Title
Towards Gigabit and Green 802.11 Wireless Networks

Author
Pefkianakis, Ioannis

Publication Date
2012

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Towards Gigabit and Green 802.11 Wireless Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Ioannis Pefkianakis

2012

Abstract of the Dissertation

Towards Gigabit and Green 802.11 Wireless Networks

by

Ioannis Pefkianakis

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2012

Professor Songwu Lu, Chair

Wireless is an increasingly dominant communication medium. The continued quest for wireless connectivity in a multitude of mobile devices, along with the emerging bandwidth hungry applications, has resulted in a huge growth of the wireless traffic. Multiple-Input Multiple-Output (MIMO) is considered the dominant technology to provide gigabit wireless links, and to accommodate the increasing demand of speed over wireless. By using multiple transmit and receive antennas, MIMO can support more reliable and faster communication. But how efficient are the current MIMO systems?

Our experiments with commodity MIMO 802.11n devices reveal that, the current MIMO wireless is low speed and energy hungry. The fundamental reason for MIMO devices' poor performance is the use of legacy 802.11a/b/g, single antenna designs over the multiple antenna, MIMO 802.11n setting. Specifically, the existing designs used over the new MIMO 802.11n devices, are oblivious to MIMO unique communication characteristics. They do not also consider that, MIMO speed comes at the cost of increased power consumption, proportional to the number of antennas.

In order to investigate solutions to these problems, this dissertation first experimentally studies the unique features of MIMO wireless and their impact on existing

designs' performance. Then, it revises the key mechanisms that control speed and energy over MIMO wireless, named *Rate Adaptation*, and *MIMO Energy Save*, and develops three systems. History-Aware Robust Rate Adaptation (HA-RRAA) is our first step towards gigabit wireless. It opportunistically selects the best goodput PHY transmission rate for legacy 802.11a/b/g networks by introducing novel mechanisms to capture short-term channel dynamics. Different from HA-RRAA, our MIMO Rate Adaptation (MiRA) proposal, seeks to identify the best goodput PHY transmission rate in MIMO 802.11n networks by considering the unique features of MIMO. Finally, MIMO Energy Save seeks to select the optimal antenna setting at runtime to minimize energy consumption. Our proposals depart from existing designs in three fundamental ways. They manage the unique MIMO communication modes in a distinct manner. They consider new metrics, to capture the tradeoffs between speed and power consumption. Our proposals also apply novel learning mechanisms to capture the wireless channel dynamics.

There are three main contributions in this dissertation. First, it builds a strong connection between wireless communication theory and wireless system design. Specifically, this dissertation provides the first experimental study of fundamental MIMO wireless communication tradeoffs (i.e. diversity vs. spatial multiplexing MIMO modes, speed vs. number of antennas) using 802.11n standard-compliant commodity testbeds. Then, it uncovers their impact on existing designs' performance. Second, it proposes novel and practical rate adaptation and energy save designs that consider MIMO unique characteristics, and are able deliver high performance gains. Third, this dissertation provides the first implementation and evaluation of MIMO rate adaptation and energy save using 802.11n standard-compliant commodity devices. The high performance gains in real world settings make our proposals a significant step towards gigabit and green wireless networks.

The dissertation of Ioannis Pefkianakis is approved.

Babak Daneshrad

Lixia Zhang

Mario Gerla

Songwu Lu, Committee Chair

University of California, Los Angeles

2012

To my parents, Michalis and Efi, and my brother Kostis

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGMENTS

I arrived at UCLA six years ago with very little knowledge about what research, contribution, and impact really mean. I am deeply indebted to my advisor Prof. Songwu Lu, for very carefully guiding me through my Ph.D. study, and teaching me the value of quality research. I feel that for every hour I put into my research Prof. Lu put two. His insights, insistence on simplicity, and great patience with my writing and experiments definitely laid the foundations for the results presented in this dissertation.

I am very grateful to my Ph.D. committee members Prof. Lixia Zhang, Prof. Mario Gerla, and Prof. Babak Daneshrad, for their insightful comments from the early stage of my research. Prof. Zhang taught me the importance of being precise and emphasizing on the big picture of a problem. Prof. Gerla's sharp technical comments and friendly discussions further motivated my research. Prof. Daneshrad's expertise in MIMO technologies, was very important for my MIMO rate adaptation and energy save work.

I would like to especially thank Suk-Bok Lee, who has been not only a great labmate and an inspiring colleague, but also an amazing friend. Suk-Bok taught me how to stay calm and positive under high pressure. I would like also to thank Yun Hu, Starsky Wong, Hao Yang, and Chi-Yu Li, who are the key contributors in making rate adaptation and energy save work, a reality. I also feel very fortunate to have worked with many bright colleagues at UCLA as Chunyi Peng, Junmei Qu and Xueyong Xu.

I am greatly indebted to my internship mentors Vaduvur Bharghavan and Thomas Woo. During my internship in Meru Networks, Dr. Bharghavan helped me to lay the foundations for my MIMO wireless networking research. Dr. Woo with his sharp questions forced me to think harder about the systems I design and build.

Teaching has been a very important part of my Ph.D. life, and has helped me to evolve as a scientist, as an engineer, and as a person. I would like to especially thank the Professors that assigned me teaching duties; Donald Browne, Ben King, Peter Reiher, Songwu Lu, David Smallberg, Ani Nahapetian.

Finally, I am very grateful to my parents, Michalis and Efi Pefkianakis and my brother, Kostis Pefkianakis, for their support, patience, and love. Without them this work would have never been accomplished.

VITA

2001–2006 Diploma in Computer Engineering and Informatics, University of Patras, Greece.

2006–2009 M.S. in Computer Science, University of California, Los Angeles.

PUBLICATIONS

1.*MIMO Rate Adaptation in 802.11n Wireless Networks.*
I. Pefkianakis, Y. Hu, S. H.Y. Wong, H. Yang, S. Lu.
ACM MOBICOM 2010.

2.*What is Wrong/Right with IEEE 802.11n Spatial Multiplexing Power Save Feature?*
I. Pefkianakis, C. Li, S. Lu.
IEEE ICNP 2011.

3.*Towards History-Aware Robust 802.11 Rate Adaptation.*
I. Pefkianakis, S. H.Y. Wong, H. Yang, S. Lee, S. Lu.
IEEE Transactions on Mobile Computing, Jan. 2012.

4.*Window-based Rate Adaptation in 802.11n Wireless Networks.*
I. Pefkianakis, Y. Hu, S. Lee, C. Peng, S. Sakellaridi, S. Lu.
ACM Mobile Networks and Applications, Jan. 2012.

5.*Towards MIMO-Aware 802.11n Rate Adaptation.*

I. Pefkianakis, S. Lee, S. Lu

IEEE/ACM Transactions on Networking, to appear.

CHAPTER 1

Introduction

Wireless is an increasingly dominant communication medium. International Telecommunication Union reported 5.9 billion mobile-cellular subscriptions at the end of 2011 [1], which corresponds to 87% of the world population. Mobile computing devices need to serve increasingly demanding networked applications (video conferencing, multiplayer 3-D games, cloud-supported mobile augmented reality [2]), which current wireless speeds struggle to accommodate. For example, a HD 1080p video's data bit rate is 50Mbps, while the current 802.11a/g devices offer approximately 30Mbps peak MAC-layer throughput. So, the continued quest for wireless connectivity in a multitude of mobile devices, along with the emerging bandwidth hungry applications, has resulted in a huge growth in wireless traffic. In October 2010, FCC pointed out that Cisco Systems, the Yankee Group and Coda Research projected that mobile data traffic in 2014 will be 35 times the volume of traffic in 2009 [3].

Designing very high speed wireless that offers high quality-of-service (QoS) constitutes a significant research and engineering challenge. We can, in principle, meet the Gbps data rate requirement if the product of bandwidth (measured in Hz) and spectral efficiency (measured in b/s/Hz) is greater than 10^9. However, employing sufficiently high bandwidth to increase wireless speed poses significant limitations. First, spectrum is a scarce and expensive resource. The spectrum allocated between 2-6GHz for WiFi and other license-free applications does not exceed 0.5GHz, and as a result it does not allow for channels wide enough for gigabit speeds. Opening up new spectrum at frequencies higher than 6GHz can offer a potential solution for high bandwidth wireless communication (IEEE 802.11ad at 60GHz band [4]). However, transmissions at very high frequencies are extremely prone to atmospheric attenuation, which renders non-line-of-sight (NLOS) and long distance wireless links unusable.

An emerging technology known as Multiple-Input Multiple-Output (MIMO), offers significant promise in making Gbps wireless links in NLOS environments a reality. MIMO adds the space dimension to the current 2-dimensional (frequency and time) wireless communication to improve performance. It uses multiple transmit and receive antennas to support two main modes of operation. *Spatial Diversity* achieves more reliable communication by supplying to the receiver multiple independently faded replicas of the same information symbol. *Spatial Multiplexing* transmits independent information symbols in parallel from the multiple antennas to boost the transmission speed. As MIMO constitutes a significant technological breakthrough, it has been adopted from both wireless LAN (802.11n [6]) and cellular network markets (LTE [5]). The current IEEE 802.11n standard [6] supports MIMO with 4 antennas and 600Mbps rates, while the upcoming 802.11ac standard [7] will allow for 8 antennas and higher than 6Gbps rates. But, how far are we from MIMO gigabit speeds?

Our experiments with MIMO 802.11n devices reveal that current MIMO is low speed and energy hungry. First, we identify a significant drop between the speed that the current MIMO radio can support (physical transmission rate) and the achieved MAC-layer speed (goodput[1]). Specifically, our experiments with commodity MIMO 802.11n devices show a 56% drop between the speed at the MAC layer and the speed that the PHY layer can support at interference-free access point-client settings. Second, the current low MIMO speed comes at a high energy budget, which hinders MIMO deployment at the mobile device side. Our measurements show that, a

[1]Goodput is defined as effective throughput by excluding protocol overheads.

MIMO 802.11n radio can deplete a smartphone's battery in less than two hours, when all its components (i.e., display) but the 802.11n radio are OFF. But why is current MIMO slow and energy hungry? Our study reveals the use of legacy (single-antenna 802.11a/b/g) designs, over the new MIMO 802.11n setting, which can lead to significant performance degradation. To this end, the *goals* of this dissertation are twofold:

- Uncover the limitations of existing designs to achieve high speeds at low energy cost over the new MIMO 802.11 setting.
- Design, implement, and evaluate new solutions which can utilize MIMO speed gains at a low energy budget.

In Section 1.1 we present the roadmap towards high speed, energy efficient MIMO 802.11 wireless.

1.1 Roadmap to the Solution

The goals of this dissertation are to uncover the limitations of existing designs and to present solutions towards high speed (gigabit), energy efficient (green) 802.11 wireless. Our study of gigabit 802.11 communication focuses on the key mechanism that controls the speed over wireless, named *Rate Adaptation* (RA). IEEE 802.11 specifications mandate multiple transmission rates/speeds at the physical layer (PHY). Rate adaptation, which exploits such multi-rate capability, dynamically selects the best goodput rate, based on the time-varying and location-dependent channel quality. When the signal (signal-to-noise ratio - SNR) is strong, rate adaptation must switch at a high rate option, to utilize channel capacity [12]. When the signal is getting weaker (e.g. the wireless client is moving away from the access point) rate adaptation must switch at a low rate option to avoid exceeding the channel capacity, which will result in excessive packet loss.

Our study on rate adaptation starts from the legacy single-antenna 802.11a/b/g setting. We first expose the limitations of existing designs to address the dynamics of the 802.11a/b/g wireless channel (multipath fading and interference) using real experiments. We then design, implement and evaluate a robust 802.11a/b/g rate adaptation. After understanding the dynamics of the legacy 802.11a/b/g network, we extend our view to the MIMO 802.11n setting. Our experiments reveal a significant impact of MIMO communication characteristics on existing rate adaptation algorithms' performance. To this end, we design, implement and evaluate MIMO 802.11n rate adaptation, which considers the unique characteristics of the MIMO 802.11n channel. Unfortunately, MIMO speed comes at a cost of increased MIMO power consumption proportional to the number of antennas. Our study towards green 802.11 communication seeks to identify the energy optimal antenna setting at runtime. We next elaborate on the different components of our study.

Legacy 802.11a/b/g rate adaptation We first experimentally study rate adaptation in the legacy (single antenna) 802.11a/b/g setting. Our key finding is that, practical state of the art algorithms do not adequately utilize the knowledge of wireless channel short-term performance. This results in transmissions at low speed rates. To this end, we design and implement a new History-Aware Robust Rate Adaptation Algorithm (HA-RRAA). HA-RRAA uses short-term loss ratio to opportunistically guide its rate change decisions, and an adaptive time window to limit transmissions at low speed rates. HA-RRAA outperforms existing practical algorithms with goodput gains up to 51.9% in field trials. The next question we seek to answer is how do legacy rate adaptation designs perform over the MIMO setting?

MIMO 802.11n rate adaptation We next experimentally evaluate state of the art practical rate adaptation algorithms over the MIMO 802.11n setting. To our sur-

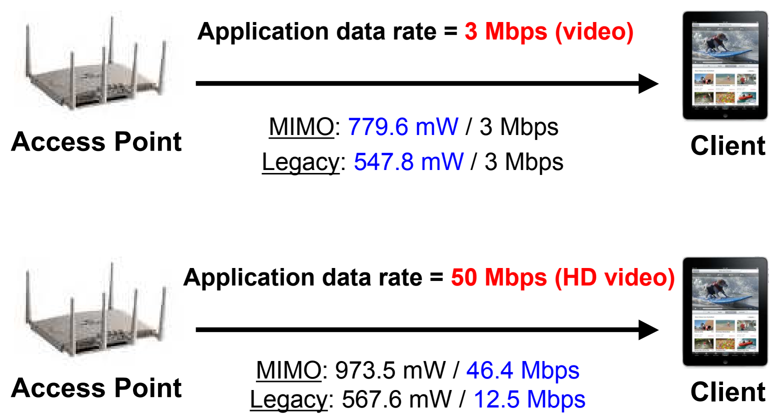


Figure 1.1: MIMO vs. legacy energy tradeoff.

prise, popular RAs give lower speeds than the best goodput fixed-rate scheme. The fundamental problem is that all such algorithms do not properly consider the inherent features of MIMO modes (diversity and spatial multiplexing), which exhibit very different loss characteristics. As a result, they transmit at rates other than the optimal (best goodput rate) sacrificing the MIMO channel capacity. To this end, we design implement and evaluate MIMO rate adaptation (MiRA). Different from existing algorithms, MiRA manages diversity and spatial multiplexing in a distinct manner. Using a novel probing scheme it is able to identify the optimal rate with low overhead. Our experiments with commodity MIMO 802.11n testbeds show that MIMO rate adaptation can outperform existing designs with 73.5% goodput gains in field trials.

From performance to Watt per performance The goal of rate adaptation is to maximize transmission speed over wireless. However, is speed the right metric? Our experiments uncover an important tradeoff between MIMO speed and power consumption. Although MIMO speed increases with the number of antennas, our measurements reveal a monotonic increase of MIMO power consumption with the number of antennas as well. In the first scenario of Figure 1.1, both legacy and MIMO settings can accommodate the offered 3Mbps video data rate. However, the legacy receiver saves

30% power over the MIMO receiver providing a better user experience. A realistic gauge of quality of user experience is the per-bit energy consumption (joule/bit). Per-bit energy consumption is defined as the ratio between the total consumed energy and the delivered bits during any data transfer. In the first scenario of Figure 1.1, legacy receiver saves 30% energy over MIMO. However, when the video data rate increases (50Mbps), the MIMO receiver achieves 3.7 times higher goodput than the legacy receiver. MIMO gains compensate the additional MIMO receiver power consumption and give 54% energy savings over legacy. Our case study of Figure 1.1 uncovers a dilemma between legacy and MIMO 802.11. Which is the most energy efficient?

In the second part of this dissertation (Chapter 6) we revise our metric from performance to Watt per performance. Then, we design MIMO energy save which seeks to identify the energy optimal antenna setting at runtime using a low-cost informed probing scheme.

1.2 Contributions

In this dissertation we seek to build a strong connection between wireless communications and wireless protocol design. To this end, we first provide a deep understanding of 802.11 wireless channel and radio features using real experiments and study their impact on the performance of existing protocols. We then propose novel algorithms that harness the insights gained from our experimental results and analysis. Our implementation and evaluation with IEEE 802.11 standard-compliant testbeds show that our designs can deliver large gains in practical settings.

1.2.1 Experimental study of 802.11 MAC and channel features

This dissertation builds on a strong understanding of the characteristics of the 802.11 MAC layer and wireless channel.

Legacy 802.11a/b/g channel We first study 802.11a/b/g channel using commodity hardware. Our results reveal significant channel fluctuations even in static, indoor, interference-free settings. Existing rate adaptation algorithms present limitations to capture these short-term channel dynamics, which results in poor goodput performance.

MIMO 802.11n channel *This dissertation provides the first experimental study of fundamental MIMO wireless communication tradeoffs using MIMO 802.11n commodity testbeds.* We first uncover the tradeoffs between diversity and spatial multiplexing MIMO modes. Then, we experimentally study the diversity and spatial multiplexing gains as a function of the number of antennas. Our work departs from wireless communication theory studies in three key ways. *a)* First, our study reveals that findings which are considered norms in wireless communications may not apply in a practical 802.11n setting. For example, signal-to-noise ratio (SNR) metric which has been used in theory to differentiate MIMO modes [10] has limitations to identify the best goodput MIMO mode in real 802.11n systems. *b)* Second, our study uncovers new factors as 802.11n MAC-layer frame aggregation which can affect the performance of MIMO modes. *c)* Finally, different from theoretical work we study the impact of MIMO channel dynamics in MAC-layer algorithms' (i.e. rate adaptation) performance.

802.11 MAC-layer features Our study uncovers 802.11 MAC-layer and radio features, which play a key role in 802.11 wireless networks' performance. At the MAC-layer, 802.11n frame aggregation used to amortize protocol overheads has a significant impact on goodput performance. On the radio, our measurements reveal a monotonic increase of power consumption with the number of antennas. This increase in power consumption significantly affects MIMO systems' energy performance.

1.2.2 Novel algorithms for rate adaptation and MIMO energy save

This dissertation provides a critique on practical, state of the art algorithms for rate adaptation and MIMO power save. After identifying the Achilles' heel of existing algorithms, it suggests a fresh angle on how to design gigabit and green 802.11 wireless networks. We highlight the main algorithmic contributions below.

- *HA-RRAA* has been designed for legacy 802.11a/b/g networks, and uses short-term loss ratio to opportunistically adjust the rate based on the wireless channel quality. The key feature of HA-RRAA is its adaptive time window, which prevents transmissions at low goodput rates.

- *MiRA* is among the first practical rate adaptation designs for MIMO 802.11n wireless networks. Different from existing algorithms, MIRA manages MIMO diversity and spatial multiplexing modes in a distinct manner. This allows for MiRA to identify the best goodput rate at low probing overhead.

- *MRES* introduces Watt per performance as the evaluation metric of a MIMO system. It seeks to save energy by identifying the energy optimal antenna setting using an informed probing scheme.

1.2.3 Implementations with 802.11 standard-compliant testbeds

This dissertation provides the first implementation and evaluation of MIMO rate adaptation and energy save using 802.11n standard-compliant commodity devices. Our proposed designs are practical in three ways. First, they are 802.11 standard-compliant and have been implemented in 802.11 commodity hardware. Second, they do not make any assumptions about the implementation of the underlying 802.11 radio and its features. Finally, they do not require channel feedback (SNR) from the receiver, which is not supported by the current 802.11 systems. We summarize the major experimental results below.

- We evaluate HA-RRAA in both controlled static and mobile settings and realistic field trials where various sources of dynamics coexist in a complex manner. The comparison of HA-RRAA with state of the art practical designs as ARF [17], SampleRate [23] and RRAA [34], shows goodput gains from 6% to 52% in realistic field trials.

- We compare MiRA and several MIMO RA alternatives with both popular practical legacy [23, 34] and MIMO [32] rate adaptation algorithms. We conduct our experiments in various scenarios with static/mobile clients, hidden terminal stations, under different MIMO configurations with both TCP and UDP traffic. Our MIMO RA proposal shows 73.5% goodput gains in realistic field trials.

- We compare MRES with designs that represent two different philosophies. First, we enable all the antennas at both sender and receiver seeking to maximize performance (speed). Second, we compare MRES with the IEEE 802.11n Spatial Multiplexing Power Save (SMPS) feature. SMPS has been proposed by the 802.11n standard, and seeks to save MIMO power (Watt), by switching from "many" to a "single" antenna setting. MRES yields 37% energy savings in a two-antenna 802.11n receiver.

1.3 Organization of the Dissertation

This dissertation can be divided and read in different ways depending on what the reader is looking for. The order of the chapters reflects the transition from legacy to gigabit and further to green, gigabit wireless. Chapter 2 provides background knowledge for the 802.11 standard, rate adaptation, and energy save. Chapter 3 presents the experimental setup and methodology used throughout this study. Chapter 4 presents our first step towards gigabit wireless. We first study the performance of existing legacy 802.11a/b/g rate adaptation algorithms and then we design, implement, and evaluate the HA-RRAA algorithm. In Chapter 5, we shift our focus from legacy 802.11a/b/g to MIMO 802.11n wireless networks. We first identify the limitations of existing RAs to perform well over the MIMO setting. Then, we design, implement, and evaluate MIMO rate adaptation for 802.11n wireless networks. In Chapter 6, we revise our metric to evaluate an 802.11n system from performance to Watt per performance. We next design, implement, and evaluate MIMO receiver energy save which seeks to identify the energy optimal antenna setting at runtime. Finally, Chapter 7 concludes this dissertation and provides our future directions. Although this dissertation is structured to guide the reader one step at a time towards gigabit and green 802.11 wireless, all chapters can be studied independently as well.

This dissertation can be of interest for both an academician and a practicing engineer. Chapter 5 shows that MIMO diversity and spatial multiplexing modes exhibit different characteristics. Chapter 6 uncovers a significant tradeoff between speed and power consumption. These can fundamentally change our philosophy of how to build protocols over the MIMO setting, as we discuss in Chapter 7. Besides a new design philosophy, this dissertation can serve as a tutorial for implementing algorithms in 802.11 wireless drivers. Chapters 4, 5, 6, uncover 802.11 drivers' unique features, implementation challenges, and solutions for rate adaptation and MIMO energy save.

CHAPTER 2

Background

This dissertation considers both infrastructure and ad-hoc 802.11 wireless networks. An 802.11 wireless local area network (WLAN) is subdivided into cells (called Basic Service Set or BSS). In infrastructure mode, wireless clients communicate through an access point (AP), which serves as a bridge to a wired network infrastructure, as shown in Figure 2.1. Both clients and access points use 802.11a/b/g/n interfaces. In ad-hoc mode, there are no APs and wireless clients communicate directly with each other. The wireless clients can be either static or roam between APs, while the APs are typically static. We next summarize the IEEE 802.11 features related to our study, and give an overview of prior work on 802.11 rate adaptation and energy save.

2.1 IEEE 802.11 Standard

The IEEE 802.11 standard specifies the physical (PHY) and medium access control (MAC) layers of the protocol stack. Based on PHY and MAC layer designs, 802.11 is divided in different standards, named with different letters (e.g. 802.11a/b/g/n). In this section, we present the 802.11 PHY and MAC layer features, related to our study.

802.11 PHY The IEEE 802.11 PHY layer operates either at 2.4GHz band for 802.11b/g/n or at 5GHz band for 802.11a/n. Each frequency band is subdivided in smaller frequency bands, named channels. For example 2.4GHz band in USA is di-

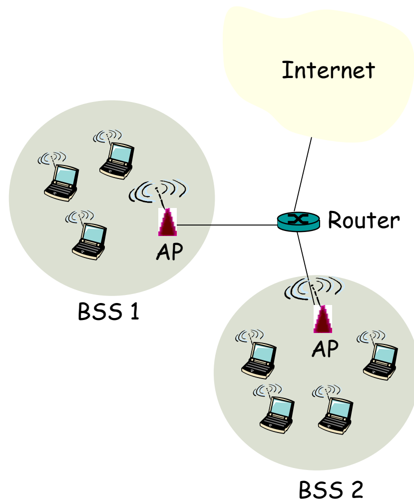


Figure 2.1: IEEE 802.11 LAN architecture.

vided in 11, 20MHz channels. Only channels 1, 6 and 11 are non-overlapping as we present in Figure 2.2. The 802.11a/g/n physical layer is based on Orthogonal Frequency Division Multiplexing (OFDM). OFDM partitions the 20MHz 802.11 channel (or carrier) into 64 subcarriers of 312.5KHz each, such that every subcarrier can be considered of as a separate narrowband channel. In 802.11 OFDM, data is sent on the subcarriers using the same modulation, coding scheme and transmit power.

IEEE 802.11 standards allow for multiple PHY transmission rates. An 802.11b device can use four rate options of 1, 2, 5.5, 11Mbps. An 802.11a device can use eight rate options of 6, 9, 12, 18, 24, 36, 48, 54Mbps. An 802.11g device can use all twelve rate options. The new IEEE 802.11n allows for rates up to 600Mbps, while the upcoming 802.11ac will support rates up to 6.93Gbps. The PHY transmission rate R can be calculated by the following equation:

$$R = 12 \cdot BW_f \cdot N_{SS} \cdot N_b \cdot R_C \cdot GI_f \tag{2.1}$$

BW_f is the channel bandwidth factor. BW_f is 1 and 2.25 for 20MHz, and 40MHz channel bandwidths, respectively. N_{SS} represents the number of spatial streams. The

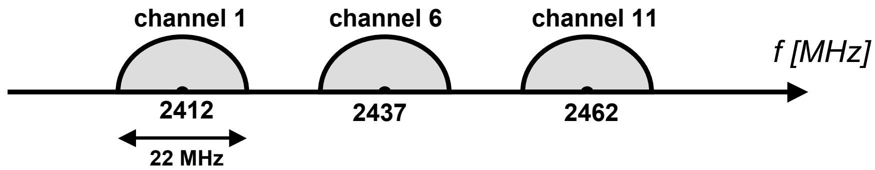


Figure 2.2: Channels of 2.4GHz band.

legacy 802.11a/b/g standards support only one spatial stream. As we describe in Section 2.2, MIMO 802.11 allows for higher number of streams, and as a result higher rates. N_b is the coded bits per OFDM subcarrier. It is 6 for 64-QAM, 4 for 16-QAM, 2 for QPSK and 1 for BPSK modulation schemes, which are supported by 802.11a/g/n. The code rate R_C of a forward error correction code, is the proportion of the data-stream that is useful (non-redundant). Finally, the guard interval GI is used to ensure that, distinct transmissions do not interfere with each other. The guard interval factor GI_f is 1, 1.11 for 800ns and 400ns guard intervals, respectively.

The rate to be used for transmission, is communicated from the transmitter to the receiver in the *Signal* field of the 802.11 PLCP header, of an 802.11 transmission (Figure 2.3). The rate is decided on the MAC layer, by the rate adaptation algorithm as we discuss in Section 2.3.

802.11 MAC The default operation mode for wireless LAN/ad-hoc networks is the Distributed Coordination Function (DCF), which applies Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In CSMA/CA, a station senses the wireless channel, and transmits only when the channel is free. A successful DATA frame transmission is acknowledged by an ACK frame. Specifically, when an 802.11 sender senses a free channel for DIFS (DCF Interframe Space) time interval, it transmits the entire frame. Upon successful reception of a DATA frame, the 802.11 receiver returns an ACK frame after SIFS (Short Interframe Space) time interval. If the 802.11

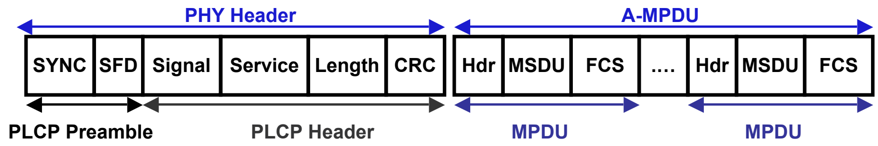


Figure 2.3: IEEE 802.11n frame format.

sender senses a busy channel, it starts a random backoff timer. The timer counts down only when the channel is free. Finally, the DATA frame is transmitted when the timer expires. Timer increases or decreases upon a failed/successful transmission respectively. The impact of backoff timer on 802.11 performance has been studied in [78].

The hidden terminal problem: Carrier sensing cannot always prevent packet collisions. In the hidden terminal case, two or more senders can be in the range of an intended receiver, but out of the range of each other. In Figure 2.4(a), while station B transmits to C, station D can act as a hidden terminal, as it cannot sense station B transmission. As a consequence, any transmission of D will result in a collision at C.

There have been many recent proposals that, seek to address interference and hidden terminals [68–72]. The state of the art solutions are either not 802.11 standard-compliant (e.g. [69, 71]), or they require PHY-layer modifications and additional hardware (e.g. [68–72]), which make them impractical for commodity 802.11 devices. To address the hidden terminal problem, the IEEE 802.11 standard proposes the RTS/CTS feature, which seeks to reserve the area around the sender and receiver for the duration of the packet exchange. A station wishing to send data, initiates the process by sending a Request to Send frame (RTS). The receiver replies with a Clear to Send frame (CTS). Stations that overhear the RTS, CTS frames, defer their transmissions during the frame exchange, by setting their Network Allocation Vector (NAV). Figure 2.4(b) illustrates the RTS/CTS handshake process.

RTS/CTS signaling overhead includes the interframe spacing intervals (SIFS,

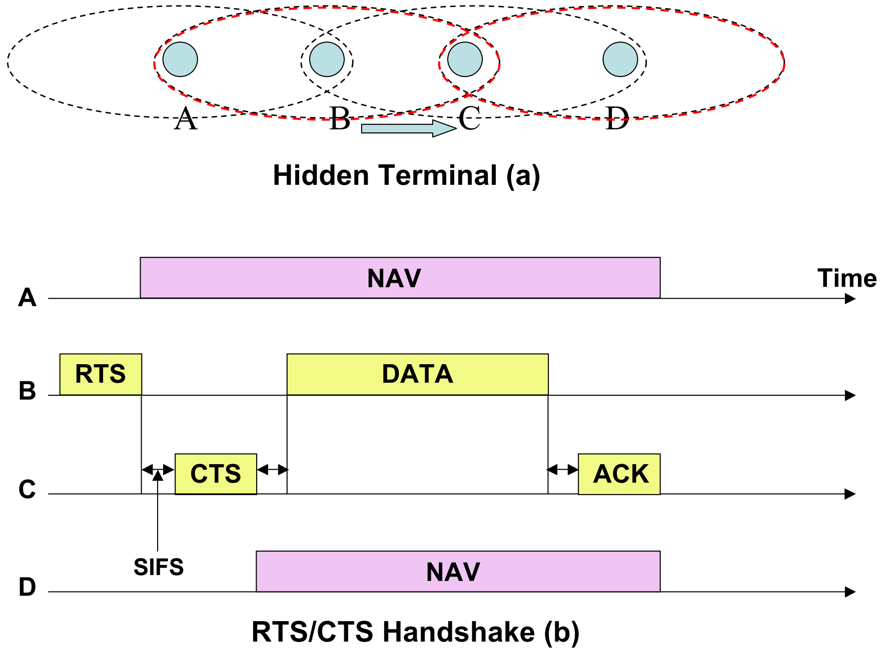


Figure 2.4: Hidden terminal and RTS/CTS handshake.

DIFS), and the transmission time of RTS/CTS frames (Figure 2.4(b)), which are transmitted using the basic low rate (24Mbps in our platform). As this overhead is significant, RTS/CTS is often turned off by default in commercial APs and wireless adapters. Our proposed MIMO rate adaptation, uses a low cost RTS/CTS scheme to address collision losses (Chapter 5). Specifically, it leverages the 802.11n frame aggregation feature to detect collision losses, and selectively enables the RTS/CTS feature.

2.2 IEEE 802.11n New Features

The new IEEE 802.11n and the upcoming 802.11ac standards incorporate several new features to boost performance. The most important are Multiple-Input Multiple-Output (MIMO), channel bonding at PHY-layer and frame aggregation at MAC-layer.

MIMO IEEE 802.11n and 802.11ac PHY uses multiple transmit and receive an-

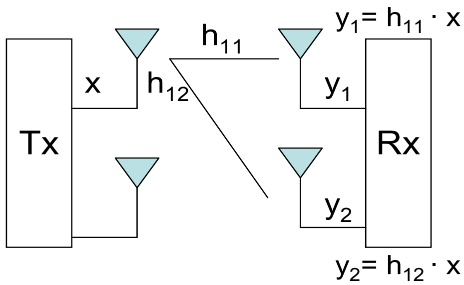


Figure 2.5: Receive diversity.

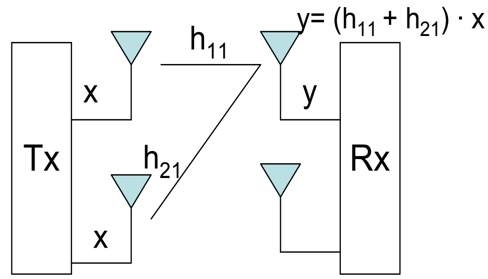


Figure 2.6: Transmit diversity.

tennas to support two MIMO modes of operation. *Spatial diversity* transmits a single data stream from each transmit antenna, leveraging the independent fading over multiple antenna links, to enhance signal diversity. *Spatial multiplexing* (SM) transmits independent and separately encoded spatial streams from each of the multiple transmit antennas, to boost performance. IEEE 802.11n standard supports up to four spatial steams, while the upcoming 802.11ac will support up to 8 streams. MIMO modes may be also combined, and diversity and SM gains can be obtained simultaneously [9].

Spatial diversity: Diversity techniques can be applied at both receiver and transmitter sides. Figure 2.5 illustrates receive diversity for a 2-antenna receiver (1x2 system). Each antenna receives a copy of the transmitted signal, modified by the channel H between the transmitter and receiver. The coefficient h_{ij} of the channel matrix H, is a complex number that represents the path gain from transmit antenna i to receive antenna j. The simplest diversity method is *Selection Combining* (SEL), which considers only the strongest signal for packet reception, and ignores the others. The more sophisticated *Maximal-Ratio Combining* (MRC), combines the signal at the receiver, and produces an SNR that is the sum of the antenna SNRs. Specifically, the receiver multiplies the received signal $\vec{y} = \vec{h}x + \vec{n}$ by the unit vector $\vec{h}^*/\|h\|$, where $\vec{h}^*$ denotes the complex conjugate of $\vec{h}$ and $\vec{n}$ is the noise vector. This operation scales each antenna's signal by its magnitude, and rotates the signals into the same phase reference

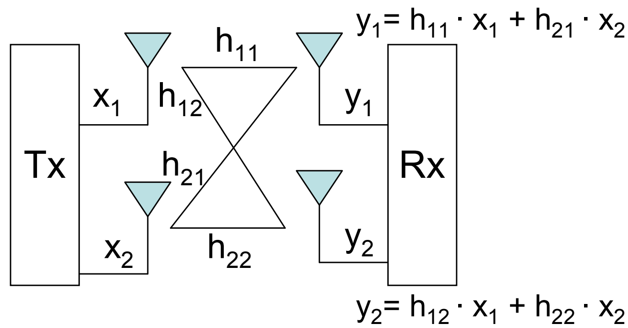


Figure 2.7: Spatial multiplexing.

before adding them. An example of MRC can be found in [80].

Transmit diversity uses multiple transmit and a single receive antenna (Figure 2.6). The transmitter equivalent of SEL is to select the single transmit antenna with the strongest signal. In the equivalent to MRC transmit diversity, the transmitter precodes the signals by delaying them to change the phase such that, they will be combined constructively at the receiver's antenna. It also weights them such that, transmit power is allocated to each spatial path based on its SNR. The disadvantage of transmit diversity over receive diversity is that, the transmitter must know the channel H beforehand in order to select between antennas or to precode the signals. This feedback may not be available in an 802.11 device.

Spatial multiplexing: Different from diversity, in spatial multiplexing mode the transmitter sends independent signals/streams x_i simultaneously from the different antennas (Figure 2.7). The PHY transmission rates linearly increase with the number of streams N_{SS} from the equation 2.1. We can express the received signal as a linear system, using the channel matrix H, the transmitted signal vector $\vec{x}$, the received signal vector $\vec{y}$ and the noise vector $\vec{n} : \vec{y} = H\vec{x} + \vec{n}$. In order to decode the multiple streams, we need simply to solve this linear system. The transmitted signal $\vec{x}$ is estimated as $H^{-1}\vec{y} = \vec{x} + H^{-1}\vec{n}$. The matrix H will be invertible if the different spatial paths are

Setting	Capacity (bits/sec)
SISO	$BW \cdot log_2(1 + SNR)$
Diversity (1xN or Nx1)	$BW \cdot log_2(1 + SNR \cdot N)$
Diversity (NxN)	$BW \cdot log_2(1 + SNR \cdot N^2)$
Spatial Multiplexing (NxN)	$BW \cdot N \cdot log_2(1 + SNR)$

Table 2.1: Channel capacity of diversity and spatial multiplexing modes.

independently faded, making stream decoding feasible.

The capacities of the wireless link for diversity and SM modes for an ideal channel, are presented in Table 2.1. BW is the channel bandwidth, while SNR is the signal-to-noise ratio. The number of antennas is N. At low SNR locations, diversity is usually preferred. At high SNR locations, SM allows for faster transmissions.

Channel-bonding IEEE 802.11n can simultaneously use two separate channels to transmit data, thus doubling the rate in principle. So, while the legacy 802.11a/b/g devices use a single 20MHz channel, 802.11n can operate in the 40MHz mode over two adjacent channels, one as the control and the other as the extension. The upcoming 802.11ac can support up to 160MHz channels.

However, as we can see from Figure 2.2, all the 40MHz channels are partially overlapping in the 2.4GHz band, as opposed to the 20MHz channels 1, 6 and 11, which are non-overlapping. Thus using 40MHz channels can lead to throughput degradation due to increased interference with neighboring channels. Recent proposals [62, 63] seek to dynamically assign channel bandwidths, to address the interference problem. Channel bandwidth assignment is out of the scope of this dissertation.

Frame aggregation IEEE 802.11n seeks to amortize protocol overhead over multiple frames. To achieve this, it packs several data frames in a single aggregated frame. There are two levels of aggregation; a) aggregate MAC protocol service unit (A-MSDU) and b) aggregate MAC protocol data unit (A-MPDU). The main difference

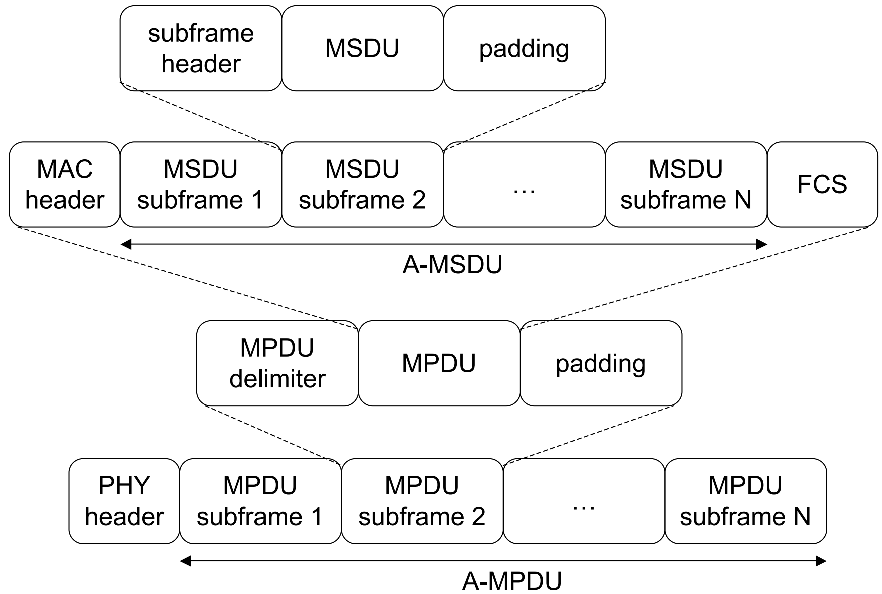


Figure 2.8: IEEE 802.11n frame aggregation.

between an MSDU and an MPDU is that the former corresponds to the information imported to or exported from the upper part of the MAC sublayer from or to the higher layers, respectively. Different from the MSDU, the MPDU is related to the information that is exchanged from or to the PHY by the lower part of the MAC.

A-MSDU: MSDU aggregation allows for multiple MSDUs to be sent to the same receiver concatenated in a single MPDU. This improves the efficiency of the MAC layer, especially when there are many small MSDUs, such as TCP acknowledgments.

A-MPDU: MPDU aggregation joins multiple MPDU subframes with a single leading PHY header. We define as *aggregation level*, the number of MPDUs in an A-MPDU. A key difference from A-MSDU aggregation is that, A-MPDU functions after the MAC header encapsulation process. The maximum A-MPDU size is 65,535 bytes. A successful A-MPDU transmission is acknowledged by a single BlockAck frame. BlockAck includes a bitmap field of 128 bytes, where each MPDU is mapped using two bytes. So the maximum number of MPDUs that a BlockAck can acknowledge is

64.

The two levels of 802.11n frame aggregation are presented in Figure 2.8, while an evaluation of 802.11n frame aggregation can be found in [79]. This dissertation provides a study of the A-MPDU aggregation level (Chapter 5). It shows that, although aggregation can indeed amortize protocol overheads, it can make rate adaptation less adaptive to fast channel dynamics.

Fast MCS feedback The 802.11n standard also supports MCS feedback (MFB) mechanism, which provides channel state feedback from the receiver to the transmitter. When the MFB field has a value in the range 0 to 126, it represents the Modulation Coding Scheme (MCS), that the transmitter can use for transmission. However, MCS feedback mechanism is optional. When an 802.11n receiver decides not to provide MCS feedback, it will set MFB equal to 127. In Chapter 5, we evaluate MCS feedback, and present its limitations in a practical setting.

2.3 Background on Rate Adaptation

Rate adaptation is a mechanism unspecified by the IEEE 802.11 standards, yet critical to the system performance by exploiting the multi-rate capability at the PHY layer. It selects the best goodput transmission rate based on the wireless channel quality. The challenges that RA needs to overcome are twofold. First, the wireless channel can dynamically change because of multipath fading, mobility and interference. Second, RA has to select the best rate from a wide set of available rate options. As rate adaptation is the key mechanism to utilize channel capacity, it has been an active research topic for more than 15 years. In this section we revisit the solution space and categorize the existing designs, while in Chapters 4, 5 we critically examine commonly adopted design guidelines using real experiments.

2.3.1 Solution space for RA

At its core, each rate adaptation algorithm should possess at least two basic mechanisms. *Estimation* mechanism either directly estimates the best transmission rate based on the current channel (e.g. using SNR), or indirectly infers the best rate by gauging how well the currently chosen rate performs (e.g. using frame loss statistics). *Action* mechanism decides when and how the transmission rate is updated given the outcome of channel estimation. Based on how these two mechanisms are implemented, we can categorize rate adaptation designs into several general approaches.

Estimation: Rate adaptation algorithms can be classified based on the different layers, information units and techniques they use to estimate the wireless channel.

Which layer to use? Algorithms can be classified based on the protocol stack layer they use. PHY-layer approaches can utilize SNR feedback (RBAR [19], OAR [20], CHARM [25], FARA [28], ESNR [33]), bit error rate (BER) information (SoftRate [27]), or other signaling information as signal distortion (AccuRate [29]). MAC-layer approaches use frame transmission successes/ losses to indirectly infer the channel quality (ARF [17], AARF [18], ONOE [21], SampleRate [23], Atheros MIMO RA [32], RRAA [34]). Hybrid approaches combine both PHY and MAC-layer information (HRC [26]).

Which information unit to use? Cross layer designs utilize symbol level information to estimate channel quality (SoftRate, FARA, AccuRate). The remaining designs use MAC-layer frames to measure SNR or loss. These designs can be further classified in the designs that use DATA frames as ARF and SampleRate, or signaling frames as RTS/CTS (e.g. RBAR and OAR).

How to estimate? SNR-based designs translate the measured SNR into a best transmission rate based on pre-defined mappings. Loss-based designs estimate the channel quality based on the outcome of previously transmitted frames. *Deterministic pattern* approaches treat consecutive frame successes/ losses as an indication of good/bad channel condition (e.g. ARF). *Statistical metric* approaches use long-term or short-term frame statistics to statistically estimate the best possible rate (e.g. SampleRate).

Action: There are two main approaches to adjust the rate upon channel estimation. *Sequential RA* designs, adjust one rate option at a time. So, they move to the next higher/lower rate when channel becomes good/bad respectively. *Best RA* designs switch directly to rates which yield the best performance.

2.3.2 An overview of RA designs

We broadly classify RA designs as SNR-based and loss-based.

SNR-based designs: RBAR [19], one of the first proposed SNR-based designs leverages the RTS/CTS exchange to estimate SNR at the receiver side. CHARM [25] leverages reciprocity of the wireless channel to estimate average SNR at the receiver using packets overheard from the receiver. So, it avoids the overhead of RTS/CTS, and enables implementation on commodity cards. FARA [28] uses per-frequency SNR measurements to enable a transmitter to use different bitrates across different OFDM subbands. ESNR [33] has been designed for Multiple-Input, Multiple-Output (MIMO) 802.11n systems and uses Channel State Information (CSI) feedback, available from the receiver to the transmitter only in 802.11n systems. SNR-based solutions that require feedback from the receiver, are not 802.11a/b/g standard-compliant and as a result they have not been popular in commercial 802.11 systems.

Loss-based designs: This class of algorithms utilizes PHY-layer [27] or MAC-layer [17,18,23,32,34] loss feedback to decide the next rate for transmission. This dissertation focuses on practical 802.11a/b/g/n algorithms (ARF [17], SampleRate [23], Atheros MIMO RA [32], RRAA [34]), which make decisions solely based on the

MAC-layer ACK frame sent upon successful delivery of a DATA frame. ARF and SampleRate are *probing-based* designs in which a few data frames are occasionally transmitted at a rate different from the current one to *probe* the channel. ARF sends a probe packet at a rate higher than the current one, upon either ten consecutive transmission successes or when a timer expires. If the probe packet succeeds, it increases the transmission rate. ARF decreases the rate upon two consecutive transmission failures. SampleRate maintains the expected transmission time for each rate, and updates it after each transmission. A frame is transmitted at the rate that currently has the smallest expected transmission time. Different from probing based algorithms, RRAA uses a short-term loss ratio to assess the channel and opportunistically adapts the runtime transmission rate to dynamic channel variations. Finally, Atheros MIMO RA selects the best goodput rate based on loss statistics, while it upper-bounds probing and rate selection.

A recent proposal [41] seeks to achieve optimal, collision-resilient RA, without requiring channel state feedback, using rateless codes. Strider [41] has limitations to be applied at an 802.11 setting. First, it requires new encoder/decoder components and MAC-layer protocol changes, which are not supported by IEEE 802.11 standards. Second, its decoder's complexity grows with the density of the modulation scheme. This poses new challenges for the upcoming 802.11ac, which supports up to 256-QAM. Finally Strider has not been designed for MIMO. This dissertation focus on the study of practical, IEEE 802.11 standard-compliant designs.

2.3.3 Departures from the state of the art 802.11 RAs

The rate adaptation proposals presented in this dissertation (HA-RRAA, MiRA), belong to the loss-based designs. This decision is based on our experimental study presented in Chapters 4, 5, which shows the limitations of SNR-based proposals to be used in real 802.11 settings. However, HA-RRAA and MiRA depart from the existing loss-based designs [17, 18, 23, 32, 34] in the following key ways. First, our proposals apply history-aware, prioritized probing schemes, to identify the best goodput rate at low probing cost. MiRA is able to address the unique loss characteristics of diversity and spatial multiplexing MIMO modes, by applying a Zigzag probing scheme. HA-RRAA and MiRA are able to eliminate transmissions at rates that consistently offer lower goodput, by applying an adaptive time probe interval. Different from existing designs [23, 32, 34], our rate adaptation proposals use events to trigger probing, and rapidly adapt to channel dynamics.

Although interference detection and reaction mechanisms are usually decoupled from rate adaptation, our study shows a significant impact of collision losses on rate selection process. Different from existing proposals [17, 18, 23, 32], MiRA and HA-RRAA integrate mechanisms that, can differentiate channel from interference losses, and react by selectively enabling RTS/CTS. Finally, our rate adaptation algorithms utilize the unique features of 802.11 protocols. For example, MiRA considers 802.11n frame aggregation features both in rate selection and collision detection process.

2.4 Background on 802.11 Energy Save

IEEE 802.11 energy efficient designs have been widely studied both on the infrastructure [43–45] and client sides [46–58].

802.11 infrastructure energy save Infrastructure energy save proposals [43–45] seek to save energy consumed in the 802.11 infrastructure components (i.e., APs, controllers). SEAR [44] and Wake-on-WLAN [45] adopt the resource on demand strategy. They seek to save energy for idle APs, which do not serve any traffic. Specifically, they strategically power on and off APs to save energy, based on users' demand. SEAR forms clusters of APs that are in close proximity. Then, it strategically powers on and off APs that belong to the same cluster, based on the traffic demand that the cluster needs to serve. Different from SEAR, which requires a central controller, in Wake-on-WLAN design, each AP independently makes decisions to power itself off, when it does not see any clients in its vicinity. Turn on/off APs solutions have been designed for legacy 802.11a/b/g networks, and they do not consider MIMO power consumption. Designing MIMO energy save for the 802.11 infrastructure is part of our future work.

802.11 client power save On the wireless client side, there are three main directions to save power in an 802.11 network. First, MIMO 802.11 speed gains come at the cost of increased power consumption, due to the added complexity of MIMO circuit blocks. MIMO circuitry power consumption is proportional to the number of antennas [64]. MIMO power save seeks to identify the most power efficient antenna setting. Second, an 802.11 interface consumes power even when it does not transmit or receive any data, while sensing the channel for incoming transmissions. Idle power save seeks to save 802.11 power consumption during idle times, when the adapter does not transmit or receive any data. Third, power consumed on power amplifiers is proportional to the transmit power [60]. Transmit power save, dynamically adjusts the transmit power in an 802.11 device. We next elaborate on these directions.

MIMO power save: The IEEE 802.11n standard [6] specifies a new Spatial Multiplexing Power Save (SMPS) feature to save MIMO power consumption. SMPS allows for a station to operate with only one active receive chain for a large period of time. It supports two operation modes. In the *Static SMPS* mode, the station retains only a single receive chain, and forces the transmitter to send using only diversity single stream rates. In the *Dynamic SMPS* mode, the receiver switches to multiple receive chains before every multiple stream transmission, which is preceded by a RTS/CTS handshake. It switches back immediately to one active chain, when the frame sequence ends. In

Chapter 6, we expose SMPS limitation to save both power and energy in practical settings.

Different from SMPS which only considers MIMO power consumption, Snooze [58] switches antenna settings according to MIMO speed (airtime utilization). Specifically, it switches to the next higher receiver antenna option, when airtime utilization is higher than a threshold. By increasing the number of antennas, Snooze seeks to increase MIMO speed, and as a result to accommodate the offered application data source rate. Our case study of Chapter 6 shows that, chain selection solely based on speed, can lead to energy sub-optimal antenna setting. Except from antenna management, Snooze schedules the sleep and wake-up intervals, of the clients connected to an AP.

Idle power save: There are many recent proposals [46, 51–53], which seek to save idle power consumption of 802.11 interfaces. In the legacy IEEE 802.11a/b/g Power Save Mode (PSM), clients can sleep adaptively, and wake up only when they intend to transmit, or expect to receive packets. The access point buffers downlink packets, and transmits them only when the client wakes up. Different from the legacy PSM, the 802.11n standard [6] introduces the Power Save Multi-Poll (PSMP) feature, which allows for clients to operate as a group rather than individually. PSMP schedules both downlink (DL) and uplink (UL) traffic for multiple PSMP-capable stations in a PSMP sequence. During a PSMP sequence, a station shall not be able to receive and transmit frames at the times outside its scheduled DL and UL periods. PSMP supports two operation modes. In *Scheduled PSMP*, the AP periodically initiates a PSMP sequence, to serve periodic QoS traffic. In *Unscheduled PSMP*, the AP may initiate a PSMP sequence for PSMP-capable, awake stations, at any time.

Transmit power save: Transmit power control designs [55, 56] seek to save power consumption by decreasing the transmit power. However, lowering the transmit power, while maintaining the same PHY transmission rate, may result in higher packet losses and lower speeds. On the other hand, increasing transmit power amplifies interference. A joint rate adaptation and transmit power control design has been proposed in [55]. Our MIMO energy save study presented in this dissertation, assumes fixed transmit power. Joint power control and antenna selection is part of our future work.

2.4.1 Departures from the state of the art 802.11 energy save

The 802.11 energy save design proposed in this dissertation (MRES), seeks to save receiver energy on MIMO 802.11 wireless clients. Different from transmit power control designs [55, 56], MRES assumes fixed transmit power, which is the default setting in the most commodity 802.11 devices. We leave the joint antenna and transmit power selection, as a future work. MRES also focuses only on antenna selection, and can work with any idle power save design, as the new IEEE 802.11n PSMP. Interestingly, our study in Chapter 6 shows that, MRES can increase sleep time opportunities and lead to energy savings, when it works in concert with idle power save solutions.

Recent proposals (SMPS [6], Snooze [58]) apply antenna selection to save energy at 802.11n receivers. SMPS seeks to save power consumed in MIMO circuit blocks, by switching from “many” to a “single” antenna setting. Snooze [58] switches antenna settings according to MIMO speed (airtime utilization). For example, Snooze will switch at a higher antenna configuration to accommodate the offered application data source rate. Our case study presented in Chapter 6 shows that, antenna selection solely based on MIMO speed, or power consumption can lead to energy sub-optimal antenna selection. MRES departs from these proposals by considering both speed and power in antenna management.

CHAPTER 3

Experimental Methodology

The results presented in this paper are obtained from real experiments. We next describe our experimental platform, setting and methodology.

Experimental platform The sender for all our experiments is a programmable AP platform, which uses Atheros AR5416 2.4/5 GHz 802.11a/b/g/n capable chipset. The 802.11 MAC is implemented in the FPGA firmware, to which we have access. The platform has several appealing features that facilitate our research on rate adaptation and energy save. First, we can implement our own algorithms, and run them at the AP. Second, we can perform per-frame tracing of various metrics of interests, such as frame hardware retries and per-antenna SNR values. Third, we can configure many different parameters in real time on a per-frame basis such as: a) the maximum retry count, b) RTS option, c) the transmission rate for each frame retry. Finally, the feedback delay from the hardware layer is small, which implies that timely link-layer information is available to rate adaptation and energy save. We repeat our experiments with various wireless clients, which use Broadcom, Marvell and Atheros chipsets. For each experimental setting we describe our wireless client characteristics.

Our AP supports all the 802.11n new features, presented in Section 2.2, besides MCS feedback. It allows for diversity, single-stream (SS) and spatial multiplexing, double-stream (DS) modes. It also supports three antennas (RF chains). Its available rate options can go up to 130Mbps and 300Mbps for 20MHz and 40MHz channels

MCS	Modulation	Code Rate	Mode	Rate (Mbps) at 40MHz	Rate (Mbps) at 20MHz	A-MPDU Max Size (bytes) at 40MHz	A-MPDU Max Size (bytes) at 20MHz
0	BPSK	1/2	SS	13.5	6.5	6684	3216
1	QPSK	1/2	SS	27	13	13368	6434
2	BPSK	1/2	DS	27	13	13360	6430
3	QPSK	3/4	SS	40.5	19.5	20052	9650
4	16-QAM	1/2	SS	54	26	26738	12868
5	QPSK	1/2	DS	54	26	26720	12860
6	16-QAM	3/4	SS	81	39	40104	19304
7	QPSK	3/4	DS	81	39	40080	19300
8	64-QAM	2/3	SS	**108**	52	53476	25740
9	16-QAM	1/2	DS	**108**	52	53440	25736
10	64-QAM	3/4	SS	121.5	58.5	60156	28956
11	64-QAM	5/6	SS	135	65	66840	32180
12	16-QAM	3/4	DS	162	78	80160	38600
13	64-QAM	2/3	DS	216	104	106880	51472
14	64-QAM	3/4	DS	243	117	120240	57890
15	64-QAM	5/6	DS	270	**130**	133600	64320
16	64-QAM	5/6	DS	**300**		148400	

Table 3.1: 802.11n rate options for 20/40MHz channels.

respectively. Table 3.1 shows our AP's rate options and their characteristics. Our AP and all the 802.11n adapters used for wireless clients in this dissertation, use MRC for receive diversity, as described in Section 2.2. The transmit diversity algorithm for our AP is Cyclic-Delay Diversity (CDD). CDD transforms spatial diversity into frequency diversity. Specifically, the signal is cyclically shifted via the available antennas, to address intersymbol interference.

Frame aggregation with BlockAck (i.e., ACK for A-MPDU) feedback is supported from both our AP and our 802.11n wireless clients. Upon receiving a BlockAck, the rate adaptation module gets feedback including the number of MPDUs in the transmitted A-MPDU (called as $nFrames$) and the number of MPDUs received with errors (called as $nBad$). If the entire A-MPDU is lost, the number of hardware retries (called as $retries$) is also available. We can then compute *Sub-Frame Error Rate* as $SFER = \frac{nFrames \times retries + nBad}{(retries+1) \times nFrames}$. Our AP's Atheros driver upper-bounds the A-MPDU

Procedure 1 Atheros MIMO RA

```
maxRate ← forthHighestRate;
probeInterval = 50;
while true do
    on_recv_blockACK(retries, nFrames, nBad);
    SFER ← (retries*nFrames+nBad) / ((retries+1)*nFrames);
    SFER_R(avg) ← 7/8 * SFER_R(avg) + 1/8 * SFER;
    if isProbe&&retries == 0&&2 * nBad < nFrames then
        maxRate ← next_higher_rate (maxRate);
        probeInterval = probeInterval/2;
    else if !isProbe&&SFER_R(avg) > 55% then
        maxRate ← next_lower_rate (R);
        probeInterval = 50;
    end if

    maintain_monotonicity(R);
    R = find_best_thr_rate(maxRate);
    if probeTimerFires&&R == maxRate then
        R = next_higher_rate(R);
        isProbe = true;
    end if
    reduce_SFER_for_all_rate(1/8, 50);
end while
```

(In lines 5–6 and 10: $SFER \leftarrow \frac{retries*nFrames+nBad}{(retries+1)*nFrames}$; $\overline{SFER_R} \leftarrow \frac{7}{8} * \overline{SFER_R} + \frac{1}{8} * SFER$; $!isProbe\&\&\overline{SFER_R} > 55\%$.)

size, such as the ratio between the A-MPDU size and the transmission rate, to be equal for every rate option. As a result, for the max A-MPDU size, it guarantees equal air-time for every transmission. The A-MPDU size upper-bounds for our driver's 40MHz and 20MHz rate options, are presented in the last two columns of Table 3.1. The maximum air-time for each rate is approximately 4ms. In Chapter 5, we evaluate rate adaptation for different A-MPDU sizes. We then uncover the tradeoff between remaining adaptive to channel dynamics and amortizing protocol overheads with frame aggregation.

The default rate adaptation algorithm for our AP platform is Atheros MIMO RA [32]. It selects the best goodput rate based on the weighted moving average SFER performance of each rate. SFER statistics of each rate are updated after transmitting at this rate. There is also an aging mechanism, which periodically (50ms) reduces SFER statistics for each rate by a 1/8 factor. The candidate rates for selection are upper-bounded by a $maxRate$. Upon high/low SFER of the current selected rate R, $maxRate$ can be decreased/increased by one rate option, respectively. The pseudocode of Atheros MIMO RA algorithm is presented in Procedure 1. In Chapter 5, we evaluate Atheros MIMO RA, and uncover its limitations.

Experimental setting We conduct all our experiments both in a campus setting and in RF chamber, a RF shielded room isolated from external RF noises and interferences. We perform both controlled experiments and field trials. We perform controlled experiments at midnight to minimize the impact of external factors, such as signal interference (as verified by our sniffer) and people walking around. Field trials represent more realistic scenarios, in which various sources of dynamics co-exist in a complex manner. We conduct both static and mobile client experiments, under interference-free and hidden terminal settings. The static settings are used to evaluate the stability and robustness of an algorithm, i.e., whether it can stabilize around the optimal setting. The mobility settings evaluate how responsive an algorithm is in adapting to significant channel variations perceived by mobile clients. The hidden terminal settings assess how an algorithm performs under collision losses. We conduct each experiment for more than 8 runs and the results presented are averages over all runs. The frame (MPDU) size used for our tests is 1.5KB.

CHAPTER 4

History-Aware Robust 802.11a/b/g Rate Adaptation

This chapter studies rate adaptation (RA) for legacy 802.11a/b/g systems. Our experiments with commodity 802.11 testbeds, show that popular, practical algorithms [17, 23, 34] perform poorly even in static client, indoor scenarios. The fundamental problem is that, real-world wireless networks exhibit rich channel dynamics, including random channel errors, mobility-induced channel variation, and contention from hidden stations. Existing RAs have limitations to capture these channel dynamics, and as result they can even perform worse than a fixed-rate scheme.

In this chapter, we first conduct a systematic experimental study and simple analysis to examine three popular design guidelines followed by practical 802.11a/b/g RA algorithms. These guidelines include: *1)* the decrease of the transmission rate upon severe packet loss, *2)* the use of PHY metrics to decide new transmission rate *3)* the use of long-term smoothened operation to produce the best average performance. Our experiments surprisingly show that each of the above three seemingly valid guidelines has its own Achilles heel. In fact, even under mild link-layer contention, these designs not only have limitations to facilitate throughput improvement, but also may reduce the throughput and aggravate channel contention, by falsely triggering rate decrease.

We further study the performance variations of the wireless channel. We observe time periods where the channel can accommodate high transmission rates, to be succeeded by significant channel quality degradation. Popular 802.11a/b/g RA solutions may have limitations to prevent transmissions at high loss rates upon bad wireless channel. They may not also switch fast at higher rates, when channel quality improves. This can result in up to 29% goodput loss of existing RA designs over the fixed best goodput rate. To address these challenges, we design and implement a *History-Aware Robust Rate Adaptation Algorithm* (HA-RRAA). HA-RRAA builds upon RRAA [34], and leverages short-term loss ratio to provide not only fresh but also dependable information to estimate the channel quality. It introduces novel mechanisms which improve RRAA performance under dynamic channels and hidden terminals. HA-RRAA applies an adaptive time window to limit the excessive number of transmissions at high loss rates, while remaining responsive to intense channel dynamics. It also leverages the per-frame RTS option in the 802.11 standards and use a cost-effective, adaptive RTS filter to suppress collision losses with low overhead. Our experiments show that HA-RRAA consistently outperforms popular 802.11a/b/g standard-compliant RAs, with 51.9% goodput gains in realistic field trials.

This chapter makes three contributions. It studies the wireless channel dynamics using 802.11 testbeds, and uncovers their impact in existing RAs performance. It proposes HA-RRAA, a new design which can successfully address these dynamics. It compares HA-RRAA with state of the art RAs using real experiments in scenarios with static/mobile stations, TCP/UDP flows, with/without hidden stations, and in controlled/field trial environments.

The rest of this chapter proceeds as follows. Section 4.1 describes our experimental setting. Section 4.2 examines three key design guidelines of existing algorithms. Section 4.3 studies the short-term channel past performance and its impact on rate selection. Section 4.4 presents the design of our proposed HA-RRAA scheme. Section 4.5 describes our implementation and evaluation efforts. Finally, Section 4.6 concludes the chapter.

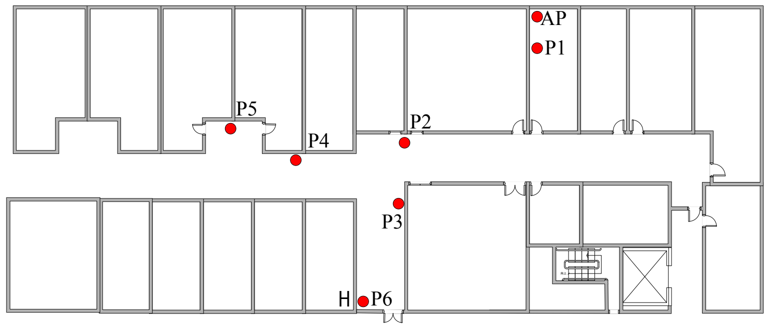


Figure 4.1: Experimental floorplan.

4.1 Experimental Setting

We conduct all our experiments in a campus setting, presented in Figure 4.1. We also conduct SNR measurements in RF chamber. We place our clients in various locations between spots P1 and P6, while our access point located at AP spot, serves in the most cases as the sender of the wireless traffic. For our controlled hidden-terminal experiments we place an AP at location H, which periodically broadcasts frames. Our experimental methodology and our AP device capabilities are described in Chapter 3. At the client side, we use various adapters as Linksys WPC600N 802.11a/b/g/n, CISCO Aironet 802.11a/b/g and AirPort Extreme, Atheros (0x168C, 0x86) adapter. Finally, we conduct each experiment for multiple runs and the results presented are averages over all runs. The average standard deviation for controlled performance (goodput) experiments was smaller than 0.6 Mbps.

4.2 Critique on Existing Design Guidelines

State of the art rate adaptation algorithms have been using several design guidelines as discussed in Section 2.3.1. In this section, using case studies we show that while such guidelines are useful in certain presumed scenarios, they can be misleading in other cases. In the worst case, they yield unexpected, erroneous results.

4.2.1 Case study 1: Decrease transmission rate upon intense packet loss

A fundamental design guideline which has been widely applied in almost all existing algorithms (e.g. [17, 18, 21, 23, 26]) says that upon severe packet loss, rate adaptation should decrease its current transmission rate. The original motivation for this rule is that, whenever the link condition between the sender and the receiver deteriorates and thus incurs significant losses at the current rate, the sender switches to lower rates to adapt to the worsening channel condition.

The above rule is easily broken in practice when hidden terminals exist. Hidden terminals can cause significant loss at the receiver independently of the channel quality. This subsequently triggers rate adaptation at the sender to decrease its rate according to the stated guideline. However, the sender should not decrease its transmission rate upon hidden-station induced losses, because this action will not solve the contention problem. In fact, reducing the rate will make channel contention even worse because it prolongs the transmission time for each packet, which aggravates channel collisions and further reduces the transmission rate.

Our controlled hidden terminal experiments verify the above statements. In our hidden station setting, an 802.11a access point H broadcasting packets at location P6, acts as a hidden terminal to an 802.11a client located at P2. UDP traffic sent from our AP to client at P2, collides with H's broadcast frames. To change the intensity of the hidden terminal setting, we vary the data source rate of the hidden access point H. In Table 4.1 we present how the popular ARF algorithm performs when hidden terminal H is disabled, when H data source is set to 2Mbps and when it is set to 4Mbps. In the modest interference setting of 2Mbps, 59% of the frames are transmitted at rates lower than 36Mbps which results in an 29.65% increase in loss comparing to non-

Rates (Mbps)	ARF Rate Distribution(%)	ARF HT (2Mbps) Rate Distribution(%)	ARF HT (4Mbps) Rate Distribution(%)
6		29	100
9		4	
12		8	
18		8	
24		10	
36	3	9	
48	57	8	
54	40	24	
Goodput (Mbps)	27.93	4.91	0.0015
Loss (%)	17.62	47.27	99.97

Table 4.1: Hidden terminals' effect on rate adaptation.

interference case. In the intense interference scenario, ARF considers collisions as channel losses and it transmits 100% of the frames at 1Mbps. We present more hidden terminal results in Section 4.5.2.

The fundamental problem is that rate adaptation may experience much richer set of packet loss scenarios in practice, which are well beyond the simplistic one of only fading/path loss envisioned by the original designs. The guideline of decreasing rate upon severe packet loss does not apply in other loss scenarios. The RA solution has to differentiate various losses and react accordingly.

4.2.2 Case study 2: Use PHY-layer feedback to infer new transmission rate

There have been many RA proposals [19,20,25–29,33], which utilize PHY-layer feedback to estimate channel quality. However, several of these solutions [19,20,27–29,33] are not 802.11a/b/g standard-compliant. They require explicit feedback from the receiver to the transmitter, which is not available in 802.11a/b/g systems. Moreover, fine PHY-layer feedback may not be exposed from the hardware to the firmware of an 802.11 device, where RAs are implemented. PHY-layer feedback is per-bit confidence information in SoftRate [27], SNR of each OFDM subband in FARA [28] and symbol level dispersion information in AccuRate [29]. As a result, the above solutions have been evaluated solely using simulations and software radio testbeds, rather than commodity 802.11a/b/g adapters.

Signal-to-noise ratio (SNR) metric calculated from RSSI and noise floor feedback on the transmitter side in the 802.11a/b/g drivers, can still be used for channel estimation (CHARM [25]). However, there are significant challenges that these SNR-based algorithms need to overcome. SNR measurements in commodity 802.11 systems may be inaccurate due to hardware calibration and interfering transmissions [25, 30]. To evaluate the SNR-fluctuations under a stable, interference-free channel, we place an 802.11a client approximately 3 meters from our AP in RF chamber. We then create uplink UDP traffic (from client to AP) and we measure the SNR values of back-to-back 1.5Kbyte received UDP frames. We fix the rate at 54Mbps, while we ensure very small gaps ($< 0.6ms$) between consecutive frame transmissions. Frame loss is negligible ($< 0.003\%$). From a trace of 550 frames presented in Figure 4.2, we observe SNR variations which can go up to 4dB between consecutive transmissions. The minimum, maximum SNR observed, is 33dB, 38dB respectively. This large variation can easily lead to more than one rate option deviation from the optimal rate, when translating SNR to transmission rate, based on the goodput versus SNR mappings (Figure 7 of [35]). We come to similar conclusions by experimenting with different transmission rates and distances between client and AP. Except from SNR fluctuations, SNR-BER relationship can change with different propagation environments. Specifically, the SNR measured at the beginning of the packet may not capture the variation in SNR during the frame transmission due to fading. As a result SNR-based protocols require in-situ training to perform efficiently across different propagation environments [24].

To evaluate solutions that use PHY-layer feedback, we compare SoftRate [27] with our proposed HA-RRAA in Section 4.5.2, using ns-3 simulations.

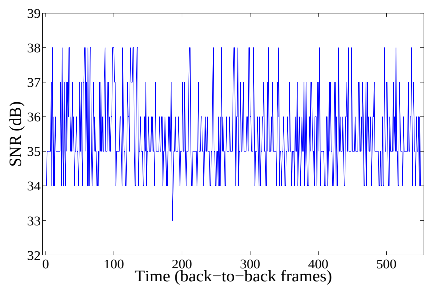


Figure 4.2: Measured SNR in RF chamber over time.

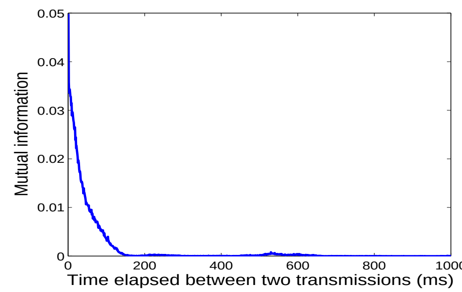


Figure 4.3: Mutual information of two packets separated by x ms in time.

4.2.3 Case study 3: Use smoothened long-term operations to evaluate performance

Another design direction suggests to use long-term smoothened operation in the presence of random losses over the channel. The long-term smoothened operation can refer to either *rate estimation* or *rate change* action, or both. In rate estimation, this rule recommends to use long-term statistical information to estimate the optimal transmission rate. For example, popular algorithms as ONOE [21] and SampleRate [23] both collect packet-level statistics (in terms of loss and throughput) over a period of one to ten seconds. In rate change decision, this rule suggests to only change rates infrequently, say once every 1 or 10 seconds. In both cases, the underlying hypothesis is that long-term estimation/action will smoothen out the impact of random errors and lead to best average performance. Our experiments and analysis based on information theory invalidate both rules.

Our experiments first reveal that long-term rate estimation and rate change action over large sampling periods will not yield best average performance. The experiment is conducted using the ONOE algorithm implemented in MADWiFi. ONOE uses one second as the default sampling interval. It changes its rate based on the packet-level loss statistics collected over each sampling period. In our experimental setting, the

Sampling intervals (ms)	5000	1000	500	100
UDP Goodput (Mbps)	14.9	15.3	16.5	17.1

Table 4.2: Performance of ONOE with different sampling intervals.

sender located at a sibling to P2 spot of Figure 4.1, uses ONOE to send packets to the AP. We vary the sampling period and the results are given in Table 4.2. The table clearly shows that small sampling period of 100ms actually produces the best average performance in the long term. Using large sampling period may lead to 12.9% throughput reduction. In fact, similar results have also been reported in early studies (Figure 3-5 of [23]). One reason for this performance drop is that the algorithm is unable to exploit the short-term *opportunistic gain* over the wireless channel, which typically occurs at the time scale of hundreds of milliseconds.

We next apply the concept of *mutual information* [74] to show that long-term rate estimate over large sampling periods does not help even in the presence of random loss. Mutual information indicates the mutual dependency of two random variables, i.e., how much information one random variable can tell about the other. We treat the transmission success/failure event at a given time as a random variable and calculate the mutual information for two events at different time instants. For our experiments we disable rate adaptation and the frame retry, and record the time for each success/failure transmission. We then calculate the mutual information for each pair of packets separated by an interval of x ms. Figure 4.3 plots the mutual information evolution with respect to different x. The figure shows that their mutual information becomes negligible when two packets are separated by more than 150ms over time. This implies that the success/failure event occurred 150ms earlier can barely provide any useful information for the current rate estimation. We also conduct similar experiments at different locations. All results show that mutual information diminishes when the sampling period becomes larger than 150$\sim$250ms. We can conclude that large sampling periods, ranging from a few seconds to tens of seconds, do not lead to more accurate rate estimation. In this case, assigning different weighting factors for samples over time, becomes a challenging issue.

We finally show that long-term, infrequent rate change decision may also lead to performance penalty. We set up a mobility scenario where an 802.11a client is moving between locations P1 and P5 at approximately constant pedestrian speed of 1m/s. The traffic is UDP from the AP to the client. We next compare ARF and SampleRate implemented in our AP as in MADWiFi. First, while [23] averages the transmission time over a 10-second window, the MADWiFi SampleRate implementation uses exponentially weighted moving average (EWMA) without any window. Second, while [23] suggests per-packet rate decision, the rate is only changed every 2 seconds or upon four consecutive losses in the implementation. Both ARF and SampleRate use relatively short-term rate estimation. ARF sends a probe packet no later than 15 transmissions. SampleRate implementation uses EWMA with a factor of 0.05, which implies that roughly only the recent 50 samples carry major weights in the estimation. However, the rate change actions in both algorithms are quite different. ARF allows for rate change every 10 or 15 packets, while SampleRate takes 2 seconds to switch to a new rate (unless four consecutive losses trigger rate decrease). These different design directions have a significant impact in performance under our mobility scenario. Our experimental results show that, the average UDP goodputs for ARF and SampleRate are 20.6Mbps and 18.8Mbps, respectively. So ARF performs 9.6% better than SampleRate in the mobile client case, which shows that the delayed rate-change decisions hurt the responsiveness of SampleRate.

4.3 The Importance of Learning from Short-Term History

State of the art rate adaptation algorithms do not adequately utilize the knowledge of channel's short-term past performance. RRAA and ARF decide the transmission rate of the subsequent frames, solely based on the performance of the current transmission rate, without considering the outcome of the past transmissions at adjacent rates. AARF [18] seeks to fix the above limitation of ARF by doubling the probing threshold, when a probe packet fails. RRAA+ [31] enhances RRAA by increasing/decreasing a probability $p[R]$ of transmitting at a rate R, when transmissions at R succeed/fail respectively. Finally, SampleRate seeks to limit sampling at high loss rates by excluding from selection for 10 seconds (MADWiFi implementation), rates which faced 4 successive failures. However, the mechanisms used from AARF, RRAA+ and SampleRate to capture channel's past performance, do not adequately address the two main dimensions of the problem.

When does loss happen? RRAA+ and SampleRate falsely consider *rate's and not channel's past performance* to limit transmissions at high loss rates. Specifically, they update the performance of a rate R, only when they transmit at R and not when there is an indication that the channel has changed. Maintaining stale history about a rate's performance, can lead to goodput degradation especially in the scenarios of intense channel dynamics (e.g. mobility).

How severe is the loss? The above history-aware mechanisms adapt probing at high loss rates, when transmissions at these rates fail. However, they do not consider how significant was the loss. For example RRAA+ will halve $p[R]$ when it moves to lower rates independently if the loss ratio for R was 40% or say 100%. AARF will double the probing threshold independently if the probe frame was hardware retried 1 or 10 times, while SampleRate does not distinguish cases where probe frame will face less or more than 4 successive failures.

Rates	RRAA Distribution(%)	RRAA+ Distribution (%)	SampleRate Distribution(%)	ARF Distribution(%)	Fixed Rate Goodput (Mbps)	Fixed Rate Loss (%)
6					5.39	0.64
9					7.74	1.54
12					10.24	0.49
18					14.73	0.80
24	1	1	6	5	18.65	1.96
36	53	96.5	87	65	25.64	3.41
48	46	2.5	6	29.5	12.49	62.84
54			1	0.5	0	100
Goodput (Mbps)	18.20	25.6	23.87	21.85		
Loss (%)	33.13	2.99	9.28	22.03		

Table 4.3: Performance of RRAA, SampleRate, ARF at location P3.

We verify the above limitations in the following section using real experiments.

4.3.1 A case study

We start our study on history-aware rate adaptation by raising two simple questions. How important is for RA designs to consider channel's short-term historical performance? Are the existing RAs history-aware? To systematically answer these questions, we conduct fixed rate experiments in many different locations and we study on a per-frame granularity their run-time loss and goodput performance. To ensure that our observations are attributed to channel dynamics and not to collisions from hidden stations, we switch to 5GHz band on channel 36, which was interference-free during our experiments, as verified by our sniffer.

Our experiments show that there are time intervals where a transmission rate's performance can be highly dynamic, which can be followed by time intervals where a rate's performance is relatively stable (longer than 10 seconds in our experiments). This behavior is attributed to intense channel dynamics, which can change in different environments as stated in [24]. In Figure 4.4 we present a 13-second trace of frame loss evolution of a scenario where client was placed at location P3 and the rate was fixed at 48Mbps. We observe that frame loss presents big variations during the first 5

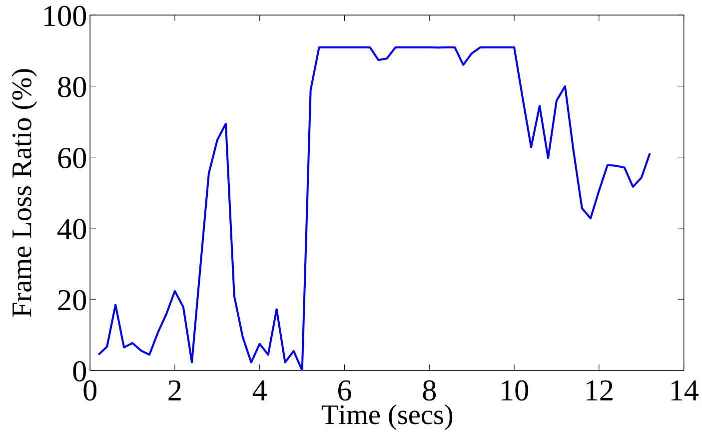


Figure 4.4: Frame loss ratio of 48Mbps over a 13 seconds trace. Each point is an average over 200 milliseconds.

seconds of our trace, while it is relatively stable after the 5th second. More specifically during the first 5 seconds, frame loss can vary from 0% to 69.4%, while it rapidly increases after that. From 5.4th to 10th second loss ranges from 86% to 91%, while overall from 5.2th to 13th second loss is greater than 42%. The average frame loss of our trace is 54.9%.

An efficient RA algorithm should be both highly responsive to rapid channel changes and should be able to limit the number of transmissions at high loss rates. In the previous example, rate adaptation should switch to 48Mbps when its loss is low and limit transmissions at this rate when loss is constantly very high (after 5th second). But, how do state of the art RAs perform in this scenario? We first extensively evaluate the performance of all 802.11a rates at location P3 and we present the results in Table 4.3. From the Table, we observe that rates smaller than 48Mbps give very low frame loss, while 48Mbps gives a significant average loss of 62.8%. Second, we evaluate and study the performance of RRAA, RRAA+, SampleRate, and ARF at that location.

As RRAA and ARF do not keep any state about rates other than the current one, they keep transmitting at high loss rates. From Table 4.3 we see that RRAA and ARF transmit 46% and 30% of the frames respectively at low goodput 48Mbps and 54Mbps rates. As a result, RRAA and ARF give 29% and 14.8% goodput loss respectively, over the fixed best goodput rate, which is 36Mbps (Table 4.3). SampleRate is proven slightly more efficient, as it still transmits 7% of the frames at high loss 48Mbps and 54Mbps rates. This results in 6.7% goodput loss over the best fixed goodput rate. RRAA+ yields the highest goodput among the evaluated algorithms, by transmitting only 2.5% of the frames at 48Mbps. Although RRAA+ is proven to be efficient in our case study setting, our extensive experiments presented in Section 4.5.2, verify the limitations of RRAA+ design discussed above. Interestingly, they also show that its guideline to halve $p[R]$ upon failure, can lead to rate under-selection (selection of rates lower than the best goodput rate).

Based on the lessons learned from our case studies, we design History-Aware RRAA, which seeks to limit transmissions at high loss rates, while remaining adaptive to intense channel dynamics.

4.4 Design History-Aware Rate Adaptation

In this section, we present the design of our History-Aware RRAA (HA-RRAA) algorithm. HA-RRAA builds upon RRAA algorithm presented in [34]. RRAA uses a *short-term loss ratio* to assess the channel and opportunistically adapt the runtime transmission rate to dynamic channel variations. Short-term loss ratio allows for RRAA to be robust against random loss, while remaining responsive to fast channel changes. RRAA calculates the loss ratio in a time estimation window ($ewnd$) as:

$$P = \frac{\#\_lost\_frames}{\#\_transmitted\_frames} \tag{4.1}$$

It also uses two fixed thresholds to sequentially move to higher or lower rates. RRAA will increase/decrease the current transmission rate by one option, if the loss ratio P is lower/higher than a threshold P_{ORI}/P_{MTL}. Finally, RRAA leverages the 802.11 RTS option in an adaptive manner to filter out collision losses with small overhead.

HA-RRAA departs from RRAA in the following two key ways. It leverages an *adaptive time window* to capture short-term channel past performance and avoid probing at low goodput rates. It leverages the per-frame RTS option in the 802.11 standards and use a cost-effective, adaptive RTS filter to suppress collision losses with low overhead. We next elaborate on these mechanisms.

4.4.1 Adaptive time window

Motivated by the 802.11 binary exponential backoff, adaptive time window ($twnd$) mechanism: a) exponentially increases a timer upon transmission failure, b) resets the timer upon transmission success, c) bounds the timer in $[0, T_{max}]$. First, an exponential increase of time window upon failure, allows for our scheme to eliminate the rates that consistently offer lower goodput, by transmitting at these rates less frequently over time. Second, by bounding and reseting appropriately the time window, our mechanism remains adaptive to fast channel dynamics. Adaptive time window is set as $T_R = T_C \times 2^{exp}$. The exponent factor exp represents the number of times that moving from a rate R to the next higher rate fails (results in moving downward at R). It is upper-bounded by 10 in our prototype. T_C represents the minimum estimation window ($ewnd$).

History-Aware RRAA applies the adaptive time window to RRAA, to limit transmissions at the adjacent high loss rates. Our basic adaptive time window mechanism also utilizes the short-term loss statistics offered by RRAA to capture the magnitude of losses, by linearly increasing time window with loss. The revised adaptive time window is expressed as:

$$T_R = T_C \times 2^{exp} \times max(1, \frac{P}{P_0}) \tag{4.2}$$

where P is the short-term loss ratio of the rate R and P_0 is a loss normalization factor set to 10% in our prototype.

HA-RRAA maintains only one time window for the next higher rate R_T of the current transmission rate R. Every time that HA-RRAA moves downward from a rate R_T to R, it will update time window based on the equation (4.2), while it will increase exponential by one. HA-RRAA will reset time window for a rate R_T in two cases: a) When transmissions at R_T are successful, meaning that they do not trigger HA-RRAA to move downward at rate R. b) When channel's further deterioration will trigger HA-RRAA to move from R to the next lower rate. HA-RRAA algorithm is presented in Procedure 2.

4.4.1.1 Handling mobility and hidden terminals

HA-RRAA further improves RRAA in mobility and hidden terminal settings as well.

Fast adaptation: To boost RRAA's responsiveness to fast channel deterioration, we enhance HA-RRAA with fast adaptation mechanism. We maintain a small window of frames (min{ewnd,10} frames in our prototype) and we compute the loss ratio inside this window. If the loss ratio P is greater or equal to a threshold P_{Thresh}, HA-RRAA directly moves downward to the next lower rate. We set $P_{Thresh} = 90\%$ for our implementation.

Cost-effective adaptive RTS filter: RRAA maintains a RTS window (RTSwnd), in which all frames are sent with RTS on. Initially RTSwnd is set to 0 and then it is updated as follows. When the last frame is lost without RTS, RTSwnd increments by one because the cause of the loss was probably collisions. However, when the last

Procedure 2 $HA - RRAA$: Input (ACK Frame),Output (R)

```
R=highest_rate;
timer=ewnd(R); fastimer=min{ewnd(R),10};
while true do
    rcv_tx_status(last_frame);
    P = update_loss_ratio();
    if timer==0 || (fastimer≤0 && P ≥ P_Thresh) then
        if P > P_MTL || P ≥ P_Thresh then
            if R!=R_T then
                reset(exp, T_R);
            end if
            T_R =update_twnd(P,exp);
            R_T = R; exp++;
            R = next_lower_rate(R);
        else
            if R==R_T then
                reset(exp, T_R);
            end if
            if P < P_ORI and T_R==0 then
                R = next_high_rate(R);
            end if
        end if
        timer = ewnd(R); fastimer=min{ewnd(R),10};
    end if
    send(next_frame, R);
    timer−−; fastimer−−; T_R−−;
end while
```

frame transmission was lost with RTS, or succeeded without RTS, RTSwnd is halved because the last frame clearly did not experience collisions.

HA-RRAA further improves RRAA's adaptive RTS (A-RTS) mechanism to address hidden terminals at a lower cost. Although A-RTS seeks to mitigate signaling overhead by selectively turning on RTS, there can be still significant overhead in the cases where frame's transmission time is small comparing to RTS/CTS transmission overhead. HA-RRAA uses a cost-effective adaptive RTS scheme, which follows the general paradigm of A-RTS, but without blindly turning on RTS, to avoid significant overhead especially observed at high rate options. Instead, it turns on RTS only when the overhead is outweighed by the potential savings. HA-RRAA first estimates the RTS/CTS overhead (T_{RCTS}), which is the channel time used for transmitting

Procedure 3 Cost-Effective Adaptive RTS

```
RTSWnd=0;
RTScounter=0;
while true do
    rcv_tx_status(last_frame);
    if !RTSOn and !Success then
        RTSWnd++;
        RTScounter=RTSWnd;
    else if RTSOn xor Success then
        RTSWnd=RTSWnd/2;
        RTScounter=RTSWnd;
    end if
    if RTScounter > 0 && T_frame ≥ 1.5 · T_RCTS then
        TurnOnRTS(next_frame);
        RTScounter−−;
    end if
end while
```

RTS/CTS signaling messages. Second it computes the transmission time of the frame as $T_{frame} = \frac{FRAME}{R} + T_{overhead}$ where $FRAME$ is the MAC-layer frame size, R is the transmission rate and $T_{overhead}$ includes the various 802.11 protocols overheads (SIFS, DIFS, ACK). Finally, HA-RRAA will turn RTS on only if the following condition holds: $T_{frame} \geq k \cdot T_{RCTS}$, where k is a benefit/cost ratio set to 1.5 in our prototype. The intuition behind this condition is that, without RTS/CTS, the frame may need at least one retry to get through, when collision occurs. The pseudocode of the cost effective adaptive RTS filter is presented in Procedure 3.

4.4.1.2 Putting everything together

In Figure 4.5 we present the complete architecture of HA-RRAA. Upon the reception of MAC-layer feedback, loss estimation module updates: a) loss ratio estimation for the selection of the next transmission rate, b) history information module to set the adaptive time window and c) mobility fast adaptation module to handle drastic channel changes. It also interacts with the cost effective adaptive RTS filter to update RTS window, as described in Section 4.4.1.1.

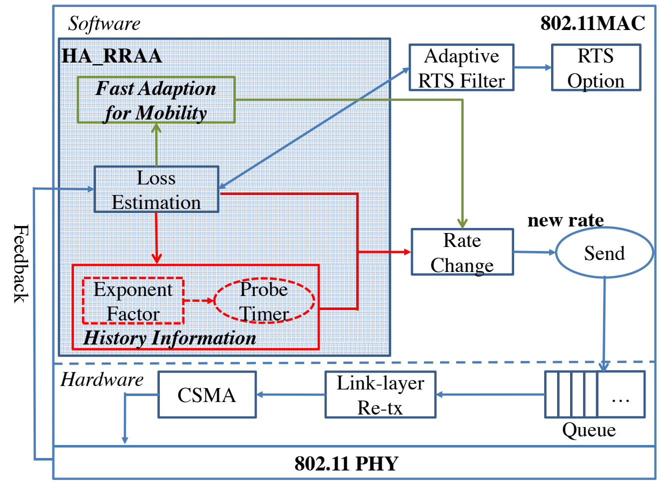


Figure 4.5: HA-RRAA architecture.

4.5 Implementation and Evaluation

We implement HA-RRAA on a programmable AP platform and we compare them with RRAA, RRAA+, SampleRate and ARF in controlled testbeds and field trials. We next present our implementation and evaluation efforts.

4.5.1 Implementation

There are two non-trivial challenges that our implementation must address. First, our AP platform avoids floating point operations, thus the runtime short-term loss ratio and the associated two thresholds are not directly applicable. To address this issue, we count the number of lost frames, rather than calculate the decimal loss ratio. Specifically, we maintain a counter to record the number of lost frames within the current estimation window, while the loss ratio thresholds are translated into the number of frame losses.

Second we need to incorporate Atheros' *software (SW) retries* [32] with the RA algorithms. SW retries are pairs of {rate, number of hardware (HW) retries}. When

	RRAA	RRAA+	SampleRate	ARF
Static UDP	(2.2-41.1)%	(0.4-6.7)%	(1.3-83.9)%	(2.4-39.6)%
Static TCP	(1.7-25.1)%	(0.7-41.6)%	(5.2-55.0)%	(0.1-33.6)%
Mobility	-	4.8%	8.6%	-
Hidden Terminal	up to 8.4%	(4.7-21.7)%	(15.4-28.5)%	50.1% - $\times$ 1145
Field Trial	(1.5-5.8)%	(12.4-24.8)%	(4.9-6.0)%	(3.6-51.9)%

Table 4.4: Performance gains of HA-RRAA over state of the art RA designs.

the rate selected from RA algorithm fails (ACK is not received), a software retry will re-send the data frame in the next lower rate, in an attempt to get the frame through. If the rate selected from RA algorithm (say 54Mbps) fails, our platform will first HW retry the frame two times at 48Mbps, two times at 36Mbps and then four times at 24Mbps, if the previous attempts fail as well. In our implementation we consider that a failure at a lower rate implies a failure at higher rates as well. For example, if the selected rate R (say 54Mbps) fails two times and R_- (say 48Mbps) fails one time, then the total failed frames considered in RRAA's loss ratio will be three. A successful transmission for a rate R, is considered a transmission of zero HW retries.

4.5.2 Evaluation

In this section we compare the different RAs both in controlled static, mobile settings and field trials, using real experiments. The results are averages of multiple back-to-back runs whose standard deviation varies from 0.004 to 1.05 Mbps in controlled settings and from 0.17 to 2.2 Mbps in the field trials (as shown by the error bars in Figures 4.6-4.11). All the algorithms are implemented on the AP side and traffic is downlink (from AP to client). HA-RRAA's goodput gains over the other 802.11a/b/g standard-compliant designs, are summarized in Table 4.4.

To evaluate solutions that use PHY-layer feedback and are not 802.11a/b/g standard-compliant, we compare SoftRate [27] with our proposed HA-RRAA, using

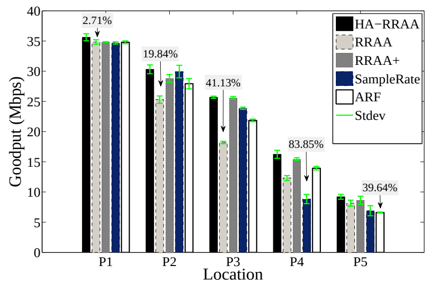


Figure 4.6: Static 802.11a client at UDP setting.

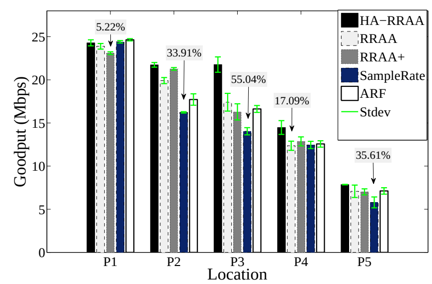


Figure 4.7: Static 802.11a client at high volume (4 flows) TCP setting.

ns-3 simulations.

Static clients We first compare RA designs in five different locations (P1-P5) on a 5GHz interference-free channel. In Figures 4.6, 4.7, 4.8, we present the goodput performance of the five algorithms for UDP, intense TCP (4 flows) and sparse TCP (1 flow) traffic respectively. We observe that HA-RRAA outperforms all the other algorithms in all the locations. For UDP traffic HA-RRAA gives goodput gains up to 41.1% over RRAA, up to 6.7% over RRAA+, up to 83.9% over SampleRate and up to 39.6% over ARF. In static TCP setting, goodput gains are significant as well and can go up to 55%.

In static settings, HA-RRAA's goodput gains over other solutions, can be mainly attributed to adaptive time window mechanism. Specifically, HA-RRAA gives significantly lower average losses over the other RAs, in the most of our static UDP and TCP settings, by avoiding transmission at lossy rates. Compared with the history-oblivious designs, HA-RRAA presents up to 29.7% lower average loss than RRAA and up to 22.4% lower average loss than ARF. Although SampleRate considers past performance before sampling higher rates as we discuss in Section 4.3, it yields higher up to 9.1% average losses than HA-RRAA. For the location P3 of our case study presented in Section 4.3.1, HA-RRAA gives significant better performance than RRAA,

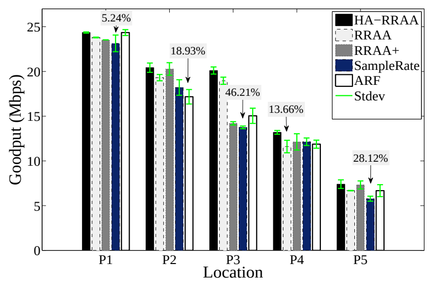


Figure 4.8: Static 802.11a client at low volume (1 flow) TCP setting.

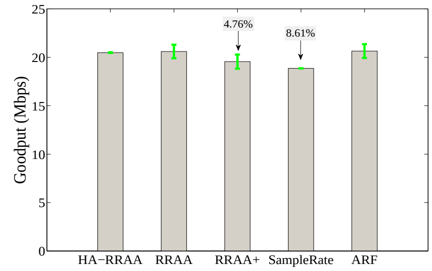


Figure 4.9: Mobile 802.11a client at UDP setting.

SampleRate and ARF (Figure 4.6) by transmitting only 2.2% of the total frames at the lossy 48Mbps rate. Although RRAA+'s performance comes close to HA-RRAA in many of the locations, our experiments show that it may select rates lower than the optimal, in various traffic and location settings for two reasons. First, RRAA+ tends to be conservative by halving $p[R]$ upon failure. Second, it suffers from stale probability $p[R]$ statistics. The negative effects of these two observations are most evident in the multiple- and single-flow TCP experiments at location P3, where RRAA+ transmits on average 40%, 58% of the frames at rates lower than 36Mbps, while the average best goodput rate is 36Mbps. Note that TCP traffic's bursty or sparse nature may affect the channel estimation in $ewnd$ and may result in different rate distributions compared with UDP.

In some scenarios, as in our case study setting presented in Section 4.3.1, ARF performs better than RRAA (20% goodput gains). Although both algorithms are history-oblivious, ARF is proven more conservative in moving to higher lossy rates and it may also move faster to lower rates upon severe frame loss.

Mobile clients In our mobility setting, client is moving between locations P1 and P5 at approximately constant pedestrian speed of 1m/s. The channel selected is interference-free and traffic is UDP. From Figure 4.9, we observe that our adaptive time window mechanism does not have any negative effect when client is moving closer to AP. So, HA-RRAA performs similar to RRAA and ARF. On the other hand, RRAA+ does not have any efficient mechanism to reset its stale statistics, which makes it less responsive. As a result HA-RRAA outperforms RRAA+ by 4.7%. An ideal setting to evaluate our proposed fast adaptation mechanism, is a vehicular network scenario when client is moving very fast away from the AP. We leave this as a future work.

Hidden terminal In this section we evaluate HA-RRAA in a controlled hidden terminal setting. In our interference scenario, an 802.11a client broadcasting packets at P6, acts as a hidden terminal to the 802.11a client at P2, which is the receiver of UDP traffic from the AP. To change the intensity of the hidden terminal setting, we vary the data source rate of the hidden station. In Figure 4.10 we present the performance of the implemented algorithms in a modest and an intense hidden terminal scenario. In the modest setting (1Mbps data source rate), HA-RRAA is the clear winner over the other designs, with goodput gains up to 50.1%. In the low interference level scenario, the goodput gains of 8.4% of HA-RRAA over RRAA can be attributed to the cost effective A-RTS filter of HA-RRAA compared with the simple A-RTS filter of RRAA. In the very intense hidden terminal scenario, HA-RRAA is slightly worse than RRAA (3.2%) because its adaptive time window can be increased upon collision losses, making HA-RRAA to transmit at lower rates compared with the optimal transmission rate. Overall because Adaptive RTS filter, RRAA, HA-RRAA and RRAA+ give significantly better performance than ARF and SampleRate.

Field trials We also conduct a series of uncontrolled field trials to understand how well the RAs perform under realistic scenarios, in which various sources of dynamics co-exist in a complex manner. Our field trial uses two static clients at locations P2 and P4 and a third client initially placed at P3, which we periodically move between locations P1 and P5. We run four sets of experiments and each lasted at least

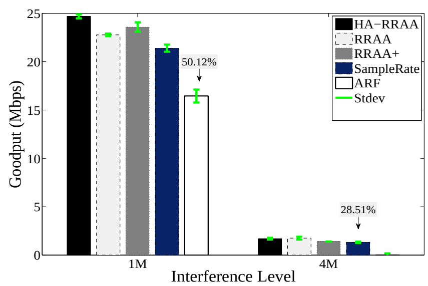


Figure 4.10: Hidden terminal setting.

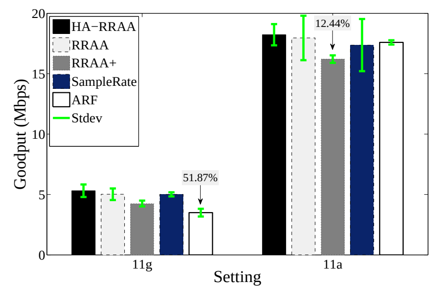


Figure 4.11: Field trials for 2.4GHz and 5GHz bands.

half an hour both at 2.4GHz (channel 1) and 5GHz bands (channel 36). Traffic is single flow TCP. At 2.4GHz band the channel was heavily loaded as we sniffed 17 APs from channel 1 to 11. Under this high interference environment HA-RRAA gives up to 5.8%, 24.8%, 6%, 51.9%, goodput gains over RRAA, RRAA+, SampleRate and ARF respectively, as presented in Figure 4.11. At the less congested 5GHz band the performance of all algorithms is significantly better. HA-RRAA gives up to 12.4% goodput gains over the other algorithms as well.

Simulations We next compare HA-RRAA with SoftRate [27], using ns-3 simulations. SoftRate uses confidence information calculated by the PHY-layer (SoftPHY hints), which are exported to higher layers to estimate the channel BER. Receiver communicates this BER estimate to the sender on a per-packet basis, which finally picks the best goodput rate. Authors in [27] use software radio traces, which specify the SoftPHY hints that are required for BER computation. As software radio traces are not available, we calculate BER based on SINR-BER curves [73]. Given that the simulation propagation environment is fixed and there are not any hardware calibration or interference issues that can affect SINR, we argue that our simulated SoftRate can perform similar to the one proposed in [27].

In our evaluation scenario, an 802.11b AP sends TCP traffic to an 802.11b client.

Setting	HA-RRAA Goodput (Mbps)	SoftRate Goodput(Mbps)
Static	5.56	5.6
Low Mobilty	2.21	3
High Mobilty	2.05	2.91

Table 4.5: HA-RRAA vs. SoftRate under static, mobile settings.

We compare HA-RRAA with SoftRate under static and mobility scenarios. In the mobility case, the AP remains static while the client is moving with 3mph, 80mph for the low and high mobility setting respectively. The results are presented in Table 4.5. Interestingly, in the static case where channel is stable, both algorithms give similar goodputs and rate distributions (>80% of total frames at 11Mbps). In our mobility setting, SoftRate can adapt the bit rate on a per-frame basis and yields up to 42% goodput gains over HA-RRAA.

4.6 Summary

This chapter provides our first step towards gigabit wireless, by studying legacy 802.11a/b/g rate adaptation, using 802.11 standard-compliant programmable testbeds. We first critique three popular design guidelines of existing algorithms, while we also experimentally study the short-term dynamics of the 802.11 wireless channel. The key insight learned is that, a RA algorithm has to capture short-term channel's performance, to infer different loss behaviors and to take adaptive reactions accordingly. To this end, we design HA-RRAA, which applies adaptive time windows to capture the short-term channel dynamics. It also differentiates fading from interference packet loss, by applying a low overhead RTS filter.

HA-RRAA is a practical design in three key ways. First, it is 802.11 standard-compliant. Second, it does not require receiver-side feedback, which is not supported by 802.11a/b/g standard. Finally, it leverages MAC-layer frame loss feedback, which is available in any commodity 802.11 driver. Our real experiments show that, HA-RRAA consistently outperforms popular 802.11a/b/g standard-compliant solutions with goodput gains up to 51.9% in field trials.

CHAPTER 5

From Legacy 802.11a/b/g to MIMO 802.11n Rate Adaptation

IEEE 802.11n standard adopts Multiple-Input Multiple-Output (MIMO) technology to further enhance its PHY-layer capability. Using multiple transmit and receive antennas, it supports both *Spatial Diversity* oriented single-stream (SS) and *Spatial Multiplexing* driven, multiple-stream (double-stream (DS) in our platform) operation modes. Together with channel bonding of two adjacent channels, 802.11n offers a much wider range of transmission rate options up to 600Mbps. The wider span and larger number of rate options, make MIMO 802.11n rate adaptation (RA) more challenging than legacy 802.11a/b/g RA. MIMO rate adaptation has to adjust not only the Modulation-Coding Scheme (MCS), but also the MIMO mode at runtime based on the channel quality.

In this chapter, we identify issues and propose solutions for MIMO-based RA in 802.11n systems. Our work started with a simple question. Can we simply apply RA algorithms, which have been shown to work well for the legacy 802.11a/b/g networks, to the MIMO setting? Our experiments on standard-compliant 802.11n AP platform show that, both popular legacy RAs (RRAA [34], and SampleRate [23]) and MIMO RAs (Atheros MIMO RA [32], used in 802.11n Atheros chipsets) have significant limitations. To our surprise, all three algorithms offer 28% to 44% lower goodput, defined as effective throughput by excluding protocol overhead, than the best fixed-rate scheme. *The fundamental problem is that all such algorithms do not properly consider the inherent characteristics of SS and DS MIMO modes, which exhibit very different loss patterns.*

Our extensive experiments both in a campus environment and in RF chamber uncover a non-negligible, non-monotonic relation between loss and rate in 802.11n MIMO scenarios, when considering all rate options and ignoring operation modes. That is, although rate increases, loss does not monotonically grow with rates in different modes due to inherent MIMO characteristics [8–10]. This results in existing RAs to transmit at rates lower than the best goodput rate. However, within each SS/DS mode, the monotonic behavior between loss and rate still largely holds.

In this chapter, we first design MiRA, a new practical RA algorithm for 802.11n networks. MiRA is 802.11n standard-compliant and can be implemented using commodity 802.11n hardware. It does not require any channel state feedback from the receiver, and it does not make any assumption about the MIMO radio implementation. MiRA addresses loss non-monotonicity by applying a novel zigzag RA scheme, which opportunistically zigzags between intra- and inter-mode RA operations. It starts by sequentially probing rates of the current MIMO mode exploiting loss monotonicity across individual modes. When it cannot further improve goodput in its current mode, MiRA performs inter-mode RA by exploring the other DS/SS mode. It uses a new adaptive probe interval mechanism to limit probing at low goodput rates, while it also exploits 802.11n frame aggregation feature and BlockAck to differentiate collision from channel losses. In addition to MiRA, we design and evaluate several alternatives to MIMO RA. Window-based rate adaptation (WRA) runs an independent RA in each MIMO mode in parallel, to address loss non-monotonicity, while it maintains and adjusts a rate selection window to identify the best goodput rate with limited probing cost. MIMO-SampleRate uses SNR measurements to differentiate between SS/DS modes.

Our experiments in indoor controlled static/low-mobility settings and field trials confirm the performance gains of MIMO-mode aware RAs, under various MIMO configurations. Specifically, MiRA consistently outperforms RRAA, SampleRate and Atheros MIMO RA with goodput gains up to 124.8%, 182.2% for static and mobile clients, respectively. In the field trials, MiRA and WRA achieve goodput gains up to 73.5% over the other legacy and MIMO RA algorithms.

The rest of this chapter is organized as follows. Section 5.1 introduces our experimental setting. Section 5.2 studies a simple case of applying existing RA algorithms in the 802.11n setting, and Section 5.3 reports the findings on characteristics of diversity and spatial multiplexing modes. Section 5.4 presents the design of MiRA, and Section 5.5 discusses several MIMO RA alternative solutions. Section 5.6 describes our implementation and evaluation, while Section 5.7 reviews the related work. Section 5.8 concludes the chapter.

5.1 Experimental Setting

We conduct all the experiments on a programmable AP platform, which uses Atheros AR5416 2.4/5 GHz MAC/BB MIMO chipset. Our AP supports single-stream (SS), double-stream (DS) modes and rates up to 300Mbps. Our testbed supports frame aggregation and 20/40MHz channels, as well. For our study, we implement both legacy [34], [23] and MIMO RAs [32] on the AP side. We provide more information about our experimental platform in Chapter 3. We repeat our experiments with different 802.11n clients; Buffalo WLI-CB-AG300NH 802.11a/b/g/n wireless adapter is based on Marvell 802.11n chipset, Linksys WPC600N 802.11a/b/g/n and Airport Extreme wireless adapters use Broadcom chipset. The results presented in this chapter

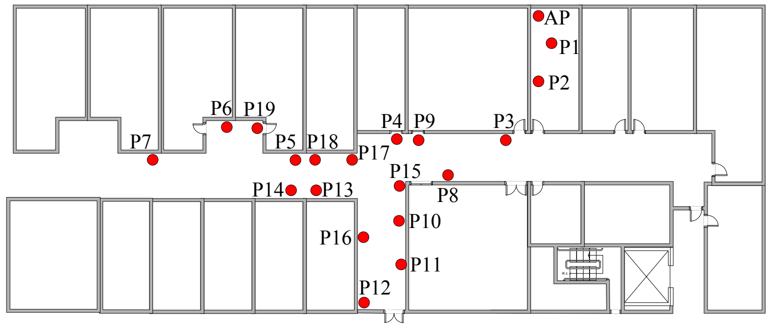


Figure 5.1: Experimental floorplan.

are from Airport Extreme adapter, which supports up to 270Mbps rates.

We conduct our experiments in both a campus setting and in RF chamber. Figure 5.1 shows the floorplan of the campus building we run the experiments. Spots P1 to P19 represent different locations where the clients are placed. In all the experiments unless it is explicitly mentioned, we initiate downlink back-to-back UDP traffic (from the AP to client) with 1.5KB MPDUs. Channel bandwidth is set to 40MHz and aggregation is enabled. We also configure the AP at the interference-free (as verified by the sniffer) 5GHz band, on channel 36.

5.2 A Case Study

We started our work by examining how well the existing RA algorithms work in the 802.11n MIMO setting. The goal is to understand which factors in these RA schemes lead to their performance gain or loss and which MIMO characteristic is the root cause. To illustrate our findings, we first present a case study, while we discuss more comprehensive results in Section 5.3. In our case study setting, the AP transmits back-to-back A-MPDUs at a static client located at P4. We studied three representative RA algorithms. RRAA [34] and SampleRate [23] have been shown to work well in the legacy 802.11a/b/g scenarios. Atheros MIMO RA is a new algorithm used in 802.11n Atheros

Rates (Mbps)	Atheros RA	RRAA	SampleRate	Fixed Rate Goodput (Mbps)	Fixed Rate SFER
MCS2 (40.5SS)				36.23	0.12%
MCS3 (54SS)	49%			49.08	0.20%
MCS9 (54DS)				48.87	0.12%
MCS4 (81SS)				72.94	0.07%
MCS10 (81DS)				72.64	0.06%
MCS5 (108SS)	51%			96.46	0.15%
MCS11 (108DS)		47%	89%	96.31	0.16%
MCS6 (121.5SS)		53%	4%	74.01	17.92%
MCS7 (135SS)			7%	36.56	54.61%
MCS12 (162DS)				128.46	4.31%
MCS12 (216DS)				5.71	96.73%
Goodput (Mbps)	71.40	85.36	91.95		
SFER	0.59%	13.24%	7.25%		

Table 5.1: Rate distribution, goodput and SFER of existing RA algorithms at P4.

chipsets. We also conducted fixed-rate experiments at every 802.11n rate option.

Table 5.1 summarizes the results of these experiments. Unfortunately, all three RA algorithms perform worse than the best fixed-rate scheme, with 28% to 44% lower goodput. The goodput at the best fixed rate is $128.5Mbps$, while Atheros RA gives $71.4Mbps$, RRAA offers $85.4Mbps$, and SampleRate gives $91.9Mbps$. These results clearly indicate that the existing RA algorithms cannot be effectively applied in 802.11n networks.

It turns out that, all three RA algorithms were transmitting at rates lower than the best rate choice. Table 5.1 states that the goodput at 162DS is $128.5Mbps$, while the goodput at 108SS, 108DS, 121.5SS and 135SS are only $96.5Mbps$, $96.3Mbps$, $74Mbps$ and $36.6Mbps$, respectively. Obviously, a good RA should transmit most of its frames at 162DS rather than at other rates. However, as illustrated in Table 5.1, the rate distribution of each RA, which provides the percentage of data frames transmitted at a given rate, shows the opposite results. SampleRate transmits 89% of frames at 108DS, RRAA transmits 53% and 47% at 121.5SS and 108SS. The Atheros MIMO

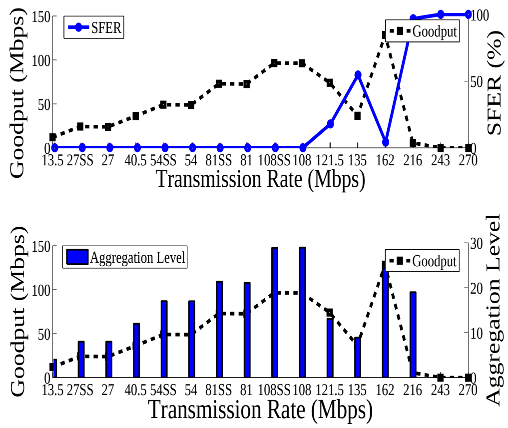


Figure 5.2: SFER non-monotonicity in high cross-mode rates (location P4).

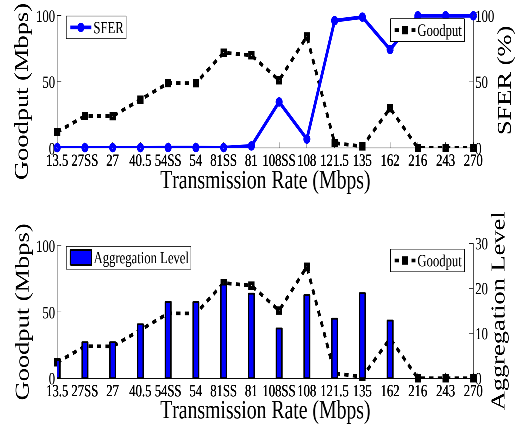


Figure 5.3: SFER non-monotonicity in same cross-mode rates (location P10).

RA is even worse, transmiting 51% at 108SS and 49% at 54SS, and not using 162DS at all.

We next examine what happens at rate 162DS and other rates. Our experiments, plotted in Figure 5.2, reveal that two factors play a critical role: non-negligible, non-monotonic relation between Sub-Frame Error Rate (SFER) and rate, and frame aggregation. Figure 5.2 shows that, SFER does not monotonically increase as the rate grows from 121.5 to 162 Mbps. The frame loss SFER is only 4.3% at 162DS, but is 54.6% at 135SS, 17.9% at 121.5SS and 0.15% at 108SS and 108DS. This finding in 802.11n MIMO settings is clearly different from that in the legacy 802.11a/b/g systems. Aggregation level is another factor that affects goodput. Figure 5.2 states that, the average aggregation level is 27 MPDUs at 162DS but is 15 MPDUs at 121.5SS. This (11.3MPDU) larger aggregation level also leads to goodput improvement as the amortized per-frame overhead is smaller. With both low SFER and high aggregation level, 162DS significantly outperforms other rates.

Once we discovered the two factors of non-monotonic SFER and frame aggregation level, we further look into why existing RA designs have difficulty in identifying and staying at the best rate that offers highest goodput. The RRAA algorithm [34] assumes that SFER monotonically increases with rate. Therefore, RRAA assumes that higher rates would yield higher losses when evaluating the rate 121.5SS. This is true for 135SS but not true for 162DS. As a result, it never probes 162DS that has smaller SFER and highest goodput. Atheros MIMO RA also assumes monotonicity in that all rates above the current rate R have no smaller SFER. When probing, it upper bounds the candidate rates for selection (maxRate) by probing one rate higher than the current best goodput rate R. By analyzing actual packet traces, we observe that probing fails at 135SS and maxRate is set at 121.5SS for most transmissions. Consequently, Atheros MIMO RA transmits almost all of the frames at rates lower than 121.5Mbps. SampleRate [23] randomly samples diverse rates via probing, but suffers from stale statistics on the goodput and SFER at a rate as it updates statistics only by probing these rates. *It consequently transmits at rates below 135Mbps* as shown in Table 5.1. Moreover, the SampleRate MADWiFi implementation bounds sampling to at most 2 rates higher than the current rate. It thus does not update stale statistics for rates greater than 135Mbps and transmits most data at 108DS. Even when we relaxed SampleRate's sampling bound, it may still suffer from stale statistics and probing overhead, as we discuss in Section 5.6.3.

5.3 Studying MIMO Characteristics in 802.11n Systems

The above case study shows that the fundamental reason for RA under-performance is the inherent MIMO characteristics [8–10]. We next repeat our case study scenario, by placing the client in various locations of the floorplan of Figure 5.1 and we present a thorough study on 802.11n characteristics.

Location	$SFER_{121.5SS}$ (%) SNR (dB)	$SFER_{135SS}$ (%) SNR (dB)	$SFER_{162DS}$ (%) SNR (dB)
P3	0.39% 42.97 (dB)	7.99% 40.64 (dB)	0.33% 41.53 (dB)
P8	0.27% 29.69 (dB)	11.90% 30.80 (dB)	0.39% 31.22 (dB)
P4	17.92% 21.67 (dB)	54.61% 22.41 (dB)	4.31% 22.15 (dB)
P10	96.29% 17.38 (dB)	98.99% 16.75 (dB)	74.50% 17.79 (dB)

Table 5.2: SFER non-monotonicity w.r.t. rate in cross modes.

5.3.1 SFER non-monotonicity in SS and DS

Our experimental results show that, different from the legacy 802.11a/b/g systems, *there exhibits a non-negligible, non-monotonic relation between the rate option and SFER in 802.11n MIMO settings* when considering all rates in both SS and DS modes. SFER does not monotonically increase when the transmission rate increases. The non-monotonicity appears more distinctive under two scenarios: (i) in the high-rate region (e.g., $\geq$ 121.5SS) as shown in Figure 5.2, and (ii) at same rates in different modes (e.g., 108SS and 108DS) as shown in Figure 5.3. Representative examples of these two cases are illustrated in Tables 5.2 and 5.3. Table 5.2 shows that the non-monotonicity in SFER is particularly severe between three adjacent cross-mode rates (i.e., 121.5SS, 135SS, 162DS). In four locations P3, P4, P8, and P10 (we show a subset of results due to space constraints), SFER increases as the rate increases from 121.5SS to 135SS, but drops significantly as the rate further moves to 162DS. SFER drops 50.3% at P4 when switching from 135SS to 162DS. Similar results are also observed in the RF chamber, where 121.5SS and 135SS have up to 6.4% and 8.1% higher SFER than 162DS, respectively. Non-monotonicity also exhibits in the same-rate pairs. The

Location	P10	P13	P14	P11	P7
	SFER(%)	SFER(%)	SFER(%)	SFER(%)	SFER(%)
	SNR(dB)	SNR(dB)	SNR(dB)	SNR(dB)	SNR(dB)
MCS1 (27SS)	0.19%	0.30%	0.61%	4.95%	10.95%
	17.10(dB)	14.93(dB)	12.96(dB)	12.34(dB)	7.03(dB)
MCS8 (27DS)	0.23%	0.31%	0.52%	17.79%	25.143%
	13.40(dB)	14.09(dB)	12.51(dB)	14.09(dB)	7.10(dB)
MCS3 (54SS)	0.25%	1.41%	1.19%	7.44%	100%
	16.1(dB)	12.34(dB)	12.87(dB)	10.60(dB)	-
MCS9 (54DS)	0.25%	0.72%	9.23%	16.73%	100%
	14.82(dB)	12.16(dB)	12.19(dB)	12.16(dB)	-
MCS4 (81SS)	0.19%	10.14%	25.60%	27.88%	100%
	17.05(dB)	11.95(dB)	11.58(dB)	11.95(dB)	-
MCS10 (81DS)	1.54%	10.03%	37.04%	37.15%	100%
	16.59(dB)	12.17(dB)	13.29(dB)	11.79(dB)	-
MCS5 (108SS)	34.83%	99.09%	97.69%	97.85%	100%
	16.13(dB)	11.64 (dB)	13.15(dB)	11.64(dB)	-
MCS11 (108DS)	6.68%	82.88%	93.60%	98.24%	100%
	15.02 (dB)	11.71(dB)	13.47(dB)	11.71(dB)	-

Table 5.3: SFER w.r.t. different cross-mode rate pairs.

SFER difference can be as large as 28.2% (location P10), as shown in Table 5.3. Note that this non-monotonic behavior is not caused by SNR variations. Table 5.2 and 5.3 show that the SNR values only exhibit minor differences at a given location.

The root cause for the behavior is that SS and DS are based on different communication approaches [8]. Thus it is unlikely that they will exhibit similar loss-rate relations by simply merging them together via the numerical value of the transmission rate. In contrast, our extensive experiments reveal that the monotonicity between SFER and rate still largely holds in individual SS and DS modes. Figures 5.4, 5.5 and 5.6, 5.7 show that SFER monotonicity is restored for the individual DS and SS modes, for the locations P4 and P10, respectively. Although Lampe et al. [11] theoretically

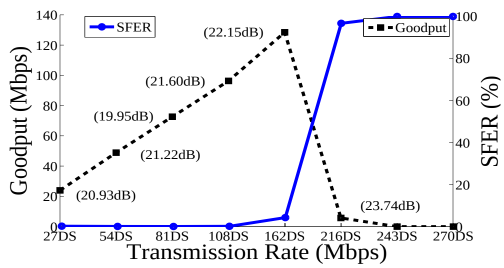


Figure 5.4: SFER monotonicity in DS mode (location P4).

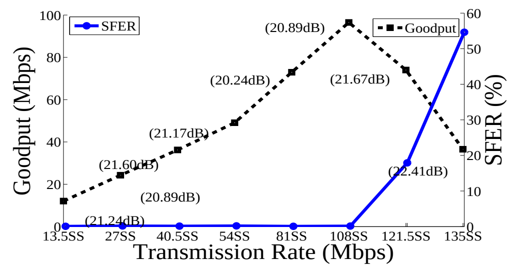


Figure 5.5: SFER monotonicity in SS mode (location P4).

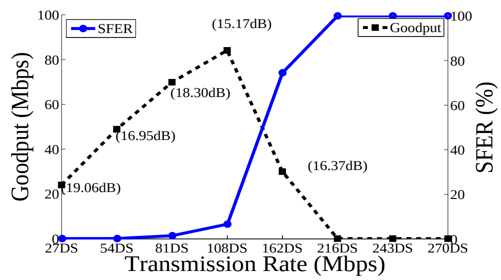


Figure 5.6: SFER monotonicity in DS mode (location P10).

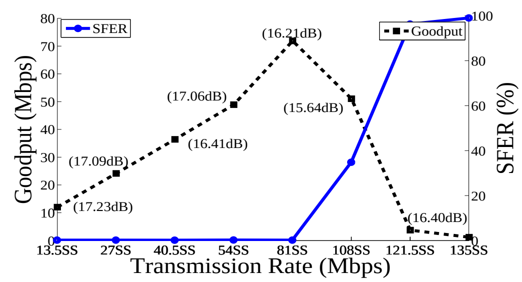


Figure 5.7: SFER monotonicity in SS mode (location P10).

showed that loss monotonicity may not hold for the single-mode SISO case, our experimental results advocate that loss monotonicity can safely be assumed in a practical MIMO setting.

The non-monotonicity between SFER and rate has profound implications for 802.11n rate adaptation design. Many existing RA algorithms implicitly assume the monotonicity between SFER and rate. For example, one popular mechanism is to sequentially probe upward/downward the rates, and adjust the rate based on the probing result. Its underlying premise is that, the packet error rate goes higher as the rate increases, and there is no need to probe/use higher rate if the current one performs poorly. While this mechanism works reasonably well in the legacy system, it does not work in the dual-mode MIMO settings. An efficient rate adaptation design should be able to handle this non-monotonic SFER behavior.

5.3.2 SS/DS mode selection

The above findings indicate that MIMO RA design should differentiate the two MIMO modes. The next issue is to identify possible conditions under which SS underperforms or outperforms DS. Several theoretical studies [8–10] have shed lights on it via examining the tradeoff between Diversity and Spatial Multiplexing gains. Our goal is to find the answer via experiments in the 802.11n setting.

The comparison between SS and DS mode summarized in Tables 5.2 and 5.3, shows that SNR can serve as a coarse-grained indicator to decide which mode is more likely to be the winner. In low-SNR regions (say, $< 13dB$ in our setting), SS is more likely to outperform DS. In these low-SNR, far-away locations, SS is the winner over DS with 5% or more goodput gain in 85.7% of locations tested, while its goodput and loss are similar to DS in the remaining locations. The winning SS rates span the broad set of 13.5SS, 27SS, 40.5SS, 54SS, and 81SS. The average goodput gain is 15.6% but varies from 6% to 40.2%. In high-SNR regions (say, $> 16dB$ in our setting), DS is more likely to outperform SS. In fact, in almost all cases, DS is the winner over SS, with the average goodput gain being 33.2%. The actual goodput gain varies from 17% to 60.4%. The winning DS rates span the broad set of 108DS, 162DS, 216DS, 243DS, 270DS, and 300DS.

One should be cautious in applying the above findings, because they simply show the general trend rather than claim which specific mode is the winner in all cases. In fact, there is always the gray area where either can be the likely winner. Moreover, there are several non-trivial challenges in finding a good SNR threshold to be used to select between SS and DS modes. We will elaborate on these issues in Section 5.6.3.

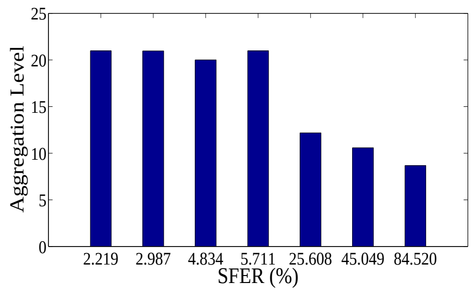


Figure 5.8: SFER vs. aggregation level.

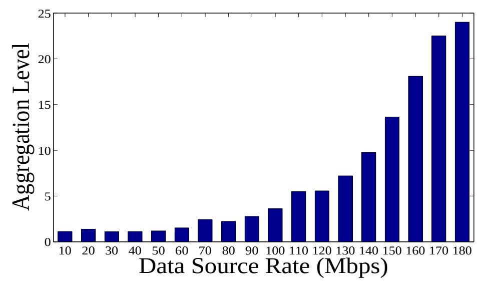


Figure 5.9: Traffic source vs. aggregation level.

5.3.3 On frame aggregation

Our study on frame aggregation reveals interesting findings due to its interplay with rate adaptation. Our experiments uncover that not only traffic source but also SFER have a significant impact on aggregation level. When SFER is negligible, traffic source determines aggregation level by affecting the number of MPDUs available in software queue. Figure 5.9 presents aggregation level evolution with traffic source in a scenario where rate is fixed to 243Mbps and loss is smaller than 2%. However higher SFER can have both positive and negative impact on frame aggregation. Higher loss may raise aggregation level, by increasing lost MPDUs accumulated in software queue for retransmission. To verify this hypothesis, we fix the rate to 135SS and we use a smaller data source (60Mbps). We next vary the SFER by switching to different locations. When the loss is small (5.3%), medium (29.2%), excessive (99.5%), the average aggregation level is 3.0, 10.5, 18.9 MPDUs, respectively. Loss may have a negative impact on aggregation as well. Figure 5.8 plots the evolution of aggregation level with SFER in a setting, where rate was fixed to 81SS and the data source was aggressive enough to ensure full software queue. We see that high SFER dropped the average aggregation level from 21 MPDUs to 8.7 MPDUs in the experiment. It turns out that, the limiting factor here is the Block ACK Window (BAW) specified by the 802.11n stan-

Rates (Mbps)	RRAA ($\%$)	RRAA-Limited ($\%$)	Aggr. Bound (#MPDUs)
MCS0 (13.5SS)	2	1	4
MCS1 (27SS)	3	0.5	8
MCS8 (27DS)	1	0.5	8
MCS2 (40.5SS)	8	14	13
MCS3 (54SS)	23	42	17
MCS9 (54DS)	29	26	17
MCS4 (81SS)	11	1	26
MCS10 (81DS)	23	15	26
MCS5 (108SS)			35
MCS11 (108DS)			35
Goodput (Mbps)	24.22	35.60	
SFER ($\%$)	46.61	24.83	
Avg. Aggr. level	19.36	11.81	

Table 5.4: Rate distribution and performance for RRAA and RRAA-Limited at P6.

dard. BAW moves forward as long as MPDUs with sequence numbers inside the BAW are acknowledged, similar to the sliding window scheme in TCP. However, if the first MPDU with sequence number Seq within BAW is lost and to be retransmitted, then all followup A-MPDUs can only aggregate frames within the window of BAW, i.e., with sequence numbers less than $Seq + 64$, where 64 is maximum number of frames aggregated in a single frame in 802.11n. If there are four followup aggregate frames, the aggregation level is only 16 MPDUs on average. Therefore, the position Seq of the lost MPDU affects the aggregation level for the followup frames.

Since higher aggregation can lead to higher goodput due to amortized overhead, the RA designs may naturally try to maximize the aggregation level. However, our experiments show that this is not always the best strategy. High aggregation level makes RA less responsive to fast channel dynamics thus reducing the effective goodput. Table 5.4 presents the performance of RRAA and RRAA-Limited at location P6. RRAA-Limited upper-bounds aggregation level in proportion to the rate (as shown in the last column of Table 5.4 for 1.5MB MPDUs). Thus the maximum A-MPDU size divided by the transmission rate (air time) should be equal at each rate. This aggregation algorithm is used from Atheros driver. From Table 5.4, the average aggregation level for RRAA is 7.6 MPDUs larger than RRAA-Limited. However, RRAA-Limited offers 46.9% goodput gain over RRAA, even with smaller aggregation. Our traces show that, RRAA experienced 21.8% higher SFER than RRAA-Limited. Table 5.4 indicates that RRAA is less responsive to fast channel dynamics and transmits 34% of frames at 81 Mbps, which yields 86.3% SFER. Higher aggregation at this lossy rate hurts goodput. So for our experiments, we upper-bound aggregation in proportion to rate (similar to Atheros). We leave the study of aggregation impact on RA's responsiveness to channel dynamics, as a future work.

5.4 Design

MiRA seeks to identify and set its transmission rate to the best rate option, which offers the highest goodput under dynamic channel conditions. Unlike other RA algorithms, MiRA uses a novel zigzag scheme, which opportunistically switches between intra- and inter-mode RA operations, to address the 802.11n MIMO characteristics. When performing probe and rate change, it first stays in its current SS/DS operation mode and adapts the rate upward/downward. This intra-mode RA effectively exploits the feature of monotonicity between loss SFER and rate in the same mode. When it cannot improve further in the same mode, MiRA performs inter-mode RA by switching its RA operation to the other operation mode. It further uses two-level prioritized probing to reduce the penalty of excessive probing at bad rates. Finally, MiRA detects collisions from channel errors based on the loss pattern learned from the 802.11n frame aggregation and BlockAck, without using the RTS/CTS mechanism. We now elaborate on each operation in detail.

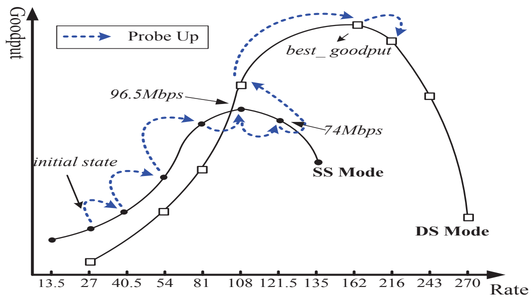


Figure 5.10: Example for Zigzag RA: Rate upward trajectory upon better channel.

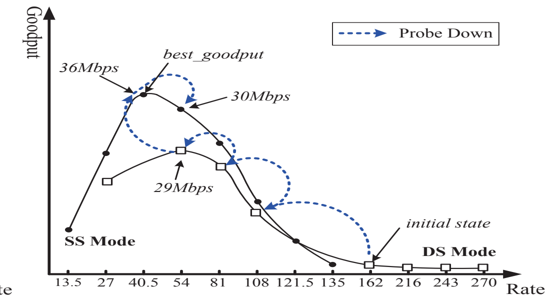


Figure 5.11: Example for Zigzag RA: Rate downward trajectory upon worse channel.

5.4.1 Zigzag RA: Intra- and inter-mode RA

MiRA zigzags between SS and DS modes. It favors intra-mode over inter-mode operations when there is a need to probe and change the rate (e.g., sudden change in goodput or probe timer expires). It probes upward/downward within the current mode until it sees no further chance for goodput improvement. After intra-mode operations are completed, it then performs inter-mode RA by probing and changing rate to the other mode. As a result, when channel dynamics call for rate adjustment, MiRA moves upward/downward in one mode, switches to the other mode and moves upward/downward until the goodput limit within the mode. Then it may switch its mode back, and continues the process as time goes. In both intra-mode and inter-mode operations, MiRA uses probing-based estimation to identify the best goodput and adjust the current rate accordingly. Zigzag RA is illustrated in Procedure 4, while downward, upward and cross-mode operations are presented by the examples of Figures 5.10 and 5.11.

Suppose the starting rate is 27SS at time t_0, again at location P4. Upon detecting a better rate, MiRA moves upward in the SS mode. It continues to probe upward as long as the estimated goodput keeps on increasing, thus going through the probing

Procedure 4 $ZigZagRA$: Input (BlockAck), Output (r)

```
update-stats( BlockAck, r)
collision-detection-and-reaction( BlockAck, r)

//zigzag RA: intra- and inter- probing
//isProbe: a variable indicating whether the last frame is a probe
//probeSeq: a list of rates already probed
if isProbe = true then
    update-priority-probing-timer( BlockAck, r)
    if intra-mode-RA-finished(probeSeq) = false then
        (r, isProbe, probeSeq) ← next-intra-rate(r, probeSeq)
    else if inter-mode-RA-finished(probeSeq) = false then
        (r, isProbe, probeSeq) ← next-inter-rate(r, probeSeq)
    else
        //finish probing, select the best rate among the probes
        (r, isProbe, probeSeq) ← best-rate(r, probeSeq)
    end if
    return r
end if

if probe-timer-expired() = true then
    //adaptive probing timer expires
    (r, isProbe, probeSeq) ← timer-expired-rate()
else if G_r(t) ≤ \overline{G_r}(t) − 2 · σ_r(t) then
    //channel becomes good
    (r, isProbe, probeSeq) ← next-higher-intra-rate(r)
else if G_r(t) ≥ \overline{G_r}(t) + 2 · σ_r(t) then
    //channel becomes bad
    (r, isProbe, probeSeq) ← next-lower-intra-rate(r)
else
    //remain in current rate
    isProbe ← false
    probeSeq ← r
end if
return r
```

sequence at rates of 40.5SS, 54SS, 81SS, 108SS. When it further probes 121.5SS that gives the goodput $74Mbps$, it does not see a higher or equal goodput than 108SS (offering $96.5Mbps$ in Table 5.1). MiRA thus completes the intra-mode RA operation within SS mode. Subsequently, MiRA zigzags to the DS mode by first probing at 108DS, which is the lowest DS rate whose loss-free goodput is higher than $96.5Mbps$. Within the DS mode, It further probes upward to 162DS and 216DS. It finally sets the transmission rate at 162DS since 216DS delivers lower goodput than 162DS, thus completing the upward operations.

When the channel condition worsens at time t_1 (say, the best rate for goodput now becomes 40.5SS). MiRA detects reduced goodput and high loss SFER at its current rate 162DS. It thus probes downward along its current mode DS via the sequence of 108DS, 81DS, and 54DS. Based on the goodput estimate (say, 30Mbps) at 54DS, MiRA does not further probe downward at 27DS since the loss-free goodput at 27DS is lower than the current estimated goodput. MiRA then zigzags to the SS mode after identifying the best goodput rate in the DS mode is 54DS. Upon inter-mode probing, MiRA first probes 40.5SS, since it is the lowest SS rate whose loss-free goodput is higher than the estimated goodput of the best rate 54DS. The goodput estimate at 40.5SS turns out to be the highest $36Mbps$ so far. In SS mode, MiRA further probes upward at 54SS, which only offers goodput estimate $29Mbps$. MiRA thus zigzags through DS and SS modes, and settles down at the best rate 40.5SS.

The zigzag RA scheme in MiRA needs to address two issues: (1) How to decide which rates, in the same mode or across the mode, to probe? (2) How to estimate the goodput based on the probing results while taking into account the effect of aggregation? We next elaborate on both issues.

5.4.1.1 Prioritized probing

Different from existing RA solutions, MiRA devises a novel, prioritized probing scheme to address MIMO related cross-mode characteristics. It also applies adaptive probing to dynamically adjust the probing interval based on the measured SFER and recent probing history, in order to reduce excessive probing to bad rates. MiRA addresses four issues in its probing scheme: (1) When to initiate probing? (2) What rates to probe? (3) How to probe the candidate rates in both modes? and (4) How to avoid excessive overhead?

Probing triggers MiRA triggers probing and subsequent goodput estimation using both event-driven and time-driven mechanisms. It starts probing whenever it observes significant change in the measured goodput at the current rate. Specifically, it probes downward (to a lower rate) when $G_r(t) \leq \overline{G_r}(t) - 2 \cdot \sigma_r(t)$, where $G_r(t)$ is the measured goodput for rate r at time t, $\overline{G_r}(t)$ is the moving average of the goodput, and $\sigma_r(t)$ is the moving average of the standard deviation of the goodput. Similarly, it probes upward (to a higher rate) when $G_r(t) \geq \overline{G_r}(t)+2\cdot\sigma_r(t)$. Alternatively, when the probing timer for a given rate option expires, MiRA initiates probing at that given rate. In essence, MiRA uses time-driven probing to update stale information on goodput statistics, and event-driven probing scheme to quickly track sudden channel variations. To remain adaptive, MiRA uses a single A-MPDU to probe the selected rate. Given that there are enough frames in the software queue, an A-MPDU can carry up to 64 MPDUs, which are sufficient for the probe to collect accurate loss statistics [34].

Candidate rates for probing MiRA opportunistically selects the candidate set of rates to probe at a given time. When probing upward, it first starts from the immediate, higher rate option within the same mode. Then it sequentially goes to each higher rate option, exploiting the fact that SFER in monotonic within one mode. The intra-mode probing stops at the highest rate option if its next higher rate has a goodput estimate smaller than the highest goodput estimate obtained so far. It then initiates inter-mode probing, starting from the lowest rate, which loss-free goodput is higher than the highest goodput estimate so far. This zigzag operation allows MiRA to handle SFER non-monotonicity in cross modes. In the example of Figure 5.10, the candidate rate set is {40.5SS, 54SS, 81SS, 108SS, 121.5SS, 108DS, 162DS, 216DS} when the upward probing starts from 27SS. Note that in inter-mode probing, the goodput estimate at 108SS is about $96Mbps$, higher than the loss-free goodput at 81DS. Therefore, the inter-mode probing in DS mode starts from 108DS. In downward direction, probing starts from the immediate lower rate within the same mode. It sequentially goes to each lower rate until its highest goodput estimate so far is larger than the next lower rate. This implies the best goodput estimate so far is larger than the loss-free goodput that the lower rate may offer. In the example of Figure 5.11, the candidate rate set is $\{$108DS, 81DS, 54DS, 40.5SS, 54SS$\}$ when the downward probing starts from 162DS. Note that the goodput estimate at 54DS is about $30Mbps$, so MiRA does not probe 27SS whose loss-free goodput will be lower than the goodput estimate at 54DS. Therefore, MiRA initiates inter-mode probing. To this end, 40.5SS is chosen first since its loss-free goodput is better than $30Mbps$. It then probes upward at 54SS which offers lower goodput estimate, so it finally identifies the best rate as 40.5SS.

Two-level probing priority MiRA ranks the sequence of rates to be probed within each mode and across modes using a two-level priority scheme. The first-level priority addresses intra-mode and inter-mode probing. In MiRA, intra-mode probing is always given higher priority and takes precedence over inter-mode probing. Therefore, probing in MiRA always starts to probe other rates in the same mode (SS or DS). The second-level priority manages probing order among candidate rates in the same mode. MiRA always gives higher priority to the rate option closer to the current rate. Therefore, it always probes the adjacent rate first, and then the next higher/lower rate in the same mode when probing upward/downward. In a sense, MiRA stays in the middle between sequential rate adaptation (e.g., RRAA) and best-rate RA (e.g., SampleRate): It differs from RRAA in that it may leap to the best rate nonsequentially; it differs from SampleRate in that it still probes sequentially among rate candidates.

Adaptive probing interval Similar to HA-RRAA (Section 4.4.1), MiRA applies an adaptive probing interval to limit transmissions at low goodput rates. It uses two mechanisms of *loss-proportional* and *binary exponential growth* to adaptively set the probing intervals for three eligible rates; the two adjacent intra-mode rates and one inter-mode rate. These three rates are used for probing upward and downward in the current mode, and probing in the other mode. The inter-mode rate is the smallest rate in the other mode which loss-free goodput is larger than the goodput at the current rate. Consider the current rate 54SS at time t_2 in Figure 5.10, the adaptive probing intervals are set for three rates: 81SS and 40.5SS used for intra-mode, and 54DS which loss-free goodput is larger than the goodput 30Mbps at 54SS. As MiRA adapts its rate upward or downward, these three rates are also changed accordingly.

Whenever the probe to these three rates results in a smaller goodput than the current transmission rate, the probing interval for rate r is adjusted based on the following formula:

$$T(r) = T_0 \cdot \min(2^k, 2^{10}) \cdot \max(1, \frac{l(r)}{l_0}) \tag{5.1}$$

where T_0 is the minimum probing interval (say, 2ms in our implementation), $l(r)$ is the current loss percentage SFER at rate r, l_0 is a threshold parameter for loss percentage (say, 10% in our implementation), and k denotes the number of probes to rate r. The update rule states that, the probing interval increases in proportion to the loss percentage $l(r)$ once it exceeds the minimum loss threshold. Moreover, as the number of probes to rate r increases over time, the probing interval grows exponentially but is upper bounded by 2^{10}. The binary exponential growth eliminates the rates that consistently offer lower goodput by probing to these rates less frequently over time. Together, these two mechanisms effectively reduce the probing frequency to the bad rates, thus limiting the associated performance penalty.

Whenever the probe to one of these three rates yields higher goodput, MiRA resets the probing interval and moves to the new best rate. It subsequently applies the same update rule to the three new probe rates.

5.4.1.2 Goodput estimation

The moving average and deviations of the goodput at probe rate r is computed as follows:

$$\begin{aligned}\overline{G_r}(t) &= (1-\alpha)\cdot\overline{G_r}(t-1)+\alpha\cdot G_r(t)\\ \sigma_r(t) &= (1-\beta)\cdot\sigma_r(t-1)+\beta\cdot|G_r(t)-\overline{G_r}(t)|\end{aligned}$$

where $\alpha = \frac{1}{8}$ and $\beta = \frac{1}{4}$ are two parameters. Note that the instantenous goodput depends on the aggregation level, which may vary a lot from one aggregate frame to another. Using the aggregation level measured from the current probe may lead to fluctuating and inaccurate estimation. To address this issue, we use the moving average of the aggregation level:

$$\overline{A_r}(t) = (1-\alpha)\cdot\overline{A_r}(t-1)+\alpha\cdot A_r(t)$$

where $A_r(t)$ is the measured aggregation level (in terms of frames) for the current probing frame. Based on this aggregation estimate, we compute the goodput as:

$$G_r(t) = \frac{DATA\cdot\overline{A_r}(t)\cdot(1-SFER)}{T_{overhead}+\frac{DATA\cdot\overline{A_r}(t)}{r}}$$

where $DATA$ is the payload size of a MAC-layer frame, and $T_{overhead}$ is the various 802.11n protocol overhead (related to $DIFS$, $SIFS$, $BlockAck$, etc.).

5.4.2 Handling hidden terminals

Recent studies [24, 34] have shown that interference-induced data losses can adversely affect the rate adaptation operations. In such cases, reducing the rate upon losses may exacerbate collisions since the transmission takes a long time at lower rates. Thus, a good RA design should differentiate between channel fading losses and collision losses. This holds for the MIMO case as well.

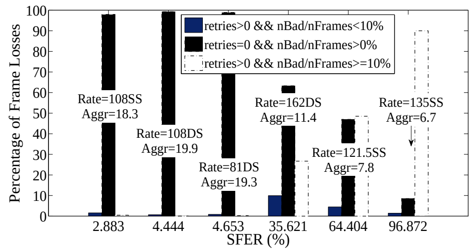


Figure 5.12: Loss patterns w/o interference.

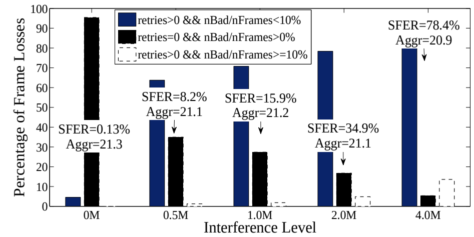


Figure 5.13: Loss patterns with interference.

Collision detection MiRA takes a novel approach to collision detection by exploiting the unique MIMO features of frame aggregation and BlockAck. During our extensive experiments, we have observed that channel fading losses and collision losses tend to exhibit very different patterns (uniform and near-binary, respectively). As an illustrative example, Figure 5.12 shows the loss patterns of UDP traffic from the AP to a client located at P15 without interference, while Figure 5.13 presents the loss patterns under a hidden terminal setting. Our hidden station is located at P12 and varies the interference level, by transmitting frames at different rates (from 0.5Mbps to 4Mbps). We categorize the frame losses into three types, based on the number of retries and the loss rate in the last retry. These results (and similar ones at other locations) reveal a distinct pattern of collision losses:

$$retries \geq 1 \quad AND \quad \frac{nBad}{nFrames} < 10\% \tag{5.2}$$

That is, the last aggregate frame experienced at least one retry, yet in the last retry, it was received with very mild subframe loss. The root cause of the above interference loss pattern can be attributed to the corruption of the PHY header upon collisions, thus causing the entire A-MPDU to be lost [75].

These findings provide us a simple heuristic to infer the possible occurence of collisions, by checking the above condition against each aggregate frame transmission.

While this heuristic is shown to be quite effective in our experiments (detailed in Section 5.6), it may lead to incorrect detection results occasionally (categorizing fading losses as collisions, or vice versa). To improve the detection accuracy, MiRA relies on repeated collision indications during a short timespan, rather than a single instance. To this end, MiRA maintains a dynamic interference observation window ($IFWnd$), which is normally set to 0. Whenever an aggregate frame satisfies Condition (5.2), MiRA suspects collisions and thus initializes $IFWnd$ to a pre-defined value (say, 3 in our implementation). For the subsequent $IFWnd$ aggregate frames, if any of them exhibits the collision pattern again, MiRA will confirm the collisions and trigger the reactions, as described below. Otherwise, $IFWnd$ decrements by one for each frame not satisfying Condition (5.2), until $IFWnd$ reaches 0.

Two alternatives to collision detection, using adaptive RTS filter [34] and SNR [24], both have downsides in the 802.11n MIMO case. An MIMO device typically operates at much higher rates than the legacy 802.11b/a/g device, thus the relative overhead of RTS/CTS grows much larger. Because the adaptive RTS/CTS scheme turns on RTS/CTS regardless of date rate or frame size, it introduces significant overhead with high rates and/or small frames. On the other hand, the SNR-triggered approach requires the sender to obtain fine-grained, per-frame accurate SNR information from the receiver, which is not available in current 802.11n systems. Moreover, 802.11 systems only measure SNR for successfully received, not collided frames at the receiver.

Cost-effective collision reaction Similar to HA-RRAA cost-effective A-RTS filter (Section 4.4.1.1), MiRA takes a cost-effective approach to whether to turn on RTS/CTS protection, by enabling it *o*nly when the potential gain outweighs the overhead. It first estimates the RTS/CTS transmission time (T_{RCTS}) and A-MPDU's transmission time as $\frac{|AMPDU|}{R}$ where an A-MPDU with size $|AMPDU|$ is transmitted at rate R. MiRA will turn on RTS/CTS only if $\frac{|AMPDU|}{R} \geq k \cdot T_{RCTS}$, where k is a benefit/cost ratio (say, 1.5 in our prototype). Benefit/cost ratio k represents the minimum number of collisions that need to happen, to compensate the cost of enabling RTS/CTS. If this condition is not met, MiRA resets $IFWnd$, without turning on RTS/CTS.

MiRA further amortizes the RTS/CTS overhead over multiple aggregate frames. This is done by setting the NAV (Network Allocation Vector, supported by all 802.11 standards) as the transmission time of multiple back-to-back aggregate frames in the buffer. To reduce the negative impact of stealing fair access from other competing devices, our prototype limits the amortization to two large back-to-back aggregate frames, though more aggressive amortization is feasible.

5.5 Alternative Designs for MIMO 802.11n Rate Adaptation

In this section we discuss alternative designs to MiRA, which take different approaches to address SFER non-monotonicity between cross-mode rates. Our first approach is a parallel MIMO mode RA scheme, which conceptually runs an independent RA in each MIMO mode and selects the best goodput rate from all modes. We also discuss several other alternatives, which use SNR, extensive probing at all MIMO modes, or 802.11n fast MCS feedback to overcome SFER non-monotonicity.

5.5.1 Window-based 802.11n RA

Window-based rate adaptation (WRA) seeks to address SFER non-monotonicity between cross-mode rates, by running an independent RA in each MIMO mode in parallel. To limit probing at low goodput rates, WRA maintains and adjusts a *Sliding Rate-Selection Window* ($RSWnd$) in each MIMO mode. $RSWnd$ includes the current best candidate transmission rates of each mode. WRA then opportunistically selects at run-time, the best goodput rate among the best SS/DS candidate rates. We next elaborate

WRA design.

5.5.1.1 Sliding window best rate selection

WRA design is based on a rate-selection window, specified by *[minRate, maxRate]*. RSWnd is introduced to reduce probing overhead based on recent rate history. Specifically, it seeks to prevent transmissions at high loss rates located above *maxRate* bound, or at low goodput rates located below *minRate* bound. To achieve this, RSWnd is adjusted to the rate set that offers high long-term average goodput. WRA then selects the best goodput rate within this set, that offers highest opportunistic goodput gains according to the instantaneous channel variations. To overcome loss non-monotonicity observed in cross-mode rates, WRA maintains and adjusts different RSWnds for both SS and DS modes. It sequentially moves rate-selection window up/down for each mode independently upon low/high frames losses, to accommodate higher/lower rates respectively. We then need to answer the following questions. (1) How to decide the best-throughput rate? (2) When to trigger window movement? How long it will be? (3) What is the appropriate window size?

Best-throughput rate selection WRA selects the best-throughput rate r among the rates included in SS and DS RSWnds as $Th_r \cdot (1 - \overline{SFER_r})$, based on the moving average loss statistics $\overline{SFER_r} = \frac{7}{8} \cdot \overline{SFER_r} + \frac{1}{8} \cdot SFER_r$. Th_r represents MAC-layer throughput, while SFER is calculated based on equation presented in Section 3. Note that WRA avoids using long-term aggregation statistics to select the best rate. So, it can remain adaptive to fast channel dynamics. To this end, it also devises an aging mechanism to periodically reset long-term SFER statistics as we discuss next.

Triggers for window movement WRA uses a time-driven mechanism to trigger SS and DS windows' movement. When a timer expires, it first computes the highest throughput inside each RSWnd. It then moves the window downward, if the best throughput is lower than the loss-free throughput of $minRate-1$. Alternatively, WRA moves RSWnd upward, if its upper bound rate $maxRate$, yields low SFER (less than 15% for our prototype). Note that the timers for moving upward and downward may be different, as we discuss next. To remain responsive upon rapid channel deterioration, WRA also introduces a fast reaction mechanism, which immediately moves the SS/DS window downward, if the last SS/DS transmission has experienced excessive retries, respectively. Note that fast reaction mechanism only moves the window downward, to embrace lower rate options, without necessarily decreasing the actual rate in use. Thus it is much more robust than early RA algorithms (e.g., ARF [17]), which immediately decrease the rate upon consecutive sub-frame losses.

Length of window movement WRA moves each RSWnd upward or downward by one rate option at a time. The rationale behind this design choice, is that loss still monotonically increases/decreases when we move to higher/lower rates across an individual MIMO mode. So WRA, cautiously explores new rate options one by one.

Impact of window size WRA's RSWnd is fixed to two rate options for each MIMO mode, based on our experiments. It is interesting to note that, via the choice of window size, WRA balances between responsiveness to fast channel changes and probing overhead. With a window size of one, WRA degenerates to a RRAA-like algorithm, which transmits using the same rate for a short-term time window. This design choice affects responsiveness to channel dynamics happening in finer time granularity. With window size of ∞, WRA is similar to SampleRate-Unbounded theme described in Section 5.5.2, which needs to pay a significant probing cost to keep updated the performance statistics of the big rate span offered by MIMO 802.11n.

Adaptability vs. probing overhead To remain responsive to fast channel variations, WRA applies an aging mechanism that resets periodically (every 50ms in our

Procedure 5 $SingleModeRA$: Input (BlockAck, MimoMode), Output (r)

```
getRSWndBounds(MimoMode, &maxRate, &minRate);

//Fast Reaction Mechanism
if txFailed(MimoMode, BlockAck) then
    move_RSWnd_down(maxRate, minRate);
else if down_timer_fired() && maxThr(maxRate, minRate) < lossfreeThr(minRate-1) then
    //Timer Expired and Channel is Bad
    move_RSWnd_down (maxRate, minRate);
else if up_timer_fired() && SFER_maxRate < 15 then
    //Timer Expired and Channel is Good
    move_RSWnd_up (maxRate, minRate);
end if

update_probe_timer(BlockAck, r);

return best_RSWnd_rate(maxRate, minRate);
```

prototype) the loss statistics as $\overline{SFER_r} = \frac{7}{8} \cdot \overline{SFER_r}$, giving the opportunity to high goodput rates, to be probed again. However, this may increase transmissions at low goodput rates, as well. To address this issue WRA uses MiRA's adaptive probing interval presented in Section 5.4.1.1. When a transmission at rate r fails (no BlockAck is received), probing interval is updated with $l(r)$ set to $\overline{SFER_r}$. Probing interval is also reset for r upon a successful transmission to this rate. So, WRA eliminates the rates that consistently offer lower goodput by probing to these rates less frequently over time.

5.5.1.2 Putting everything together

WRA runs RSWnd operations described in Procedure 5 for each MIMO mode and finally selects the best goodput rate between the best SS and DS rates. For our case study scenario of Section 5.2, SS, DS RSWnds will be set to [108SS, 121.5SS], [108DS, 162DS] respectively and WRA will select 162DS as the best goodput rate. An improvement or deterioration in channel quality, will cause the windows to sequentially move to higher or lower rate options respectively.

We identify two different design philosophies comparing WRA to MiRA. First, WRA seeks to address SFER non-monotonicity between cross-mode rates by running in parallel across different MIMO modes. For our case study setting described in Section 5.2, DS RSWnd will include the best goodput 162DS rate. On the other hand, MiRA applies a zigzag mechanism to overcome this issue. Second, MiRA will transmit at the selected rate for a short-term time window, until timers expire or events are triggered. However, upon adjusting rate selection windows, WRA can switch among different rate options on per-AMPDU basis. This design choice can be more adaptive to fast channel dynamics, but it may come at a higher probing overhead as we discuss in Section 5.6.3.

WRA's operations across an individual MIMO mode differ from legacy RAs. Compared with RRAA, which transmits using the same rate over a short-term window, WRA chooses among a set of rates on per-AMPDU granularity, to exploit the intra-mode channel opportunistic gains. Differently from SampleRate, which applies random probing at different rates, WRA probes only inside RSWnd. So, WRA stands in the middle between sequential and best-rate adjustment design. Inside the RSWnd, WRA jumps at the best goodput rate. However, it moves the window sequentially to accommodate higher/lower rates upon better/worse channel.

5.5.2 Other design options for MIMO RA

Now we discuss several alternative design approaches to MIMO RA, which can be implemented as extensions to legacy 802.11a/b/g RAs. The first approach searches for the best rate option within a pre-specified range of rates spanning all modes. However, it is nontrivial to properly pre-select the range: the smaller the rate range, the higher probability the optimal rate is missed; the larger the rate range, the bigger the probing overhead. We will further examine a *tuned SampleRate* design in this category in

Section 5.6.3. The second approach exploits the observation made in Section 5.3.2 and takes a SNR-based mode selection. When the measured SNR is lower (or higher) than a threshold, it chooses the SS mode (or DS mode) and uses the conventional RA within the mode. One challenge for this approach is how to set the thresholds, which change with different operation environments, as we demonstrate in Section 5.6.3. The third approach uses the fast MCS feedback mechanism supported by the 802.11n standard, where the receiver can communicate the best rate option to the sender. In Section 5.6.3 we discuss implementation limitations and design challenges, which fast MCS feedback needs to address in the current 802.11n chipsets.

5.6 Implementation and Evaluation

In this section, we describe MiRA's implementation and evaluate its performance using both controlled experiments and field trials.

5.6.1 Implementation

We implemented MiRA in the firmware of a programmable AP platform (about 900 lines of C code). Compared with other RA algorithms, MiRA poses two implementation challenges. First, its probing mechanism requires frame transmission and rate control, which are two separate modules in the driver, to be synchronized on a per-AMPDU basis. We maintain an additional binary state for each client (other states kept at AP are per-client statistics), which is set upon collision losses and checked for each AMPDU transmission. The second challenge is that, the NAV for RTS cannot be directly set by the transmission module of the driver. To reserve the wireless channel, we use Atheros' *Virtual more Fragment* interface, which consists of a virtual more-fragment bit (vmf) and a $burst\_duration$ parameter. Atheros uses this interface

	Atheros RA	RRAA	SampleRate
Static UDP	(3.4-82.3)%	(2.9-71)%	(1.1-104.5)%
Static TCP	(9.1-107.9)%	(5.9-37.5)%	(14.7-124.8)%
Mobility UDP	116.1%	30.2%	182.2%
Mobility TCP	72.5%	4.9%	94%
Hidden Terminal	(79.4-1094)%	up to 6.5%	(33.8-983)%
Field Trial	(46.35-67.4)%	(16-28.9)%	(19.4-73.5)%

Table 5.5: Goodput gains of MiRA over existing RAs.

to enable frame bursts. Upon collision losses, if channel reservation is possible we set the vmf bit as 1 and $burst\_duration$ as the transmission time of the aggregated frames that NAV in RTS protects (the reception time of BlockAck is also included in $burst\_duration$). The virtual more-fragment bit goes down to the hardware queue together with the burst of aggregate frames.

5.6.2 Performance evaluation

In this section, we compare MiRA with RRAA [34], SampleRate [23] and Atheros MIMO RA [32]. For RRAA, we disabled its adaptive RTS/CTS filter, except in the hidden terminal settings, to avoid goodput degradation which was observed to be up to 12.2% during our experiments. These experiments were conducted in various scenarios with static/mobile clients, hidden terminal stations, under different MIMO configurations with both TCP and UDP traffic. All the algorithms were implemented on the AP side. The results show that MiRA consistently outperforms existing alogithms in all scenarios, with goodput improvement up to 73.5% in field trials. The performance gains of MiRA over existing RAs are summarized in Table 5.5.

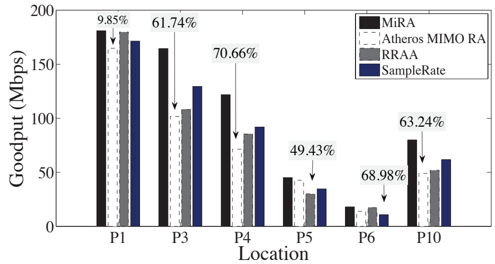


Figure 5.14: 3×3 /5GHz/UDP static setting.

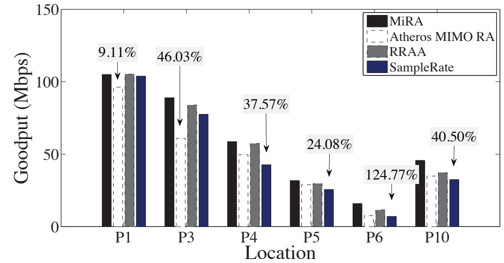


Figure 5.15: 3×3 /5GHz/TCP static setting.

5.6.2.1 Static clients

We first evaluate the RA algorithms with static clients at multiple locations. We conduct these experiments during midnight and, in the 5GHz band cases, we select interference-free channels, as verified by the sniffer. We also perform tests with various antennas configurations. The channel bandwidth is set to 40MHz in all experiments unless explicitly specified.

UDP/3×3 Antennas/5GHz case Figure 5.14 plots the UDP goodput measured at 6 different locations (as marked in Figure 5.1) with 3×3 antennas at 5GHz band and the maximum MiRA goodput gains over the other designs. We see that MiRA performs better than other algorithms at all locations, with goodput gains up to 70.7% over Atheros RA, 54.2% over RRAA, and 68.9% over SampleRate. Except from the closest client-to-AP location where all RAs tend to transmit at high rates, MiRA delivers significant gains which can go up to 70.7% at location P4.

UDP/2×2 Antennas/5GHz case To assess the impact of antenna configurations, we also evaluate the system with 2×2 antennas, again at 5GHz band. Our experiments show that MiRA still outperforms other algorithms at all locations, with the goodput gains varying from 15.2% to 104.5%. In 3×3 configuration, in the same layout (location, RA algorithm), we observe up to 43% higher goodput compared with the 2×2 configuration; this gain is attributed to additional signal redundancy offered by the third antenna.

TCP/3×3 **Antennas/5GHz case** We also conduct experiments with one flow TCP traffic. Figure 5.15 shows that MiRA gives significant TCP goodput gain over others, up to 107.9% over Atheros MIMO RA, 37.5% over RRAA, and 124.8% over SampleRate. Similar to the UDP scenario, MiRA offers high gains in all locations, starting from 24.1% (location P5) to 124.8% (location P6).

UDP/3×3 **Antennas/2.4GHz/40MHz case** We also test 2.4GHz channels. Setting the channel to 40MHz in 2.4GHz results in partially overlapping channels. During this experiment, we set our AP on Channel 1. We sniff many other APs on other channels: twelve on Channel 1, two on Channel 4, eight on Channel 6, six on Channel 9 and nine on Channel 11. The goodput performance and gains of MiRA vary from 9.6% to 57.7% at five locations, as shown in Figure 5.16. We see that losses and goodput degradation are significant compared with the 5GHz band due to highly uncontrolled interference.

UDP/3×3 **Antennas/2.4GHz/20MHz case** We finally repeat the experiments of the 2.4GHz band setting using 20MHz channel. For the 20MHz channel case, MiRA gives also significant gains which are up to 36.9% over Atheros RA, 70% over RRAA, and 80.3% over SampleRate. Even with 20MHz channel the highest goodput observed was 43Mbps because of the intense interference.

From our experiments, we identify additional aspects that contribute to the performance gains of MiRA.

Effective probing Most existing RAs do not have any efficient mechanism to learn from short-term past channel's performance, which can lead to significant amount of transmissions at low goodput rates. For example, at location P4 of our

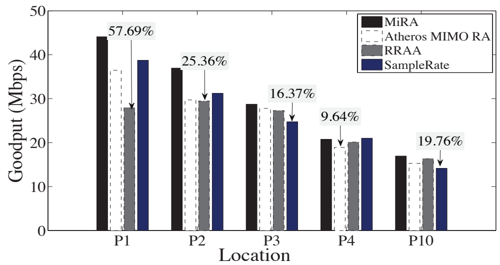


Figure 5.16: 3×3 /2.4GHz/40MHz/UDP static setting.

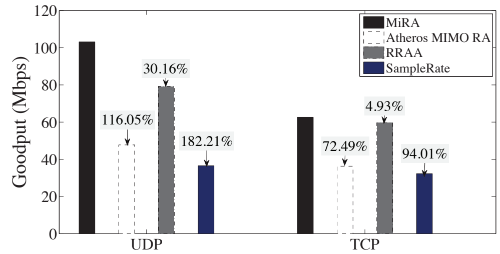


Figure 5.17: Goodput in mobility setting.

case study, RRAA sends 53% of MPDUs at 121.5Mbps, which exhibits significant loss ($SFER_{121.5} = 17.9\%$ from Table 5.1). Similar behavior is observed in other scenarios such as location P10, where RRAA transmits 31.7% of MPDUs at 108SS that presents 34% average loss. Similar to RRAA, SampleRate also uses non-adaptive probing, despite less aggressive than RRAA. In contrast, the *adaptive probing mechanism* of MiRA prevents it from excessively transmitting at lossy rates. MiRA transmits only 2% and less than 3% of MPDUs at low-goodput rates, at locations P4 and P10, respectively.

Handling SFER non-monotonicity By zigzagging between MIMO modes, MiRA avoids to get trapped at lower rates in loss non-monotonicity scenarios. In our case study setting at location P4, MiRA transmits 96% of frames at 162Mbps which is on average the best goodput rate. In contrast, other algorithms transmit their frames at rates lower than 162DS. At location P10 where non-monotonicity is exhibited between 108SS and 108DS rates, MiRA transmits 79% of its MPDUs at 108DS, which is the average winner (Figure 5.3), differently from other RAs which transmit at most 1% at this rate. We also observed that Atheros MIMO RA may occasionally get trapped at lower rates because of SFER same-rate-pair non-monotonicity (say 54SS/DS) in some locations. The Atheros algorithm *a priori* ranks all rates to be probed in the particular

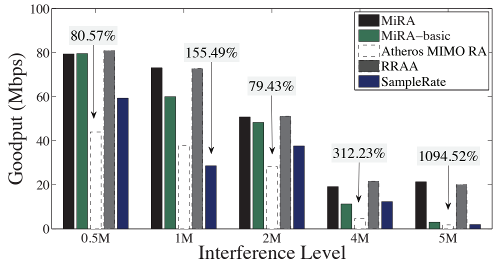


Figure 5.18: Goodput in hidden terminal.

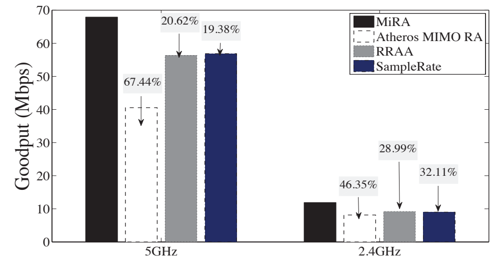


Figure 5.19: MiRA performance in field trials.

order, say 54DS has higher ranking than 54SS but lower than 81SS in the implementation. Consider that the current probe upper-bound rate is set as 54DS in the Atheros algorithm. If 54SS gives better performance than 54DS, the algorithm may get trapped at 54SS (Table 5.1).

5.6.2.2 Mobile clients

In order to gauge the responsiveness of MiRA to fast channel dynamics, we carry a client and walk from P1 to P6 and then come back at approximately constant speed of 1m/s. Figure 5.17 plots the goodput of the four RAs for both UDP and single-flow TCP traffic. MiRA offers goodput gains up to 116.1% over Atheros RA, up to 30.2% over RRAA, and up to 182.2% over SampleRate. As discussed in Section 5.4, MiRA uses (i) *moving average* to detect significant channel changes, (ii) only one AMPDU to probe, which is transmitted in a relatively short period and typically contains enough samples, and (iii) resetting statistical history upon rate changes. Consequently, MiRA quickly adapts to channel dynamics due to mobility.

5.6.2.3 Setting with hidden terminals

We next evaluate whether MiRA can successfully infer collision losses and adjust the rate accordingly in the hidden terminal scenario of Section 5.4.2. We also compare MiRA with MiRA-basic (MiRA without interference module) to evaluate the performance of our interference module. In the hidden terminal setting, we also turn on RRAA's Adaptive RTS filter. Figure 5.18 presents the gains of MiRA at five interference levels where we vary the traffic intensity of the hidden terminal. We observe that MiRA is very effective in intense interference scenarios (4Mbps and 5Mbps), where it gives up to 11.9 times higher goodput over Atheros MIMO RA and SampleRate. MiRA performs similar to RRAA, without having to pay the RTS/CTS overhead of RRAA's adaptive RTS filter. Finally, MiRA gains over MiRA-Basic range from 5.1% to 599.9%. MiRA big gains are attributed to its selective RTS mechanism, which limits collision losses and prevents MiRA from probing down by misinterpreting interference as channel losses. MiRA yields up to 32.5% smaller average loss compared with Atheros MIMO RA and SampleRate. By avoiding probing down in high interference scenarios, MiRA still transmits at high rates under heavy collisions. With 4Mbps interference, Atheros MIMO RA transmits 94% of frames at the lowest rate 13.5Mbps, while MiRA only transmits 6% at this rate.

5.6.2.4 Field trials

We also conduct uncontrolled field trials under realistic scenarios, where various sources of dynamics coexist in a complex manner. In our field trial, we use 3 static clients, at locations P4, P10, and P17, and we move an 802.11n client on a regular basis based on the mobility scenario of Section 5.6.2.2. We use TCP traffic to evaluate each RA for about an hour. During our experiments, the physical environment was highly dynamic as people walk back and forth. Figure 5.19 shows the comparison results

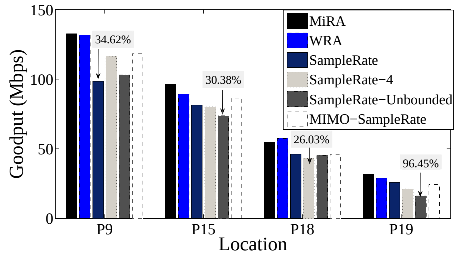


Figure 5.20: MIMO RA alternatives in static settings.

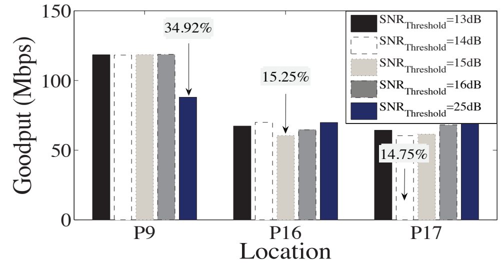


Figure 5.21: MIMO-SampleRate vs. different SNR thresholds.

both for 5GHz band and the more dynamic and congested 2.4GHz band. MiRA gives goodput gains up to 67.4% over Atheros MIMO RA, 28.9% over RRAA, and 32.1% over SampleRate.

5.6.3 Assessing MIMO RA alternatives

We next assess the alternative MIMO RA designs presented in Section 5.5 in a wide SNR range of controlled settings (from 10dB to 30dB) and field trials, at 5GHz band, with 40MHz channels and 3x3 antennas. In field trials, we use 3 static clients, at P15, P4 and a sibling to P8 location (P8a). We also move a client on a regular basis from location P3a (sibling to P3) to P7 through P4, at pedestrian speed. Traffic is UDP, TCP for controlled, field trial settings, respectively.

Window-based RA Algorithm Our experimental results presented in Figure 5.20 show that WRA can give 5.3% goodput gains over MiRA (location P18). Interestingly, we observe that the average SNR at P18 is 14dB and belongs to [13dB, 16dB] SNR range, in which our experiments have not identified a winning MIMO-mode (see Section 5.3.2). WRA is more adaptive in this setting, as it can switch between rates of different modes on a per-AMPDU granularity. In the remaining locations, MiRA is able to converge to the best rate with less probing overhead and yields gains from 0.7%

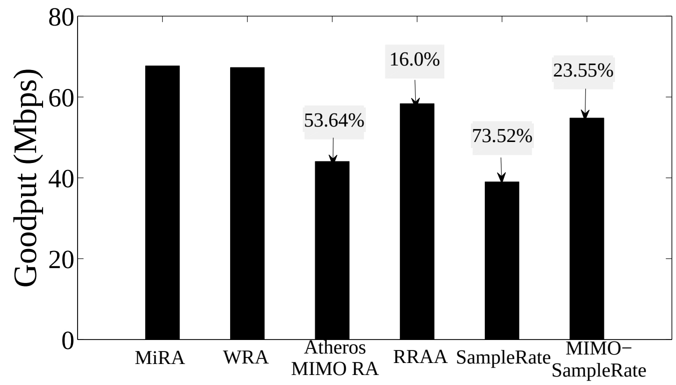


Figure 5.22: MIMO RA alternatives in field trials.

to 9.2% over WRA. In field trials presented in Figure 5.22, WRA performs similar to MiRA, while it gives gains up to 72.5% over the other designs.

Tuned SampleRate Algorithm By upper-bounding sampling up to 2 rates higher than the current rate, SampleRate has limitations to address SFER non-monotonicity as stated in Section 5.2. To address this issue, we implement SampleRate-4 that enlarges the sampling bound to 4, and SampleRate-Unbounded that allows for search among all the rates larger than the current rate. Figure 5.20 indeed shows that by expanding its search scope, SampleRate-4 achieves goodput gain of 18% over SampleRate at P9. However, SampleRate-4 does not perform as well in the other three locations. At P19, SampleRate delivers 21.2% goodput gain over SampleRate-4. SampleRate-Unbounded is even worse, incurring goodput reduction up to 37.3%. Trace analysis reveals that SampleRate transmits 87% of frames at the high-goodput rates (40.5Mbps, 54Mbps), whereas SampleRate-4 transmits only 50% at these rates. SampleRate-Unbounded transmits 9.5% of frames at almost 100% loss. Sampling of these expanded rates consequently incurs higher probe penalty.

SNR-based Mode Selection RA Our proposed MIMO-SampleRate selects the SS/DS mode based on a pre-selected SNR threshold, (14dB in our implementation) measured from the received ACK frames and averaged over all antennas.[1] This design exploits the findings of Section 5.3.2, where in low/high SNR range SS/DS is more likely to be the winning mode. Once the mode is chosen based on SNR, SampleRate algorithm is used within the selected mode.

MIMO-SampleRate achieves higher or similar goodput compared with tuned-SampleRate, with goodput gains up to 20% in static settings (Figure 5.20) and 40.4% in field trials (Figure 5.22). However, MiRA still outperforms MIMO-SampleRate with goodput gains up to 30% in statics settings and 23.6% in field trials. These results indicate that the SNR-based MIMO-SampleRate can achieve good performance, while retaining its operation simplicity. However, our experiments show that, there may not be an optimal SNR threshold to give the highest goodput in all the settings. The best SNR threshold values may also depend on the operation environment. Figure 5.21 shows the goodput performance of different SNR thresholds at locations P9, P16, P17, whose average SNRs are 23dB, 16dB, 14.5dB, respectively. By choosing a high SNR threshold, say 25dB, at location P9, we exclude DS rates (including the highest-goodput rate 162DS), thus incurring goodput degradation up to 34.9% compared with using [13dB, 16dB] thresholds at P9. However, choosing SNR thresholds in [13dB, 16dB] does not guarantee the best performance in other locations. At P16 and P17, 25dB threshold outperforms other choices up to 15.3% and 14.8% respectively. This is attributed to the algorithm's fluctuation between SS/DS modes when using other threshold values. For example, at P17, while 25dB threshold transmits more than 98% of the frames at 81SS (which is the best goodput rate), other threshold values give sub-optimal rate distributions.

RA Based on Fast MCS Feedback Fast MCS feedback (MFB) supported by the IEEE 802.11n standard, can be used for receiver-initiated rate adaptation based on

[1] A more sophisticated design that uses two thresholds, gives similar results.

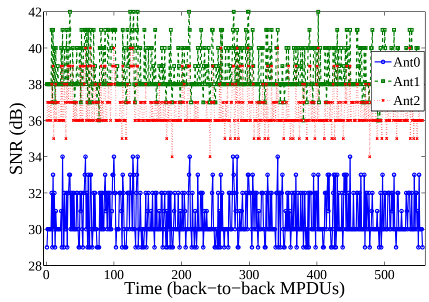


Figure 5.23: Control channel per-antenna SNR in static RF-chamber settings.

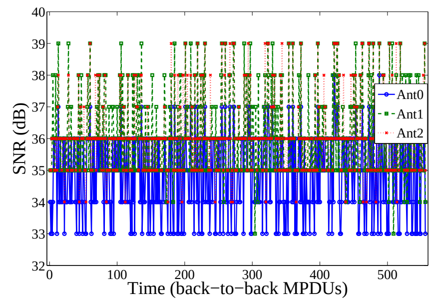


Figure 5.24: Extension channel per-antenna SNR in static RF-chamber settings.

per-frame feedback [10, 33, 36–38]. Although MFB is more adaptive to fast channel dynamics compared with MiRA, it faces certain practical limitations and design challenges. First, MFB is an optional feature and its implementation is vendor-dependent. Our available APs which use proprietary drivers from Atheros (AR5416 chipset) and Broadcom (BCM47XX, BCM53XX chipsets) and popular open source drivers, as Intel's iwlagn and Atheros' ath9k, do not currently implement MFB algorithms. As a result, loss-based, transmitter-side RAs as MiRA are required when MCS feedback is not available. Second, various metrics used in MFB algorithms, as uncoded bit error rate or per-subcarrier SNR (a survey can be found in [38]) are not available in many commodity 802.11n drivers.

There are also design challenges, which receiver-side RAs need to address. Our experiments in RF chamber reveal large SNR variations, when SNR is calculated from received signal strength and noise floor. Figures 5.23, 5.24 plot the per-antenna received SNR for the control (the primary 20MHz channel) and the extension channel in a static 3×3 setting, when an 802.11n client sends back-to-back UDP 1.5KB MPDUs at the AP (aggregation is disabled). The rate is fixed at 135Mbps and the time gap between consecutive frames is less than 0.35 milliseconds. We observe that SNR variations can be up to 5dB between consecutive frames both for control and extension channel, which can impact the decision of the best goodput rate. SNR fluctuations in commodity 802.11 devices have been verified by independent studies [25, 30, 33], and can be attributed to multipath and hardware calibrations. A more accurate SNR calculation requires per-subcarrier SNR feedback, which may not be available in the current commodity 802.11n drivers. Finally, SNR-BER relations vary with different propagation environments. Consequently, SNR-based solutions require in-situ training to perform well across different propagation environments [24].

Ideally, MFB is communicated on a per-transmission basis. However, is there any protocol overhead? What is the impact of delayed MFB in RA performance? Kant et al. in [38] show that MFB delays can lead to more than 40% throughput decrease. We further investigate MFB in our future work.

5.7 Related Work

There have been several rate adaptation proposals [17, 23–25, 27, 28, 33, 34] in recent years. Many of them target the legacy 802.11a/b/g networks [23–25, 34], or take a cross-layer approach [27, 28] by using PHY-layer feedback to select the best goodput rate. These algorithms are not designed for MIMO systems and they do not consider MIMO modes and 802.11n frame aggregation. The early work on MIMO RA [10,33,36–38] takes the receiver-based approach by exploiting the MCS feedback. Although, these approaches can be more adaptive to fast channel dynamics, they have not been widely adopted by commodity 802.11n systems, due to their practical limitations (5.6.3). Transmitter-based approaches have been proposed as well. ARFHT [39] selects the best MIMO mode based on SNR differences among receive antennas. It assumes MIMO channel reciprocity as it measures SNR at the sender based on received frames. ARFHT faces the challenges of using SNR feedback raised in Section 5.6.3.

Atheros MIMO RA selects the best goodput rate based on SFER statistics, while it upper-bounds probing and rate selection.

There are also a few experimental studies relevant to this work. In [33], authors study packet error rate/SNR relation, without focusing on the performance of different MIMO modes. In [40], experiments are based on a testbed that supports only a limited set of 802.11n features. Finally, theoretical studies on MIMO communications [9, 10], seek to characterize the theoretical tradeoffs of MIMO systems, often in the limiting cases. In contrast, our study uses real experiments to examine the behavior of 802.11n MIMO devices.

5.8 Summary

In this chapter, we empirically study MIMO rate adaptation using an IEEE 802.11n compliant, programmable AP platform. The key insight learned, is that diversity-oriented SS mode and spatial multiplexing-driven DS mode exhibit different features and cannot be managed indistinctly. Existing RA solutions do not properly consider characteristics of SS and DS, thus suffering from severe performance degradation. To this end, we first propose MiRA, a new zigzag RA algorithm that explicitly adapts to the SS and DS modes in 802.11n MIMO systems. We also design and evaluate window-based and SNR-based MIMO RA solutions. Our experiments in controlled testbeds and field trials show clear gains of MIMO-mode aware RAs. In a nutshell, our work is among the first to examine MIMO RA in a practical setting using programmable 802.11n commercial hardware. We expect that our effort will stimulate more community effort on MIMO RA to exploit the full capacity of MIMO communication.

CHAPTER 6

Towards Green MIMO 802.11n Wireless

The recent IEEE 802.11n standard [6] has opened the venue for fully leveraging Multiple-Input Multiple-Output (MIMO) technology in wireless LANs. An 802.11n device by supporting 4 spatial streams[1], can deliver high rates up to 600Mbps. The upcoming IEEE 802.11ac standard [7] will allow for higher than 6Gbps rates, by supporting 8 spatial streams. However, do more spatial streams/antennas offer better user experience?

In our first case study of Figure 1.1 discussed in Chapter 1, both legacy and MIMO receivers can accommodate the offered 3Mbps application data source rate. However, legacy receiver saves 30% power over MIMO, providing better user experience. Our study reveals that the cause of MIMO poor user experience is MIMO circuitry power consumption, which grows with the number of active RF chains. Specifically, our measurements with commodity 802.11n devices show that, an 802.11n receiver can deplete a smartphone battery in less than two hours, when all its components (i.e., display) but the 802.11n radio are OFF.

To address this issue, the 802.11n standard specifies a new Spatial Multiplexing Power Save (SMPS) feature, which seeks to save power at the receiver by retaining only one active RF chain. The rationale behind SMPS is intuitive and simple; "Maintain only one RF chain to minimize receive power consumption". Our experiments

[1] The number of available spatial streams determines the minimum number of antennas supported by a MIMO device.

show that, SMPS can indeed achieve its goal, by saving up to 1.15W over multiple active receive chains, in certain scenarios. However, can SMPS save energy over multiple active receive chains? Interestingly, our experiments show that the power hungry multiple receive chains can yield 78.6% energy savings over SMPS. This is observed in scenarios where MIMO speed compensates the additional MIMO power consumption. A realistic gauge of the tradeoff between power consumption and speed (goodput) is the per-bit energy consumption (joule/bit). Per-bit energy consumption is defined as the ratio between the total consumed energy and the delivered bits during any data transfer.

In this chapter, we experimentally study the tradeoff between MIMO speed and power consumption, by uncovering step by step the *"good'*, *"ugly"* and *"bad"* of SMPS feature. We then design and implement MIMO Receiver Energy Save (MRES), which seeks to identify and set the most energy-efficient chain setting for the receiver at runtime. The core of MRES is a low-overhead sampling scheme, which excludes those chain configurations that are highly unlikely to yield energy savings. Our prototype experiments show that MRES outperforms SMPS, with energy savings up to 37%.

The rest of the chapter is organized as follows. Section 6.1 discusses MIMO power consumption and introduces the 802.11n Spatial Multiplexing Power Save feature. Section 6.2 presents our experimental platform and methodology. Sections 6.3, 6.4, 6.5, discuss the potential benefits (the "good"), dangers (the "ugly") and drawbacks (the "bad") of SMPS. Section 6.6 presents our proposed MIMO Receiver Energy Save algorithm, while Section 6.7 presents our implementation and evaluation efforts. Finally, Section 6.8 discusses the related work and Section 6.9 concludes the chapter.

6.1 IEEE 802.11n SMPS

The IEEE 802.11n standard uses Multiple-Input Multiple-Output (MIMO) technology to support high date rates up to 600Mbps. It uses multiple transmit and receive RF chains to support two modes of operation. *Spatial Diversity* transmits a single data stream from each chain, thus leveraging independent fading over multiple links to enhance signal diversity. *Spatial Multiplexing* (SM) transmits independent and separately encoded spatial streams from the multiple chains to boost throughput. The performance gains of MIMO are achieved at the cost of increased power consumption due to the added complexity of MIMO related processing and circuits. The power consumption along a signal path P_c, includes the power consumption of all the amplifiers P_{PA} and circuit blocks P_b [64]:

$$P_c = P_{PA} + P_b, \tag{6.1}$$

where the circuit power consumption P_b is in proportion to the number of transmit (N_t) and receive (N_r) RF chains.

The IEEE 802.11n standard [6] specifies a new Spatial Multiplexing Power Save (SMPS) mechanism to improve power efficiency. SMPS allows for a station to operate with only one active receive chain for a large period of time. We next describe SMPS feature and its implementation by popular vendors.

6.1.1 SMPS feature

A station consumes more power on all active receive chains, even though they are not necessarily required for the actual frame exchange. The 802.11n SMPS feature, seeks to reduce MIMO power consumption at the receiver, by allowing it to operate with only one active receive chain for a significant portion of time. It supports two modes of operation.

Static mode In the static mode, the station retains only a single receive chain and forces the transmitter to send using only diversity single-stream rates. An 802.11n station may use the SMPS action frame to communicate its SM Power Save state to the access point (AP). It may also use the SMPS bits of its Association Request to achieve the same purpose.

Dynamic mode In the dynamic mode, a station enables its multiple receive chains when it receives the start of a frame sequence addressed to it. Such a frame sequence shall start with a single-stream individually addressed frame that requires an immediate response and that is addressed to the station in dynamic mode. RTS/CTS can be used for that purpose [6]. So in dynamic mode, the receiver switches to multiple receive chains when it receives a RTS addressed to it and switches back immediately to one active chain, when the frame sequence ends. A drawback of the dynamic mode is that a station cannot distinguish between a RTS/CTS sequence that precedes a MIMO transmission and any other RTS/CTS.

We start our work by asking the following questions.

1. Does SMPS achieve its goal, to save power over multiple active receive chains? Do power savings come for free?

2. Can SMPS save energy over multiple active receive chains as well? In what scenarios?

We next elaborate on the *"good'*, *"ugly"* and *"bad"* of SMPS feature.

6.1.2 SMPS implementation

IEEE 802.11n provides the basic SMPS mechanism and leaves two open questions for the vendors. When do you send SMPS action frame or RTS/CTS to switch chain settings in static and dynamic modes respectively? In our receiver, which uses Intel's Wireless WiFi 5100A/G/N adapter and the open source iwlagn driver, SMPS can be enabled manually by the user. Our transmitter, which is a commercial AP based on Atheros chipset, precedes with RTS only multiple-stream frame transmissions. Whether the transmission rate will be diversity single-stream or spatial multiplexing multiple-stream, is determined by the underlying rate adaptation algorithm.

The second open issue is, what chain setting to select? SMPS defines switching from *one to many* active chains and vice versa, but never defines what is the "many". For example the RTS frame used in dynamic mode, does not explicitly specify the number of chains that should be activated at the receiver. Our receiver device switches to the maximum available chains upon the reception of a RTS. Finally, it is out of the scope of the SMPS to determine the number of active chains on the transmitter side. The standard configuration of our AP is three active transmit chains. Our experiments show that different implementation choices can have a significant impact on 802.11n SMPS performance.

6.2 Experimental Setting

We conduct our experiments using two types of 802.11n devices. Our transmitter is a programmable 802.11n AP platform, which uses Atheros AR5416 2.4/5 GHz MAC/BB MIMO chipset and has three RF chains. Our receiver uses an Intel Wireless WiFi 5100A/G/N adapter and a modified version of Intel's open source iwlagn driver. The receiver has two available RF chains. Both transmitter and receiver plat-

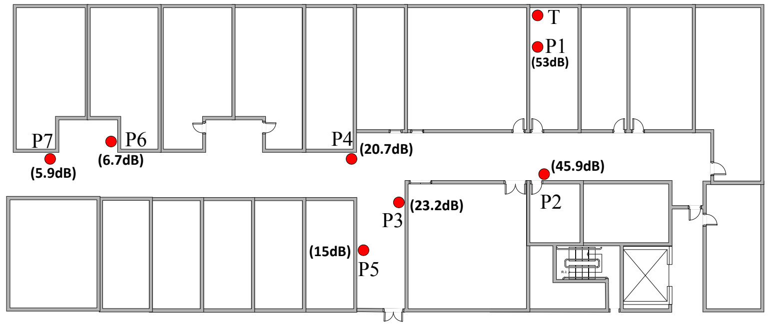


Figure 6.1: Experimental floorplan.

forms allow for both single stream (SS) and double stream (DS) MIMO modes, with transmission rates up to 300Mbps over 40MHz channels. We provide more information about our experimental testbed in Chapter 3.

We conduct our experiments in a campus setting shown in Figure 6.1. Spots P1 to P7 represent different locations where the receiver is placed. The AP is always located at T. For each experiment, we collect frame loss, aggregation, goodput, SNR and power consumption data. To measure the power consumption at the receiver, we use Intel's PowerTOP running on Linux [42]. We disable all other unnecessary applications and hardware at the laptop to improve accuracy. The receiver consumes 1.18W, 1.61W for one and two active chains, respectively, when remaining idle. This 36.4% increase in idle power consumption when switching from one to two chains is also confirmed by another independent study [57].

To single out the impact of idle period on power and energy consumption, we also compute results for two operation modes of 802.11n adapter. At *Doze OFF* mode, the 802.11n adapter remains idle during idle periods, resulting in P_{idle} power consumption. At *Doze ON* mode, the 802.11n adapter switches to the sleep mode during idle, resulting in near-zero power consumption. Doze ON mode may not be always feasible in reality. Fine-grained switching between sleep and active, say, between consecutive frame transmissions, may not be feasible due to switching overhead and delays which can degrade application performance [59]. For example 802.11 PSM, NIC wakes up at the granularity of beacon intervals (100ms). However, we show results for Doze ON mode as a benchmark in our study; they help us to understand the impact of transmission time on power and energy consumption.

6.3 "The Good": SMPS Potential Power Savings

In this section, we seek to answer whether the SMPS feature indeed saves power compared with multiple active receive chains. We first conduct a simple case study at a controlled interference-free setting (location P2). We evaluate the Doze OFF mode here, while we elaborate on Doze ON in the following sections. Our results presented in Figure 6.2 show that, retaining one active receive chain can always save power from 0.5W to 1W, compared with multiple receive chains, in Doze OFF. Therefore, the static SMPS mode, which retains only one chain to save power, is proven correct. The dynamic SMPS mode yields smaller up to 0.4W power savings, over multiple active receive chains. Consequently, the next issue to examine is whether the static mode is always better than the dynamic mode in terms of power consumption. Our case study of Figure 6.2 shows that, the dynamic mode always consumes from 0.2W to 0.7W more power than the static mode in Doze OFF.

Our case study reveals the impact of two factors on power consumption: a) *number of active chains* and b) *application data source rate*. To substantiate our findings, we conduct extensive experiments with various source rates and $N_t \times N_r$ settings. We analyze our experimental results by modeling the receiver power consumption as:

$$P_{rx} = P_p + P_c, \tag{6.2}$$

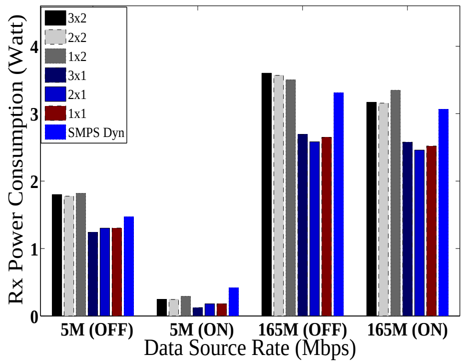


Figure 6.2: Receiver's power consumption (high SNR location P2).

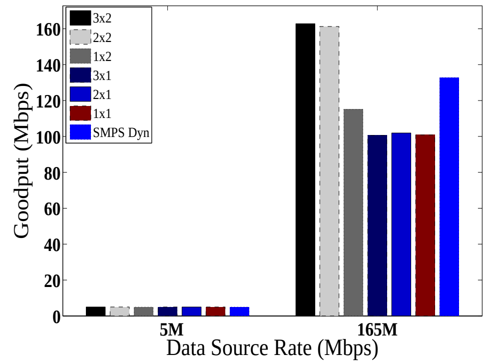


Figure 6.3: Goodput (high SNR location P2).

where P_c and P_p are the MIMO circuitry and processing power consumption, respectively. P_p includes processing in the network protocol stack, and is proportional to CPU utilization U_{CPU}. It can be estimated as $P_p = U_{CPU} \cdot P_f$, where P_f is a system power coefficient per CPU utilization unit.

Number of active chains Our extensive experiments show that, for a given source rate, fixed number of transmit chains N_t and in Doze OFF mode, power consumption monotonically increases with the number of receive chains N_r. Specifically, two active receive chains, can consume 1.15W more power compared with one receive chain. The amount of savings depends on source rate as we discuss next. This increase is mainly attributed to MIMO circuitry power consumption P_c [64]. As a result, static SMPS always yields power savings over multiple chains in Doze OFF, by operating with one active receive chains for long time intervals. Dynamic SMPS always gives power savings up to 0.5W over multiple receive chains, when it operates in Doze OFF as well. The fact that dynamic mode activates a single receive chain only when idle, or when transmissions are diversity, single stream, can justify its smaller power savings compared with the static mode.

The impact of source rate When the offered traffic volume increases, the difference in power consumption P_{rx} between $N_t \times 2$ and $N_t \times 1$ grows from 0.5W to 1W when data source increases from 5M to 165M (Figure 6.2). First, the volume of received frames can increase with the number of receive chains N_r under high sources, as we show in Section 6.4. This makes the gap between processing power consumption P_p between $N_t \times 2$ and $N_t \times 1$ to grow. In our case study, the CPU utilization was approximately 3% higher for $N_t \times 2$ over $N_t \times 1$ settings at 165M, while it was similar at the low 5M source. Second, the gap between power consumption P_c increases, under high volume traffic as well. This is attributed to the fact that MIMO circuitry needs to remain active for a larger fraction of time. We can conclude, the gap in power consumption between two and one receive chain grows with source rate, in Doze OFF. As a result, the potential power savings for static SMPS can increase at higher source rates. However, data source may have the complete opposite effect in dynamic SMPS power consumption. Increasing data source reduces receiver's idle time and as a result its opportunities to operate with a single active receive chain. This can reduce dynamic SMPS potential savings over multiple active receive chains. From Figure 6.2 we observe that the gap in P_{rx} between $N_t \times 2$ and dynamic mode, shrinks from 0.4W to 0.2W when data source increases from 5M to 165M.

Our first set of findings can be summarized as:

Finding 1 Regarding power consumption at the receiver,

1a. *Static SMPS always saves power from 0.5W to 1.15W at the receiver, over multiple active receive chains, in the Doze OFF mode. Its power-saving margin increases with increasing data source rate.*

1b. *Dynamic SMPS always saves power from 0.2W to 0.5W at the receiver over multiple active receive chains in the Doze OFF mode. Its power-saving margin* *may increase with decreasing data source rate.*

1c. *Static SMPS always saves power from 0.1W to 0.7W over dynamic SMPS in the Doze OFF mode. The reason is that dynamic mode can switch to a single receive chain only when idle, or when transmissions are diversity, single stream.*

6.4 "The Ugly": SMPS Goodput Performance

Unfortunately SMPS power savings do not come for free. Our case study reveals that the price for saving receive power is a significant decrease in speed. Specifically, 3x2 yields 61.8% goodput gains over $N_t \times 1$ settings and 22.6% over dynamic SMPS, at 165M source, as shown in Figure 6.3. We identify three main factors that affect goodput: a) *MIMO gains*, b) *signaling overhead*, c) *application data source rate*.

MIMO gains MIMO gains can be further classified as Spatial Multiplexing (SM) and Diversity gains, observed at high, low SNR scenarios respectively. SM can increase the rate of communication by sending multiple independent spatial streams from the multiple RF chains. Diversity improves the reliability of reception, by transmitting a single data stream from each chain [9].

Spatial multiplexing: *Static Spatial Multiplexing Power Save (SMPS) does not exploit Spatial Multiplexing MIMO gains.* Maintaining only one active receive chain in static SMPS, limits the transmitter to use only SS bit-rates, which can go up to 135Mbps, significantly lower than 300Mbps, which is our platform's highest DS rate. In our case study scenario, 3x2 transmits 100% of the total frames at DS rates, which results in 61.8% goodput gains over $N_t \times 1$ settings. Our experiments at various high SNR locations (SNR>30dB) and various transmit chain N_t configurations, reveal goodput gains from 14.1% to 61.8% of $N_t \times 2$ over $N_t \times 1$ settings, as shown in Table 6.1. Dynamic mode can still utilize spatial multiplexing, by preceding a DS

	Low SNR	Medium SNR	High SNR
3x2 over 3x1	up to 47.9%	up to 49%	up to 61.8%
2x2 over 2x1	up to 18.5%	up to 54.8%	up to 58.2%
1x2 over 1x1	up to $\times 3.8$	up to 11.3%	up to 14.1%
3x2 over Dyn.	up to 62.7%	up to 47.9%	up to 22.6%

Table 6.1: Spatial multiplexing and diversity goodput gains.

transmission with RTS.

A monotonic increase in goodput with the number of active chains, has been also verified theoretically. In spatial multiplexing mode and given perfect channel state information, capacity has been shown to grow linearly with $min(N_t, N_r)$ [13, 14]. Although the rate of growth may change for different SNRs, the linear relation between capacity and the number of chains still holds [15]. Without perfect channels or under data source rate constraints, there is a saturation point where, increasing the number of active chains does not boost capacity [16].

Diversity: *Static SMPS does not exploit receiver Diversity MIMO gains.* Maintaining only one active receive chain in static SMPS, decreases the reliability of reception. At low SNR settings (SNR$\leq$15dB) where diversity gains are maximized, two active receive chains give from 18.5% up to 3.8 times higher goodput compared with one receive chain, as shown in Table 6.1. In medium SNR range (15dB$<$SNR$\leq$30dB), goodput gains of $N_t \times 2$ over $N_t \times 1$ settings are mainly attributed to diversity as well and can go up to 54.8%. Diversity goodput gains of two over one active receive chain, for a representative medium SNR location P4 and low SNR location P7 of our floorplan, are presented in Figures 6.5, 6.8, respectively.

A monotonic increase in goodput with the number of active chains, is theoretically verified for diversity as well. In diversity mode, the error probability function can be

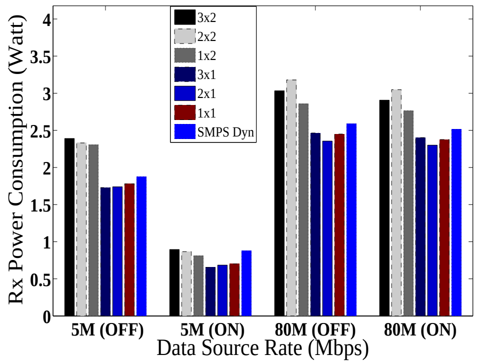


Figure 6.4: Receiver's power consumption (medium SNR location P4).

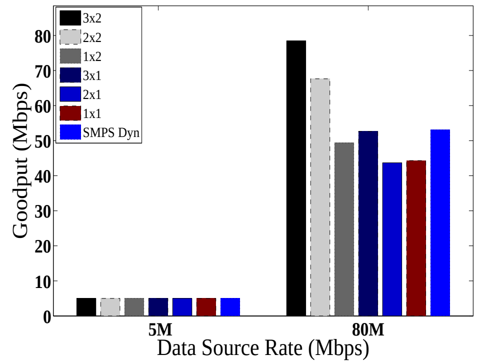


Figure 6.5: Goodput (medium SNR location P4).

expressed as $P_e = \frac{1}{SNR^{N_r \cdot N_t}}$ [9]. Then the goodput G is given by $G = R \cdot (1 - P_e) = R \cdot (1 - \frac{1}{SNR^{N_r \cdot N_t}})$, where R is the bit-rate and SNR is the signal-to-noise ratio. As the error probability P_e decays with the exponent of the diversity gain factor $N_r \cdot N_t$, goodput increases with the number of active chains.

Signaling overhead Dynamic SMPS is able to exploit spatial multiplexing and diversity MIMO gains by switching from one to many receive chains on a per-transmission basis, but at a high RTS/CTS overhead. For our case study scenario, 96.5% of the total frames transmitted at DS rates, need to be preceded by RTS. This results in 22.6% goodput gains of 3x2 over dynamic SMPS. Our experiments at various SNR locations and data source rates, show that 3x2 can achieve from 22.6% to 62.7% goodput gains over dynamic SMPS, as shown in Table 6.1.

Our simple analysis shows that RTS/CTS handshake is proven expensive, when it precedes every MIMO transmission. We model the transmission time of an 802.11n aggregate MPDU frame (A-MPDU) as $T_{tx} = T_{overhead} + \frac{MPDU \cdot A_R(t)}{R}$, where $T_{overhead}$ includes the various 802.11n protocol overheads (DIFS, SIFS, Preamble, PLCP, RTS/CTS, ACK) and R is the transmission rate. Aggregation level A_R is the number

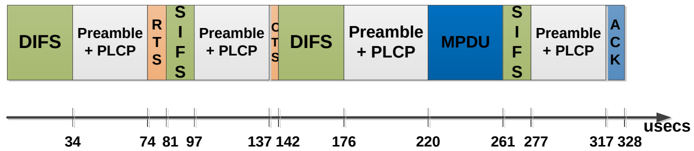


Figure 6.6: 802.11n RTS/CTS frame exchange.

of MPDUs packed in an A-MPDU. If we assume that there is no frame aggregation (A_R=1), R=300Mbps and MPDU is 1.5KBytes, we can observe from Figure 6.6, that 43.3% of the total transmission time is allocated for the RTS/CTS handshake. Even in the scenario of full frame aggregation where A-MPDU is 64KBytes, RTS/CTS overhead allocates 28.1% of the total transmission time.

Data source rate Our experiments have revealed significant goodput gains of $N_t \times 2$ over $N_t \times 1$ settings and dynamic SMPS. However, these gains are upper-bounded by the offered data source rate, as we observe in Figures 6.3, 6.5, 6.8, at low source rates.

Our second set of findings can be summarized as:

Finding 2 *Multiple active receive chains can give from 11.3% to 3.8 times higher goodput compared with SMPS, when data source rate does not upper-bound achieved goodput. These gains can be attributed to:*

2a. *Spatial multiplexing and diversity gains of multiple over one active receive chain in the static mode.*

2b. *RTS/CTS overhead, which dynamic mode needs to pay before every MIMO transmission.*

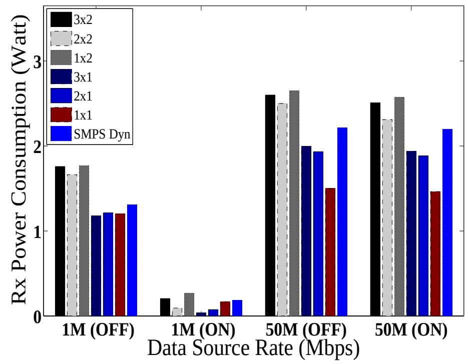


Figure 6.7: Receiver's power consumption (low SNR location P7).

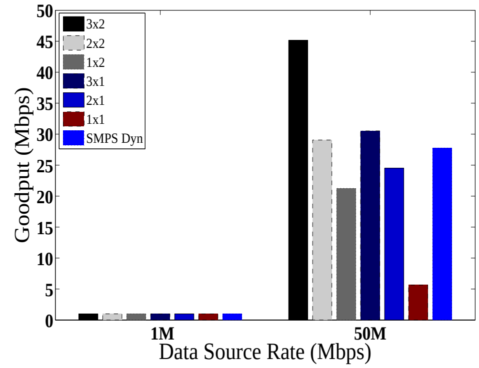


Figure 6.8: Goodput (low SNR location P7).

6.5 "The Bad": SMPS Potential Losses

We now shift our attention to two potential drawbacks of SMPS, which come from the interplay between power consumption and goodput. First, our study so far has revealed power savings of SMPS over multiple active receive chains in Doze OFF mode. However, are these savings observed in Doze ON as well? Second, our study has been focused on SMPS receive power consumption. However, is SMPS energy-efficient?

6.5.1 SMPS power consumption in Doze ON

Interestingly, our experiments reveal that SMPS may not save power, compared with multiple active receive chains, when the receiver operates in Doze ON mode.

Dynamic SMPS: For our case study scenario of Figure 6.2, dynamic SMPS consumes from 0.1W to 0.3W more power compared with the other chain configurations at 5M, Doze ON case. Our traces reveal a significant impact of RTS/CTS overhead on dynamic SMPS power consumption performance. Dynamic SMPS transmits 96.5% of the total frames at spatial multiplexing DS rates, which are preceded by RTS/CTS. This signaling overhead increases the transmission time of the same amount of data from 5.1% to 7.2% (Table 6.2) compared with the other chain configurations and as a result it decreases sleep time opportunities. During this active time, the receiver in dynamic SMPS maintains two active chains to receive DS frames, while the faster chain settings can save power by switching to Doze ON. Our case study result is verified in various settings, where dynamic SMPS can require up to 8% more time compared with other chain settings, to transmit the same amount of data. The impact of RTS/CTS overhead on idle time, is significant at low source rates. When the data source rate approaches or overcomes the effective goodput (e.g. at 165M of our case study), the idle time between dynamic SMPS and remaining configurations is almost the same.

Static SMPS: Although static SMPS can still save power compared with multiple active receive chains in Doze ON, its savings drop significantly. For example at location P7 (Figure 6.7), 2×2 consumes only 0.01W more power than 2×1 setting at 1M. The SMPS power savings drop at P7, because $N_t \times 2$ settings require up to 10% less time to transmit the same amount of data, compared with $N_t \times 1$ configurations.

Our third set of findings can be summarized as:

Finding 3 On power consumption at the receiver,

3a. *Static SMPS power savings can drop to 0.01W compared with multiple active receive chains, in the Doze ON mode. Receiving with a single chain, results in 10% less sleep time opportunities of static SMPS over multiple active receive chains.*

3b. *Dynamic SMPS can consume 0.3W more power, compared with multiple active receive chains in the Doze ON mode. RTS/CTS overhead required prior to a MIMO transmission, results in 8% less sleep time opportunities of dynamic*

	3x2	2x2	1x2	3x1	2x1	1x1	Dyn.
Idle Time	96.6%	95.1%	94.8%	94.5%	95.2%	95.3%	89.4%

Table 6.2: Idle time for 5Mbps source rate, at location P2.

SMPS over multiple active receive chains.

6.5.2 SMPS energy consumption

Our experiments show that saving power does not necessarily result in saving per-bit energy E_{rx} formulated as:

$$E_{rx} = \frac{P_{rx}}{G} \tag{6.3}$$

In our case study setting, although two active receive chains are more power hungry compared with one active chain (Figure 6.2), they yield the lowest per-bit energy consumption at 165M, as indicated by the text arrows in Figure 6.9. Specifically, 3x2 yields energy savings defined as the decrease in per-bit energy consumption, from 12.8% to 24% over static SMPS ($N_t \times 1$ setting) and from 11.3% to 15.6% over dynamic SMPS. The savings can be attributed to the goodput gains of 3x2 over static (61.8% gains), and dynamic (22.6% gains) SMPS, which compensate for its additional power consumption.

By studying the interplay between power consumption and goodput, we end up with two interesting conclusions. *First, the fastest RF chain setting may not be the most energy efficient.* In the scenarios where source rate can be accommodated by a single receive chain, $N_t \times 1$ settings are more energy-efficient than the faster $N_t \times 2$ configurations. This is observed for source rates 5M or smaller at locations P2, P4, P7. Dynamic SMPS can still give higher power consumption in Doze ON and as a result higher energy consumption performance, at low source rates, compared with static

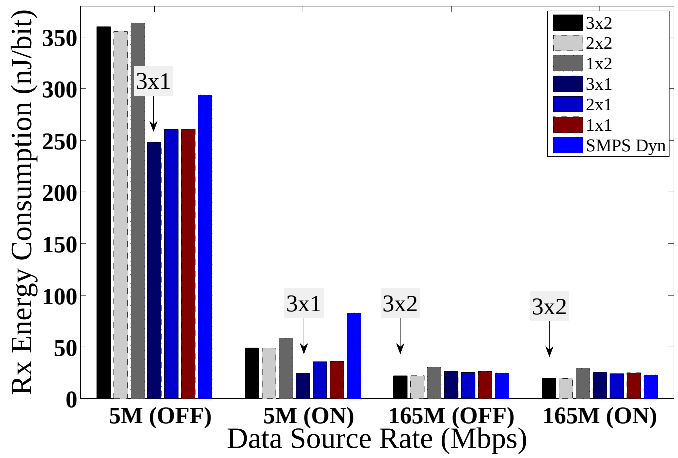


Figure 6.9: Receiver's energy consumption (high SNR location P2).

SMPS and multiple active receive chains. *However, the most power hungry RF chain setting may not be the least energy efficient.* When source rate does not limit achieved goodput of multiple active receive chains, $N_t \times 2$ settings are energy optimal as shown in Figures 6.9, 6.10, 6.11. In these scenarios, 3x2 can give from 12% to 78.6% energy savings over static and dynamic SMPS.

Our experiments uncover important implementation implications on SMPS performance. For a fixed number of receive chains N_r, goodput monotonically increases with the number of transmit chains N_t, as well. Activating three chains at the transmitter, can yield up to 5.4 times higher goodput comparing to one active transmit chain. This goodput gain observed for 3x1 over 1x1 at location P7, can significantly affect the performance of SMPS as it results in 75.5% energy savings as shown in Figure 6.11.

Our fourth set of findings can be summarized as:

Finding 4 *Saving power does not necessarily result in saving energy. Multiple active receive chains can give from 12% to 78.6% per-bit energy savings over SMPS. This is observed when the offered data source rate is equal or higher than the maximum*

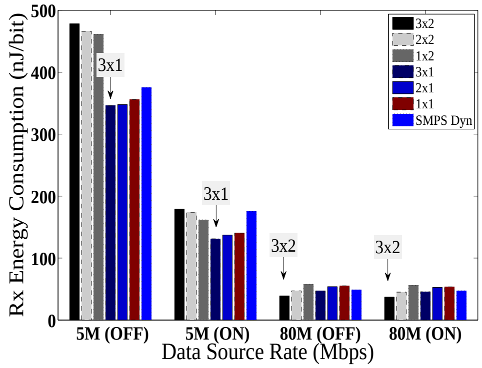


Figure 6.10: Receiver's energy consumption (medium SNR location P4).

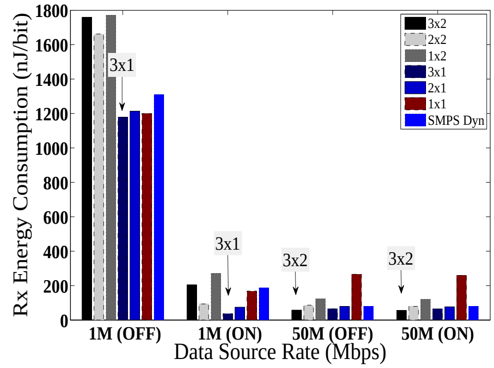


Figure 6.11: Receiver's energy consumption (low SNR location P7).

achievable goodput of multiple receive chains. SMPS needs to consider both consumed power and achieved goodput to save energy.

6.6 Design

In this section, we present MIMO Receiver Energy Save (MRES) scheme, which seeks to identify and set the most energy efficient chain setting for the receiver at runtime. A critical design challenge is to converge to the receiver's most energy efficient setting with small sampling overhead. MRES devises a novel, low-overhead sampling scheme, which improves over exhaustive sampling all possible chains, in Doze OFF mode. It opportunistically evaluates the receiver chain options and excludes those chain configurations that are highly unlikely to yield energy savings. We next describe MRES operations.

6.6.1 MIMO Receiver Energy Save sampling

Traffic-driven sampling MIMO Receiver Energy Save main component is a low-overhead sampling scheme. Its main design principal is that *the most energy efficient is the lowest chain setting, which can accommodate the offered source rate, in Doze OFF*. It derives from Finding 1, which shows a monotonic increase in power consumption with the number of receive chains N_r in Doze OFF, given a fixed number of transmit active chains N_t. So MRES traffic-driven sampling *sequentially* samples upward (higher number of active chains), starting from the lowest chain setting. It terminates sampling when a chain's moving-average achieved throughput $\overline{Thr_{chain}}$ is the same as the moving-average source rate $\overline{srcRate}$ ($\overline{Thr_{chain}} \geq \alpha \cdot \overline{srcRate}$) The smoothing factor α is set to 0.95 in our prototype. The pseudo-code of our scheme is presented in Procedure 6. MRES scheme needs to address two important issues: a) When is sampling triggered? b) How long will sampling last and how will its outcome be evaluated?

Sampling triggers MRES triggers sampling and subsequent chain evaluation, using both time- and event-driven mechanisms. To prevent high overhead from switching chains on a per-transmission basis (Findings 2b, 3b), it samples periodically (3 seconds in our prototype) to identify the best-energy chain. To be adaptive to MIMO channel and data source rate dynamics, MRES triggers sampling whenever it observes significant change in the measured throughput of the current chain. Specifically, it triggers sampling when $Thr_{chain}(t) \leq \overline{Thr_{chain}}(t) - 2 \cdot \sigma_{chain}(t)$ or $Thr_{chain}(t) \geq \overline{Thr_{chain}}(t) + 2 \cdot \sigma_{chain}(t)$. $\overline{Thr_{chain}}$, Thr_{chain} are the moving-average and current achieved throughput at time t respectively, while $\sigma_{chain}(t)$ is the throughput standard deviation. Event-driven sampling is proven critical in dynamic traffic scenarios (e.g. VoIP, bursty web traffic) to reduce idle energy consumption.

Sampling should be long enough for RA to first identify the best rate (T_{RA} mil-

Procedure 6 MRES: Input (chain, doze), Output (best_chain)

// Update stats upon the reception of a BlockACK frame
update-stats(Thr_{chain}, $srcRate$, $chain$);

if (event-triggers($\overline{Thr_{chain}}$, Thr_{chain}, σ_{chain})
|| sample-timer-expired()) && is_sample = false **then**
    chain = lowest-chain();
    init-sample-period(T_P);
    is_sample $\leftarrow$ true;
end if

if is_sample && sample-period-ended() **then**
    (best_chain) $\leftarrow$ best-energy-chain(best_chain, chain);
    if ($\overline{Thr_{chain}} \geq \alpha \cdot \overline{srcRate}$ && doze=OFF)
    || chain = highest-chain() **then**
        is_sample $\leftarrow$ false;
        sample-timer-reset();
    else
        (chain) $\leftarrow$ next-higher-chain(chain);
        init-sample-period(T_P);
    end if
end if

return best_chain;

liseconds) and then to evaluate its performance (T_E milliseconds). It should be also short enough to limit transmissions at high-energy chain settings. MRES sets its sampling period $T_P = T_{RA} + T_E$, where T_{RA} is RA algorithm dependent. It also updates the measured throughput and source rate of a given chain setting as $\overline{Thr_{chain}} = \frac{3}{4} \cdot \overline{Thr_{chain}} + \frac{1}{4} \cdot Thr_{chain}$ and $\overline{srcRate} = \frac{3}{4} \cdot \overline{srcRate} + \frac{1}{4} \cdot srcRate$ every 20ms. When the best rate is reached, our prototype uses 6 samples to update the moving averages and sets T_E to $120ms$.

Metric MRES estimates the per-bit energy consumption of a chain setting using

Equation (6.3). Instead of goodput G, it uses measured throughput $\overline{Thr_{chain}}$ at the sender. Finally, the chain setting with minimum E_{rx} is selected for transmission.

Sampling cost reduction MRES limits sampling cost by preventing transmissions at high-energy chains. Sampling cost is proportional to the sampling time at energy sub-optimal chain settings which is expressed as $T_{sp} = T_{RA} + T_E + 2 \cdot T_{comm} + T_{ant}$. The time to identify the best rate T_{RA} is RA specific. For example, RRAA [34] evaluates every rate option for approximately 15ms. So in the worst case scenario under a stable wireless channel, $T_{RA} = 255ms$ given that all the available rate options of our platform are 17 for 40MHz channel bandwidth. The total sampling period is then $T_P = T_{RA} + T_E = 375ms$. After MRES identifies that the sampled chain is not the most energy efficient one, it requires T_{ant} time until the receiver hardware switches to the optimal receive chain (35usecs in our system) and T_{comm} time for each of MRES handshake messages in order to commit the new setting. In a ideal scenario with no interfering traffic, $T_{comm} = 59.7usecs$, given that MRES management frame size is 360bytes and is transmitted at 24Mbps in our platform. So sampling cost is $375.2ms$ for each energy sub-optimal sampled receive chain. In the scenario where the optimal is the lowest receive chain, MRES can exclude $N_r - 1$ energy-sub-opimal chains from sampling. Without MRES low-overhead sampling, MRES would transmit up to 37.5% of the total time at energy sub-optimal chains, given that the RA is set to RRAA, $N_r = 4$ and sampling interval is 3 seconds.

Traffic-driven Sampling in Doze ON Power consumption monotonic relationship with increasing number of receive chains N_r, may not hold in Doze ON. Let's assume $T_{tx,i}$ is the transmission time of M bits when i receive chains are active. From our analysis and experiments discussed in Section 6.3, we formulate $T_{tx,i} = T_{tx,i+1} + T_{idle}$, where T_{idle} is the idle time of the higher chain $i+1$ upon the completion of its transmission. The per-bit energy consumption for i, $i+1$ receive chains is $E_{rx,i} = \frac{E_p + T_{tx,i} \cdot P_{c,i}}{M}$

and $E_{rx,i+1} = \frac{E_p + T_{tx,i+1} \cdot P_{c,i+1} + T_{idle,i+1} \cdot P_{dozeON}}{M}$ respectively. E_p is the processing energy consumption, which is assumed to be similar for i, $i+1$ settings, given that the amount of bits M to be processed is the same. P_{dozeON} is the power consumption is Doze ON mode, which for simplicity is considered negligible. Our proposed low-overhead sampling holds in Doze ON for chains i, $i+1$ that can accommodate the offered source rate, only if $E_{rx,i} \leq E_{rx,i+1} \Rightarrow P_{c,i+1} \geq \frac{T_{tx,i}}{T_{tx,i+1}} P_{c,i}$. Although the relation $P_{c,i+1} > P_{c,i}$ is known in advance [64], transmission time $T_{tx,i+1}$ depends on rate R and aggregation level A_R (Section 6.4) which may be different between chain i and $i+1$. To ensure that the energy optimal chain setting will be identified, MRES takes a conservative approach and disables traffic-driven sampling in Doze ON.

6.6.2 MIMO Receiver Energy Save mechanism

MRES introduces a new management frame, as neither the SMPS action frame nor the RTS/CTS of SMPS modes can be used without modifications. First, they have not been designed to support chain setting exchange information. Second, they do not communicate power consumption, which is necessary information for computing energy consumption. To address these issues we propose a new management action frame presented in Figure 6.12. The *Energy Save Enabled* bit is set to 1 to enable the energy save mechanism. Using *Available Chains* and *Active Chains* bits, the receiver informs the transmitter for the number of its available and currently active chains. *Chain Feedback* bits are only set by the transmitter to activate the appropriate number of receive chains. Two bits can accommodate four spatial streams available in 802.11n. Finally, the optional *Power* field PW, is used to communicate receiver power consumption information. It is a 15 bit unsigned integer, which represents the power consumption of a single active chain in milliwatts. PWI_1, PWI_2, PWI_3, are 11 bit unsigned integers, which represent the additional power consumption in milliwatts of 2, 3, 4 active

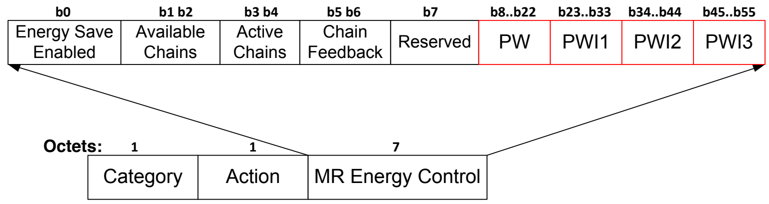


Figure 6.12: MRES frame format.

receive chains over 1, 2 ,3 chains respectively. For example the power consumption of N_r active receive chains is calculated as $PW + \sum_{j=1}^{N_r-1} PWI_j$. The difference in power consumption between adjacent chain settings does not exceed 1.15W in our experiments, and can be represented by 11 bits. If PW field is not used, transmitter needs to estimate receiver chains' power consumption.

When the transmitter receives a MRES action frame, it sets receiver's energy save status, active, available chains and power consumption information if available, while it ignores *Chain Feedback*. The MRES frame sent by the receiver, does not require any response. When transmitter requires from the receiver to switch chains, it sends a MRES action frame with the *Chain Feedback* bits set to the selected chain setting. Upon the reception of the MRES frame, the receiver commits the new chain setting and it forms a new MRES frame with all but *Chain Feedback* field set to the new values. Only when the transmitter receives the MRES response, it commits the new receiver's chain setting.

6.7 Implementation and Evaluation

In this section, we first describe the implementation of MRES. Then, we compare it with static, dynamic SMPS and our system's 3x2 default configuration, using both real experiments and trace-driven simulations.

6.7.1 Implementation

We implement MRES in approximately 400 and 200 lines of code on the transmitter, receiver side, respectively. Due to hardware constraints to support the Doze ON, we only evaluate the OFF mode in our experiments. An issue to overcome is the estimation of the data source rate, which can accurately be measured only when it does not exceed the effective throughput. In the case where source rate is higher than the effective throughput, MRES checks for buffer overflows. Buffer overflow implies that source rate cannot be accommodated by the current chain setting.

Besides our proposed traffic-driven sampling, we also apply an adaptive sampling scheme, which seeks to eliminate chain settings that consistently incur high energy consumption. Our scheme keeps a separate timer for the two available receive chains of our testbed. MRES samples and updates the energy consumption of a given setting only after its timer expires[2]. After sampling a setting yields higher energy consumption than the current best one, its timer is exponentially increased. MRES prevents a chain setting from being completely excluded by a) upper bounding the timer to 8 seconds, b) resetting the timer when sampling a chain setting results in lower energy consumption than the current lowest one.

6.7.2 Performance evaluation

We now compare MRES with SMPS implemented as described in Section 6.1.2 and with our system's default 3x2 configuration. We first conduct experiments with one transmitter and one receiver, in the campus setting of Figure 6.1. We evaluate the proposed solutions in terms of receiver per-bit energy consumption, in static, mobility scenarios, with various 802.11n configurations and different RAs, with low, high

[2]Timers are considered only for time- and not event-driven sampling.

	Static SMPS	Dynamic SMPS	3x2
Static UDP	(1-36.8)%	(0.7-32)%	(0.4-34.2)%
Static TCP	(10.1-11.7)%	(9.7-21.3)%	(11.3-20.8)%
Mobility	14.4%	9.1%	14.9%
Simulation	up to 12.2%	(15-60.5)%	(7.4-35.4)%

Table 6.3: Energy savings of MRES over alternative designs.

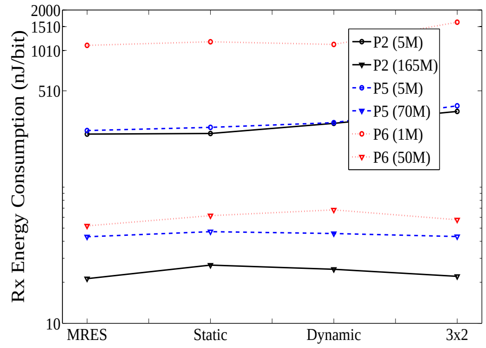
volume UDP and TCP traffic. The experimental results show that MRES consistently outperforms alternative solutions in all scenarios, with energy savings from 0.7% to 36.8% and from 0.4% to 34.2% over SMPS and 3x2 configurations, respectively. It also offers goodput gains up to 67.5% in all the examined scenarios over static mode and goodput gains up to 37.6% in 70% of the tested scenarios over dynamic mode. Finally, MRES consumes from 0.02W to 0.6W less power in 83.3% of the tested scenarios over dynamic mode. It also never consumes more than 0.15W compared with static SMPS in 95% of the examined scenarios.

We also run simulations for two reasons. First, they allow for us to compare the designs in larger network topology. Second, they enable us to assess the Doze ON mode, which is not available in our platforms. Simulation results show up to 60.5% energy saving of MRES over SMPS in both infrastructure and ad-hoc network scenarios. The MRES energy savings are summarized in Table 6.3.

6.7.2.1 Static clients

We first evaluate MRES for static clients, over both interference-free 5GHz channels verified by our sniffer and the highly congested 2.4GHz band. The channel bandwidth is set to 40MHz and rate adaptation to MiRA in all experiments unless explicitly stated.

UDP/5GHz case Figure 6.13 plots the per-bit energy measured at high-, medium-



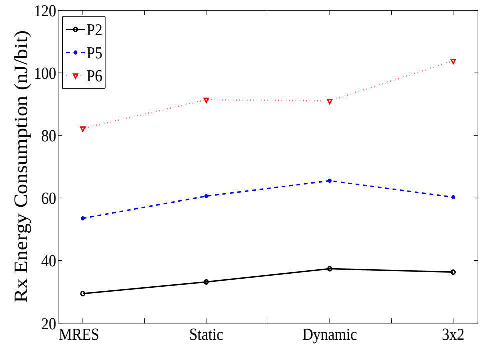


Figure 6.13: Receiver's energy consumption (UDP/5GHz - Log-scale).

Figure 6.14: Receiver's energy consumption (TCP/5GHz).

and low-SNR locations (marked in Figure 6.1), over the 5GHz band and for high and low UDP traffic sources. MRES consistently outperforms alternative algorithms, with energy savings up to 36.8% over static SMPS, 32% over dynamic SMPS and 34.2% over 3x2. Its savings come from its ability to identify the most energy-efficient chain setting for the receiver at low sampling cost.

Figure 6.15 plots the chain distribution along with the receiver power consumption and goodput for locations P2, P5, P6. For our case study location P2, we observe that MRES gives close to optimal distribution, by transmitting almost 100% of its frames at 3x1, 3x2 settings, for the low- (5M), high- (165M) volume UDP sources respectively. For locations P5, P6, MRES selects the average energy optimal 3x1 setting, for the low volume UDP traffic. Under higher traffic volume and intense MIMO channel dynamics observed usually at low SNRs, MRES can switch between one and two active receive chains. MRES is able to identify the most energy efficient chain setting, with low sampling overhead. It gives from 10.6% to 59.6% goodput gains over static SMPS in the examined locations, while it outperforms dynamic SMPS at P2, P6, as well. The goodput gain of dynamic SMPS over MRES at location P5, is attributed to the fact that MRES selects 3x1 for 92.7% of its transmissions and not to its sampling cost.

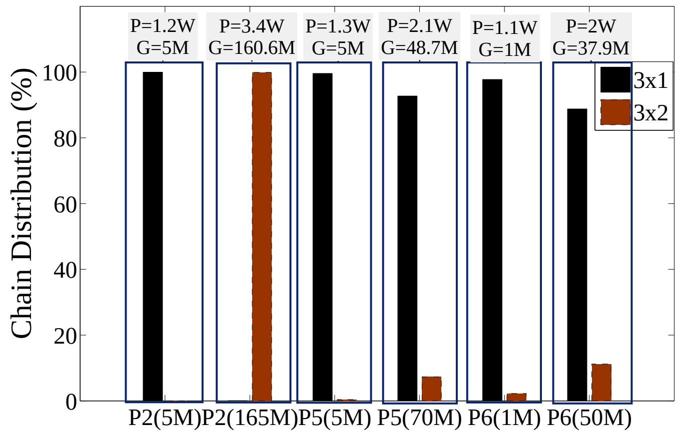


Figure 6.15: MRES chain distribution.

TCP/5GHz case We also conduct experiments with four TCP flows. Figure 6.14 shows that, MRES produces energy savings up to 11.7% over static SMPS, up to 21.3% over dynamic SMPS, and up to 20.8% over 3x2.

UDP/2.4GHz case We then switch to the congested 2.4GHz band (channel 11), where we sniff more than 20 APs on channels 1 to 11. We change channel width to 20MHz to mitigate interference caused by overlapping 40MHz channels [61]. The per-bit energy consumption of different algorithms for locations P2 and P4 is presented in Figure 6.16. The higher per-bit energy consumption compared with the 5GHz settings can be attributed to lower goodput, which does not exceed 54.7Mbps. MRES still outperforms SMPS and 3x2 designs with savings up to 36.8% and 29.4% respectively.

Impact of rate adaptation We finally evaluate the various strategies using both legacy 802.11a/b/g RAs (RRAA [34], SampleRate [23]) and MIMO 802.11n RAs (MiRA, Atheros MIMO RA [32]), which we have prototyped on our testbed. Figure 6.17 plots per-bit energy at the medium SNR location P3 using 90Mbps UDP source. MRES consistently outperforms SMPS and 3x2 with savings up to 32% and 12.8%

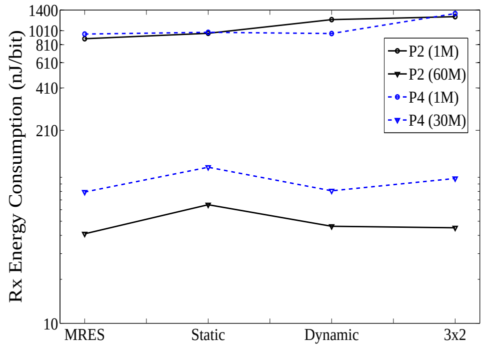


Figure 6.16: Receiver's energy consumption (UDP/2.4GHz - Log-scale).

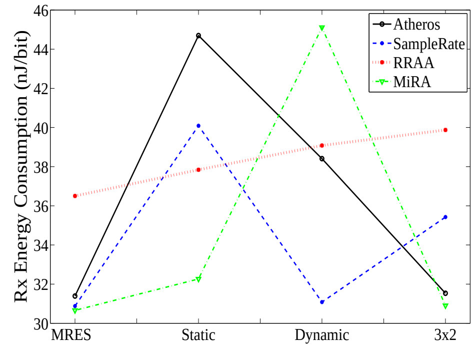
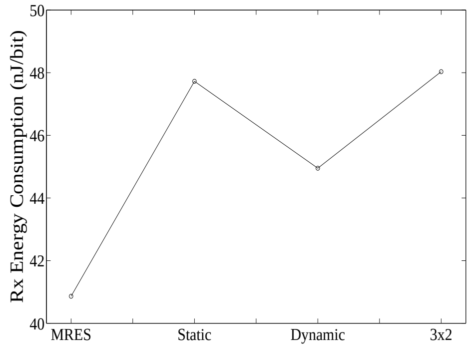


Figure 6.17: Receiver's energy consumption. MRES over various RAs.

respectively, independently of the underlying RA scheme. Our traces reveal that chain distributions and as a result receiver power consumption for MRES, are almost the same for all RA algorithms. What varies among the tested RAs, is the rate distribution and as a result the goodput. The maximum energy savings of MRES over static and dynamic SMPS are observed over Atheros (29.8%) and MiRA (32%) respectively.

6.7.2.2 Mobile clients

To gauge the responsiveness of MRES upon MIMO channel dynamics, we carry a client and walk from P1 to P7 through P3, P5 and then come back at approximately constant, pedestrian speed of 1m/s. Figure 6.18 plots the per-bit energy consumption of our four schemes using 100Mbps UDP source. MRES offers 14.4%, 9.1% of energy savings over static, dynamic SMPS respectively and 14.9% over 3x2 configuration. Our event-driven sampling is fairly responsive to our pedestrian mobility scenaro, without incurring high sampling overhead or low goodput. Characteristically, 3x2 gives only 7.9% goodput gains over MRES, which cannot offset 3x2 setting's higher power consumption.



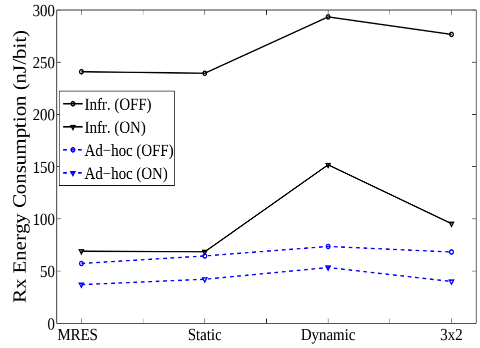


Figure 6.18: Receiver's energy consumption under mobility.

Figure 6.19: Network's energy consumption. Infrastructure and ad-hoc settings.

6.7.2.3 Trace-driven simulations

We next use trace-driven simulations to assess MRES in larger infrastructure and ad-hoc networks. We collect real channel and power consumption traces, by placing the AP at T but moving the client across multiple locations in the campus setting of Figure 6.1. For each location, we measure the goodput, frame loss, aggregation, SNR and power consumption. To extend our simulation to three receive chains, we estimate a) power consumption of three chains based on the difference between power consumption of two and one chain, b) goodput to be similar to 3x2 setting. We test various traffic volume scenarios.

We feed the traces to a customized 802.11a/g/n simulator written in C++. In the infrastructure setting, the AP is located at T, while clients are randomly deployed in our campus setting. We vary the number of clients from 9 to 15. Figure 6.19 plots the per-bit energy for a 9-client topology and for both Doze ON and OFF modes. The network energy consumption is calculated based on the total power consumption of all nodes and the network's aggregate goodput. MRES performs similar to static SMPS, while it outperforms dynamic SMPS and 3x2 with energy savings 54.9% and 28.2% respectively. In the ad-hoc scenario, we randomly deploy 50 nodes in a 1000m x 1000m area. We vary the number of traffic flows from 10 to 30 among randomly selected transmitter and receiver pairs. To emulate the MIMO channel using our traces, we map the distance between two communicating nodes with an SNR value, corresponding to a given goodput, frame loss and aggregation performance. Figure 6.19 plots the network's per-bit energy for a 10-flow setting. MRES outperforms SMPS and 3x2 with energy savings up to 30.6% and 16.1%, respectively.

6.8 Related Work

Energy efficient algorithms have been widely studied in the legacy 802.11 wireless networks [43–49, 52]. However, the problem remains largely unexplored in the MIMO 802.11n systems. Recent proposals (SMPS [6], Snooze [58]) apply antenna selection to save energy at 802.11n receivers. SMPS seeks to save power consumed in MIMO circuit blocks, by switching from "many" to a "single" antenna setting. Snooze [58] switches antenna settings according to MIMO speed (airtime utilization). However, our study shows that, RF chain selection solely based on MIMO speed, or power consumption can lead to energy sub-optimal chain setting. MRES departs from these proposals by considering both speed and power in chain management.

There have been several theoretical studies focused on energy-efficient MIMO systems [64–67]. They seek to find a theoretical transition point, where the most energy-efficient MIMO setting changes. The crossover point can be expressed as the tradeoff between MIMO gains, which come at the cost of increased power consumption. While [64, 66] focus on the system's energy consumption, [65] considers uplink energy-efficient transmissions. Different from these efforts, we focus on experimental studies, while proposing new energy save solutions for the 802.11n receivers.

Early experimental work on identifying factors that affect 802.11n energy consumption on commodity hardware has been reported in [57]. Different from our study, the authors do not consider the impact of data source rate and Doze OFF, ON modes in their per-bit energy and power consumption measurements. They do not propose new designs as well.

6.9 Summary

In this chapter, we discuss the tradeoff between MIMO power consumption and speed, by presenting a critique on the newly proposed 802.11n Spatial Multiplexing Power Save feature. Our experiments with standard-compliant 802.11n devices uncover two important insights. First, the fastest RF chain setting may be the least energy efficient. Second, the most power hungry RF chain setting may be the most energy efficient. To this end, we propose a MIMO receiver energy saving scheme, which seeks to identify the energy optimal antenna setting at a low cost. We compare MRES with two design philosophies. The first seeks to increase performance by turning all the antennas on. The second philosophy switches from “many” to a “single” antenna setting to save MIMO power consumption (SMPS). MRES gives 37% energy savings over both philosophies for a 2-antenna receiver.

CHAPTER 7

Conclusion

Multiple-Input Multiple-Output (MIMO) offers significant promise in making Gbps wireless links a reality. However, our experimental study with MIMO 802.11n commodity devices reveals that, the current MIMO is low speed and energy hungry. The root cause is the use of legacy (single antenna) designs over the new MIMO (multiple antenna) setting. This dissertation advocates for the need of novel designs over the MIMO setting, by illustrating the MIMO unique characteristics and their impact on current network speed and energy performance. We conclude this dissertation by summarizing the key insights learned from our study, and by examining the remaining challenges.

7.1 Lessons Learned and Departures

The current MIMO 802.11 wireless networks are designed using the legacy 802.11a/b/g wireless networks as the blueprint. The existing designs abstract the wireless channel as a 2-dimensional (frequency and time) communication link, and graft the legacy wireless protocols onto the new MIMO setting. As our study shows, MIMO wireless channel is fundamentally different from legacy. To this end, this dissertation seeks to answer two questions. a) Why should we consider novel designs over the MIMO setting? b) What are some of these novel design ideas?

The insights gained using experiments with commodity 802.11 devices can be summarized as follows.

- **MIMO modes exhibit different characteristics** Our experimental study with commodity MIMO 802.11n devices uncovers that the diversity and spatial multiplexing MIMO modes exhibit different loss characteristics. Different from legacy, loss is not monotonic with respect to transmission rate over the MIMO setting. State of the art rate adaptation solutions are designed though by assuming loss monotonicity, and as a result they have limitations to identify the best goodput transmission rate.

- **MIMO speed is expensive** Our experimental study with commodity MIMO 802.11n devices uncovers that MIMO speed comes at a cost of increased power consumption. MIMO power consumption is proportional to the number of active antennas. Designs that either seek to boost performance by activating all the antennas, or to reduce MIMO power consumption by turning off all but one antenna (SMPS) can be energy sub-optimal.

- **802.11 channel exhibits rich dynamics** Our experimental study with commodity 802.11 devices reveals intense short-term channel dynamics for both legacy and MIMO settings. These dynamics are attributed to multipath fading and interference from neighboring nodes. The additional degrees of freedom of a MIMO 802.11n radio make these dynamics even more intense. Varying the number of active antennas and spatial streams can drastically change MIMO gains. Changing the channel bandwidth has a double impact on speed. Wider channels allow for higher PHY transmission rates, while at the same time, they can increase interference. 802.11 channel dynamics can also interact in a complex manner with the new 802.11n standard features, as frame aggregation. Existing designs have limitations to address 802.11 channel dynamics, which results in significant performance degradation.

Based on our findings we revise rate adaptation and energy save, which lay the foundations for gigabit and green wireless. Our proposals depart from state of the art designs in the following key ways.

- **Differentiate MIMO modes** The fundamental departure of our MIMO rate adaptation proposal is that, it manages diversity and spatial multiplexing modes in a distinct manner. First, it classifies the transmission rates in different MIMO modes based on the number of streams. Then, by applying sequential and zigzag probing, it can identify the optimal (best goodput) rate in each individual mode and the optimal mode, respectively. This allows for MiRA to overcome loss non-monotonicity observed in cross MIMO mode rates.

- **Consider new metrics** The focus of MIMO is higher speed over wireless. However is speed the right metric? Our study shows that, MIMO can lead to poor user experience because its higher speed comes at a cost of additional power consumption, proportional to the number of antennas. Our MIMO receiver energy save proposal considers both the MIMO speed and power consumption of a MIMO 802.11 device. It seeks to improve user experience by selecting the antenna setting with the lowest per-bit energy consumption.

- **Learn from history** Our proposed designs seek to avoid selecting sub-optimal settings by analyzing the short-term network performance. This involves a) learning and b) differentiating events. Both rate adaptation and energy save use adaptive channel learning to avoid transmissions at low performance settings. Learning requires classification and independent management of different events. For example, our rate adaptation designs seek to differentiate channel fading from interference losses, and appropriately enable RTS/CTS. Our rate adaptation and energy save proposals classify goodput variations to trigger the selection of a new setting.

This dissertation integrates the above ideas into practical, 802.11 standard-compliant solutions. Specifically, it integrates rate adaptation and energy save within the current device drivers and provides prototype implementations of the proposed designs. It also presents testbed evaluations that show large performance gains in comparison with the state of the art algorithms. Thus, we believe that this work is a significant step towards the future gigabit and green wireless network.

7.2 Open Issues and Future Work

The systems presented in this dissertation are a significant first step towards gigabit and green wireless networks. Our experimental findings uncover also novel problems, whose solution is of major importance for the future high speed, energy efficient, secure networks. We next elaborate on these new directions.

Rate adaptation Our MIMO rate adaptation proposal is able to identify the best goodput rate at a low probing cost by applying a prioritized adaptive probing scheme. First, sequential probing allows for MiRA to identify the best goodput rate R across a single MIMO mode, and prunes all the remaining rate options above R . Zigzag RA prevents probing the rates, whose loss-free goodput is smaller than the current best goodput performance. Finally, adaptive probing interval prevents MiRA from transmitting at rates, which continuously offer low performance. However, probing may still have limitations to adapt to very fast channel dynamics (e.g. vehicular mobility scenarios), where the channel can change in microsecond scales [24]. In our future work, we plan to examine rate adaptation that utilizes fast MCS feedback provided by IEEE 802.11n and 802.11ac standards. The key challenge across this direction is to overcome a) interference, b) multipath, c) hardware calibrations, which can poisson the SNR measurements (Section 5.6.3).

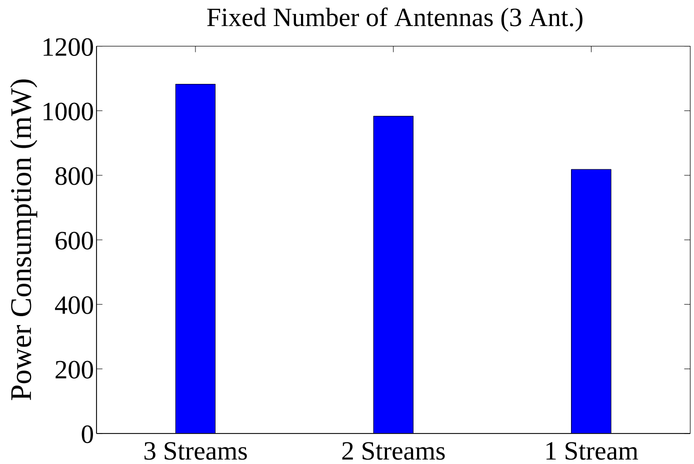


Figure 7.1: Receiver's energy consumption as a function of spatial streams.

Energy save Our MIMO Receiver Energy Save design seeks to save energy at the client side by selecting the energy optimal receiver antenna setting. However, our experiments reveal that saving energy at the receiver side calls for collaboration between the transmitter and the receiver. From Figures 6.9, 6.10, 6.11 we observe that the energy optimal antenna setting for the receiver always requires 3 active antennas at the transmitter side. However, activating all the transmit antennas can result in high transmit energy consumption. As a future work, we examine energy save solutions for both infrastructure and mobile device sides. To achieve that, we need to select the energy optimal system (transmitter and receiver) antenna setting.

Putting speed and energy together In this dissertation we design rate adaptation and energy save as two independent MAC-layer components. Our MIMO energy save solution can work independently of the underlying rate adaptation algorithm and vice versa. However, our experiments with commodity 802.11n hardware reveal that rate adaptation has a profound impact on energy consumption. Figure 7.1 shows a monotonic increase in receiver power consumption with the number of spatial streams, for a fixed number of antennas. As a result, rate adaptation that seeks to maximize speed may lead to high energy consumption. Designing joint rate adaptation and energy save algorithms is part of our future work.

7.2.1 From single-user to system view

The focus of this dissertation is to optimize speed and energy for the single-user case (single transmitter-receiver scenario). Our ongoing and future work seeks to extend our findings to the Multi-user (MU) MIMO case, which allows a terminal to transmit (or receive) signal to (or from) multiple users in the same band, simultaneously. The problem of rate adaptation in multi-user case is significantly different for two reasons. First, MU-MIMO requires channel state information at the transmitter (CSIT). CSIT while not-essential in SU-MIMO channel, is of critical importance to most downlink multi-user precoding techniques. Acquiring and utilizing timely CSIT feedback in a practical setting remains an open problem. Second, MU-MIMO allows for spatial sharing of the channel by many users. The scheduling procedure associated with the selection of a group of users that will be served simultaneously, adds more complexity to the problem.

7.2.2 Beyond the gigabit radio

The key insight gained from this dissertation is that, we need to depart from the simplistic view of MIMO as a pure physical layer technology. To this end, we design MIMO rate adaptation and energy save, which seek to optimize speed and energy at the MAC-layer. In our future work, we are planning to go beyond the MAC and examine how we need to revise the upper protocol layers. Specifically, we seek to answer two questions. What should routing look like over MIMO? Is MIMO gigabit network secure?

The tradeoffs between diversity (transmission range) and spatial multiplexing (speed) modes, and the speed impact on power consumption can fundamentally change the way that we forward packets and design routing metrics. Although, spatial multiplexing can provide the fastest links from a source to a destination node, diversity can be used along with opportunistic routing [76] and networking coding [77] architectures to allow for reliable and high performance end-to-end communication. Moreover, the fastest MIMO routing path may not be the most energy efficient one, because of additional MIMO power consumption. Designing green routing is still an open challenge.

MIMO multipath environment amplifies existing security vulnerabilities by generating a richer link signature space. Signal from authorized devices will spill over longer distances from devices' domain, allowing for malicious parties to overhear and interfere with existing transmissions. In the next generation wireless, the norms in network security for Authentication, Authorization, Confidentiality, Integrity need to be revised.

References

[1] The World in 2011. ICT Facts and Figures. *http://www.itu.int/*.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies. The Case for VM-based Cloudlets in Mobile Computing. *IEEE Pervasive Computing, 2009.*

[3] Federal Communications Commission. Mobile Broadband: The Benefits of Additional Spectrum. *OBI Technical Paper No. 6, October 2010.*

[4] IEEE P802.11ad/D0.1: Enhancements for Very High Throughput in the 60GHz Band. Draft 0.1, June 2010.

[5] 3GPP TR 25.913. Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN). *http://www.3gpp.org/*.

[6] IEEE P802.11n, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Enhancements for Higher Throughput. 2009.

[7] IEEE P802.11ac, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Enhancements for Very High Throughput for Operation in Bands below 6 GHz. 2011.

[8] A. J. Paulraj, D. A. Gore, R. U. Nabar and H. Bolcskei. An Overview of MIMO Communications - A Key to Gigabit Wireless. *Proc. IEEE*, 92(2), 2004.

[9] L. Zheng and D. Tse. Diversity and Multiplexing: A Fundamental Tradeoff in Multiple Antenna Channels. *IEEE Trans. Info. Theory*, 49(5), 2003.

[10] R. W. Heath and J. Paulraj. Switching Between Diversity and Multiplexing in MIMO systems. *IEEE Trans. Commun.*, 53(6), 2005.

[11] M. Lampe, T. Giebel, H. Rohling, W. Zirwas. PER-Prediction for PHY Mode Selection in OFDM Communication Systems. *IEEE Globecom 2003*.

[12] David Tse, Pramod Viswanath. Fundamentals of Wireless Communication. *Cambridge University Press, 2005*.

[13] G. J. Foschini, M. J. Gans. On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas. *Wireless Personal Commun.: Kluwer Academic Press*, 6(3), 1998.

[14] E. Telatar. Capacity of Multi-Antenna Gaussian Channels. *Eur. Trans. Telecomm. ETT*, 10(6), 1999.

[15] C. Chuah, D. Tse, J. Kahn, R. Valenzuela. Capacity Scaling in MIMO Wireless Systems Under Correlated Fading. *IEEE Trans. Inform. Theory*, 48(3), 2002.

[16] T. Marzetta, B. Hochwald. Capacity of a Mobile Multiple-Antenna Communication Link in Rayleigh Flat Fading. *IEEE Trans. Inform. Theory*, 45(1), 1999.

[17] A. Kamerman and L. Monteban. WaveLAN II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, Summer 1997.

[18] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. *ACM MSWiM'04*.

[19] G. Holland, N. Vaidya and V. Bahl. A Rate-Adaptive MAC Protocol for Multihop Wireless Networks *ACM MOBICOM'01*.

[20] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knighlty. Opportunistic Media Access for Multirate Ad Hoc Networks. *ACM MOBICOM'02*.

[21] Onoe Rate Control. http://madwifi.org/browser/trunk/ath rate/onoe.

[22] J. Kim, S. Kim, S. Choi and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. *IEEE INFOCOM'06*.

[23] J. Bicket. Bit-rate Selection in Wireless Networks. *MIT Master's Thesis*, 2005.

[24] J. Camp and E. Knightly. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-layer Implementation and Experimental Evaluation. *ACM MOBICOM'08*.

[25] G. Judd, X. Wang and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. *ACM MobiSys'08*.

[26] I. Haratcherev, K. Langendoen, R. Lagendijk and H. Sips. Hybrid Rate Control for IEEE 802.11. *ACM MobiWac'04*.

[27] M. Vutukuru, H. Balakrishnan and K. Jamieson. Cross-Layer Wireless Bit Rate Adaptation. *ACM SIGCOMM'09*.

[28] H. Rahul, F. Edalat, D. Katabi and C. Sodini. Frequency-Aware Rate Adaptation and MAC Protocols. *ACM MOBICOM'09*.

[29] S. Sen, N. Santhapuri, R. R. Choudhury. AccuRate: Constellation Based Rate Estimation in Wireless Networks. *USENIX/ACM NSDI'10*.

[30] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A Practical SNR-Guided Rate Adaptation. *IEEE INFOCOM'08*.

[31] K. Ramachandran, R. Kokku, H. Zhang, M Gruteser. Symphony: Synchronous Two-Phase Rate and Power Control in 802.11 WLANs. *ACM MobiSys'08*

[32] M. Wong, J. M. Gilbert and C. H. Barratt. Wireless LAN using RSSI and BER Parameters for Transmission Rate Adaptation. *US patent* 7,369,510, 2008.

[33] D. Halperin, W. Hu, A. Sheth, D. Wetherall. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. *ACM SIGCOMM'10*.

[34] S. H. Wong, H. Yang, S. Lu and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. *ACM MOBICOM'06*.

[35] D. Qiao, S. Choi, and K. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Trans. on Mobile Computing*, 1(4), 2002.

[36] W. Xi, A. Munro and M. Barton. Link Adaptation Algorithm for the IEEE 802.11n MIMO System. *Networking*, 2008.

[37] F. Peng, J. Zhang and W. Ryan. Adaptive Modulation and Coding for IEEE 802.11n. *IEEE WCNC'07*.

[38] S. Kant and T. L. Jensen. Fast Link Adaptation for IEEE 802.11n. *Aalborg University, Master's Thesis*. 2007.

[39] Q. Xia, M. Hamdi, and K. B. Letaief. Open-Loop Link Adaptation for Next-Generation IEEE 802.11n Wireless Networks. *IEEE Trans. Veh. Technology*, 58(7), 2009.

[40] W. Kim, M. Khan, K. Truong, S-H. Choi, R. Grant, H. Wright, K. Mandke, R. Daniels, R. Heath and S. Nettles. An Experimental Evaluation of Rate Adaptation for Multi-Antenna Systems. *IEEE INFOCOM'09*.

[41] A. Gudipati, S. Katti. Strider: Automatic Rate Adaptation and Collision Handling. *ACM SIGCOMM'11*.

[42] Intel PowerTOP Tool for Linux. *http://www.lesswatts.org/projects/powertop/*

[43] A. Jardosh, G. Iannaccone, K. Papagiannaki, and B. Vinnakota. Towards an Energy-Star WLAN Infrastructure. *IEEE HotMobile'07*.

[44] A. Jardosh, K. Papagiannaki, E. M. Belding, K. C. Almeroth, G. Iannaccone, B. Vinnakota. Green WLANs: On-demand WLAN Infrastructures. *ACM MONET*, 14(6), 2008).

[45] N. Mishra, K. Chebrolu, B. Raman, A. Pathak. Wake-on-WLAN. *ACM WWW'06.*

[46] Z. Zeng, Y. Gao, P. R. Kumar. SOFA: A Sleep-Optimal Fair-Attention scheduler for the Power-Saving Mode of WLANs. *IEEE ICDCS'11*

[47] Kyu-Han Kim, A. W. Min, D. Gupta, P. Mohapatra, J. Singh. Improving Energy Efficiency of Wi-Fi Sensing on Smartphones. *IEEE INFOCOM'11*.

[48] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, R. Gupta. Wireless Wakeups Revisited: Energy Management for VoIP over Wi-Fi Smartphones. *ACM MobiSys'07*.

[49] J. Liu and L. Zhong. Micro Power Management of Active 802.11 Interfaces. *ACM MobiSys'08*.

[50] N. Balasubramanian, A. Balasubramanian, A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. *ACM IMC'09*.

[51] X. Zhang, K. G. Shin. E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks. *ACM MOBICOM'11*.

[52] F. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: Exploiting High Bandwidth Wireless Interfaces to Save Energy for Mobile Devices. *ACM MobiSys'10*.

[53] Y. He, R. Yuan, X. Ma., Jun Li. The IEEE 802.11 Power Saving Mechanism: An Experimental Study. *IEEE WCNC'08*.

[54] A. Rahmati and L. Zhong. Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer. *ACM MobiSys'07*.

[55] D. Qiao, S. Choi, A. Jain, K. G. Shin. MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h. *ACM MOBICOM'03*.

[56] J. Gomez, A. T. Campbell, M. Naghshineh, C. Bisdikian. Conserving Transmission Power in Wireless Ad Hoc Networks. *IEEE ICNP'01*.

[57] D. Halperin, B. Greenstein, A. Sheth, D. Wetherall. Demystifying 802.11n Power Consumption. *USENIX HotPower'10*.

[58] K. Jang, S. Hao, A. Sheth, R. Govindan. Snooze: Energy Management in 802.11n WLANs. *ACM CONEXT'11*.

[59] R. Krashinsky, H. Balakrishnan. Minimizing Energy for Wireless Web Access with Bounded Slowdown. *ACM MOBICOM'02*.

[60] H. L. Krauss, C. W. Bostian, F. H. Raab. Solid State Radio Engineering. *John Wiley & Sons, 1980*.

[61] V. Shrivastava, S. Rayanchu, J. Yoon, S. Banerjee. 802.11n under the Microscope. *ACM IMC'08*.

[62] S. Rayanchu, V. Shrivastava, S. Banerjee, R. Chandra. FLUID: Improving Throughputs in Enterprise Wireless LANs through Flexible Channelization. *ACM MOBICOM'11*.

[63] M. Arslan, K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, S. Addepalli, K. Papagiannaki. Auto-configuration of 802.11n WLANs. *ACM CoNEXT'10*

[64] S. Cui, A. J. Goldsmith, A. Bahai. Energy-efficiency of MIMO and Cooperative MIMO in Sensor Networks. *IEEE Journal on Selected Areas in Commun.*, 22(6), 2004.

[65] H. Kim, C.-B Chae, G. de Veciana, R. W. Heath Jr. A Cross-Layer Approach to Energy Efficiency for Adaptive MIMO Systems Exploiting Spare Capacity. *IEEE Transactions on Wireless Commun.*, 8(8), 2009.

[66] B. Bougard, G. Lenoir, A. Dejonghe, L. V. der Perre, F. Catthoor, W. Dehaene SmartMIMO: An Energy-Aware Adaptive MIMO-OFDM Radio Link Control for Next-Generation Wireless Local Area Networks. *EURASIP J. Wireless Commun. Networking*, 2007(3), 2007.

[67] W. Gabran, B. Daneshrad. Hardware and Physical Layer Adaptation for a Power Constrained MIMO OFDM System. *IEEE ICC'11*.

[68] S. Gollakota, D. Katabi. ZigZag Decoding: Combating Hidden Terminals in Wireless Networks. *ACM SIGCOMM'08*.

[69] S. Gollakota, S. D. Perli and D. Katabi. Interference Alignment and Cancellation. *ACM SIGCOMM'09*.

[70] S. Gollakota, F. Adib, D. Katabi, S. Seshan. Clearing the RF Smog: Making 802.11 Robust to Cross-Technology Interference. *ACM SIGCOMM'11*.

[71] S. Sen, R. R. Choudhury, S. Nelakuditi. No Time to Countdown: Migrating Backoff to the Frequency Domain. *ACM MOBICOM'11*.

[72] M. Jain, J. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, P. Sinha. Practical, Real-time, Full-Duplex Wireless. *ACM MOBICOM'11*.

[73] G. Pei and T. Henderson. Validation of ns-3 802.11b PHY Model. May 2009. http://www.nsnam.org/∼pei/80211b.pdf

[74] T. M. Cover and J. A. Thomas. Elements of Information Theory. *John Wiley & Sons, 1991*.

[75] G. Ramakrishna, W. David, G. Ben and S. Seshan. Understanding and Mitigating the Impact of RF Interference on 802.11 Networks. *ACM SIGCOMM'07*.

[76] S. Biswas and R. Morris. ExOR: Opportunistic Multi-Hop Routing for Wireless Networks. *ACM SIGCOMM'05*.

[77] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft. XORs in The Air: Practical Wireless Network Coding. *ACM SIGCOMM'06*.

[78] V. Bharghavan, A. Demers, S. Shenker, L. Zhang. MACAW: A Media Access Protocol for Wireless LAN's. *ACM SIGCOMM'94*.

[79] D. Skordoulis, Q. Ni, H. Chen, A. P. Stephens, C. Liu, A. Jamalipour. IEEE 802.11n MAC Frame Aggregation Mechanisms for Next-Generation High-Throughput WLANs. *IEEE Wireless Commun.*, 15(1), 2008.

[80] D. Halperin, W. Hu, A. Sheth, D. Wetherall. 802.11 with Multiple Antennas for Dummies. *ACM SIGCOMM CCR'10*.