously broadcast the same RTS packet that indicates the transmission duration and clients' addresses. Then, all selected clients return the CTS packet. The (common) header part of the CTS packet contains the duration of this transmission attempt, and is sent by all clients synchronously to reserve channel from transmitters in neighboring clusters. The second part is sent sequentially by each client (following the order of addresses in the RTS), and contains the channel gain information from the CPs to clients. After completing the data transmission, ACK is also sent sequentially by each client in a similar manner. While such an RTS/CTS/ACK scheme incurs overhead, employing it for the channel estimation procedure (inevitable for netMIMO) amortizes its overhead. Further, as will be clarified in our evaluation, the overhead is negligible compared with the netMIMO gains.

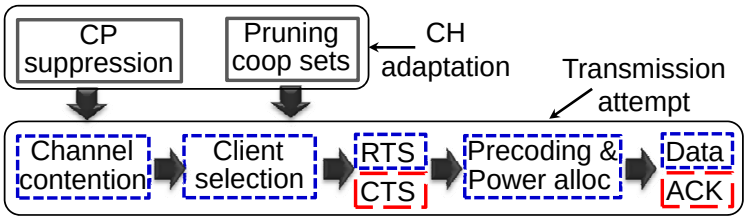

Figure 6.7: Summary of NEMOx operations.

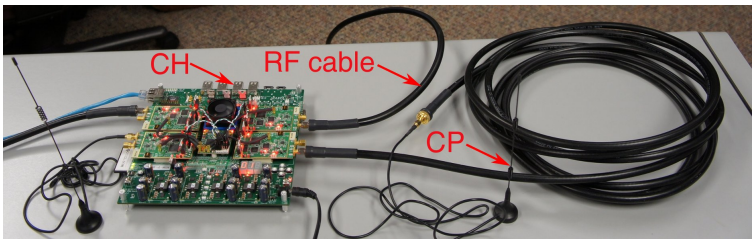

Figure 6.8: Building a NEMOx cluster using WARP.

6.6 Prototyping NEMOx

Flow of Operations in NEMOx. Fig. 6.7 summarizes NEMOx's flow of operations throughout one transmission attempt. The CH periodically runs the CP suppression algorithm (Sec. 6.5.4) to exclude certain CPs from channel contention to promote reuse across clusters. The CH continuously contends for channel access (running the distributed channel access algorithm in Sec. 6.4) on behalf of all the CSets consisting of the CPs after the pruning operation (Sec. 6.4.4). The CSet that wins contention in each cluster will start its netMIMO transmission attempt immediately. It first uses the client selection algorithm (Sec. 6.5.2) to determine the set of clients to serve jointly through netMIMO. Then, the CPs in the winning set initiate the RTS/CTS exchange with the selected clients, estimate the corresponding channel matrix (Sec. 6.5.5), and the CH computes the precoding matrix along with power allocation (Sec. 6.5.1). It then executes the netMIMO concurrent data transmission, followed by ACK reception from the clients - the latter being used to infer transmission success/failure and update contention parameters.

Implementation. We have implemented a prototype of NEMOx on the WARP

[76] software radio platform. We use the original WARP board as a CH, and deploy the CPs (as a DAS) by extending the WARP antennas (with a radio board) by up to 30ft using LMR-400 50Ω coaxial cables and SMA male-to-female connectors (Fig. 6.8). Based on the WARPlab driver, we have implemented a full-fledged multi-user MIMO-OFDM modulation/demodulation library to support NEMOx. Fig. 6.9 shows the basic components and flow of operations in our implementation.

Transmit path. In the transmit path, we first run the NEMOx client-selection algorithm to determine the set of CPs/clients to be used for netMIMO transmissions. Then, the digital bits of each client are mapped to symbols via BPSK. NEMOx's joint precoding and power-allocation algorithm is then performed, in the frequency domain (on the symbols carried by each OFDM subcarrier). The precoded symbols for each CP are then modulated using OFDM and sent over the air.

Receive path. Each client runs the receiver path that detects the packets and then estimates the channel from each CP. A packet follows a similar format with 802.11ac [6]. It starts with a short-preamble used for self-correlation-based packet detection, and then a sequence of long-preambles, sent consecutively by each CP, and used for estimating the frequency offset between CP and the client, as well as the channel phase/amplitude distortion to each OFDM subcarrier. These channel estimation results are fed back to the CH to perform the intra-cluster optimization (Sec. 6.5). Following the long-preambles is an additional preamble sent by all CPs concurrently and used to estimate the composite channel created by channel distortion and precoding. Based on the estimation results, the receiver demodulates the OFDM symbols and decodes the digital bits therein. Within each OFDM symbol, 4 pilot subcarriers (with known bits) are sent by the transmitter and used for correcting residual errors in the frequency offset estimation using long-preambles.

Hardware limitations. Due to limitations of the WARP testbed, our NEMOx prototype incurs a longer channel estimation delay than is expected in a typical

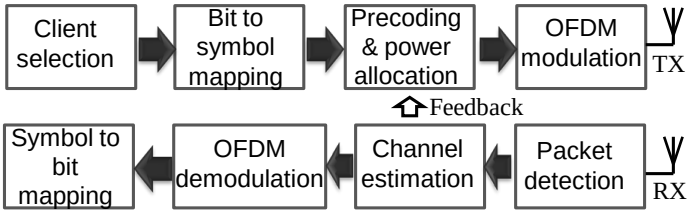

Figure 6.9: NEMOx prototype components on WARP.

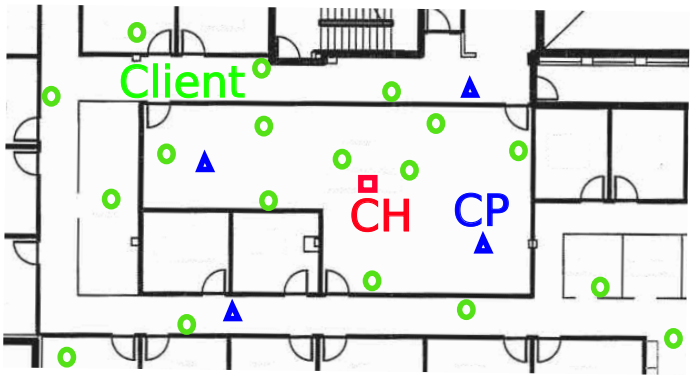

Figure 6.10: Map of NEMOx testbed.

hardware implementation. The channel matrices are fed from clients to a central controller via Ethernet. The central controller (a PC desktop running the NEMOx PHY-layer signal processing modules) computes the precoding matrices accordingly and then writes the modulated signals to the transmit buffer of each NEMOx CP (a WARP radio board) for transmission. This entire cycle takes around 3 seconds when 4 CPs and 4 clients are running concurrently. However, our experiments were conducted in the night and consistent with the results of an existing measurement study [14], wherein the channel is stable over several seconds, and a substantial netMIMO gain can still be achieved. Tightening the channel estimation delay would serve to further improve the results.

6.7 Evaluation

The goal of this section is to evaluate (i) NEMOx' scalability within a cluster and the underlying contribution of its intra-cluster optimizations; and (ii) effectiveness of

NEMOx' MAC and its resulting scalability across clusters.

6.7.1 Field-test of NEMOx

Our testbed is located in an indoor office environment (floor map shown in Fig. 6.10). Each NEMOx CH connects to up to 4 CPs, and all clients are single-antenna WARP nodes. To isolate ambient interference, all experiments are run over a 2.4GHz ISM band (channel 14) unused by other devices.

6.7.1.1 Scaling within a Cluster

We first verify how the netMIMO scheme in NEMOx allows its performance to scale with the number of CPs in a single cluster. We compare it against *(i)* MultiAP, the single-antenna WLAN using CSMA to achive spatial reuse; and *(ii)* state-of-the-art MU-MIMO scheme with a single transmitter and co-located antennas [6] (CAS). To make a fair comparison by preserving the number of transmitters, NEMOx's CPs are placed in the same locations as the MultiAP's transmitter (AP) locations. The CAS scheme places all antennas at the CH to serve the same number of clients distributed over the same network area as NEMOx and MultiAP. Comparison with other netMIMO schemes is deferred to Sec. 6.7.2, where the impact of spatial reuse on scalability is also captured for multiple clusters.

We increase the number of CPs from 2 to 4, and the number of clients accordingly. Clients' locations are randomly selected from the spots in Fig. 6.10, but the mean distance of all clients to the topology center (CH) increases with topology index. For each client configuration, we use the NEMOx and CAS's net rate gain over the MultiAP scheme as performance metric. We obtain the SINR of received symbols (using mean magnitude of positive BPSK symbols as the signal magnitude and variance of these symbols as noise power) and then convert it to achievable rate using Shannon's formula. When running MultiAP, clients are assigned to the nearest APs and the

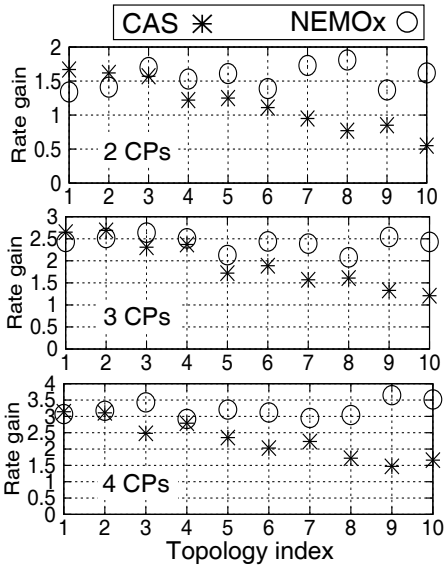

Figure 6.11: Multiplexing gain.

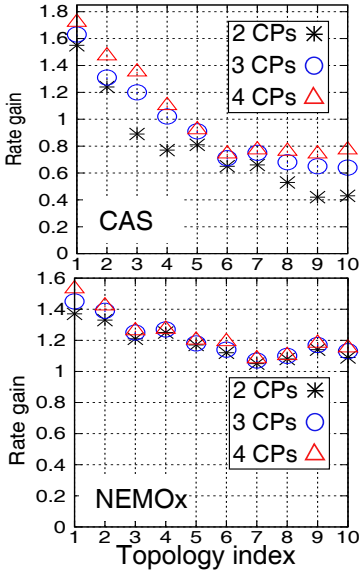

Figure 6.12: Diversity gain.

throughput is computed by assigning an equal access rate for competing APs.

From the results in Fig. 6.11, we observe that for each topology, NEMOx's rate gain increases proportionally with the number of CPs. With 2, 3 and 4 CPs, the mean rate gain (averaged over all clients and topologies) is 1.5, 2.4, and 3.1 for NEMOx, but only 1.1, 1.9 and 2.2 for CAS. *NEMOx's rate gain shows scalable performance within a cluster and comes from the tight netMIMO cooperation between CPs (correspond to APs in MultiAP). While CAS can also harvest the multiplexing gain through cooperation between transmit antennas, it is beneficial only for those clients near the CH.* For other clients, signal path-loss dominates and the rate may be even lower than MultiAP. In contrast, NEMOx sees a higher rate gain for such clients, because CPs are distributed and have shorter distance to clients, and less power is needed for canceling inter-stream interference when the CP is close to the target client. Note that the actual multiplexing gain in NEMOx is less than the theoretical bound (*i.e.*, the number of CPs), mainly because of the power needed for canceling inter-stream interference, and particularly the imperfect precoding caused by long channel estimation delay in WARP.

Besides, NEMOx can also exploit the diversity gain when the number of CPs is larger than that of clients. Fig. 6.12 shows the diversity gain (over MultiAP) of NEMOx and CAS when 2 to 4 CPs are serving a single client. When the client is near the topology center, CAS allows multiple antennas to beamform the same signal to the client, achieving up to a gain of 1.74 over MultiAP. However, as clients move farther away, the gain drops sharply (to below 1) because of the dominating path-loss effect. NEMOx circumvents such effects and the diversity gain is always above 1. Clients near the topology center experience a higher diversity gain, mainly due to relatively lower SNR (they are farther away from the CPs), where beamforming is more beneficial. Clients far away from the topology center are close to only a few CPs, thus achieving a lower gain, and increasing the number of CPs gives a marginal improvement.

6.7.1.2 Effectiveness of Intra-cluster Optimizations

Power allocation. We compare NEMOx's power allocation scheme with two other schemes for MU-MIMO networks: eq-power or equal power allocation designs the precoding matrix by inverting channel matrix ([52, 84, 102],Sec. 6.3), and opt-Amp [23] — a middle-ground solution that first inverts the channel matrix, and then optimizes the amplitude (power) of each data stream.

It has been observed that opt-Amp achieves asymptotically optimal performance for CAS [23] and eq-power achieves comparable performance in practice [14]. However, our experimental results for a cluster with 4 CPs (Figs. 6.13(a) and (b)) show that this is no longer valid for a distributed netMIMO system, where topology tends to be imbalanced. In particular, by optimizing the CPs' power allocation, NEMOx improves the mean link SINR by around 7dB when the clients are concentrated near 1 or 2 CPs (which we call *primary CPs*, Fig. 6.13(a)), and when 1 or 2 clients are served by all 4 CPs (Fig. 6.13(b)). In addition, by jointly designing the precoding

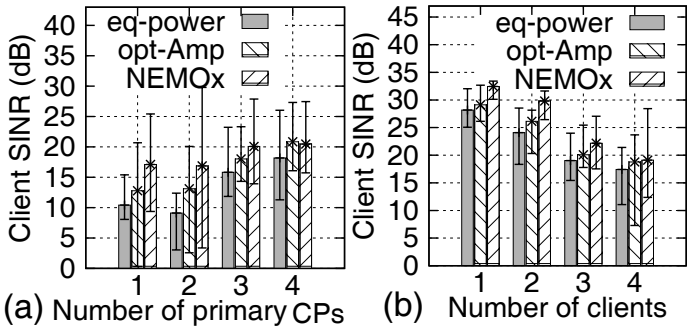

Figure 6.13: Impact of joint power allocation and precoding. Error bars show max and min of 5 client locations.

and power allocation, NEMOx achieves a substantial SINR improvement of 5 dB even over opt-Amp. The gain is especially high when the topology is imbalanced or when diversity gain is pronounced.

Client selection. An additional dimension of optimization in NEMOx is client selection. Fig. 6.14 shows the experimental results when we fix the number of CPs at 4 while varying that of clients. Compared to a naive scheme that randomly selects 4 clients, NEMOx achieves 58.9% to 75.6% higher average rate for clients. Even when the number of CPs equals that of clients, the naive scheme always selects all clients. By contrast, NEMOx may partition the clients into multiple (overlapping) groups, each having a high sum-rate and time-sharing the channel between them. Client selection becomes relatively less critical when there are many clients in the cluster. However, NEMOx tends to group clients with more balanced channel gain, thus still achieving higher gains.

CP suppression. We evaluate the effectiveness of NEMOx's CP suppression in two interfering clusters, each with 4 CPs. In the interfering region between clusters, we place a client in cluster 1, and increase the number of clients in cluster 2 (from 1 to 4) to capture scenarios where nearby clusters have different density or traffic demand. Fig. 6.15 shows the throughput of both clusters with and without CP suppression. CP suppression boosts the total network throughput by up to 51% when the topology

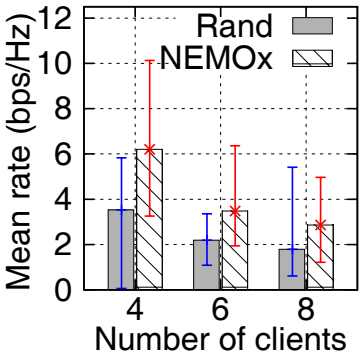

Figure 6.14: Client Selection.

Figure 6.15: CP Suppression.

involves diverse traffic demand (density) between clusters, with little sacrifice to the diversity gain for the under-utilized cluster.

6.7.2 Performance of the NEMOx MAC

While the above field test validates NEMOx' performance in small topologies, due to long response time of the WARP nodes (driven by WARPLab), we are unable to implement a real-time MAC protocol on WARP. Therefore, we complement the testbed implementation by building a detailed packet-level simulator, which extends the ns-2 802.11 module to a netMIMO PHY layer, and incorporates all the NEMOx components including the MAC and intra-cluster optimization algorithms for multi-cluster evaluation. The simulator uses an empirical channel recommended in 802.15 for 2.4GHz indoor environment [66], and a realistic SINR model that accumulates all incoming packets' signal strength to compute the interference plus noise. Under the empirical model, with -81dBm carrier sensing threshold, -96dBm noise floor, and 20dBm transmission power (typical parameters for WiFi), the equivalent carrier sensing range is 155m and transmission range 80m under 10dB SINR threshold.

The topology is generated such that clusters partially overlap to ensure full coverage of a rectangular area, the size of which varies with the number of clusters. Unless noted otherwise, each client has saturated downlink UDP traffic demand, and the number of clients is the same as that of CPs in each cluster. Packet aggregation is

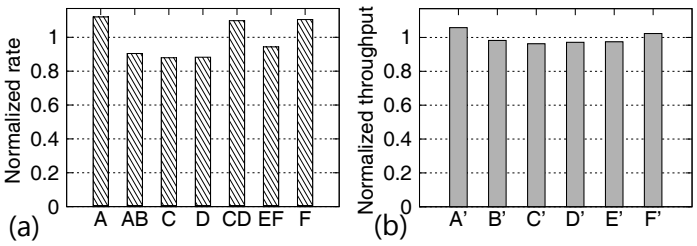

Figure 6.16: (a) CSet Access Rate, (b) Client Throughput.

used (as in high-rate standards like 802.11n and 802.11ac) and packet size is set to 4KB (further discussion in Sec. 6.7.2.3).

6.7.2.1 Fairness, efficiency and scalability

Benchmark throughput and fairness. We start with the 3-cluster symmetric network topology in Fig. 6.2 (excluding client 4) with 6 CPs and 6 clients. This topology has a well-defined proportionally fair allocation of access rates for all CPs, which can be obtained from the conflict graph of all CSets, and then solving the NCA problem. We compare the mean access rate of all CSets normalized to the optimal allocation in Fig. 6.16. The results show that *the rates yielded by the distributed channel access in NEMOx are close to the optimal proportionally fair allocation with a small deviation (around 97% to 105%).* The throughput of clients are even closer to optimum, mainly because each client in a cluster can be served by both CPs, and the intra-cell optimization framework ensures long-term proportional fairness.

Scalability in Large Topologies. We further test the scalability of NEMOx in large networks. Unless noted otherwise, the number of clients is set to the same as the number of transmitters (APs for MultiAP; CPs for NEMOx and FullCoop). For each client, we use the throughput gain of NEMOx and FullCoop (F-C) over MultiAP as the performance metric. FullCoop is representative of existing netMIMO schemes and is further supplemented with our MAC and intra-cluster optimizations. Fig. 6.17(a) shows the distribution of throughput gain while varying the CP density

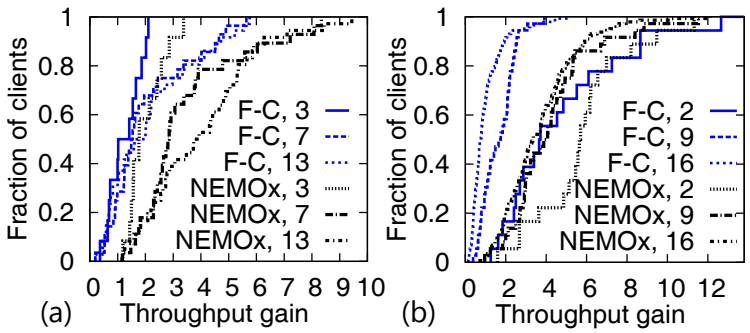

Figure 6.17: Scalability (a) varying m, (b) varying N.

m=3	**m=7**	**m=13**	**N=2**	**N=9**	**N=16**
1.45	2.33	4.06	1.59	4.20	7.75

Table 6.1: Normalized total network throughput of NEMOx.

within each cluster (fixed $N = 4$). With only 3 CPs per cluster, NEMOx's throughput gain ranges from 1.1 to 3.3, and the median throughput gain of all clients is 29.6% higher than that of FullCoop. For some clients, the throughput gain is even higher than the maximum multiplexing gain (*i.e.*, 3), because NEMOx employs proportional fairness for clients within each cluster, thereby allowing frequent serving of clients that can better exploit multiplexing gain. The throughput gain further improves as the CP density increases. *With 13 CPs per cluster, the median gain reaches 3.89, which is 2.1× that of FullCoop. It is clear that a straightforward extension of existing netMIMO schemes from single cluster (contention domain) to multiple contending clusters cannot fully exploit the multiplexing gain of netMIMO. The random access MAC in NEMOx is critical to capacity scaling in this regard.*

Fig. 6.17(b) shows how the throughput gain scales with the number of clusters (fixed $m = 8$). When increasing the number of clusters from 2 to 9, the overlapping area between clusters expands significantly, and NEMOx's throughput gain drops accordingly due to increased interference. However, further increasing the number of clusters causes diminishing loss, since the inter-cluster contention overhead is amortized and the multiplexing gain of NEMOx starts to dominate. For FullCoop, with two clusters, the gain is comparable to NEMOx. However, as the number of clusters

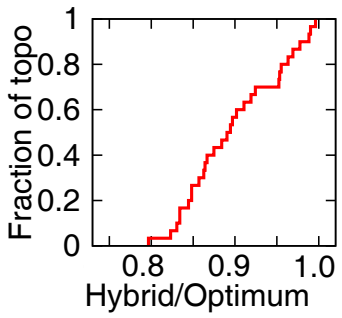

Figure 6.18: Hybrid Power Allocation.

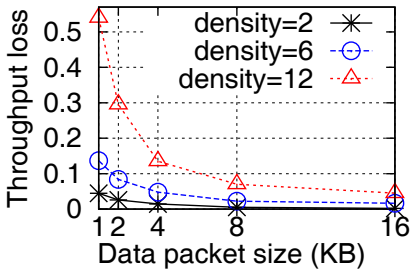

Figure 6.19: Channel-estimation overhead.

increases, spatial reuse becomes more critical, and the rigid binding of CPs causes severe performance loss. *With 9 and 16 clusters, NEMOx's median throughput gain is* 2.0× *and* 3.4× *over FullCoop.*

We also present NEMOx' normalized total network throughput corresponding to Fig. 6.17 in Table 6.1. When m (N) is varied, the normalization is *w.r.t.* its performance at $m = 1$ ($N = 1$). The results clearly indicate *the effectiveness of our MAC in allowing NEMOx' performance to scale reasonably well (sub-linear but non-saturating) in large multi-cluster topologies as well.*

6.7.2.2 Hybrid power allocation algorithm

NEMOx uses a hybrid power allocation scheme (Sec. 6.5.3) to reduce complexity. We evaluate this scheme in 40 random topologies each with a cluster density of 6 and number of clients varying between 1 and 6. We found 65% of the topologies to have a ρ-factor larger than the empirical threshold, allowing our hybrid scheme to employ the simple, equal power allocation. However, this complexity reduction comes at the cost of certain performance loss. Fig. 6.18 shows the distribution of the ratio of average throughput (per client) between hybrid power allocation and the optimum. On average, the hybrid power allocation achieves 89% throughput compared with the optimal power allocation. Thus, *the hybrid scheme is able to provide a large reduction in complexity, by trading a small loss in throughput.*

6.7.2.3 MAC layer overhead

Similar to other netMIMO schemes, NEMOx' gains come at the expense of MAC-layer overhead resulting from channel matrix feedback. In practice, the feedback overhead depends on coherence time/bandwidth, network density, and data packet size. Since our testbed experiments showed negligible performance loss even when channel-estimation delay is several seconds (consistent with other measurement studies [14]), our simulation assumes channel estimation to be valid within a period of 500ms (coherence time), and uses an empirical coherence bandwidth of 10MHz [87]. In Fig. 6.19, we vary other network parameters and evaluate the MAC overhead of NEMOx by comparing it with an oracle scheme that knows the channel without estimation. We observe that when the CP density is higher, more CPs tend to be grouped for concurrent transmission, hence introducing larger overhead. However, by increasing the packet size through frame aggregation as in high-rate MAC standards (*e.g.*, 802.11n and 802.11ac), the overhead can be reduced dramatically (we have used a packet size of 4KB in our simulations). Other techniques such as matrix compression (as in 802.11ac) can be used to further reduce NEMOx' overhead, but is left for future work.

6.8 Discussions

Compatibility with existing protocols. We made the major design choices for NEMOx with compatibility in mind. NEMOx' PHY layer is built on ZFBF, and its channel estimation and ACK mechanisms are consistent with the 802.11ac MU-MIMO standard. Its power allocation and client selection algorithms only customize the precoding matrix, and can be implemented in the driver. NEMOx' MAC uses a persistent algorithm and needs modifications to the 802.11ac AP. But it can be converted into a backoff-based algorithm by translating the persistent parameter into

backoff window size [80]. Moreover, NEMOx requires no modifications to clients. It can easily down-grade to a 802.11-compatible system by deeming each CP as the 802.11 AP and allowing the NEMOx CH to run the 802.11 MAC for them.

Uplink transmission. We have focused on improving the downlink capacity of wireless networks using NEMOx, as downlink traffic accounts for 70–80% of the traffic in enterprise networks where APs can coordinate and NEMOx is best applicable. A simple way to accommodate uplink transmissions is to allow the clients to send RTS, and CPs to defer the CTS, waiting for an opportunity when multiple uplink transmissions can run simultaneously. Such uplink transmissions can exploit SIMO decoding algorithms [122] and will be utilized in our future work.

Deployment issues. The LMR-400 cable in our testbed is one type of commercial off-the-shelf solution for building a DAS-based NEMOx CH. At 2.4GHz, it causes an attenuation of 6.8dB per 100ft (sufficient to cover a typical indoor WLAN cell). With higher-quality cables, such as LMR-1700, the attenuation can be reduced to 1.7dB, which is negligible and is outweighed by the cooperation gains from NEMOx.

Multi-Antenna CP and clients. In NEMOx, each CP or client has only one antenna. By allowing multiple antennas, the per-client throughput can be further boosted. NEMOx's inter-cluster channel access algorithm will still be valid in such a case, but the power-allocation algorithm needs to be re-designed to exploit this capability. This is a matter of our future exploration.

6.9 Related Work

Existing works in this domain can be classified under multi-user MIMO (MU-MIMO) [14, 122] and netMIMO [52, 84]. MU-MIMO has been standardized (in 802.11ac [6]) and is applicable for single-cell networks with a multi-antenna AP. [14] implemented zero-forcing beamforming (ZFBF, a common approach of realizing MU-MIMO) in the WARP software-radio platform [76]. It is shown that concurrent trans-

mission of multiple downlink streams is indeed feasible and has little inter-stream interference in common cases. [122] implemented multi-user spatial multiple access that allows clients to transmit data concurrently to a multi-antenna base station, essentially the dual version of MU-MIMO downlink beamforming. Practical netMIMO schemes usually share the same communication algorithm (*e.g.*, ZFBF) with MU-MIMO, except that the antennas are from distributed transmitters. Recently, [52] and [84] extend the information theoretic concept of interference alignment, in order to realize netMIMO in a network of mutually-interfering links. However, both assume a single contention domain where every transmitter can hear and synchronize with others.

The DAS concept has existed for years and has been studied theoretically for broadband cellular networks [131, 132, 108]. However, cellular solutions either assume the DAS is deployed in isolation (*e.g.*, leveraging dedicated spectrum) or neighboring DAS clusters are synchronized [108], thus limiting the potential for scalability in asynchronous wireless networks. In NEMOx, we aim at scaling the capacity of asynchronous wireless networks (especially wireless LANs), via hierarchical organization and decentralized scheduling of DAS clusters.

6.10 Summary

In this chapter, we have proposed NEMOx—a novel system to leverage netMIMO gains in a scalable manner in wireless networks. NEMOx organizes the network into multiple clusters, optimizes and executes netMIMO within each cluster through a DAS, and manages interference and reuse across clusters efficiently through a decentralized channel access mechanism. Our prototype implementation of NEMOx on WARP, coupled with large-scale evaluations in NS2 have shown scalable netMIMO performance both within each cluster and across the network. These indicate that NEMOx provides a promising framework for scaling the gains of netMIMO schemes

in wireless networks.

CHAPTER VII

Conclusion

7.1 Concluding Remarks

Conventional CSMA-based wireless networks adopt weakly coupled MAC and PHY layers. Although the PHY-layer technologies are constantly evolving and becoming more heterogeneous, the MAC layer simply abstracts such evolution as a change of data rate. Such an abstract interface prevents many PHY-layer advances from being translated into network-level performance improvement, and causes severe coexistence problems when new PHY technologies are deployed.

In this dissertation, we propose a joint design of the wireless MAC/PHY layers in order to overcome the limitations of CSMA networks that hinder their capacity, interoperability, and energy efficiency. We have redesigned the primitive operations in CSMA that exhibit the lack of MAC/PHY interaction. First, we propose CSMA/CR that leverages PHY layer collision-resolution to enable delay-optimal broadcast and asynchronous cooperative relaying for wireless mesh networks. Second, we design and implement ASN, a MAC/PHY mechanism that leverages fine-grained PHY layer spectrum access to improve the efficiency and fairness of spectrum sharing when different spectrum widths coexist. Further, we propose CBT, a new carrier signaling protocol that leverages PHY layer frequency flip and MAC-layer scheduling of a busy-tone signaler, in order to overcome the coexistence problem when different CSMA net-

works coexist with each other and share spectrum. In addition, we redesign the PHY layer idle listening mechanism, and enable fine-grained clock-rate management, thus substantially reducing the idle power consumption for CSMA-based WiFi networks. Finally, we introduce NEMOx, a new architecture and protocol that synthesizes existing PHY layer MIMO cooperation algorithms, and makes them scalable in large-scale multi-cell CSMA-based wireless LANs.

7.2 Future Work

The principle of MAC/PHY co-design can be extended further by simplifying the MAC/PHY interface, applying it to emerging wireless networks, and enabling PHY-aware application layer algorithms.

7.2.1 Simplifying the MAC/PHY interface

A hallmark of MAC/PHY co-design is the use of novel signal processing algorithms (*e.g.*, iterative collision resolution, frequency flip, sampling-rate invariant detection, partial spectrum sensing and decoding) that make PHY-layer capabilities usable by the MAC layer. These algorithms usually require modifications to existing hardware/firmware, and can become easier to deploy with the development of software radios.

However, even with reconfigurable radio platforms, a simple MAC/PHY interface is preferable. Since many problems in CSMA networks are caused by lack of interactions between the MAC and PHY layers, simple algorithms for enhancing such interactions may achieve the same objective of MAC/PHY co-design and boost network performance.

For example, in the proposed CBT (Ch. IV), ASN (Ch. III) and E-MiLi (Ch. V) systems, each requires the channel usage activity to be sensed without direct communication between the transmitter and receiver (which is infeasible due to their

spectrum heterogeneity). This task can be realized using a simple, unified energy sensing approach. The transmitter can send two consecutive signal pulses and use the separation between them to convey certain information (*e.g.*, duration of channel occupation, spectrum width to be used and destination address) to the receiver. Since energy sensing is a basic PHY capability of all CSMA devices, such an approach will substantially simplify the MAC/PHY co-design behind the proposed systems.

7.2.2 MAC/PHY co-design for emerging wireless networks

Wireless networking technology has been reshaping itself at an accelerating speed. The future wireless architecture will become more heterogeneous, diverging from the WiFi and cellular paradigms, and supporting a broad range of customized applications and distributed systems. To prevent recurrence of the problems in CSMA networks, emerging wireless networks should be based on the principle of MAC/PHY co-design during their early stage of development.

One example is the whitespace network, which promises to enable low-cost, high-performance mobile Internet access, but faces fundamental challenges such as the sporadic spectrum blocks. Instead of following the current trend of migrating the WiFi standard to whitespace [18], the CSMA MAC/PHY should be co-designed by taking into account the unique PHY-layer features (sporadic spectrum distribution, large transmission range, stable propagation profile, *etc.*) to improve network performance.

In addition to Internet access, future wireless networks need to support communications within large-scale distributed infrastructures. For example, future smart grid may have a built-in wireless backbone (e.g., based on whitespace networks) to reliably deliver data and control information over a wide area. For such networks, the confluence of MAC/PHY co-design and flow-level optimization may be needed to establish mesh-like connections with certain quality-of-service guarantees.

7.2.3 PHY-aware wireless applications

The philosophy of *co-design* can be extended to enhance the interaction between other network layers. In particular, many wireless and mobile applications can benefit from a comprehensive set of PHY-layer information. Existing work exploited the PHY layer signature (*e.g.*, clock drift and receiver noise) to identify different wireless devices [25]. As we observed in previous chapters, however, the PHY layer information and capabilities are much richer than a simple signature.

One example application of such information is indoor localization. The indoor wireless environment is typically filled with WiFi signals. Beside the periodic traffic due to WiFi network management (*e.g.*, the 802.11 beacons), most other traffic is introduced by human activity, which is known to exhibit certain patterns [96]. Therefore, a node can gauge its location by exploiting the spectrum usage pattern in its radio environment.

Because of their practical relevance and inter-disciplinary nature (involving signal processing, communications, and wireless networking), we believe the above directions are worthy of further exploration using the principle of MAC/PHY co-design and its extension.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] open-ZB. http://www.open-zb.net.

[2] The GNU Software Radio. http://gnuradio.org/trac/wiki.

[3] Coexistence of Wireless Personal Area Networks With Other Wireless Devices Operating in Unlicensed Frequency Bands. *IEEE Std 802.15.2*, 2003.

[4] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std. 802.15.4*, 2003.

[5] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std. 802.11*, 2007.

[6] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std. 802.11ac Draft 1.0*, 2011.

[7] J. Acharya, H. Viswanathan, and S. Venkatesan. Timing Acquisition for Non Contiguous OFDM Based Dynamic Spectrum Access. In *Proc. of IEEE DySPAN*, 2008.

[8] Yuvraj Agarwal, Ranveer Chandra, Alec Wolman, Paramvir Bahl, Kevin Chin, and Rajesh Gupta. Wireless Wakeups Revisited: Energy Management for VoIP Over Wi-Fi Smartphones. In *Proc. of ACM MobiSys*, 2007.

[9] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-Management in Chaotic Wireless Deployments. In *Proc. of ACM MobiCom*, 2005.

[10] I. F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Computer Networks*, 47(4), 2005.

[11] Analog Devices. AD9522 Data Sheet, 2008.

[12] G. Anastasi, M. Conti, E. Gregori, and A. Passarella. 802.11 Power-Saving Mode for Mobile Computing in Wi-Fi Hotspots: Limitations, Enhancements and Open Issues. *Wireless Networks*, 14(6), 2008.

[13] Mustafa Y. Arslan, Konstantinos Pelechrinis, Ioannis Broustis, Srikanth V. Krishnamurthy, Sateesh Addepalli, and Konstantina Papagiannaki. Auto-Configuration of 802.11n WLANs. In *Proc. of ACM CoNext*, 2010.

[14] Ehsan Aryafar, Narendra Anand, Theodoros Salonidis, and Edward W. Knightly. Design and Experimental Evaluation of Multi-user Beamforming in Wireless LANs. In *Proc. of ACM MobiCom*, 2010.

[15] Atheros Communications. AR5213 Preliminary Datasheet, 2004.

[16] Atheros Communications. Power Consumption and Energy Efficiency of WLAN Products, 2004.

[17] K. Azarian, H. El Gamal, and P. Schniter. On the Achievable Diversity-Multiplexing Tradeoff in Half-duplex Cooperative Channels. *IEEE Trans. on Information Theory*, 51(12), 2005.

[18] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. White Space Networking With Wi-Fi Like Connectivity. In *Proc. of ACM SIGCOMM*, 2009.

[19] John Thomson Bevan et al. An Integrated 802.11a Baseband and MAC Processor. In *IEEE ISSCC Digest*, 2002.

[20] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proc. of ACM MobiCom*, 2005.

[21] S. Biswas and R. Morris. ExOR: Opportunistic Multi-hop Routing for Wireless Networks. In *Proc. of ACM SIGCOMM*, 2005.

[22] A. Bletsas, A. Khisti, S. Member, and D. P. Reed. A Simple Cooperative Diversity Method Based on Network Path Selection. *IEEE Journal on Selected Areas in Communications*, 24(3), 2006.

[23] F. Boccardi and H. Huang. Zero-Forcing Precoding for the MIMO Broadcast Channel under Per-Antenna Power Constraints. In *IEEE 7th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2006.*, 2006.

[24] J. Bondy and U. Murthy. *Graph Theory With Applications*. Elsevier, 1976.

[25] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless Device Identification With Radiometric Signatures. In *Proc. of ACM MobiCom*, 2008.

[26] O. Brun and J-M. Garcia. Analytical Solution of Finite Capacity M/D/1 Queues. *Journal of Applied Probability*, 37(4), 2000.

[27] V.R. Cadambe and S.A. Jafar. Interference Alignment and Degrees of Freedom of the K-User Interference Channel. *IEEE Transactions on Information Theory*, 54(8), 2008.

[28] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement Driven Deployment of a Two-tier Urban Mesh Access Network. In *Proc. of ACM MobiSys*, 2006.

[29] James K. Cavers. *Mobile Channel Characteristics*. Kluwer Academic Publishers, 2000.

[30] Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra, and Paramvir Bahl. A Case for Adapting Channel Width in Wireless Networks. In *Proc. of ACM SIGCOMM*, 2008.

[31] Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra, and Paramvir Bahl. A Case for Adapting Channel Width in Wireless Networks. In *Proc. of ACM SIGCOMM*, 2008.

[32] B. S. Chlebus, Gasieniec L., A. Gibbons, A. Pelc, and W. Rytter. Deterministic Broadcasting in Unknown Radio Networks. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2000.

[33] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM MobiCom*, 2003.

[34] G. R. Danesfahani and T. G. Jeans. Optimisation of Modified Mueller and Muller Algorithm. *IEEE Electronics Letters*, 31(13), 1995.

[35] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin. On the Linear Behaviour of the Throughput of IEEE 802.11 DCF in Non-Saturated Conditions. *IEEE Communications Letters*, 11(11), 2007.

[36] William R. Dieter, Srabosti Datta, and Wong Key Kai. Power Reduction by Varying Sampling Rate. In *ACM/IEEE ISLPED*, 2005.

[37] Digi International Inc. XBee-PRO 802.15.4 OEM RF Modules. http://www.digi.com/.

[38] Prabal Dutta, Ye-Sheng Kuo, Akos Ledeczi, Thomas Schmid, and Peter Volgyesi. Putting the Software Radio on a Low-Calorie Diet. In *Proc. of ACM HotNets*, 2010.

[39] Ettus Research LLC. Universal Software Radio Peripheral (USRP). http://www.ettus.com/.

[40] P. Fan and M. Darnell. *Sequence Design for Communications Application*. Research Studies Press, 1996.

[41] FCC. Second Memorandum Opinion and Order, Sep. 2010.

[42] Shulan Feng, Heather Zheng, Haiguang Wang, Jinnan Liu, and Philipp Zhang. Preamble Design for Non-Contiguous Spectrum Usage in Cognitive Radio Networks. In *Proc. of IEEE WCNC*, 2009.

[43] Krisztián Flautner, Steve Reinhardt, and Trevor Mudge. Automatic Performance Setting for Dynamic Voltage Scaling. In *Proc. of ACM MobiCom*, 2001.

[44] The DAS Forum. In-Building Enterprise DAS. In *Wireless Infrastructure*, 2011.

[45] The DAS Forum. Augmenting Mobile Broadband. In *FCC Workshop*, 2012.

[46] Makoto Fujinami and Takuya Murakami. PSM Extension for ns-2. http://nspme.sourceforge.net/index.html.

[47] R. Gallager. *Discrete Stochastic Processes (2nd Ed.) Draft* . 2009.

[48] R. Gandhi, S. Parthasarathy, and A. Mishra. Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks. In *Proc. of ACM MobiHoc*, 2003.

[49] M. Gerla, P. Palnati, and S. Walton. Multicasting Protocols for High-Speed, Wormhole-Routing Local Area Networks. In *Proc. of ACM SIGCOMM*, 1996.

[50] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and Wei Yu. Multi-Cell MIMO Cooperative Networks: A New Look at Interference. *IEEE Journal on Selected Areas in Communications (JSAC)*, 28(9), 2010.

[51] S. Gollakota and D. Katabi. ZigZag Decoding: Combating Hidden Terminals in Wireless Networks. In *Proc. of ACM SIGCOMM*, 2008.

[52] Shyamnath Gollakota, Samuel David Perli, and Dina Katabi. Interference Alignment and Cancellation. In *Proc. of ACM SIGCOMM*, 2009.

[53] S. Gopal, S. Paul, and D. Raychaudhuri. Investigation of the TCP simultaneous-send problem in 802.11 wireless local area networks. In *Proc. of IEEE International Conference on Communications (ICC)*, 2005.

[54] James Gross, Oscar Puñal, and Marc Emmelmann. Multi-user OFDMA Frame Aggregation for Future Wireless Local Area Networking. In *Proceedings of IFIP-TC 6 Networking Conference*, 2009.

[55] R. Gummadi, H. Balakrishnan, and S. Seshan. Metronome: Coordinating Spectrum Sharing in Heterogeneous Wireless Networks. In *First International Workshop on Communication Systems and Networks (COMSNETS)*, 2009.

[56] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and Mitigating the Impact of RF Interference on 802.11 Networks. In *Proc. of ACM SIGCOMM*, 2007.

[57] Z.J. Haas and Jing Deng. Dual Busy Tone Multiple Access (DBTMA)–a Multiple Access Control Scheme for Ad Hoc Networks. *IEEE Transactions on Communications*, 50(6), 2002.

[58] D. Halperin, T. Anderson, and D. Wetherall. Taking the Sting Out of Carrier Sense: Interference Cancellation for Wireless LANs. In *Proc. of ACM MobiCom*, 2008.

[59] Yong He, Ji Fang, Jiansong Zhang, Haichen Shen, Kun Tan, and Yongguang Zhang. MPAP: Virtualization Architecture for Heterogenous Wireless APs. *SIGCOMM Comput. Commun. Rev.*, 41(1), 2010.

[60] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X.P. Costa, and B. Walke. The IEEE 802.11 Universe. *Communications Magazine, IEEE*, 48(1), 2010.

[61] J. Hou, B. Chang, D-K. Cho, and M. Gerla. Minimizing 802.11 Interference on ZigBee Medical Sensors. In *Proc. of the International Conference on Body Area Networks*, 2009.

[62] J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond Co-existence: Exploiting WiFi White Space for ZigBee Performance Assurance. In *Proc. of IEEE ICNP*, 2010.

[63] S. Huang, P. J. Wan, J. Deng, and Y. Han. Broadcast Scheduling in Interference Environment. *IEEE Trans. on Mobile Computing*, 7(11), Nov 2008.

[64] S.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang. Minimum-Latency Broadcast Scheduling in Wireless Ad Hoc Networks. In *Proc. of IEEE INFOCOM*, 2007.

[65] IEEE 802.11ac TG. http://www.ieee802.org/11/Reports/tgac_update.htm.

[66] IEEE 802.15 Working Group. Coexistence Analysis of IEEE Std 802.15.4 With Other IEEE Standards and Proposed Standards, 2010.

[67] IEEE Standard. 802.11™: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications , 2007.

[68] IEEE Standard. 802.11n: Enhancements for Higher Throughput, 2009.

[69] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination For Resource Allocation in Shared Computer Systems. Technical Report TR-301, DEC Research, September 1984.

[70] G. Jakllari, S. Krishnamurthy, M. Faloutsos, P. Krishnamurthy, and O. Ercetin. A Framework for Distributed Spatio-Temporal Communications in Mobile Ad Hoc Networks. In *Proc. of IEEE INFOCOM*, 2006.

[71] Kyle Jamieson. The SoftPHY Abstraction: from Packets to Symbols in Wireless Network Design. Ph.D. Thesis, MIT, 2008.

[72] Kyle Jamieson and Hari Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *Proc. of ACM SIGCOMM*, 2007.

[73] Bang Chul Jung, Young-Jun Hong, Dan Keun Sung, and Sae-Young Chung. Adaptive Sub-Band Nulling for OFDM-Based Wireless Communication Systems. In *Proc. of IEEE WCNC*, 2007.

[74] Sundaresan K and R. Sivakumar. Cooperating with Smartness: Using Heterogeneous Smart Antennas in Ad-Hoc Networks. In *Proc. of IEEE INFOCOM*, 2007.

[75] Min Suk Kang and Bang Chul Jung. Decentralized Intercell Interference Coordination in Uplink Cellular Networks using Adaptive Sub-Band Exclusion. In *Proc. of IEEE WCNC*, 2009.

[76] Ahmed Khattab, Joseph Camp, Chris Hunter, Patrick Murphy, Ashutosh Sabharwal, and Edward W. Knightly. WARP: a Flexible Platform for Clean-Slate Wireless Medium Access Protocol Design. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12, 2008.

[77] G. Kramer, I. Maric, and R. D. Yates. Cooperative Communications. *Foundations and Trends in Networking*, 1(3), 2006.

[78] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New Insights From a Fixed-Point Analysis of Single Cell IEEE 802.11 WLANs. *IEEE/ACM Trans. on Networking*, 15(3), 2007.

[79] J. N. Laneman and G. W. Wornell. Distributed Space-Time Coded Protocols for Exploiting Cooperative Diversity in Wireless Networks. *IEEE Trans. on Information Theory*, 49(10), 2003.

[80] J. W. Lee, M. Chiang, and R. A. Calderbank. Optimal MAC design based on utility maximization: Reverse and forward engineering. In *IEEE INFOCOM*, 2006.

[81] Li Erran Li, Kun Tan, Harish Viswanathan, Ying Xu, and Yang Richard Yang. Retransmission $\neq$ Repeat: Simple Retransmission Permutation Can Resolve Overlapping Channel Collisions. In *Proc. of ACM MobiCom*, 2010.

[82] Y. Li and X. Xia. A Family of Distributed Space-Time Trellis Codes with Asynchronous Cooperative Diversity. In *Proc. of IEEE IPSN*, 2005.

[83] C-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. In *Proc. of ACM SenSys*, 2010.

[84] Kate Ching-Ju Lin, Shyamnath Gollakota, and Dina Katabi. Random Access Heterogeneous MIMO Networks. In *ACM SIGCOMM*, 2011.

[85] Jiayang Liu and Lin Zhong. Micro Power Management of Active 802.11 Interfaces. In *Proc. of ACM MobiSys*, 2008.

[86] W. Lou and J. Wu. Toward Broadcast Reliability in Mobile Ad Hoc Networks with Double Coverage. *IEEE Trans. on Mobile Computing*, 6(2), 2007.

[87] H. MacLeod, C. Loadman, and Z. Chen. Experimental studies of the 2.4-GHz ISM wireless indoor channel. In *Proc. of Communication Networks and Services Research Conference*, 2005.

[88] R. Mahjourian, F. Chen, R. Tiwari, M. Thai, H. Zhai, and Y. Fang. An Approximation Algorithm for Conflict-Aware Broadcast Scheduling in Wireless Ad Hoc Networks. In *Proc. of ACM MobiCom*, 2008.

[89] Justin Manweiler and Romit Roy Choudhury. Avoiding the Rush Hours: WiFi Energy Management via Traffic Isolation. In *Proc. of ACM MobiSys*, 2011.

[90] Maxim. MAX2831/MAX2832 2.4GHz to 2.5GHz 802.11g/b RF Transceivers, 2010.

[91] B. McFarland, A. Shor, and A. Tabatabaei. A 2.4 & 5 GHz Dual Band 802.11 WLAN Supporting Data Rates to 108 Mb/s. In *IEEE GaAs IC Annual Digest*, 2002.

[92] Arunesh Mishra, Vivek Shrivastava, Suman Banerjee, and William Arbaugh. Partially Overlapped Channels Not Considered Harmful. In *SIGMETRICS*, 2006.

[93] T. Moscibroda, R. Chandra, Yunnan Wu, S. Sengupta, P. Bahl, and Yuan Yuan. Load-Aware Spectrum Distribution in Wireless LANs. In *Proc. of IEEE ICNP*, 2008.

[94] R. Mudumbai, D.R. Brown, U. Madhow, and H.V. Poor. Distributed Transmit Beamforming: Challenges and Recent Progress. *IEEE Communications Magazine*, 47(2), 2009.

[95] T. Nandagopal, T.E.Kim, X. Gao, and V. Bhargavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *ACM MobiCom*, 2000.

[96] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, and David Wetherall. 802.11 User Fingerprinting. In *Proc. of ACM MobiCOm*, 2007.

[97] Caleb Phillips and Suresh Singh. CRAWDAD data set pdx/vwave, 2007.

[98] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access For Wireless Sensor Networks. In *Proc. of ACM SenSys*, 2004.

[99] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai. Harmful Coexistence Between 802.15.4 and 802.11: A Measurement-based Study. In *Proc. of CrownCom*, 2008.

[100] J.D. Poston and W.D. Horne. Discontiguous OFDM Considerations for Dynamic Spectrum Access in Idle TV Channels. In *Proc. of IEEE DySPAN*, 2005.

[101] Daiming Qu, Jie Ding, Tao Jiang, and Xiaojun Sun. Detection of Non-Contiguous OFDM Symbols for Cognitive Radio Systems without Out-of-Band Spectrum Synchronization. *IEEE Transactions on Wireless Communications*, 10(2), 2011.

[102] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. Sourcesync: a distributed wireless architecture for exploiting sender diversity. In *Proc. of ACM SIGCOMM*, 2010.

[103] Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat. Learning to Share: Narrowband-Friendly Wideband Networks. In *Proc. of ACM SIGCOMM*, 2008.

[104] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based Characterization of 802.11 in a Hotspot Setting. In *Proc. of SIGCOMM E-WIND*, 2005.

[105] E. Rozner, Y. Mehta, A. Akella, and Lili Qiu. Traffic-Aware Channel Assignment in Enterprise Wireless LANs. In *Proc. of IEEE ICNP*, 2007.

[106] Eric Rozner, Vishnu Navda, Ramachandran Ramjee, and Shravan Rayanchu. NAPman: Network-Assisted Power Management for WiFi Devices. In *Proc. of ACM MobiSys*, 2010.

[107] S.G. Sankaran, M. Zargari, L.Y. Nathawad, H. Samavati, S.S. Mehta, A. Kheirkhahi, P. Chen, Ke Gong, B. Vakili-Amini, J. Hwang, S.-W.M. Chen, M. Terrovitis, B.J. Kaczynski, S. Limotyrakis, M.P. Mack, H. Gan, M. Lee, R.T. Chang, H. Dogan, S. Abdollahi-Alibeik, B. Baytekin, K. Onodera, S. Mendis, A. Chang, Y. Rajavi, S.H.-M. Jen, D.K. Su, and B. Wooley. Design and Implementation of a CMOS 802.11n SoC. *IEEE Communications Magazine*, 47(4):134 –143, 2009.

[108] Sawahashi, M. and Kishiyama, Y. and Morimoto, A. and Nishikawa, D. and Tanno, M. Coordinated Multipoint Transmission/Reception Techniques for LTE-Advanced [Coordinated and Distributed MIMO]. *IEEE Wireless Communications*, 17(3), 2010.

[109] A. Scaglione and Y.-W. Hong. Opportunistic Large Arrays: Cooperative Transmission in Wireless Multihop Ad Hoc Networks to Reach Far Distances. *IEEE Trans. on Signal Processing*, 51(8), 2003.

[110] Schneider Electrics. ZigBee WiFi Coexistence. http://www.zigbee.org/LearnMore/WhitePapers.aspx, 2008.

[111] Aaron Schulman, Dave Levin, and Neil Spring. CRAWDAD data set umd/sigcomm2008, 2008.

[112] Souvik Sen, Romit Roy Choudhury, and Bozidar Radunovic. PHY-Assisted Energy Management for Mobile Devices. In *ACM MobiSys Poster Session*, 2010.

[113] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. CSMA/CN: Carrier Sense Multiple Access With Collision Notification. In *Proc. of ACM MobiCom*, 2010.

[114] Li Shang, Li-Shiuan Peh, and Niraj K. Jha. Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks. In *Proc. of IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2003.

[115] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair. Wake on Wireless: an Event Driven Energy Saving Strategy for Battery Operated Devices. In *Proc. of ACM MobiCom*, 2002.

[116] C-K. Singh, A. Kumar, and P. M. Ameer. Performance Evaluation of an IEEE 802.15.4 Sensor Network With a Star Topology. *Wireless Networks*, 14(4), 2008.

[117] B. Sklar. *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.

[118] K. Sreeram, S. Birenjith, and P. V. Kumar. DMT of multi-hop cooperative networks-Part II: Layered and multi-antenna networks. In *Proc. of IEEE ISIT*, 2008.

[119] A. Tabatabaei, K. Onodera, M. Zargari, H. Samavati, and D.K. Su. A Dual Channel $\Sigma\Delta$ ADC with 40MHz Aggregate Signal Bandwidth. In *Proc. of IEEE ISSCC*, 2003.

[120] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W Wang, and G. M. Voelker. Sora: High Performance Software Radio Using General Purpose Multi-core Processors. In *Proc. of USENIX NSDI*, 2009.

[121] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine-Grained Channel Access in Wireless LAN. In *SIGCOMM*, 2010.

[122] Kun Tan, He Liu, Ji Fang, Wei Wang, Jiansong Zhang, Mi Chen, and Geoffrey M. Voelker. SAM: Enabling Practical Spatial Multiple Access in Wireless LAN. In *Proc. of ACM MobiCom*, 2009.

[123] Texas Instrument. CC2420 Preliminary Datasheet, 2004.

[124] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12), 1975.

[125] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[126] S. Wei. Diversity Multiplexing Tradeoff of Asynchronous Cooperative Diversity in Wireless Networks. *IEEE Trans. on Information Theory*, 53(11), 2007.

[127] A. Wiesel, Y.C. Eldar, and S. Shamai. Zero-Forcing Precoding and Generalized Inverses. *IEEE Transactions on Signal Processing*, 56(9), 2008.

[128] Lei Yang, Ben Y. Zhao, and Haitao Zheng. The Spaces Between Us: Setting and Maintaining Boundaries in Wireless Spectrum Access. In *Proc. of ACM MobiCom*, 2010.

[129] Wei Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC protocol for Wireless Sensor Networks. In *Proc. of IEEE INFOCOM*, 2002.

[130] M. Zargari et al. A Dual-Band CMOS MIMO Radio SoC for IEEE 802.11n Wireless LAN. *IEEE Journal of Solid-State Circuits*, 43(12), 2008.

[131] Jun Zhang and J.G. Andrews. Cellular communication with randomly placed distributed antennas. In *Proc. of IEEE GlobeCom*, 2007.

[132] Y. Zhang, H. Hu, and J. Luo. *Distributed Antenna Systems: Open Architecture for Future Wireless Communications*. Auerbach Publications, 2007.

[133] J. Zhu, A. Waltho, X. Yang, and X. Guo. Multi-Radio Coexistence: Challenges and Opportunities. In *Proc. of IEEE ICCCN*, 2007.