Adaptive Rate Control based on Collision and Frame Aggregation Awareness

by

Dongjin Kim

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 2, 2014

Keywords: 802.11n, MIMO, Rate Adaptation, Rate Control, Channel Adaptation, Link Adaptation, Collision, Frame Aggregation,

Approved by

Alvin Lim, *Chair*, Associate Professor of Computer Science and Software Engineering
David Umphress, Associate Professor of Computer Science and Software Engineering
Drew Hamilton, Affiliate Professor of Computer Science and Software Engineering
Xiao Qin, Associate Professor of Computer Science and Software Engineering

Abstract

The innovative advances in mobile technologies have facilitated remarkable growth in the use of mobile devices such as smart phones, e-book readers, and tablet PCs. Since a significant amount of wireless traffic is generated from these devices, the shared channel medium may suffer from serious mutual interference. In addition, various kinds of fading weaken the stability of channel quality.

Though IEEE 802.11 standards specify an available set of transmission rates as one of the approaches to accommodate channel dynamics, they do not discuss in detail when and how to use them. Rate Adaptation (RA) algorithms try to solve this decision problem by determining the most appropriate data rate for the instantaneous wireless channel condition.

The primary focus of Rate Adaptation studies for legacy standards such as IEEE 802.11 a/b/g has been on when to increase/decrease transmission rates or how to differentiate between collision-induced loss and channel disorder-induced error. However, with the advent of novel Multiple Input Multiple Output (MIMO), Channel Bonding, and Frame Aggregation/Block Acknowledgement technologies, it has become possible to enable Gigabit connectivity in wireless communication. At the same time, these new technologies add a new dimension of problems with rate adaptation. Even though much research has attempted to address the non-monotonic relationship [1] between data rate and throughput via exploring MIMO technology or by identifying the usability of another channel space for bonding [2], little research has analyzed the impact of Frame Aggregation.

This paper begins by surveying existing research on Rate Adaptation (RA) algorithms. A wide range of RAs are introduced and classified based on various design factors.

Secondly, a new RA scheme called *RACD* (Rate Adaptation with Collision Differentiation) is presented. RACD is capable of differentiating the reasons for packet errors by using

CTS-to-self control packets. After classifying the sources of errors, RACD responds accordingly with actions such as initiating RTS/CTS handshake or adjusting Contention Window size. Extensive simulation results confirm that RACD significantly improves performance, especially in simulated congestion scenarios.

Finally, the new features introduced in the state of the art IEEE 802.11n standard are overviewed and discussed. Out of the three dominant innovations specified in the new amendment, the impact of one innovation, Frame Aggregation (FA), on RA algorithm design is examined from various perspectives with a case study. Based on the analysis and findings from the case study, a novel *AARA* (Aggregation Aware Rate Adaptation) algorithm is proposed and evaluated with extensive experiments conducted in controlled and repeatable environments. The performance enhancement over representative existing RA algorithms is validated with throughput gains up to 473% in various experimental scenarios. Potential performance benefits on RA design from Frame Aggregation awareness is discussed in detail.

Although the experimental work and the proposed AARA algorithm provide insight into the significant influence of FA on RA algorithm design, a number of critical issues remain which should be considered for further research. This dissertation concludes with a discussion of potential future work related to FA and RA for IEEE 802.11 Wireless Networks.

Acknowledgments

First of all, I would never have been able to finish my dissertation without the guidance of my advisor, Dr. Alvin Lim and I would like to express my deepest gratitude to him for his caring, patience, and providing me with an excellent atmosphere for doing research.

I also thank my committee members, Dr. David Umphress, Dr. Drew Hamilton, Dr. Xiao Qin, and Dr. Shiwen Mao for their teaching, encouragement and consideration.

I would like to give my best appreciation to my friends, Ted, Se-eung, Museung, Mike, Chris, Tom, Nesha. They have given me comfort and energy for me to go through lots of difficulties.

Many thanks to my lab mates, Qing, Brandon, Jian, Kanika, Seungbae, and Song. They have been always willing to help and give their best suggestions. I would have been lonely without them.

I would also like to thank my parents and sister. They have never stopped supporting me and encouraging me with their best wishes.

Finally, I would like to appreciate the help from my wife, Mia Kim and two children, Gaeun and Junhyeok Kim for their being there with me through the good and bad times.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Rate Adaptation in 802.11 WLANs

An extensive amount of research on wireless communications has been done both in the literature and industry. Such research has led to an explosive growth in the number of wireless devices used by people who are attracted by their inherent convenience and portability.

Among different types of wireless networks, Wireless Local Area Network (WLAN), also called WiFi, has been in the limelight for short distance connections due to its convenience and relatively higher bandwidth availability. However, in addition to the inherent instability of the channel medium and signal distortion caused by movement of devices, the increasing volume of wireless traffic seriously degrades network performance. Hence, promptly and flexibly adapting to unpredictable wireless link variations has become a primary challenge of wireless link performance. Rate Adaptation (RA) is one of the channel adapting methods which dynamically change data transmission rate according to the current wireless status.

A series of WLAN standards developed by the IEEE 802.11 Working Group provide multiple data rates varied by different modulation types and coding rates. Furthermore, as technology advances, the number of available rate options is increasing. However, the IEEE Working Group did not specify an algorithm which selects the most appropriate rate to ongoing channel.

The key problem is that the fastest rate does not always provide the best performance. Typically, a higher data rate can transmit more data in the same period of time than a lower option can due to denser signal modulation, but a lower rate can reach a longer distance due to its robustness in transmission. Therefore, network efficiency is seriously affected by adaptive data rate adjustment.

1.2 Key Contributions

The goal of this dissertation is to design and propose efficient Rate Adaptation algorithms to optimize wireless link performance. To this end, Chapter 2 of this work surveys RA schemes which have been suggested to date. Important challenges involved in efficient design options are carefully reviewed. Then, two novel RA algorithms are proposed with intensive performance evaluation followed by in-depth research on how different aspects of RA design metrics affect wireless network performance.

The key contributions are as follows:

First, a wide range of the existing RA algorithms are categorized and classified hierarchically. The survey not only deals with RA algorithms, but also includes design factors which are potentially related to wireless network problems connected to RA system design. Even though numerous RA algorithms have been proposed in the literature, little effort has been made to organize and categorize those algorithms.

Secondly, a new RA scheme called RACD (Rate Adaptation with Collision Differentiation) is proposed in Chapter 4. RACD is capable of differentiating the reasons for packet errors by means of using *CTS-to-self* control packets and minimizes the throughput degradation due to misinterpretation of collision-induced failure. After classifying the sources of error, RACD responds accordingly by actions such as initiating RTS/CTS handshake or adjusting Contention Window size. Extensive simulation results confirm that RACD significantly improves performance especially in simulated congestion scenarios.

Finally, Chapter 5 overviews and discusses the new features introduced in the state of the art IEEE 802.11n standard. Out of the three dominant innovations specified in the new amendment, the impact of one innovation, Frame Aggregation (FA) on RA algorithm design is closely investigated from various perspectives with a case study. Based on the analysis and findings from the case study, a novel AARA (Aggregation Aware Rate Adaptation) algorithm is proposed and evaluated via exhaustive experiments carried out in controlled and repeatable environments. Specifically, in contrast to prior work, the test results exhibit per-AMPDU packet rate selection compared to the ideal transmission rate upon each channel condition. This chapter evaluates the performance enhancement over representative existing RA algorithms based on extensive experiments under various scenarios and discusses in detail the potential benefit on RA design from Frame Aggregation awareness.

1.3 Terms and Abbreviation

Table 1.1 lists the most frequently used terms and abbreviations in this dissertation. Note that in order to avoid confusion, the term *packet* used in this paper indicates an completed data structure which can be transmitted without additional information such as protocol headers for each layer, check sequence, and padding bits; whereas the term *frame* means an individual MPDU or MSDU before being aggregated into either A-MPDU or A-MSDU packets as presented in Figure 5.2. Similarly, *retry* is used when an individual MPDU in an aggregated packet is failed and retried, whereas *retransmission* is used when an entire A-MPDU or A-MSDU packet is lost and retried.

Table 1.1: Terms and Abbreviations

Terms	**Abbreviations**
Rate Adaptation (Link Adaptation)	RA
Modulation and Coding Scheme	MCS
Received Signal Strength Indicator	RSSI
Signal to Noise Ratio	SNR
Multi-Rate Retry	MRR
Multiple Input Multiple Output	MIMO
Spatial Diversity/Single Stream	SD / SS
Spatial Multiplexing/Multi Stream	SM / MS
Frame Aggregation	FA
Mac Protocol Data Unit	MPDU
Mac Service Data Unit	MSDU
Aggregated MPDU	A-MPDU
Aggregated MSDU	A-MSDU
Block Acknowledgement	BlockACK
Number of frames in an A-MPDU	nFrames
Number of bad frames in an A-MPDU	nBad
Packet Error Ratio for time window	PER
Frame Error Ratio in multiple A-MPDUs	FER
Frame Success Ratio in multiple A-MPDUs	FSR
A-MPDU Frame Error Ratio in a single A-MPDU	AFER
A-MPDU Frame Success Ratio in a single A-MPDU	AFSR

Chapter 2

A Survey of Rate Adaptations in 802.11 WLANs

2.1 Introduction

Rate Adaptation (also called Link Adaptation in the literature) is a set of processes to determine the most suitable transmission rate for the current channel condition out of available rate options specified in the standard.

Determining the optimal transmission rate in timely fashion is a challenge for several reasons. First, it is essential to assess the ambient channel quality using various metrics typically available in the PHY and MAC layers. Channel estimation may be (but does not have to be) followed by differentiating losses from various causes. Then, according to the channel estimation and loss differentiation, the RA decides which rate option can maximize the current link capacity. The data rate can be increased or decreased step by step in a linear manner, or the RA can monitor statistics of each potential rate option and jump directly to the best one.

Even though RA has been extensively studied in the last decade, little attempts has been made to survey the existing RA algorithms. S. Biaz and S. Wu [3] surveyed and compared different RA schemes, but their study includes only the initial RA algorithms proposed before 2008. This section extends their work to aggregate the most recent RA schemes, re-organizes them with distinct criteria, and gives a brief overview of each algorithm.

2.2 Classification of Rate Adaptation Algorithm

A remarkable number of studies have been published on rate adaptations in the literature that provide high transmission throughput in wireless networks. These algorithms can be classified into different categories according to various criteria.

One criterion for classifying RA algorithms is whether the decision-making station is the transmitter or the receiver. A transmitter-side decision rate adaptation scheme determines the best rate based only on the transmitter's local information, whereas a receiver-side scheme determines the rate based on information exchanged between the transmitter and the receiver. The receiver-side decision is relatively more accurate compared to the transmitter-side decision but has the drawbacks of a longer delay and a more complicated decision. It also requires modification of the current IEEE 802.11 standard because it needs to wait for information about the channel sent from the transmitter.

Secondly, a good number of RAs differentiate collision-induced losses from channel degrading losses, because decreasing data rate upon collision-induced losses not only degrades the network performance, but also further aggravates the channel congestion due to the longer transmission time of lower rate options. A variety of protocols studying collision avoidance mechanisms will be introduced in Section 2.2.3.

On the other hand, RA schemes also can be classified into ones before and after the IEEE 802.11n standard. Since the new standard leads to a significant breakthrough in network bandwidth capacity, it introduces unique challenges which MIMO RAs must address for effective adaptation. In this paper, the term *Legacy RA* refers to RA schemes that were used before the 802.11n standard, as opposed to *MIMO RAs* which targeted and exploited MIMO-enabled devices following the new amendment. In addition, RA protocols can also be classified by the ability to differentiate collision-induced losses from channel-induced losses, regardless of which standard they are designed for.

2.2.1 Rate Adaptation for Legacy Standards

In this section, RA protocols designed for the legacy standards 802.11a, b and g are introduced and categorized based on which protocol stack they access to infer channel condition. The parameters for channel interpretation can be obtained by either the PHY layer, the MAC layer, or both. Though the two types of RAs each have their own strengths and weaknesses, PHY metric-based RA is known to overestimate the channel status while MAC metric-based RA tends to underestimate the current state. Therefore, hybrid designs using both PHY-layer and MAC-layer metrics have been introduced to take advantage of the strengths and constraints of each system. I. Haratcherev and K. Langendoen [4] selected the optimal transmission rate using MAC-layer frame error statistics and bounds available transmission rates according to the report from channel dynamic detector using RSSI variation. Note that since the legacy standards do not have the aggregation mechanism specified in the protocol, each packet is transmitted with a single MPDU frame.

MAC-layer Metric-based Rate Adaptation Algorithms

While the PHY-layer metric based RAs gauge the link state using direct indications such as Received Signal Strength Indicator (RSSI) and Signal to Noise Ratio (SNR), MAC-layer metric based RAs (also called *Loss-based RAs* in the literature) indirectly estimate how efficiently the current or candidate rate option performs to infer the optimal transmission rate. Typically, the link quality is evaluated by checking packet loss information, which can be identified when the transceiver can neither decode nor receive the acknowledgement from the receiver.

The packet loss pattern can be measured either by counting the number of consecutive successes (or losses) or by observing the Packet Error Ratio (PER) over a sliding time window. The initial stage RA algorithms such as ARF [5], AARF [6], AARF-CD [7], and RRAA [8] employ a consecutiveness counter to assess the link condition. For example, as the earliest RA scheme, Auto Rate Fallback (ARF) [5] simply increases the data rate by one step after ten consecutive successful transmissions and decreases it upon two sequential packet losses. Using consecutiveness of packet losses as a decision guideline is intuitive and easy to implement; however, as I. Pefkianakis [9] demonstrated by well-organized experiments, this method has potential limitations due to unstable channel dynamics.

Another RA proposal interprets link condition based on Packet Error Ratio (PER) between corrupted packets and total transmitted packets over a specified time frame. Sample Rate [10] collects statistics over a period of one to ten seconds and chooses the rate which exhibits the shortest expected transmission time in the last sample period. ONOE [11] also collects the packet transmission statistics, then adjusts credit counters whenever the statistic satisfies certain conditions. If the credits reach a given threshold, the RA decides whether to use a higher or lower data rate for the next time frame. Atheros RA [12], Minstrel [13] and TERA [14] commonly measure network throughput reflecting the regulatory overhead specified in the standard and packet error rate for the sampling time window. Then the protocol switches directly to the best performing rate as opposed to making linear rate adjustments for consecutiveness based solutions.

Context-Aware Rate Section (CARS) [15] takes advantage of context information in addition to the MAC layer statistics, such as vehicle speed and distance between corresponding nodes, to achieve better performance in VANET. Also, RapidSample [16] adopts statistic-based SampleRate for static settings and employs the time difference between the last failed attempts and current transmission to identify the best rate for mobile settings. The mobility can be detected by external sensors such as GPS and accelerometers.

PHY-layer Metric-based Rate Adaptations

As opposed to the MAC-layer metric-based RAs, a wide range of RA protocols have been proposed based upon diverse metrics from the PHY layer. The most popular and widely accepted parameters are Received Signal Strength Indicator (RSSI) and Signal-to-Noise Ratio (SNR). Receiver Based Auto Rate (RBAR) [17] is proposed as the first and most representative SNR-based, closed-loop rate adaptation algorithm; this method leverages RTS/CTS control packets to exchange SNR information between corresponding stations.

A number of protocols, including Vehicular Ad-hoc Network (VANET), pay attention to mobile environments, which are different from typical office environments due to higher fluctuation and channel asymmetry. K. Lee et al [18] paved the way for simulating VANET environment by collecting SNR traces from moving vehicles and feeding them into a simulator to reflect the real channel condition. Instead of using simulations, J. Camp and E. Knightly [19] designed a custom cross-layer framework to implement the SNR-based RA protocol for VANET in their FPGA platform. In addition, P. Deshpande and S. Das [20] proposed the SNR-based RA scheme BRAVE, which predicts near future channel states based on short-term RSSI history. Recently, RFRA [20] applied a machine-learning heuristic algorithm to rate adaptation for vehicular networks to exploit statistical dependency between vehicle speed, SNR, and data rate.

Since PHY-layer metric-based RAs are typically receiver-based (closed-loop) and the receiver needs to feedback its surrounding channel state to a transceiver which is not standard-compliant, most PHY-layer protocols are either simulation-based or implemented in proprietary test-beds. On the other hand, CHARM [21] designed and implemented a transmitter-based, open-loop RA algorithm based on RSSI measurements from receiver-side. It exploits channel reciprocity, which enables the transmitter to compute the path-loss based on the additional information provided by the receiver such as transmit power and antenna gain. RAM [22] also overcame the standard incompliance using receiver-side ACK rate variance to direct transmission rate adjustment.

Accurate rate selections are made by more sophisticated symbol-level PHY-layer metrics to reflect the channel state in [23, 24, 25]. Softrate [23] computes interference-free bit error ratio based on SoftPHY hints, which are bit level confidence information, to estimate channel quality, while FARA [24] collects the SNR and BER values, respectively, of individual OFDM data sub-bands to track the channel. On the other hand, AccuRate [25] observes symbol-level dispersion between the transmitted and received symbol pair on the constellation space.

This type of RA schemes can be agile to adapt to fast-changing channel and adjust data rate; however, they have common issues which prevent vendors from implementation in practice: (1) They require cross-layer implementation and an explicit feedback mechanism for receiver-side channel measurements which are not standard-compliant. (2) Since these RAs need *a-priori*, pre-defined mappings of the ambient channel conditions between a range of PHY metrics, PER, and rate options, an in-situ hardware-specific calibration process is required to correctly reflect the surrounding channel. (3) As pointed out in [10], even though the SNR value is able to correctly interpret the current link condition, it has little predictive value for future transmissions because SNR is typically measured at the start of the packet reception and it cannot reflect variations even during the duration of transmission.

2.2.2 Rate Adaptation for the MIMO-enabled Standards

The new 802.11n/ac standards shed light on a new era of up to Giga-bps network bandwidth for WLANs. The representative techniques to enable the performance surge are three-fold: (1) Multiple Input Multiple Output (MIMO) enables the involving stations to transmit more than two data streams simultaneously, (2) Channel Bonding allows the combination of adjacent channel bandwidths to boost performance, and (3) Frame Aggregation (FA) aggregates multiple data frames into a single packet to amortize protocol overheads. More details of these new characteristics are provided in Section 5.2. Since these advanced features introduce new levels of freedom to the decision-making process, RA algorithms for the new amendments require different approaches. This section presents various RA algorithms exploring the new properties.

Receiver-based Closed-loop Approaches

In addition to the features for performance enhancement, the new standard also defines the Channel Status Information (CSI). CSI, a MCS feedback mechanism, enables the receiver to feedback to the transceiver in response to incoming packets to overcome the limitation of the legacy standards. Thanks to this rectification, the PHY-metric, receiver-based RAs can be implemented in a standard compliant way.

Effective SNR [26] develops a packet delivery model based on information from the Network Interface Controller. The receiver averages the Bit Error Ratio (BER) of 56 OFDM subcarriers and calculates the effective SNR over SNR of received signal and OFDM modulations, and then sends the CSI information back to the transceiver.

G. Martorell et al [27] designed the Fast Link Adaptation scheme, which adjusts MCS to a pre-defined target PER. The Exponential Effective SNR Mapping (EESM) and corresponding target BER is averaged over individual OFDM sub-carriers and translated to a representative Packet Error Ratio.

ARAMIS [2] employs closed-loop approach as well. It takes advantage of SNR difference between antennas along with the combined SNR to predict the channel quality. After measuring the SNR and SNR difference, it identifies a corresponding Packet Reception Rate (PRR) from an *a-priori* mapping table, then feedbacks the highest available MCS rate which satisfies the PRR threshold.

Even though the CSI feedback mechanism is clearly defined in the standard and paves the way for closed-loop, receiver-based RA algorithms, these algorithms are unpredictable because of the rapid channel variation during which the receiver detects the link and the transceiver obtains the feedback message. Moreover, since its cost inefficiency and implementation complexity keep vendors from implementing real drivers, most of the proposed algorithms are just evaluated with simulations and not implemented in practice.

Transceiver-based Open-loop Approaches

The general concept throughout legacy RAs is that Packet Error Ratio (PER) is proportional to MCS rate index where incremental rate growth can eventually reach the best option; however, this seemingly intuitive idea cannot be applied to MIMO RAs, because, given that the channel quality is not sufficient to tolerate multiple streams, single-stream MCS rates outperform the double stream rates. For example, as indicated in Table 2.1, Single Stream (SS) MCS 5 and Double Stream (DS) MCS 11 both have 52 Mbps data rate when used with Long Guide Interval (LGI) without channel bonding (default 20GHz channel bandwidth). If the channel quality is stable enough to hold a double stream, DS MCS 11 is desirable due to the robustness of lower modulation (16QAM over 64QAM for MCS 5); however, if the channel cannot tolerate double streams, SS MCS 5 must be the better choice in spite of a more aggressive modulation type. This *Non-monotonicity* between PER and data rate has been the key challenge to open-loop RA designs, as opposed to the closed-loop approaches introduces in the previous section.

MiRA [1] confirms the *Non-monotonicity* through extensive real world experiments. This study collected the per A-MPDU statistics and demonstrated that when both MIMO mode rates are used interchangeably, the A-MPDU Frame Error Ratio (AFER) does not grow monotonically as the bandwidth of the transmission rate increases. Yet the authors also validated monotonic relationship holds for the individual operation mode. Based on the result of a practical case study, the MiRA algorithm not only gathers statistics from the current MIMO mode (intra-mode), but also samples rate options of another MIMO mode (inter-mode) to avoid situations in which the RA gets stuck with the lower transmission rate rather than the potentially better rate option due to the Non-monotonicity. This algorithm also employs an adaptive probing scheme which prevents the protocol from frequent sampling for bad rates. WRA [28] follows MiRA's motivation, but suggests a different solution. It operates separated, individual RA algorithms for each MIMO modes instead of zigzagging. These two schemes are well designed to overcome the Non-monotonicity, but aggressive

Table 2.1: MCS Index Table Available in 2 Streams MIMO NIC

MCS Index	# of Streams	Modulation Type	Coding Rate	Data Rate (Mbps)			
				20GHz Channel		40GHz Channel	
				LGI	SGI	LGI	SGI
0	1	BPSK	1/2	6.50	7.20	13.50	15.00
1	1	QPSK	1/2	13.50	14.40	27.00	30.00
2	1	QPSK	3/4	19.50	21.70	40.50	45.00
3	1	16-QAM	1/2	26.00	28.90	54.00	60.00
4	1	16-QAM	3/4	39.00	43.30	81.00	90.00
5	1	64-QAM	2/3	52.00	57.80	108.00	120.00
6	1	64-QAM	3/4	58.50	65.00	121.50	135.00
7	1	64-QAM	5/6	65.00	72.20	135.50	150.00
8	2	BPSK	1/2	13.00	14.40	27.00	30.00
9	2	QPSK	1/2	26.00	28.90	54.00	60.00
10	2	QPSK	3/4	39.00	43.30	81.00	90.00
11	2	16-QAM	1/2	52.00	57.80	108.00	120.00
12	2	16-QAM	3/4	78.00	86.70	162.00	180.00
13	2	64-QAM	2/3	104.00	115.60	216.00	240.00
14	2	64-QAM	3/4	117.00	130.00	243.00	270.00
15	2	64-QAM	5/6	130.00	144.40	270.00	300.00

probing over different operation modes produces non-negligible overhead and limits further performance improvements.

ARF-HT [29] divides the solution space into MIMO mode decisions and rate adjustments in each mode and adopts different MAC-layer metrics respectively. While a credit system based on RSSI difference is used to select the MIMO operation modes, ARF-HT employs Expected Transmission Time (ETT) as SampleRate [10] for adjusting intra-mode rate selection. Similarly, RAMAS [30] divides available MAC-layer metrics used for rate selection into two independent groups: the modulation group and the enhancement group. The modulation group only takes account of modulation type and coding scheme; therefore, a monotonic relation between PER and data rate can be presumed among the rates for the same modulation group. On the other hand, the enhancement group deals only with the new improvement features in the 802.11n standard, such as the number of streams, guard interval (GI), and channel width. Two independent groups employ separate credit-based protocols which are used to decide whether to move upward or downward.

L3S[31] incorporates the Multi-Rate Retry (MRR) scheme, which is a set of tuples composed of data rate and retransmission limit. MRR enables RAs to attempt multiple transmission rates when packets are lost and retransmitted. L3S employs this MRR functionality to probe various candidate rate options to accommodate the current channel fluctuation. In addition, the probing interval is adapted based on the consecutiveness of packet success and failure.

S. Lakshmanan et al. [32] found that detrimental interference makes the MIMO process more susceptible to interference and developed a simple MAC-layer metric called Median Multiplexing Factor (MMF). MMF decides whether to use the MIMO mode rates under the current channel condition by throughput comparison between MCS rates which have the same modulation type and coding rate.

2.2.3 Rate Adaptation with Collision Differentiation

Wireless link degradation is mainly caused by: (1) signal fading, (2) *In-range* collision, (3) hidden collision, and (4) ambient interference. Signal fading is caused by either signal propagation loss by distance or *Multi-path fading*, which occur when multiple paths exist between a transceiver and receiver pair and the identical signals arrive at the receiver-side with a delay (called *Delay Spread*). On the other hand, *In-range* and hidden collisions involve additional stations besides the transmission pair. While *In-range* collision is attributed to simultaneous transmissions among stations within wireless range, hidden collision is caused by mutually unrecognized transmitters. Finally, interference is caused by random signals which the involved standard cannot decode.

RA systems should respond only to fading-induced channel dynamics. If RAs switch rates over collision-induced packet failures, the switch worsens the link condition because the lower rate increases the channel occupancy and augments the chance of collision. Many researchers have attempted to differentiate the causes of channel deterioration and introduced various metrics for the distinction.

The initial RAs [8, 33, 34, 35] employ RTS/CTS control packets to detect and respond to collision-induced packet failures. After a transceiver sends out an RTS packet after a frame loss, the packet determines the current condition by the success of CTS packet reception. DRANLD [36] used the received signal strength and Negative ACK packet (NACK) upon the RTS/CTS exchange to effectively diagnose packet losses.

Regardless of the distinction capability, RTS/CTS packets constitute a substantial overhead. Therefore, more sophisticated symbol level classification methods have been suggested. Z. Eu et al [36] correlates RSSI value with Link Quality Indicator (LQI) to classify packet transmission outcomes specifically for Wireless Sensor Networks (WSNs). LQI value is an accumulation of magnitude difference between ideal and actual signal level over individual OFDM symbols. COLLIE [37] makes the distinction by collecting and analyzing the error-bit pattern upon reception of a corrupted packet. As pointed out earlier, symbol level implementation is complicated and cannot be made in a standard-compliant way.

In order to overcome issues using symbol-level measurements in the PHY-layer, various metrics available in the MAC-layer have been proposed to diagnose collision-induced packet losses. LDRA [38] uses beacon packets which are mandated in all of the 802.11 standards for broadcasting necessary information for synchronization and power management. In addition, M. A. Y. Khan and D. Veitch [39] exploit a simple idea that if there is an ongoing protection protocol which reserves the channel proprietarily, such as fragmentation, RTS/CTS handshake, or Block reservation in 802.11e, the packet loss is caused by channel disorder. Similarly, REACT [40] defines a *green channel period* which is a congestion-free interval, and considers any loss during the green channel period to be a collision-induced loss.

On the other hand, time-domain metrics have been suggested in the literature to detect current congestion. WOOF [41] predicts the density of congestion by measuring *channel* *busy time* with statistical mitigation for unpredictable fluctuation. YARA [42] computes time difference between transmission time and effective transmission time. The *Effective transmission time* refers to the ideal transmission time where there is no contending station to attempt a channel access.

In contrast to the previous work, M. Kim and C. Choi [44] explored Frame Aggregation (FA) and Block Acknowledgement (BlockACK) functionality presented in the IEEE 802.11n standard. In this functionality, a hidden node detection mechanism differentiates losses from channel error, *In-range* collision, and hidden collision by mathematical analysis using MAC-layer parameters such as number of idle slots, missing BlockACK, and other transmissions, then reacts accordingly.

2.3 Summary

In summary, all Rate Adaptation schemes should address two core challenges: (1) how to estimate the current channel condition, and (2) how to translate the estimate into the optimal transmission rate. In this chapter, a wide range of Rate Adaptation algorithms are briefly introduced and classified by various criteria which they use for channel estimation.

Chapter 3

Problem Statement and Research Questions

3.1 Problem Statement

Wireless channels naturally suffer from different types of interference than do wired channels because the wireless signals radiated from the antenna take numerous different routes, collide with other signals, and experience attenuation while they are travelling through the channel medium. Moreover, a growing number of wireless devices make the channel more variant and unpredictable. Rate Adaptation is one of the approaches which attempt to enhance different network performance metrics by improving agility in the presence of channel dynamics. In addition to adaptability, another important factor in designing RA algorithms is differentiating the causes of packet losses, as packet failures resulting from collision should not affect the decision of appropriate data rate.

While these problems have been extensively researched, there are still many challenges to address to maximize the performance of wireless networks. This dissertation aims at examining limitations of the current RA schemes and developing effective rate control algorithms which enable responses to varying channel conditions.

Despite of considerable efforts to classify the causes of packet failures in wireless communication, challenges still remain in differentiating between *In-range* collision and hidden collision-induced packet failure. In addition, as the new IEEE 802.11n standard employs new techniques which are introduced in Section 5.2, most of the MIMO-enabled RAs proposed in the literature show limited improvement because they only focus on designing adaptive use of Spatial Multiplexing based on the MIMO feature and cannot fully profit from the new enhancements. Specifically, though Frame Aggregation aimed at amortizing the protocol overhead plays a key role in achieving intensive throughput gain, the need of active control in the aggregation level has not been a major research focus. It is obvious that RAs should consider the impact of Frame Aggregation as a primary design criteria.

3.2 Research Questions

The goal of this work is to design and implement efficient Rate Adaptation algorithms by addressing the following research questions.

- For static environments where the channel is relatively stable and an optimal fixed rate exists, how does the proposed solution effectively find and switch to the best rate while maintaining adaptability to unpredictable short-term channel fluctuation with minimal overhead?
- For mobile environments where the channel rapidly changes over time due to movement, how does the new algorithm effectively adapt to the varying link conditions to obtain an instantaneous optimal rate?
- For congested environments where multiple stations compete for the channel occupancy, how does the RA scheme intelligently distinguish the potential reason for losses in order to employ adaptable responsive actions?
  - If the channel error is caused by *In-range* collision, what is the appropriate response for the solution to activate?
  - If the channel error is caused by hidden station collision, when and how long does the RA protocol initiate RTS/CTS handshake as the solution to the hidden station problem?

In order to resolve these questions, this dissertation carries out extensive experiments and reveals several significant insights, then proposes two Rate Adaptation algorithms based on answers to the questions listed above.

Chapter 4

Rate Adaptation with Collision Differentiation for IEEE 802.11 WLANs

4.1 Introduction

Since the IEEE 802.11 Wireless Local Area Network (WLAN) has emerged as a dominant and promising technology for wireless access for mobile/portable devices, the increasing amount of wireless traffic leads to poor channel qualities. Channels deteriorate for four main reasons: (1) signal fading, (2) *In-range* collision, (3) hidden collision, and (4) ambient interference.

Signal fading is caused either by signal propagation loss by distance or *Multi-path fading*, which occurs when multiple paths exist between a transceiver and receiver pair and identical signals arrive at the receiver-side with a delay (called *Delay Spread*). The impact of multi-path fading can be either constructive or destructive depending on how the two duplicated symbols combine. On the other hand, *In-range* and hidden collision involve additional stations besides the transmission pair. While *In-range* collision is attributed to simultaneous transmissions among stations within wireless range, hidden collision is caused by mutually unrecognized transceivers. Finally, interference is caused by random signals which the involving standard cannot decode.

These causes cannot be eliminated due to the unstable nature of the wireless medium. Therefore, dynamic traffic control is necessary to make wireless communication more flexible. Recent research in the IEEE 802.11 standard improved the PHY specification that enables the standard to provide multiple data rates. Thus, stations in WLAN can select appropriate transmission rates according to the current channel conditions to achieve higher throughput.

Rate adaptation helps a transmitter choose the most appropriate transmission rate to maximize performance based on the current channel quality. The fundamental challenge of rate adaptation algorithms is how to estimate, classify and respond to varying channel conditions. When a channel is estimated to be unstable, a rate adaptation algorithm classifies the cause of channel instability and then responds with pre-defined reactive procedures. It is important for rate adaptation to properly diagnose the cause of channel degradation, because the appropriate action (e.g., increasing or decreasing transmission rate) differs based on the cause of channel instability. Many recent studies differentiate collisions from other causes using Request to-Send (RTS) probing [34, 35, 8] on the assumption that the chance of an RTS packet error is negligible due to its small size and robustness.

Hence, an RTS packet transmission fails only when a collision occurs. In this case, rate adaptation algorithms do not decrease the transmission rate because decreasing the rate will actually increase collisions due to the longer transmission time at a lower data rate. On the other hand, when a data frame (that is transmitted after a successful RTS packet) fails, rate adaptation algorithms determine that the previous error was caused by signal fading or interference and reduce the transmission rate. Additionally, in the IEEE 802.11 standard, the Exponential Back-off Algorithm requires every station to wait a random back-off period before retransmitting or sensing a busy channel. This algorithm increases the Contention Window size exponentially when a packet collision occurs. The problem is that this algorithm cannot distinguish the cause of packet losses. Therefore, a transmission failure always doubles the Contention Window size. When packet failures are caused by collision from hidden stations and channel disorder, the CW size should not be increased because these causes of packet failure are not related to the ones that the Exponential Back-off Algorithm is designed to avoid and respond to.

In-range collisions are defined as a collision caused by stations within the same wireless transmission range. This type of collision occurs when more than two transmitters within a wireless range are assigned to the same *back-off* slot of the Contention Window. On the other hand, hidden collision is defined as a collision caused by hidden stations that are

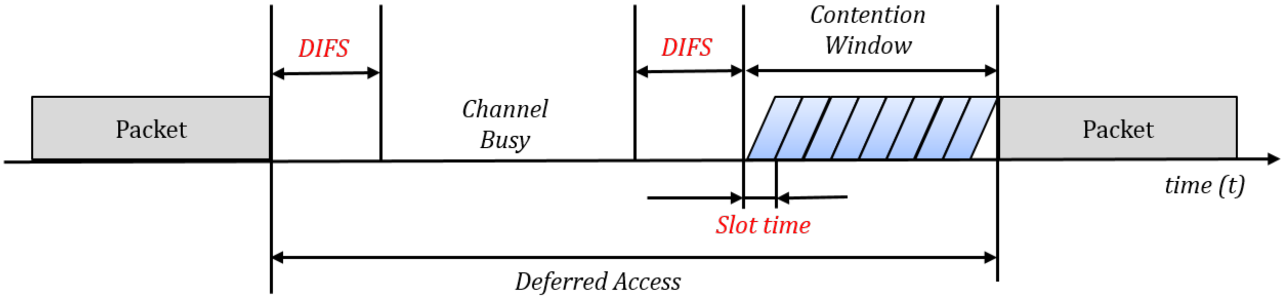


Figure 4.1: Distributed Coordination Function in IEEE 802.11 Standards

out of a transmitter's range but in the receiver's range. These collisions happen when the out-of-range transmitters are unaware of the media usage and try to send data frames.

In this paper, a novel rate adaptation algorithm *Rate Adaptation with Collision Differentiated (RACD)* is proposed. This algorithm distinguishes *In-range* collisions from hidden collisions. Moreover, *CTS-to-self* packet is used to differentiate hidden collisions from *In-range* collisions. Because a *CTS-to-self* packet is shorter than RTS/CTS packets, the decision can be made faster than with previous rate adaptation algorithms. Further, RACD combines rate adaptation with the Return to 0 on Channel Errors (R0CE) scheme presented in [43] to maximize throughput.

The rest of this chapter is organized as follows. Related work is introduced in Section 4.3. Section 4.4 details our new algorithm and Section 4.5 presents the simulation results. Then the chapter is summarized in Section 4.6.

4.2 Background

4.2.1 Distributed Coordination Function (DCF)

As opposed to a wired network which is able to detect packet collision, wireless networks cannot identify whether an error is caused by contention or channel disorder. In order to avoid collisions, the IEEE Working Group adopted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as a basic channel access protocol. However, CSMA/CA is not sufficient to address the problem effectively because multiple transmission attempts can be started right after the carrier sensing. Therefore, Distributed Coordination Function (DCF) is proposed to combine the CSMA/CD with several time intervals such as DIFS, SIFS and Back-off slots.

Figure 4.1 illustrates how DCF works in IEEE 802.11 WLANs. Upon a transmission request, a transceiver checks whether the channel is idle. If there is no ongoing transmission, it waits for DIFS, then senses the channel again. If the channel is still idle, the transceiver selects a random number within a range of Contention Window (CW) and defers access by the chosen time interval. In DCF, if a transmitter fails to transmit a packet and attempts retransmission, this failure is assumed to have occurred due to a collision and DCF doubles the Contention Window size up to predefined maximum Contention Window size. Note that regardless of all the collision avoidance mechanisms listed above, collisions occur when multiple stations are assigned to the same random back-off number (*In-range collision*). In addition, the ongoing signal can be corrupted when another transmitter which is unaware of the current transmission attempts to send packets (*hidden collision*).

4.2.2 CTS-to-self Protection Mechanism

CTS-to-self is a protection mechanism for secure communication and compatibility introduced in the interest of maximizing performance in IEEE 802.11 standard. Under the IEEE 802.11 specifications, all stations compete with each other to gain access to wireless channels. Due to the half-duplex communication characteristic of wireless communication, a collision is not detectable but just avoidable. In order to avoid collisions, a station senses a channel before sending the first packet. If the channel becomes busy, it waits for a Distributed Inter Frame Space (DIFS) with a random period of *back-off* time determined by an assigned number which ranges from zero to the size of *contention window*. As the number of stations grow, the probability of collisions increases. This increased probability is why the IEEE 802.11 standard doubles the size of the *contention window* up to a predefined maximum

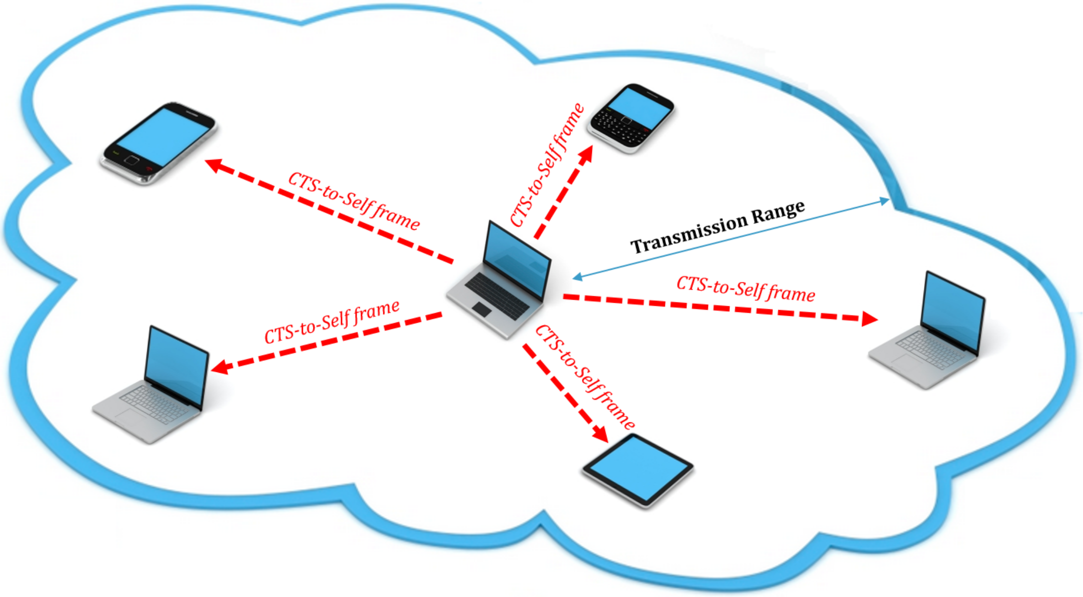


Figure 4.2: *CTS-to-self* mechanism is newly introduced in 802.11 specification for secure communication.

window size to reduce the chance of collisions. However, it cannot reduce collisions coming from hidden stations because the Clear Channel Assessment (CCA) introduced earlier cannot detect the signal from hidden stations. The RTS/CTS mechanism addresses this problem by reserving the channel within the range of hidden stations at the expense of communication overhead.

4.3 Related Work

Since Section 2 introduces and categorizes various RA algorithms, this section focuses on detailed explanation of research related to the RACD algorithm and simulative experiments presented in Section 4.5.

Auto Rate Fallback (ARF) [5] is the first Rate Adaptation algorithm and has been widely accepted for legacy WLANs because of its simplicity and ease of implementation. If there are two consecutive packet failures, which means that the transceiver does not successfully receive two ACK packets in a row, ARF automatically switches to a rate one step lower. It also switches to the next higher rate either after ten consecutive successes in transmission or after a pre-defined time interval. ARF is based on the general assumption that there is a monotonic relationship between rate bandwidth and Packet Error Rate (PER).

Adaptive Auto Rate Fallback (AARF) [6] extended the idea of ARF to improve adaptability to time-varying channel conditions. Specifically, AARF keeps changing the thresholds of consecutive success and failure at run-time. The thresholds increase using the Binary Exponential Back-off algorithm, which multiplies the thresholds by two upon the failure of a rate change attempt. In addition, it adapts rate update time intervals when stable channels hardly fluctuate.

Collision-Aware Rate Adaptation (CARA) [34] is another extension of ARF, but it focuses on differentiating collision-induced packet losses from fading-induced packet losses and handling fading-induced losses with RTS/CTS control packets. Based on the robustness of the short-length RTS/CTS packet using the lowest rate, CARA assumes that failed RTS/CTS handshake implies that the previous loss resulted from a collision. In other words, upon a packet loss, CARA initiates RTS/CTS handshake before retransmitting the missed packet. If the transmitter fails to receive data ACK after the handshake where the channel is exclusively reserved and stable enough for the lowest rate, the packet loss was due to an unsuitable data rate. It is then required to adjust the transmission rate. CARA also suggests a supplement detection mechanism using Clear Channel Assessment (CCA) measurements.

Robust Rate Adaptation Algorithm (RRAA) [8] is a MAC-layer statistic-based RA scheme enhanced with collision differentiation capability. RRAA is composed of three modules: Loss estimation, Rate Change, and Adaptive RTS filter. The *Loss estimation* module counts the number of packet losses for a short time window and computes Packet Error Ratio for channel quality assessment. The *Rate Change* module decides whether to increase or decrease data rate based on the PER estimation, determined by comparing the Max Tolerable Loss threshold (MTL) and the Opportunistic Rate Increase Threshold (ORI). Finally, the *Adaptive RTS (A-RTS) filter* selectively alternates the RTS/CTS option to mitigate collision-induced packet losses. Whenever a packet loss occurs even with RTS/CTS exchange, A-RTS filter increments the size of the RTS window by one for the RRAA to suppress severe collision losses.

As described in Section 4.2.1, Contention Window in DCF is used to mitigate the impact of *In-range* collisions. As the window size grows, the chance of collision-induced failure drops accordingly, at the expense of delay in channel access. As a result, dynamic Contention Window adjustment is required in addition to Rate Adaptation. M. Kim and C. Choi [44] initialized the Contention Window to the minimum available value whenever hidden collisions are identified based on their detection algorithm, while doubling the CW upon *In-range* collisions. P. Chaporkar et al. [43] builds and analyzes a mathematical model and shows that initiating the Back-off state to zero upon a packet loss improves the performance of Rate Adaptation algorithm.

4.4 Algorithm Design

In this section, a novel rate adaptation algorithm called Rate Adaptation with Collision Differentiation (RACD) is proposed; this algorithm chooses the most appropriate transmission rate according to current channel quality to maximize the throughput of the IEEE 802.11-based wireless networks. In order to achieve this goal, our algorithm exploits *CTS-to-self* to distinguish *In-range* collisions from hidden collisions. Section 4.2.2 analyzes the effect of *CTS-to-self* with a brief introduction to DCF. The detailed design of the RACD algorithm is presented in Section 4.4.1.

4.4.1 Rate Adaptation with Collision Differentiation (RACD)

CTS-to-self is a newly defined mechanism to reduce this overhead. It lets a transmitter set the destination address of a packet to its own address so that other stations within range do not initiate their transmissions for a specified duration. Figure 4.2 shows the packet exchange procedure in the *CTS-to-self* mechanism. Once the *CTS-to-self* packet from station A is overheard by other stations within range, stations B and C defer their transmissions

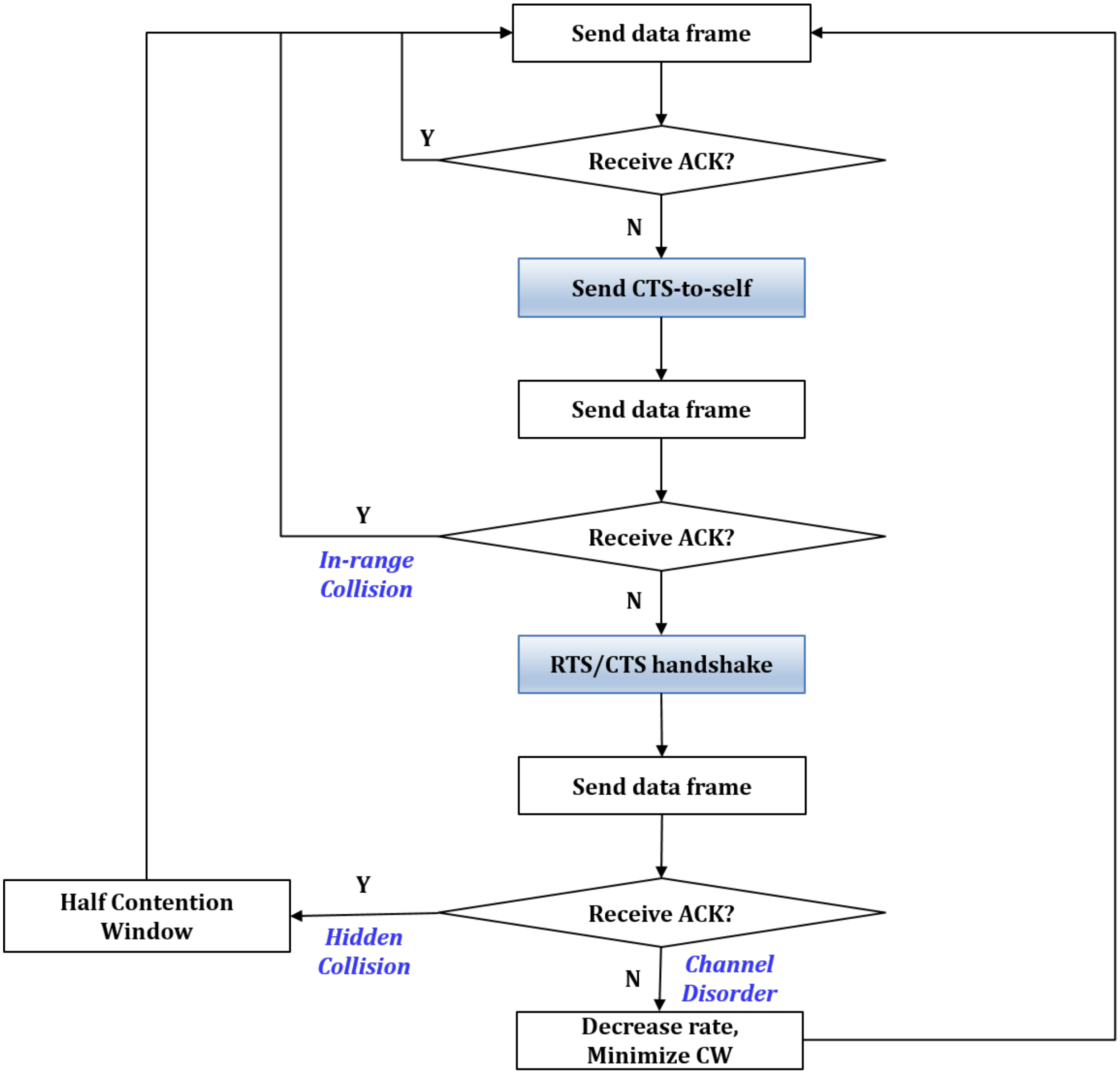
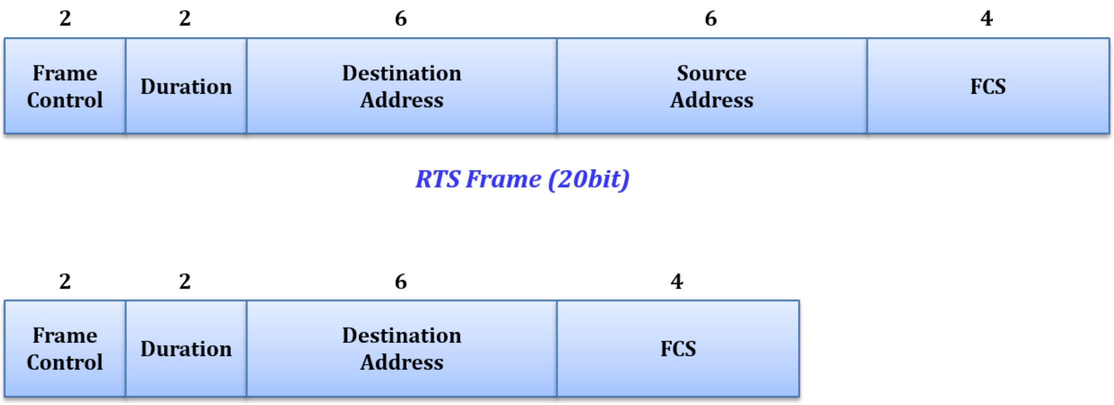


Figure 4.3: Flowchart of Proposed RACD Channel Status Differentiation



RTS Frame (20bit)

CTS and CTS-to-self Frame (14bit)

Figure 4.4: Packet Size Comparison

for the duration described in the *CTS-to-self* packet. The equation below compares the overhead of the two protection mechanisms and shows that the total transmission time of *CTS-to-self* mechanism is shorter than the RTS/CTS mechanism.

According to the packet format definition (IEEE 802.11) shown in Figure 4.4, the sizes of RTS, CTS, and *CTS-to-self* packets are 20 bits, 14 bits and 14 bits respectively. Given the basic transmission rate, 1 Mbps, the transmission time of RTS/CTS and *CTS-to-self* are 20μs and 14μs respectively. As the SIFS is defined as 10μs in IEEE 802.11, it turns out that $t_{RTS/CTS} = 54\mu s$ and $t_{CTS-to-self} = 24\mu s$. The *CTS-to-self* transmission time is reduced by 30μs, which is 55% of RTS/CTS probing time.

$$t_{RTS/CTS} = t_{RTS} + t_{CTS} + 2 \times SIFS \tag{4.1}$$

$$t_{CTS-to-self} = t_{CTS} + SIFS \tag{4.2}$$

On the other hand, the *CTS-to-self* mechanism can only avoid *In-range* collisions, not hidden collisions. Therefore, a RTS/CTS packet is used to avoid hidden collisions when *CTS-to-self* does not make a successful probe.

Table 4.1: Simulation Parameters

Simulation Parameter	**Value**
Number of Stations	$10 \sim 50$
Min Contention Window Size	15
Max Contention Window Size	1023
Data Frame Size	2Kbytes
Reference Loss	46.67dB

As discussed earlier, channel quality can be impacted by signal fading, collisions, and interference. Collisions occur either because of *In-range* stations or hidden stations. *In-range* collisions take place when two packets depart from different *In-range* stations at nearly the same time, and the packets do not arrive at the destination. The hidden collision is driven by overlapped signals from out of the wireless range area. Though these collisions have different underlying reasons, they are treated as the same cause of channel degradation in the literature. It is assumed that when packets collide, the collision affects more than two transmissions because there is a high probability that the colliding packet is longer than the entire transmitted packet and may also collide with subsequent *CTS-to-self* packets. The colliding packet is much shorter than the transmitted packet in only a small number of cases. In addition, this assumption is typically used in the RTS probing methods [34, 35] to distinguish a collision-based failure from channel-error failure.

RACD exploits the benefit of the *CTS-to-self* function to differentiate the types of collisions and adjusts the transmission rate and Contention Window size. As shown in Figure 4.3, the RACD algorithm activates when a transmitter fails to get an acknowledgement, i.e., when a packet failure occurs. It divides the channel states into (1) *In-range* collision, (2) hidden collision, and (3) *channel error* in cooperation with *CTS-to-self* and RTS/CTS packets.

First of all, RACD determines that the previous data loss was caused by an *In-range* collision when a data packet is transmitted successfully after reserving the channel with a *CTS-to-self* packet. Therefore, it increases the Contention Window size exponentially.

Second, if the data frame after *CTS-to-self* fails, the algorithm activates the RTS/CTS handshake. If the transmitted data after the RTS/CTS packets is not acknowledged, the algorithm regards the previous failure as being caused by a hidden collision. Though the hidden collision has no impact on the *back-off* counter, according to DCF, the Contention Window size is doubled after every packet loss. To address this issue, the RACD algorithm returns to the previous *back-off* stage by halving the size of Contention Window in the hidden collision state.

Finally, if there is another data frame loss after a successful RTS/CTS exchange, the RACD scheme decides that the channel suffers from fading or interference and thus requires rate adjustment. This decision is based on the assumption that the RTS/CTS transmission has negligible chance to be interrupted by channel disorder due to its short length and robustness. As a result, the transmission rate is decreased by one unit during *channel error* state. In addition, as mentioned in Section 4.3, RACD utilizes the idea in [43] to maximize throughput. Our scheme initializes the *back-off* stage to 0 and sets the Contention Window size to the minimum.

In this paper, the main focus is to find how to adapt to the channel when the channel condition grows worse. Hence, when the channel condition becomes better, e.g., when the number of consecutive successful packets reach a certain threshold, the simple ARF approach is adopted to change the rate to the next higher rate.

4.5 Simulation Results

4.5.1 Simulation Setup

The "NS-3"[45] simulator is used in our experiments to evaluate the efficiency of the algorithm which implements the *CTS-to-self* mechanism and modifies the inherent DCF function modules to support Contention Window adjustments. The PHY layer is set to WIFI_PHY_STANDARD_holland, which is used in RBAR [17] and implemented in NS-3 [45] as one of the basic configurations. In order to simulate multi-path fading effects, a constant

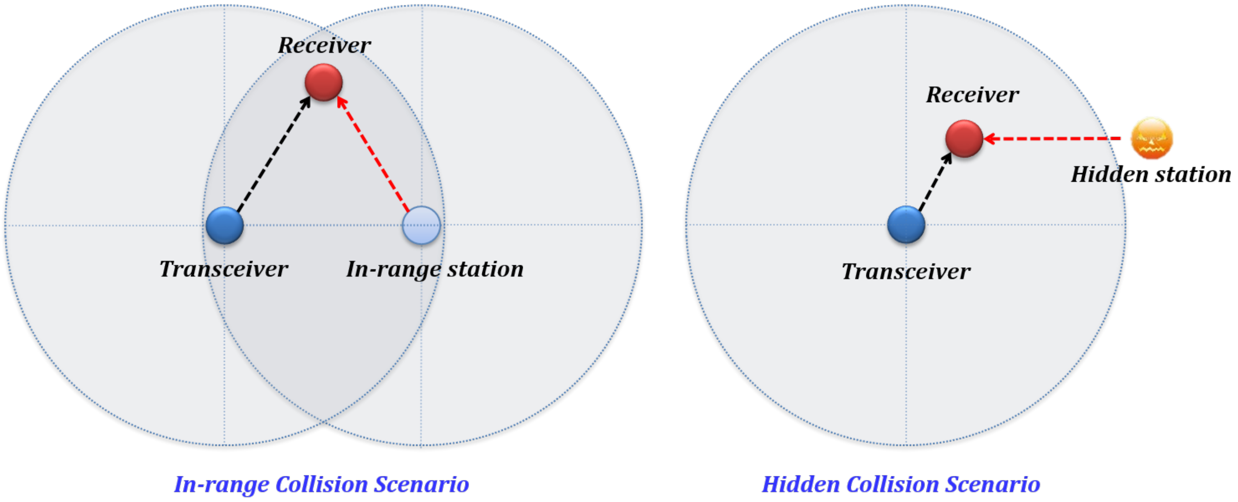


Figure 4.5: *In-range* (left) and Hidden Collision Scenario (right)

propagation delay model was used. The propagation loss was based on the *Log distance model* with a reference loss of 46.67 dB at a distance of 1m. Then all stations generated 2Kbyte data packets and sent the data packets to a single destination. Each simulation result was averaged over 20 repetitions of the experiment.

Two different scenarios were simulated: (1) an *In-range* collision scenario without hidden stations and (2) a hidden collision scenario. For the *In-range* collision scenario, a varying number of contending stations were randomly placed within a 50m-radius circle around the receiver. In the hidden collision scenario, varying number of hidden stations were randomly deployed out of the transmitter's communication range but within the receiver's communication range. The simulation results were compared with ARF [5], AARF [6], CARA [34] and RRAA [8]. In order to evaluate the effect of the Contention Window adjustment in the new algorithm, experiments with the Contention Window adjustment function turned off were conducted and compared with the original RACD.

4.5.2 Throughput Comparison with Different Schemes

Figures 4.6 and 4.7 show the averaged throughput results from different rate adaptation algorithms. Throughput is defined as the number of successfully received packets divided

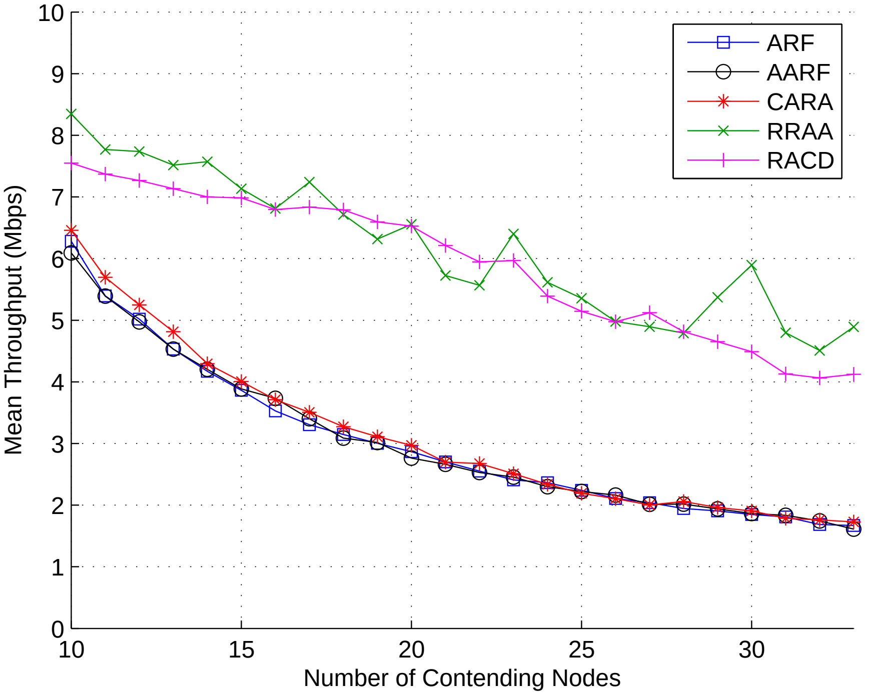


Figure 4.6: *In-range* Collision Scenario

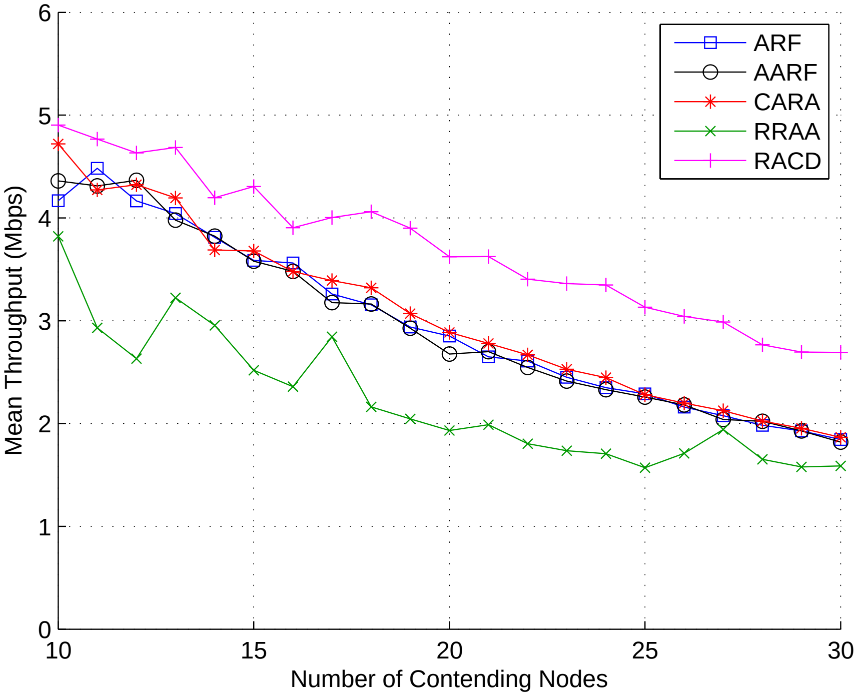


Figure 4.7: Both *In-range* and Hidden Collision Scenario

by the number of packets transmitted. To get more generalized results, the mean throughput presented in these experiments is an accumulated number of correctly received frames counted by each destination station divided by the number of stations.

Typically, all five algorithms showed a trend of throughput degradation as the number of contending stations increased. As shown in Figure4.6, RRAA performed the best out of these five rate adaptation solutions. ARF, AARF and CARA deliver almost the same throughput. RRAA outperforms RACD by 11% and the other algorithms by an average of 128% due to the effectiveness of its adaptive RTS filter.

On the other hand, as shown in Figure4.7, when there were hidden stations typical in a wireless communication environment, our RACD scheme performed best. Although RRAA was efficient in the *In-range* collision scenario, its inefficiency in the hidden station scenario shows that the adaptive RTS filter does not effectively deal with collisions caused by hidden stations. RACD gave a 67% throughput enhancement compared to RRAA and 22% compared to CARA, which exhibited the third best performance. This throughput improvement proves that our proposed RACD algorithm can effectively respond to both *In-range* and hidden collisions because of the *CTS-to-self* probing mechanism's effectiveness in differentiating two types of collisions.

4.5.3 Impact of Adjusting Contention Window

In order to evaluate the impact of modifying the DCF function by adjusting the Contention Window size, the same simulations were performed with only *CTS-to-self* mechanism without modifying Contention Window adjustments. As can be seen in Figures 4.8 and 4.9, in both cases, the scheme with Contention Window adjustment represented better throughput. The original RACD outperformed RACD without DCF adjustment by 64% and 14% for the *In-range* and hidden collision scenario respectively.

In addition, it is interesting to see that RACD without DCF adjustment performed better in the hidden station scenario than in the *In-range* scenario by an average of 12%.

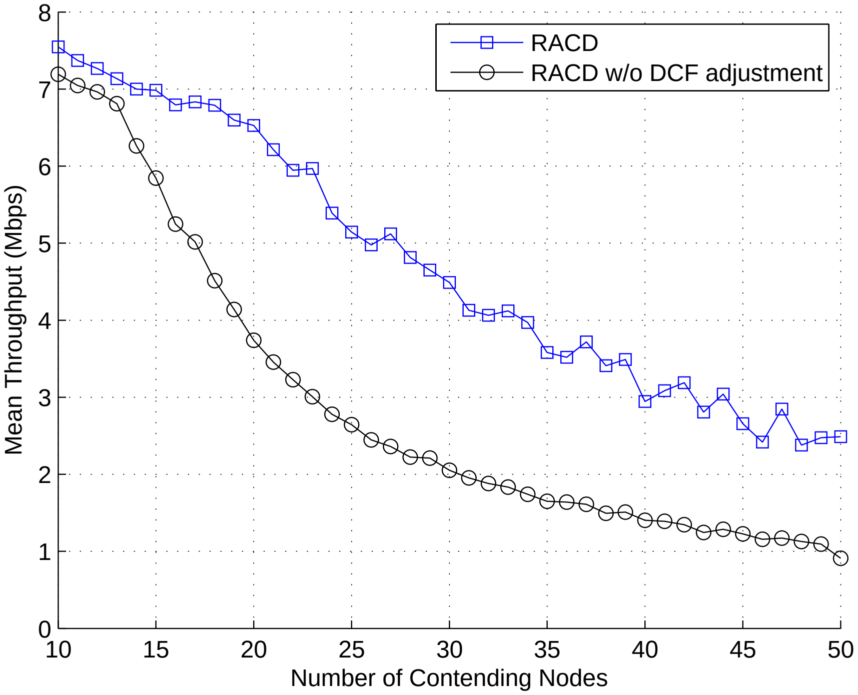


Figure 4.8: *In-range* collision Scenario Comparison

This finding confirms that the new Contention Window control scheme significantly improves the performance of our algorithm when hidden stations cause collisions.

4.6 Summary

This chapter proposed a novel Rate Adaptation with Collision Differentiation (RACD) algorithm for the IEEE 802.11 WLANs which reduces the effect of variability in wireless channels and improves data throughput. Our new scheme takes advantages of the newly standardized *CTS-to-self* mechanism and Contention Window adjustment in DCF. Simulation results show that RACD achieved a significant performance improvement in general communication environments compared to existing rate adapting schemes. The utility of *CTS-to-self* for discriminating *In-range* collisions from hidden collisions is shown through our experiments.

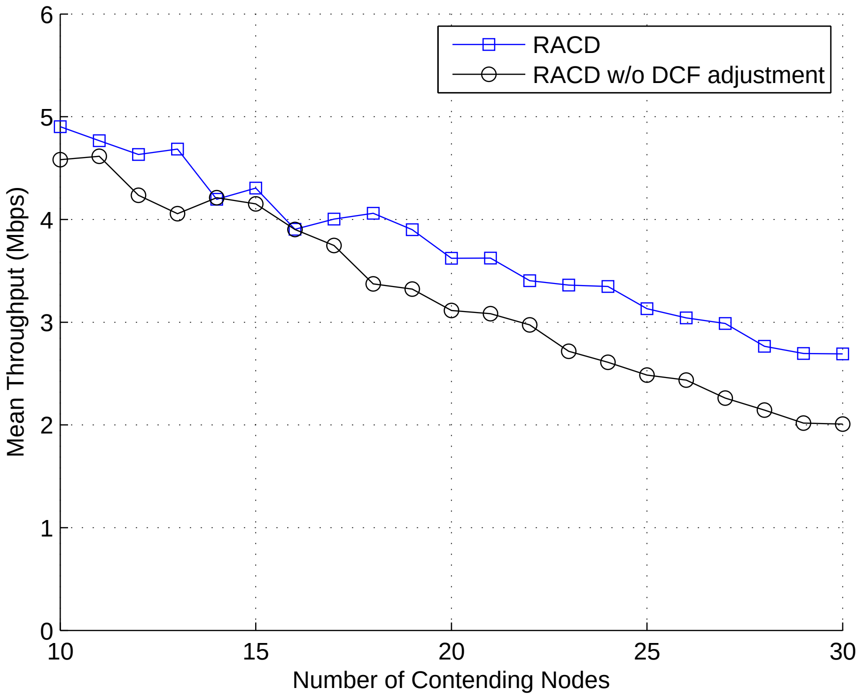


Figure 4.9: Both *In-range* and Hidden Collision Scenario Comparison

Future work should include implementing and testing our scheme in a real WLAN environment. In addition, further research work should attempt to test and analyze how this scheme affects other metrics such as fairness and global network throughput. By utilizing the *CTS-to-self* packets for distinguishing *In-range* collisions from hidden collisions, various applications can be enhanced for different purposes in various IEEE 802.11 WLANs.

Chapter 5

Aggregation Aware Rate Adaptation in IEEE 802.11n WLANs

5.1 Introduction

With the growing demand for high bandwidth data transmissions, the IEEE 802.11n standard paves a way for a new era of high bandwidth in wireless local area networks (WLANs). The foremost enhancement over the previous standards is achieved by Multiple-Input Multiple-Output (MIMO), Channel Bonding, and Frame Aggregation (FA) & Block Acknowledgement (BlockACK). By utilizing multiple antennas for transmitting data, MIMO technology can improve the transmission reliability by sending duplicated signals (*Spatial Diversity*), or can expand network bandwidth by simultaneously transmitting independent signals in parallel (*Spatial Multiplexing*).

In addition to MIMO, the new standard also doubles the network performance by combining two adjacent channels so that the participating stations use a wider 40MHz channel bandwidth instead of the basic 20MHz channel bandwidth. While the enhancement by MIMO and Channel bonding is achieved by increasing the resource utilization, Frame Aggregation (FA) & Block Acknowledgement (BlockACK) improves the performance by amortizing the network protocol overhead. Instead of sending a single frame per protocol overhead, the 802.11n stations can transmit multiple frames with a single protocol header. These enhancements improve network throughput up to 600 Mbps at the physical layer in a 40MHz channel with the maximum of four streams, and offer 32 available transmission rates to choose from.

Since there is no pre-defined specification in the standard illustrating how to select the data transmission rate, it is essential to have an Rate Adaptation (RA) process which selects the appropriate transmission rate to the current channel quality in order to exploit the benefit of this novel technology. A number of studies have been carried out with the goal of accomplishing the maximum network performance in varying channel conditions. Most of the rate adaptation schemes in the previous 802.11 standards, such as 802.11a, b, and g, focus on adjusting the Modulation and Coding Scheme (MCS) according to the channel quality; in contrast, the 802.11n rate adaptation algorithm should not only be able to adjust MCS, but also be able to determine the number of streams, identify channel bonding availability, and decide on the level of aggregation. Moreover, IEEE WGn is developing a new IEEE 802.11ac standard that further expands the concept of wireless interface of 802.11n by increasing the number of available streams to eight and utilizing wider channel bonding of up to 160MHz. Therefore, the number of degrees of freedom on Rate Adaptation is drastically increased.

The most distinctive problem in the new standard compared to the legacy RAs arises from the Frame Error Ratio (FER) non-monotonicity [1], which means that there is a non-monotonic relationship between the transmitting data rate and FER. In Single Input Single Output (SISO) systems, the FER is usually in proportion to the data rate, i.e., transmitting at a lower rate guarantees few packet failures. However, due to the added data streams, the MIMO-enabled stations exhibit non-monotonic behavior under certain scenarios. Most of the MIMO-aware RAs focus on resolving this non-monotonicity resulting from added space of freedom.

In addition to MIMO and Channel Bonding, the FA & BlockACK scheme is one of the key enhancement factors in the 802.11n standard. Much research has analyzed the impact of FA to leverage its benefit; however, a little work has been done on inter-relationships between FA and RA.

This chapter evaluates the impact of FA from a different perspective of RA schemes and suggests the need for frame aggregation level control for RA algorithms. Then using an efficient way of probing enabled by frame-level adjustment, a new Aggregation Aware Rate Adaptation algorithm is designed and evaluated to balance trade-offs between maintaining stability and agility to channel dynamics.

Table 5.1: Comparison between 802.11n and 802.11ac Standards

Standard	MIMO Streams	Channel Bonding	Available Modulation	Max Throughput
IEEE 802.11n	1-4	20-80 MHz	BPSK-64QAM	600 Mbps
IEEE 802.11ac	1-8	20-160 MHz	BPSK-64QAM, 256QAM	1300 Mbps

The remainder of this chapter is organized as follows. The background of the topic is introduced in Section 5.2 with related work reviewed in Section 5.3. Section 5.4 gives an overview of the following research work. Section 5.6 reports the findings to provide rationales behind the AARA algorithm which is detailed in Section 5.7. Detailed experiment results are presented in Section 5.8. Section 5.9 summarizes the chapter.

5.2 Background

Following intensive efforts to enhance wireless network performance, the IEEE Working Group released new amendments to the existing 802.11 standards. While the best legacy standard has potential optimal throughput up to only 54 Mbps, the new standards such as 802.11n and 802.11ac are designed to transmit a maximum of 600 Mbps and 1300 Mbps respectively. These novel enhancements come from three representative new techniques: Multiple Input Multiple Output (MIMO), Channel Bonding, and Frame Aggregation. Except one level higher order modulation, which is 256 QAM, most of the other improvements over 802.11n result from the multiplication of resources, as shown in Table 5.1.

This section briefly describes these three key MAC-layer enhancements, focusing on Frame Aggregation, which is key to achieving better performance with the proposed algorithm.

5.2.1 Multiple Input Multiple Output (MIMO)

MIMO technology is the key underlying model of IEEE 802.11n architecture and lays the foundation of higher network performance for standard-compliant stations. It leverages multiple antennas for both transmitter and receiver and takes advantage of the multipath effect which is inherently perceived as one of the interferences. MIMO operates in two different modes, *Spatial Diversity* (SD) based on single stream (SS), and *Spatial Multiplexing* (SM) using multiple streams (MS). *Spatial Diversity* is the technology that allows corresponding stations to transmit and receive duplicated information through multiple streams to improve reliability and sustainability over longer distances. The diversified identical signals encounter different spatial structures and experience varied fading, so the receiving station is able to compensate for the defects by comparing two decoded signals.

On the other hand, *Spatial Multiplexing* makes use of multiple antennas to send independent, separately encoded signals for better network performance. The network throughput can be easily multiplied by the number of streams the related stations utilize. Even though the current 802.11n standard specifies a maximum of four streams, increasing the available number of streams can be one of the simplest and most convenient ways to enhance network utilization, so the next generation 802.11ac standard allows for maximum of eight streams.

5.2.2 Channel Bonding

Similar to the way that increasing the number of possible streams improves spectral efficiency, *Channel Bonding* expands the channel bandwidth for wider utilization of the channel medium. While the legacy 802.11 standards operate only 20MHz channel bandwidth, the 802.11n standard extends the operable bandwidth to 40MHz. Therefore, it theoretically doubles the data transmission rate by utilizing one more adjacent channel band; however, for backward compatibility with legacy standards, it also operates with 20MHz.

5.2.3 Frame Aggregation and BlockACK

Even with the assumption of infinite data rate increase, the maximum potential throughput is known to be bounded at 50 Mbps due to protocol overhead by the analysis estimated in [46]. The key concept behind Frame Aggregation (FA) is reducing the frequency of wasted

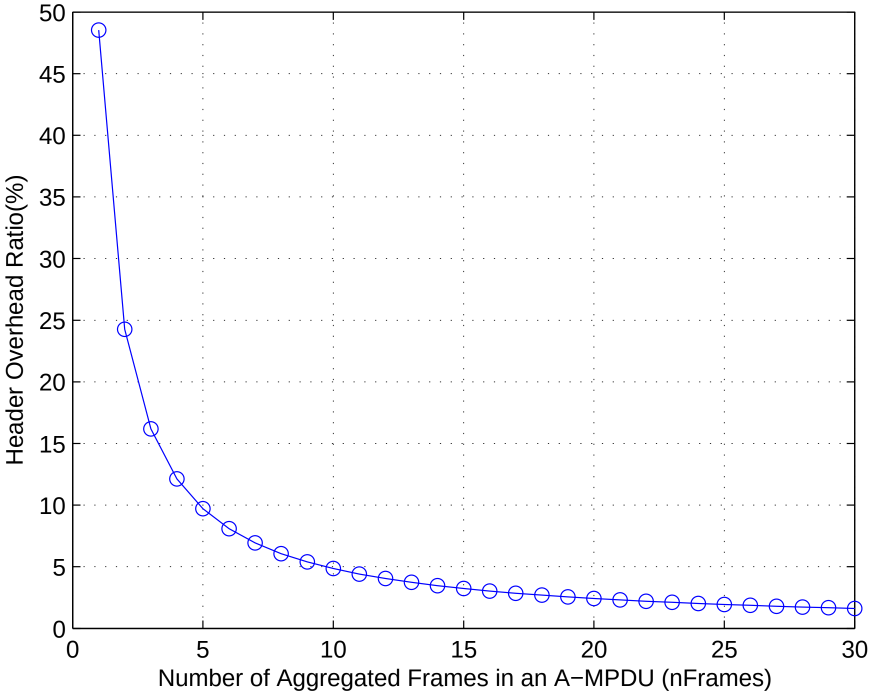


Figure 5.1: Amortized Header Overhead according to Aggregated Number of Frames

time incurred by MAC and PHY headers. As a result, the FA scheme is essential to achieve the target maximum performance specified in the standard.

In general, the MAC header is shorter than data frames; however, since the standard requires the headers to be transmitted at the one of the lowest rates to ensure transmission reliability, the overhead caused by protocol header is relatively large in legacy standards. In order to reduce this overhead, the new MAC layer of the 802.11n standard combines multiple frames to amortize the cost of non-data transmission overhead.

As can be intuitively perceived, the amortized overhead significantly decreases with the growth in the number of sub-frames of an aggregated packet. Figure 5.1 reveals that irrespective of the non-trivial protocol overhead, as the level of aggregation increases when the data rate is 104 Mbps, the ratio of the packet header transmission time to actual data transmission time drops significantly from 48% to only 1.5%.

IEEE 802.11n specifies two different aggregation mechanisms as illustrated in Figure 5.2. First, Aggregated Mac Service Data Unit (A-MSDU) concatenates multiple MAC Data Service Units (MSDUs) for a single Mac Protocol Data Unit (MPDU). A sub-frame header and padding bytes are attached to each MSDU and a collection of sub-frames completes one A-MSDU, then a common MAC header Frame Check Sequence (FCS) is added to the end. Though A-MSDU maximizes the spatial efficiency even better than A-MPDU, due to the fact that each sub-frame does not have an individual FCS to check and correct possible errors, a single point bit failure within a sub-frame can corrupt the entire aggregated A-MSDU. This problem explains why A-MSDU is not robust enough for time varying, error-prone channels.

On the other hand, as a dominant difference from A-MSDU, a sub-frame in an Aggregated MAC Physical Data Unit (A-MPDU) has an individual MAC header and FCS, as depicted in Figure 5.2. Although A-MSDU exhibits higher analytic saturation throughput than A-MPDU structure, due to the vulnerability to channel dynamics, A-MPDU equipped with better protection is more appropriate in practice.

In the Frame Aggregation scheme, the Block Acknowledgement (BlockACK) mechanism enables the aggregated MPDUs to be acknowledged by a single ACK packet instead of sending a separate ACK for each MPDU. Since the BlockACK contains a 64 bit long data structure where each bit represents the individual status of an MPDU, which is either success or failure, both A-MSDU and A-MPDU combines a maximum of only 64 sub-frames.

5.3 Related Work

In addition to MIMO RAs introduced in Section 2, this section elaborates a few representative MIMO RAs used for evaluation of the newly proposed AARA algorithm and introduces research efforts on Frame Aggregation.

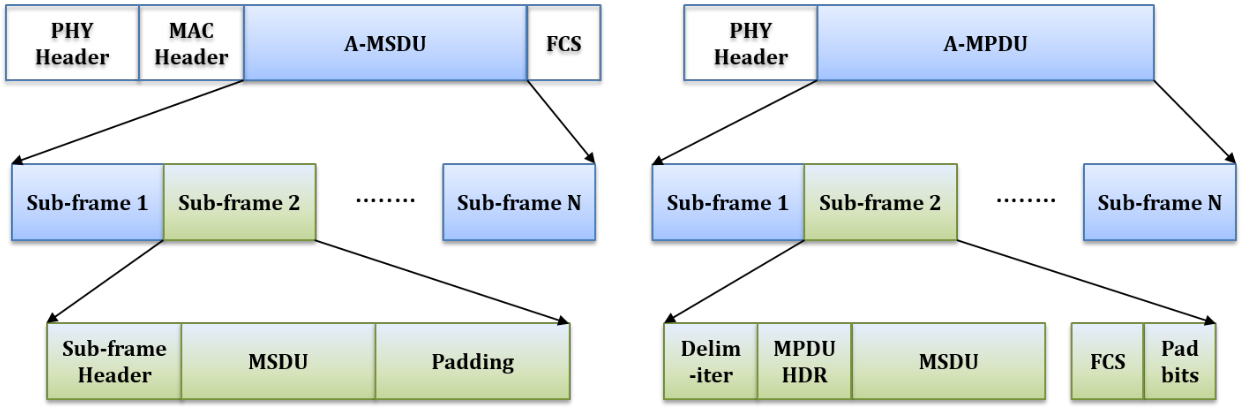


Figure 5.2: Aggregated Frame Structure: A-MSDU (Left), A-MPDU (Right)

5.3.1 MIMO-aware Rate Adaptations

Thanks to the new features introduced in the IEEE 802.11n standard, MIMO-enabled Rate Adaptation algorithms have an array of choices in deciding the best option.

RAMAS [30] divides these design choices into two categories, *modulation group*, and *enhancement group*, and each group employs an independent set of rules to determine the best option inside each group. The modulation group only takes modulation type and coding scheme into consideration so that a *monotonic relationship* between FER and MCS indices can be assumed.

The modulation group keeps track of the number of success, failures and retried MPDU frames for 30 MPDU frame transmissions. It decreases the group index either when the number of retried frames is more than the number of successful transmissions or when the error rate is less than a pre-defined credit value. Based on the guided parameters, it decreases the rate option within the same modulation group by one whenever there are six frame errors out of 30 transmitted frames. For rate escalation, it employs a credit-based routine. A credit is earned when the error ratio is less than 20%, and RAMAS increments the rate when the credit reaches 10.

The enhancement group is independent from the modulation group and only deals with the additional features for the new amendment, such as the number of streams, guard interval, and channel width. The enhancement group adjustment focuses on deciding whether the current link quality accommodates Spatial Multiplexing transmission based upon Frame Error Ratio (FER).

The open-source Atheros 9K Rate Adaptation (Ath9K RA) [12] is built in the device-drivers for the popular Atheros network interface. It is a MAC-layer metric-based, throughput-based statistical RA algorithm. First, Ath9K RA picks up the valid rate set out of available MCS rates so that they are linearly sorted by available bandwidth. For example, it chooses only 10 data rates out of 16 candidates available in two-stream MIMO adapters without channel bonding.

Ath9K RA sends out a probing packet periodically. Upon measuring the Frame Success Ratio (FSR) which is defined as the ratio of the number of acknowledged MPDUs to the total number of transmitted MPDUs for a specified time interval, Ath9K RA computes throughput for the probing rate by multiplying the FSR and potential bandwidth of the rate. Note that FSR is maintained by the popular Exponential Weighted Moving Average (EWMA) routine to reflect partial performance history. The best performing rate which provides highest throughput is chosen from among the valid candidate rate set. This algorithm also utilizes Multi-Rate Retry (MRR) for adapting short-term extensive channel dynamic. The four available slots in MRR are filled with the next four sequentially lower rates.

Minstrel HT [13] is the upgraded version taking the place of Minstrel for the IEEE 802.11n standard. Similar to Ath9K RA, it constructs MCS rate groups with varying numbers of streams and channel width, probes the channel periodically, collects MAC-layer statistics, calculates throughput, and makes the appropriate rate decision. However, Minstrel HT utilizes the full rate set to take advantage of trade-offs between a Single Stream (SS) mode, higher modulation options and Double Stream (DS) mode lower modulation options.

In addition, it attempts to probe randomly chosen rates instead of sampling the next higher rate. The throughput for Minstrel HT is predicted precisely based on various protocol overheads such as DIFS, transmission time according to the current rate and protocol headers. The noticeable difference is the usage of MRR chain. It puts the best and second best throughput rates in the first and second slots in the chain respectively, and for the third slot, uses the stable rate option which shows the highest successful rate in the SS mode.

5.3.2 Research on Frame Aggregation

As discussed in Section 5.2, the Frame Aggregation (FA) scheme has recently been introduced in the IEEE 802.11n standard in an effort to reduce the frequency of transmitting protocol overhead. Although there are many studies of FA in the literature, many of them [47, 48, 49] focus on exploring a complementary usage between the two aggregation schemes A-MPDU and A-MSDU.

B. Ginzburg and A. Kesselman [49] formulated an analytic model to estimate the potential achievement of each aggregation solution. Additionally, D. Skordoulis et al [47] proposed a hierarchical combination of A-MPDU and A-MSDU schemes for channel efficiency. A simple adaptive scheduler proposed by Selvam T and Srikanth S [48] dynamically switches the type of aggregation scheme according to relevant channel metrics and adaptively determines the level of aggregation. Rather than adapting the level of aggregation, an adaptation algorithm was proposed in [50] to find the optimal frame size in an A-MPDU packet for various channel environments.

Whereas the above approaches focus on aggregation schemes alone, there have been few research attempts to exploit the benefits of controlling Frame Aggregation over Rate Adaptation. Xin He et al. [51] presented an adaptation algorithm which jointly adjusts data rate and the size of individual frame size in an A-MPDU packet to achieve potential optimal throughput. M. Kim and C. Choi [52] suggested using a part of MPDU frames in an aggregated packet for probing the link quality in Rate Adaptation and proposed a rate adjustment direction to overcome the non-monotonic problem of MIMO RA.

In [44], FA is leveraged to differentiate the basic causes of frame failures, which are *In-range* collision, hidden station collision, and channel disorder. This study examines and analyzes both cases for an entire *A-MPDU failure* and partial MPDU frame loss and designs and evaluates a Hidden node Detection (HD) algorithm by simulative study. MiRA [1] is the first practically implemented rate adaptation providing insight into the promising relationship between RA and FA. This study found that higher aggregation level does not always guarantee better performance due to the responsiveness to dynamic channel variation, and it examined the effects of frame loss on the number of aggregating frames in A-MPDU.

5.4 Overview

In order to address the problems listed in Section 3.1 specifically from the perspective of Frame Aggregation, a case study was carried out with well-known existing RAs (Minstrel HT and Ath9K RA). The goal is to provide rationales for the algorithm design provided in Section 5.6 by identifying which design factors affect the performance of RA schemes and how those factors lead to performance variation.

All experiments were conducted with Linux-based equipments over Atheros chipset. Some standard-compliant modifications were made over the Compat-wireless driver to print out statistics in the 802.11n MIMO setting. Iperf [53] was used to collect data and all results were averaged over ten attempts, excluding the best and worst cases.

First of all, the meaning of retransmission under Frame Aggregation scheme was re-examined and the probability of *A-MPDU failure* over different numbers of aggregated frames was analyzed and verified by experiment results.

The influence of the *Multi-Rate Retry* (MRR) mechanism over the FA mechanism is investigated in Section 5.6.3. Although the inherent goal of MRR is to provide extra protection for short-term channel dynamics, MRR inadvertently causes performance limitation when it is used with the Frame Aggregation scheme.

In general, existing RAs estimate the current channel by attempting sample packets for probing. Two legacy probing mechanisms of the existing RAs are reviewed in Section 5.6.1. Problems with these probing schemes are identified and a possible solution is proposed based on the capability to control the level of aggregation.

In Section 5.6.4, the impact of congestion over Frame Aggregation is described. Differences in network throughput and retransmission ratio are illustrated for different congestion scenarios.

Based on the observations from the case study, a new rate adaptation algorithm, called Aggregation Aware Rate Adaptation (AARA), is described in Section 5.7. This section explains a detailed procedure of state transition for frame aggregation control and describes an adaptive probing scheme which can efficiently estimate the channel while avoiding the overhead. It then introduces a collision avoidance mechanism of AARA based on the discussion of A-MPDU retransmission.

The performance results from extensive experiments show that the proposed AARA outperforms the existing RAs. The throughput gain varies from 7.8% to 473.2% in various experimental settings where the channel qualities are different. Performance is enhanced by analyzing the statistics collected, better utilizing of FA, and choosing better rates based on an efficient probing scheme.

5.5 Experiment Methodology

This section introduces the experimental platform for the following case study and performance evaluation, including hardware and software setup, different experiment scenarios, and various performance metrics.

In this section, the experimental platform regarding hardware and software setup, different experiment scenarios and various performance metrics are introduced in detail which are used for the following case study and performance evaluation.

5.5.1 Experimental Platform

All experiments presented in the following sections were carried out with a Linux-based programmable platform equipped with Atheros 2.5/5 GHz MIMO-enabled chipset, which can support 16 MCS indices with a maximum of two streams. The transmission rate can reach up to 130 Mbps when used without channel bonding.

For the access point, Hostapd [54] was installed in a MSI mini-computer [55] for the programmable platform operating in 802.11n with single stream (SS)/multiple streams (MS) modes being enabled. Among the two aggregation options, A-MPDU and A-MSDU, only A-MPDU scheme is supported with BlockACK functionality and Channel Bonding operation was not enabled in the tests.

Three laptops were used in different scenarios as transmitters, and contending stations. Fedora 15 with Linux kernel 2.6.43 was built on each node and a stable version of the Compat-wireless device driver [56] was installed for wireless network communication. Iperf [53] was used to generate saturated back-to-back A-MPDU packet streams in order to estimate the maximum available network throughput. Each A-MPDU packet contained a number of 1500 byte frames based on the transmission rate.

Four different RA algorithms were tested and compared in the following sections. The widely-used Minstrel High Throughput (Minstrel HT) and Atheros 9000 (Ath9K) RA are embedded and initially available in the Compat-wireless driver kit. In addition, due to its simplicity and clarity, the RAMAS was chosen as one of the state-of-the-art MIMO RAs published in recent years. The RAMAS algorithm was implemented and validated through a wide range of tests. Finally, the Aggregation-Aware Rate Adaptation (AARA) was also

Table 5.2: Experimental Parameters

Parameter	Value
Operating System	Fedora 15
Kernel Version	Linux Kernel 2.6.43
WLAN Card	Atheros AR9300
Device Driver	Compat Wireless 3.6.8.1
Wireless PHY Standard	IEEE 802.11n
Channel Bandwidth	20 MHz
Available MCS Indices	0 (6.5 Mbps) - 15 (130 Mbps)
Frequency Bands	2.437GHz (Channel 6) / 5.2GHz (Channel 40)
Access Point Module	Hostapd 7.3
Packet Generator	Iperf 2.0.5

implemented upon the driver kit and the performance was analyzed and compared with the other three RAs. The detailed parameters for experimental platform are listed in 5.2.

5.5.2 Experimental Scenario

Due to the complication of variable wireless channel interference, it is a challenge to replay the same channel condition so that the performance of different RAs can be fairly compared. In order to clear out unexpected interference effects and ensure a controlled environment setup, all the experiments were performed with 5GHz band around midnight on the weekend when no other signal was identified by a Wireshark [57] packet sniffer equipped with the AirPcap [58] network card for enabling 802.11n packet capturing.

In order to estimate the rate decision accuracy for diverse channel states, the experiments were conducted in different environments, each of which is characterized different channel conditions. First, a set of tests were carried out in the middle of plain terrain to emulate an environment with minimized multi-path fading effect. Another set of experiments were performed in a campus building representing a typical office environment. The floor plan of the building is illustrated in the 5.3 and different locations of the client node are marked with letters A to F.

In order to analyze the impact of contention, *In-range* nodes were placed within a range where they could sense each other's signal. For a hidden station scenario, the hidden node

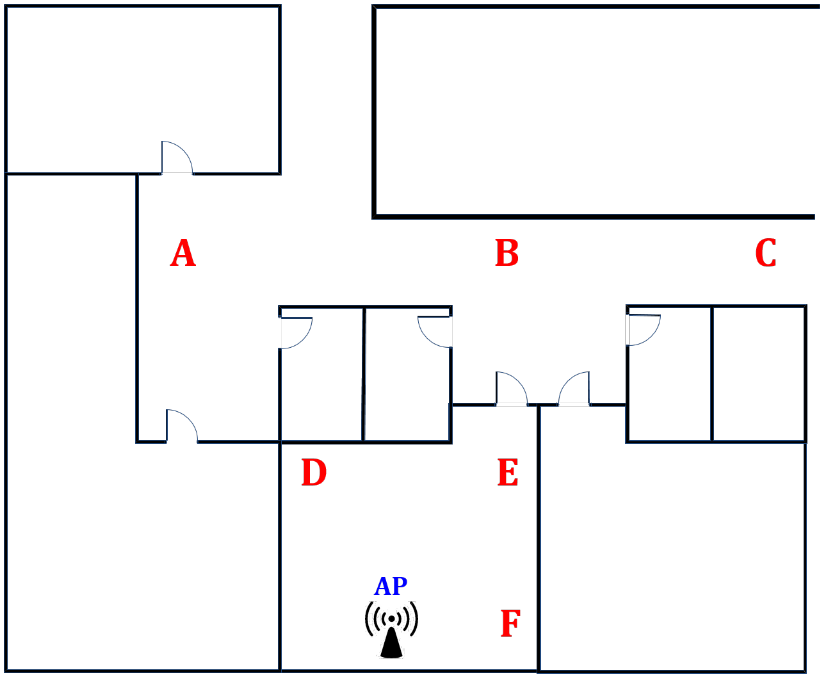


Figure 5.3: Experimental Floor Plan

was placed not only behind a concrete wall but also on a different level where there were minimized reflecting objects for multipath effect. Most of the signal was blocked except for occasional successes verified by the sniffer. The contention scenario setup (*In-range* and hidden) is illustrated in Figure 5.4.

Since different RAs cannot be executed simultaneously, it was important and challenging to create a configuration where wireless equipment could experience the same channel dynamics especially in mobile environment. For fair comparison between RAs as possible, in the mobile scenario, a laptop was placed on top of a rolling chair and was moved toward the AP in increments of 25ft as shown in Figure 5.5.

Finally, a set of experiments was conducted in the student center specifically to examine the usage of MRR chain. In the student center, the link quality changed dramatically due to contention and moving people.

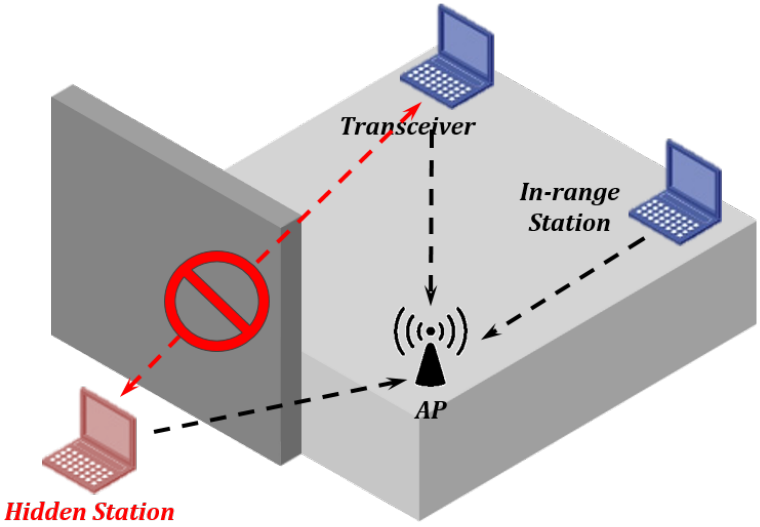


Figure 5.4: Contention Scenario Setup

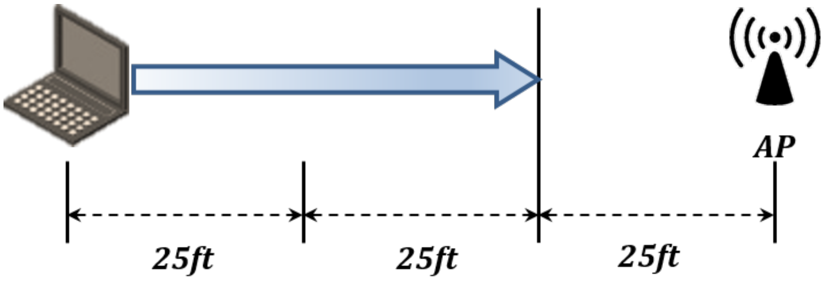


Figure 5.5: Mobile Scenario Setup

5.5.3 Performance Metric

By enabling per-A-MPDU packet control functionality, most of the available MAC layer traces were collected and printed out to the Linux system log upon receiving BlockACK packets. These layer traces included Received Signal Strength Indicator (RSSI), the difference of RSSIs between multiple antennas, transmission rates, the number of hardware retransmissions including Multi-Rate Retry (MRR) chain information, BlockACK bitmaps, RTS/CTS control packet usage, and frame payload length. This information was then passed to a parsing program implemented to compute statistics such as Frame Success Ratio (FSR) and network throughput. The FSR is simply calculated as the following formula.

$$FSR = \frac{(nSuccess \times 100)}{nFrames} \tag{5.1}$$

nBad and *nFrames* imply the number of bad frames and the total number of frames in an A-MPDU packet respectively. Note that in contrast to previous work, the retransmission counts are not included in the formula. The detailed rationale behind this decision will be provided in Section 5.6.2.

The network throughput is defined as

$$Throughput = \frac{(T_{nFrames} - T_{nBad}) \times L_{dat}}{T_{overhead} + T_{MPDU}} \tag{5.2}$$

where $T_{nFrames}$ and T_{nBad} represent the total number of frames and total number of failed frames during transmission time respectively. $T_{overhead}$ is the sum of 802.11n protocol overhead which is represented as:

$$T_{overhead} = DIFS + T_{bf} + T_{phy} + SIFS + T_{lphy} + T_{BlockACK} \tag{5.3}$$

where detailed notations are described in 5.3.

Table 5.3: Variables used in Retransmission Cost and *Goodput* Calculation

Variable	**Value**
DIFS	34.0μs
SIFS	15.0μs
Average Back-off Delay (T_{bf})	63.0μs
Mixed-mode PHY Preamble and Header Time (T_{phy})	44.8μs
Legacy-mode PHY Preamble and Header Time (T_{lphy})	24.0μs
BlockACK frame time ($T_{BlockACK}$)	11.3μs
A-MPDU MAC Header Length (T_{hdr})	$34 bytes$

In addition, T_{MPDU} includes aggregated MPDU frame header and data transmission time as depicted in the following equations.

$$T_{MPDU} = T_{nFrames} \times (T_{hdr} + T_{dat}) \tag{5.4}$$

In contrast to the network throughput defined above, the proposed algorithm uses a different metric, *A-MPDU goodput*, for performance measurement based on the computation model in order to simplify calculation and reduce computational overhead as follows:

$$G_{A-MPDU} = Thput_r^{max} \times FSR \tag{5.5}$$

$Thput_r^{max}$ represents the maximum per A-MPDU throughput associated with the data rate r, which can be pre-calculated with the assumption of no frame loss during A-MPDU packet transmission. FSR is *Frame Success Ratio* introduced in formula 5.1. Table 5.4 lists the maximum possible *nFrames* and $Thput_r^{max}$ according to transmission rates available in this study.

5.6 Case Study

As mentioned in the previous sections, FA and BlockACK are the key mechanisms in the 802.11n standard to enable high network throughput. However, since most of the current

Table 5.4: Maximum Possible *nFrames* and Maximum per A-MPDU Throughput ($Thput_r^{max}$)

MCS Index	# of Streams	Modulation/ Coding Rate	Data Rate ($Mbps$)	Max *nFrames*	$Thput_r^{max}$ (Mbps)
0	1	BPSK 1/2	6.5	1	5.8
1	1	QPSK 1/2	13	4	10.6
2	1	QPSK 3/4	19.5	6	17.7
3	1	16QAM 1/2	26	8	23.8
4	1	16QAM 3/4	39	12	35.4
5	1	64QAM 2/3	52	17	47.6
6	1	64QAM 3/4	58.5	18	54.2
7	1	64QAM 5/6	65	20	60.1
8	2	BPSK 1/2	13	4	10.6
9	2	QPSK 1/2	26	8	23.1
10	2	QPSK 3/4	39	12	35.4
11	2	16QAM 1/2	52	17	47.6
12	2	16QAM 3/4	78	25	71.1
13	2	64QAM 2/3	104	32	95.4
14	2	64QAM 3/4	117	32	108.1
15	2	64QAM 5/6	130	32	119.4

RA algorithm designs do not take FA into consideration, they cannot make fully use the potential benefits of the new enhancement.

In this section, a preliminary case study is performed to understand the impact of Frame Aggregation on rate adaptation using a few fixed rates and Minstrel HT, which is one of the most representative and popular MIMO RA algorithms. Various RA algorithm design options such as A-MPDU retransmission, Multi-Rate Retry, contention avoidance, and channel probing were re-evaluated from the perspective of Frame Aggregation. The following questions are investigated in detail.

- What are the limitations of the current probing mechanism with single short packets to obtain more reliable statistics? How can the limitations be resolved with consideration of Frame Aggregation?
- How is the meaning of retransmission in an aggregated packet transmission different from a single packet transmission and how does it have to be handled?
- Based on the answer to the above question, how does RA use Multi-Rate Retry scheme for better adaptation?
- How can contention-induced packet losses be distinguished from fading-induced losses in the aggregation-enabled scheme?
- How can the RA detect the multiplexing structure which determines whether to use a low-modulation multiple stream rate or a high-modulation single stream rate?

5.6.1 Probing with A-MPDU

Popular RA solutions examine whether other data rates (typically higher than the current option) are suitable for the existing channel performance. If the probing is successful or results in better performance, then the rate moves upward; otherwise, it is downgraded.

In contrast to RA algorithms in the legacy standard, MIMO RAs can probe the channel using either (1) a single frame testing packet or (2) a normal A-MPDU packet. Minstrel HT

Table 5.5: Probing Attempt Distribution and Success Ratio on Each Rate Option

Data Rate (*Mbps*)	Total Count	Success Count	Failure Count	FSR	Rate Usage Distribution
6.5	3	3		100%	
13	8	8		100%	
19.5	4	4		100%	
26	9	9		100%	
39	6	6		100%	**12.34%**
52	7	7		100%	**72.41%**
58.5	3	3		100%	
65	N/A	N/A	N/A	N/A	
78	19	15	4	78.95%	**15.25%**
104	25	14	11	56%	
117	34	8	26	23.53%	
130	34	0	34	0%	
Total	152	77	75	50.66%	

and Atheros RA send a 257 byte-long single short packet to determine whether the sample rate is suitable for the channel condition. The statistics are collected for a specified time interval, and then the best rate for the next transmission is chosen based on the estimated throughput. However, evaluating channel status from a single short packet is susceptible to overestimation due to the relatively high Frame Success Ratio (FSR) of the next higher option as pointed out in [9].

Table 5.5 shows the distribution of probing attempts and rate usage collected using Minstrel HT RA in the office setting. Given that the best performing rate is at least lower than 78 Mbps, as can be identified intuitively, sub-optimal rates such as 104 Mbps and 117 Mbps shows non-negligible success ratio in the experiment. Since the robustness of the shorter packet leads to a better success ratio, even inappropriate rate options may have more than a 50% chance of success, a limitation which can confuse the RA process.

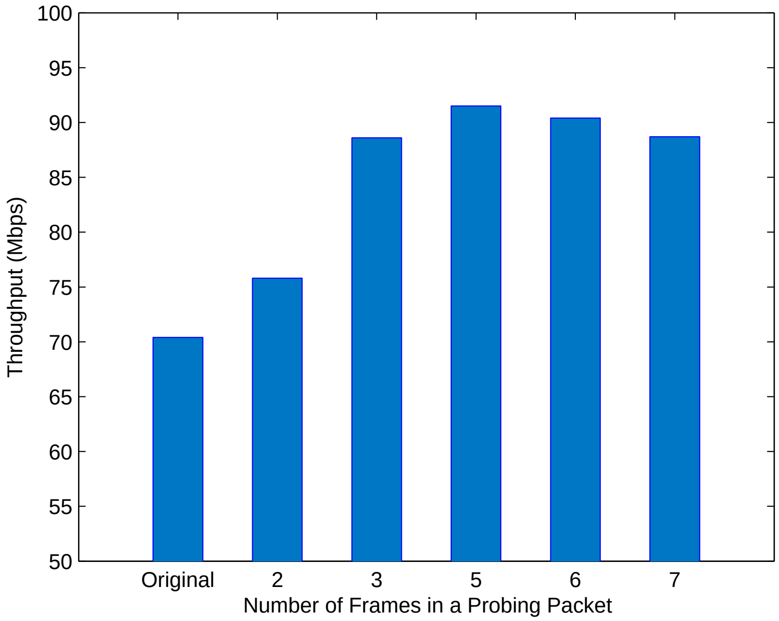
Furthermore, the single-frame packet probing has another defect around Frame Aggregation. When the aggregation module in the driver is triggered to send a probing packet, it instantly stops and closes the current aggregation process and places a single probing packet into the transmission queue. This process further degrades the utilization of the Frame Aggregation and negatively impacts the performance. A few studies such as MiRA [1] and WRA [28] have tried sampling with the normal A-MPDU packet; however, the cost of full utilization of Frame Aggregation may give rise to severe loss when the sampling rate is unsuitable for the current channel. Consequently, the level of aggregation should not be too high or too low.

To confirm the claim, another set of experiments was conducted in the office environment with no interference as verified by sniffer. Instead of using a single short frame, multiple frames were aggregated into sampling packets. Figure 5.6 illustrates the network throughput (in Mbps) against the number of frames in a probing packet. Generally, the performance appeared to be in proportion to the number of frames in probing packets, but the throughput dropped when six frames were used for probing in this experiment. As other results show the same trend with little difference of converging level, it can be concluded that the performance gain of multiple-frame probing increases to a certain level, then begins to converge and decreases.

On the other hand, random probing in RA design also leads to performance degradation. As can be observed in Table 5.5, the data rate of 65 Mbps was not tried at all during the total of 152 probing attempts; consequently, the Minstrel HT RA does not use the 65 Mbps option, which may be the optimal on this setup. Similarly, the probing attempt of 58.5 Mbps was not sufficient for RA to collect required statistics. Therefore, 72% of data transmissions were sent at 52 Mbps, which is seemingly more than two steps lower than the optimal rate.

In summary, RAs that adjust rates using short-term statistical analysis and random probing cannot avoid misjudgment and suffer from non-negligible performance degradation due to the distortion of the probing result.

Figure 5.6: Network Throughput with Varying Number of Frames in a Probing Packet



5.6.2 A-MPDU Retransmission

A-MPDU aggregates multiple numbers of MPDUs into a single packet, as overviewed in Section 5.2. The maximum number of MPDUs in each A-MPDU (called *aggregation level* or *nFrames*) is limited by specified total payload length (65,535 bytes) and by BlockACK, which can only acknowledge less than 64 MPDU frames based on the standard. The IEEE 802.11n standard additionally specifies the BlockACK sliding Window (BAW), which contains a range of sequence numbers to be transmitted in the TX queue. This window moves forward as the involved MPDU sequence numbers are acknowledged. If the first MPDU in the BAW is lost and needs to be retransmitted, the aggregation level of the following A-MPDU is limited because the BAW is not able to move forward to hold another MPDU in the queue.

The vendor implementation can further limit the aggregation level because the detailed aggregation logic is not specified in the standard. The Atheros network adapter used for this study is in compliance with regulatory 4ms transmission opportunity (TXOP) in 5

GHz band operation and the possible number of frames in an A-MPDU is further restricted according to the transmission rate involved. For example, the single-stream MCS 0, which sends data at 6.5 Mbps, cannot hold more than a single MPDU in an A-MPDU packet. On the other hand, the MIMO-enabled double stream MCS 15, which has 130 Mbps data rate, can aggregate 32 MPDUs due to the shorter transmission time of each MPDU, as listed in Table 5.4.

A-MPDU retransmission occurs when all of the frames in an A-MPDU packet have been lost (called *A-MPDU failure*). This situation is different from an individual MPDU retry where some of the MPDU frames in an A-MPDU packet are not successfully decoded by the receiver upon reception of a BlockACK. In case of individual MPDU failures, a transceiver simply duplicates and inserts the lost MPDUs into the next available A-MPDU packet. On the other hand, *A-MPDU failure* takes place when either a transmitter fails to receive a BlockACK from the receiver within a specified time duration or when the received BlockACK indicates that all the frames have been lost. As can be easily perceived, *A-MPDU failure* has more impact on the network performance than individual MPDU retry.

A-MPDU failure can be attributed to two representative reasons. First, all of the MPDU frames can be corrupted either by channel disorder or other interference. However, in contrast to the legacy standards without FA functionality where a single MPDU failure leads to a packet retransmission, a sudden channel dynamic which causes an MPDU frame error is rarely sustained for the entire A-MPDU transmission duration if the transmission rate is within a reasonable range.

Second, *A-MPDU failure* can result from simple A-MPDU physical header corruption. Only short-term disruption can cause the *A-MPDU failure.* Because the standard mandates that the PHY header to be transmitted with the lowest rate available to ensure safe delivery, the header is rarely corrupted by channel degradation except in some extreme cases. Furthermore, since the occupation of the header in an entire A-MPDU packet diminishes

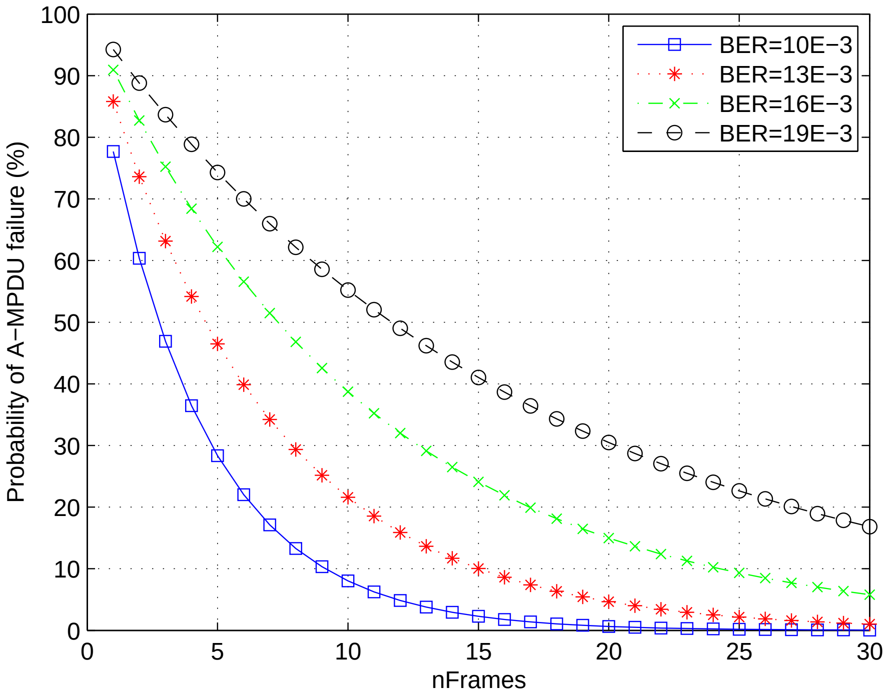


Figure 5.7: *A-MPDU Failure* Probability over *nFrames*

significantly as the aggregation level grows as shown in Figure 5.1, there is less chance of *A-MPDU failure* by header loss due to channel disruption.

The probability of *A-MPDU failure* according to Bit Error Rate (BER) regarding the channel dynamic can be expressed as

$$P_{A-MPDU failure} = \prod^{nFrames} (1 - (1 - P_{bit})^L) \tag{5.6}$$

where P_{bit} is the BER and L is the total length of A-MPDU including various protocol headers in bits. With this formula, Figure 5.7 presents the influence of *nFrames* on *A-MPDU failure* probability under different bit error rates. Noticeably, if the *nFrames* are greater than 20, the probability of *A-MPDU failure* caused by BER is lower than 30% even in erroneous channels. In other words, there is less chance of losing the entire A-MPDU packet, assuming FER of the current data transmission rate is within a normal range. Therefore, it can be assumed that an *A-MPDU failure* is mainly caused by header corruption. Table 5.6 presents

Table 5.6: Rate Distribution, *nFrames*, Retransmission Ratio, FER

Data Rate	*nFrames*	RET Ratio (%)	FER
117 Mbps	32	1.97%	0.06%
130 Mbps	32	2.71%	0.54%
130 Mbps	24	3.09%	0.71%

the A-MPDU retransmission ratio and FER with varying transmission rate and *nFrames* set during the experiment where two stations were located in proximity, the sniffer verified no contending station, and a stable channel was guaranteed by a plain terrain with no reflecting object around. As expected, the A-MPDU retransmission ratio scales with data rate, but is inversely proportional to *nFrames*.

In summary, considering that the PHY protocol header is rarely corrupted by the channel fading due the robustness of low rate for header transmission (usually 24 Mbps), it is reasonable to assume that *A-MPDU failure* is caused by other factors such as collisions and internal data queue overflow. Since these factors are not related to channel quality deterioration, the RA algorithm is required to tolerate such cases and keep the current data rate.

5.6.3 Multi-Rate Retry Scheme with Frame Aggregation

Designing an RA algorithm gives rise to two interrelated design challenges. One is determining a data rate for the first attempt, and the other is selecting the number of retransmission attempts when the first attempted rate fails. Multi-Rate Retry (MRR) functionality is available in all the 802.11 PHY-compliant devices.

Up to four different rates and retransmission counts can be chained in the MRR descriptor. The MRR chain is represented as $(r0/c0)$, $(r1/c1)$, $(r2/c2)$, $(r3/c3)$ where $r0 \sim r3$ represent data rates, and $c0 \sim c3$ represent packet retransmission counts corresponding with each rate respectively. The sum of all counts ($c0 + c1 + c2 + c3$) defines the retransmission limit after which the packet is discarded.

The MRR mechanism improves responsiveness and reliability for transient short-term channel fluctuation by re-sending the lost packets at lower transmission rates in order; however, with FA, the MRR mechanism affects the performance in a different way. Due to fairness in accessing wireless channel medium, the Atheros driver imposes a 4ms limit for maximum duration, which means that once a transmitting node obtains the channel access and gets a transmission opportunity, the slot is open for a maximum of 4ms. Hence, while the original maximum A-MPDU aggregation level is limited to 64 by the available information of BlockACK bitmap, it is further limited by the transmission rate chosen as shown in Table 5.4 to ensure an equivalent amount of air transmission time.

This restriction can provide RA with a useful opportunity to control aggregation level in A-MPDU. The maximum *nFrames* of the lowest rate in the MRR chain imposes a limit on the current level of aggregation in the next A-MPDU, and therefore RA can easily control the *aggregation level* by simply putting the lower rate in the last available slot in the MRR chain. Even though MRR-based controlling has the shortcoming that it can be set for only the pre-defined and restricted number of aggregation levels listed in Table 5.4, controlling the level of aggregation using MRR provides another level of protection against sudden channel deterioration.

Since the probability of retransmission drops significantly in proportion to aggregation level, as discussed in Section 5.6.2, the depth of MRR chain, which means the number of A-MPDU retransmission counts of each rate, should be re-evaluated as well. From the experiments in the student center where the channel changed rapidly due to contention and moving people, the chance of using the second rate in the MRR chain ranged only from 0.2% to 5.9%, and only 0.8% of data was transmitted using the third rate. As a result, it is not necessary to have more than three rates specified in the chain.

5.6.4 Impact of Congestion

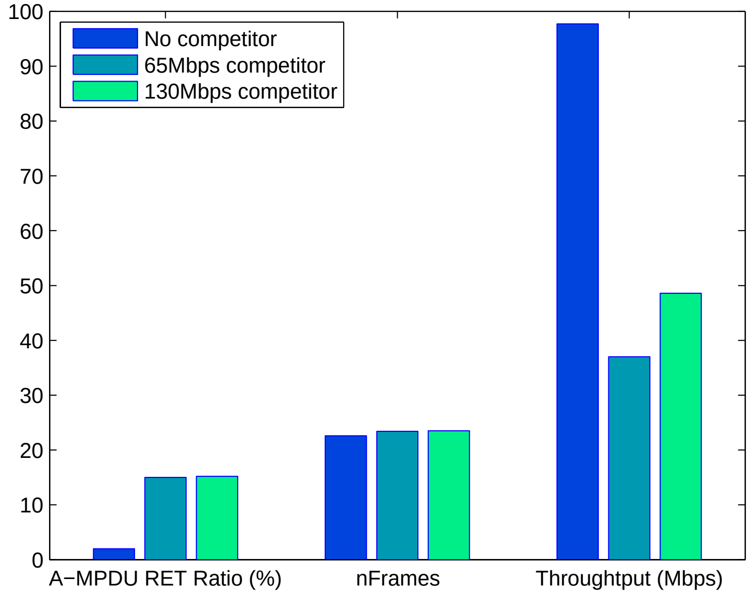
In order to examine in detail the impact of collision on aggregated packet transmission, two sets of experiments were performed with *In-range* and hidden stations respectively as shown in Figure 5.4. Minstrel HT was used for Rate Adaptation algorithm and channel fading and multi-path effects were minimized and verified with a sniffer.

Figure 5.8 illustrates throughput and A-MPDU retransmission ratio under an *In-range* contention scenario. The A-MPDU retransmission ratio is defined as retransmission counts divided by total transmission counts. The contending *In-range* station used two fixed rates, 65 Mbps and 130 Mbps. As the figure shows, the throughput dropped significantly when stations shared the channel medium. Notice that the performance was better when the contending station used the faster transmission rate (130 Mbps) than when it is used the slower rate (65 Mbps). This result occurred because the slower rate transmission occupies the channel longer; hence, the transceiver has fewer transmission opportunities. In contrast, it is interesting that the retransmission ratio was barely affected by contending stations data rate but its frequency increased by 7.6 times due to collisions. It can be easily perceived that *In-range* collision occurs only when more than two transmitting stations are assigned to the same *back-off* slot, and the chance of being in the same slot is irrelevant to the transmission rate.

The average aggregation level of three cases ranged from 22.6 to 23.5. Even though the results do not show a distinct difference because of the insufficient level of contention, the claim still holds that more congestion leads to better utilization of aggregation due to the delay in channel access.

With *In-range* stations, a collision occurs only when multiple stations are assigned to the same back-off slot, as explained in Section 4.2. Since involved stations start transmissions at the same time, there is a high probability that the header of the transmitting packet will be corrupted and trigger an entire A-MPDU retransmission. In other words, the *In-range*

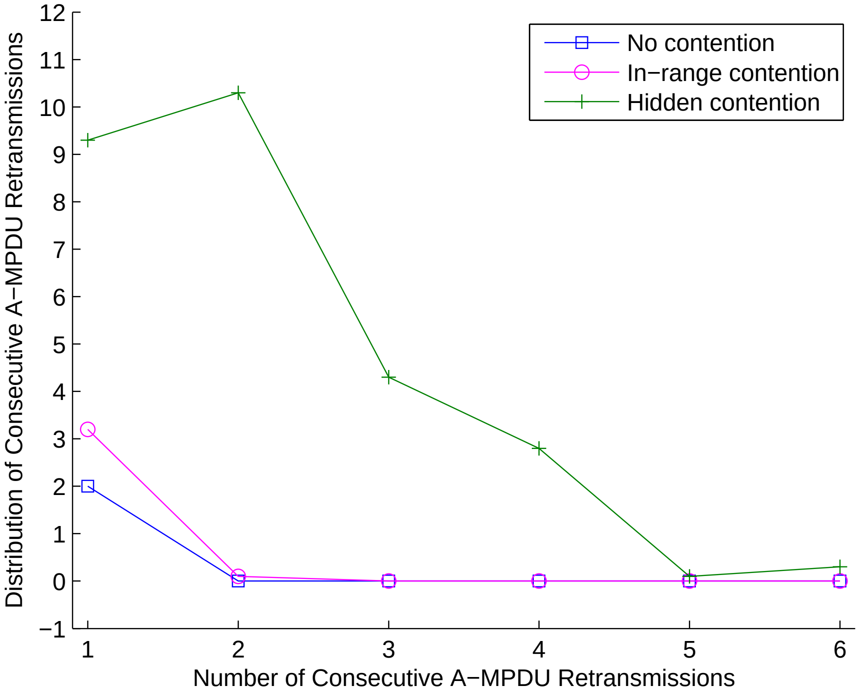
Figure 5.8: A-MPDU Retransmission, Average nFrames, and Throughput over Different Congestion Scenarios



collision may not cause a partial loss within the aggregated packet. This claim is proved by the fact that AFER of each A-MPDU transmission was negligible in the experiments.

In contrast, hidden station collision can affect both the header and body of an A-MPDU packet. Hence, it is difficult to differentiate hidden collision traces from fading-induced error; however, due to the channel obliviousness of the hidden station, the erroneous status is sustained for a certain time interval and causes a burst of packet error. Figure 5.9 illustrates retransmission ratio over total number of A-MPDU transmissions, as the x axis indicates the number of consecutive A-MPDU retransmissions. As the figure shows, congestion triggers significant A-MPDU retransmissions, and the hidden station scenario leads to the highest number of retransmissions. In addition, the consecutiveness of A-MPDU retransmission can be an important notifier of the existence of hidden stations.

Figure 5.9: Consecutive A-MPDU Retransmissions over Different Scenarios

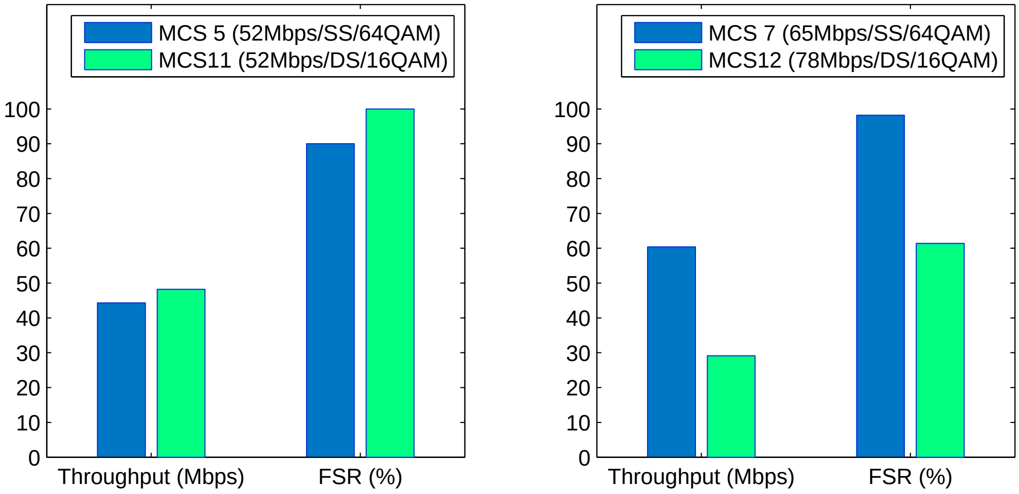


5.6.5 Multiplexing Structure

In contrast to Single Input Single Output (SISO) systems, the performance of the next higher bandwidth rate in a MIMO system does not always imply a higher network throughput because a MIMO system has multiple streams available which add another dimension to the decision plane. Making the best rate choice depends not only on modulation and coding scheme but also on the number of streams.

Multiplexing structure is a channel condition metric upon which the link quality is sufficient to allow the transceiver to transmit with multiple streams. With a multiplexing structure, the RA can prioritize a lower modulated multiple stream rate over a higher modulated single stream rate. Figure 5.10 depicts two sets of experiment results that show throughput performance and Frame Success Ratio (FSR) over different MIMO operation modes. While the Double Stream (DS) mode wins in the left, the Single Stream (SS) mode results in better performance even though the DS MCS 12 rate has a higher bandwidth and

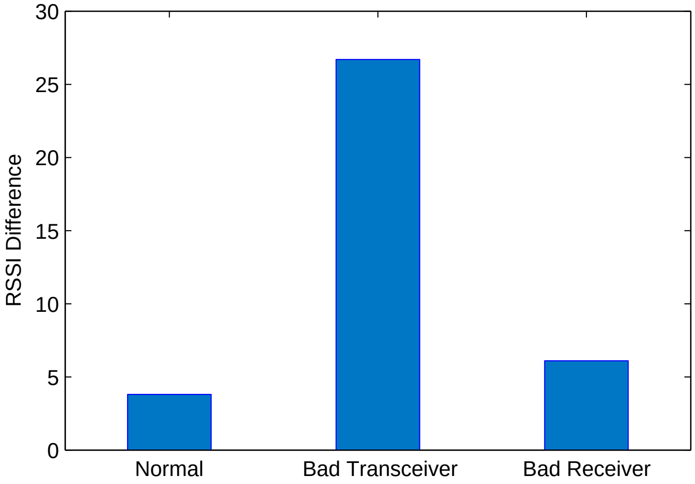
Figure 5.10: Throughput and FSR Comparison under Different MIMO modes



lower modulation scheme. Since the receiver of the second experiment was placed in a steel box and the received RSSI value was lower than a normal range, the channel did not allow DS transmission, that is, the multiplexing structure did not exist.

Identifying whether the channel has the multiplexing structure is a non-trivial challenge. In spite of unpredictable fluctuation, RSSI difference between multiple antennas can be one of the potential metrics, as shown in ARF-HT [29]. In order to verify the quality of the metric, another experiment was performed under different scenarios. One of the two antenna was disabled on either side of the transmission pair. As shown in Figure 5.11, when the receiver's antenna hurts the multiplexing structure, it does not sufficiently clarify the situation; however, when the transceiver side's antenna is disabled, the value of RSSI difference clearly indicates the problem. Therefore, even though it is difficult to identify the receiver side's problem, the RSSI difference metric is a useful metric to detect a transceiver-side defect affecting the multiplexing structure.

Figure 5.11: RSSI Differences in Various Scenarios



5.7 Aggregation Aware Rate Adaptation (AARA)

In this section, a novel new rate adaptation algorithm called Aggregation Aware Rate Adaptation (AARA) is proposed based upon the observations and findings from the case study. AARA is a standard-compliant open-loop RA scheme that involves only the transmitter for the best rate decision process. The design of AARA addresses the following issues: (1) How to estimate the channel and how to find the best-performing transmission rate for the current channel, (2) How to keep the agility to the channel dynamic while maintaining stability in adaptation, (3) How to differentiate and adapt to *In-range* and hidden collision-induced interference. The following sub-sections provide detailed discussions of these questions.

5.7.1 Best Transmission Rate Selection

AARA is a goodput-based statistical RA using probing for channel estimation. Upon reception of a BlockACK, the AARA begins to collect a specified number of A-MPDU packets and counts numbers of different variables listed in Algorithm 1. Based on the indications in BlockACK, Frame Success Ratio (FSR) is then calculated as discussed in Section 5.5.3.

Notice that the retransmission counts are not included in the FSR calculation, a unique difference from previous research [1, 28]. This difference is based on the assumption that channel deterioration not only causes A-MPDU retransmission, but also corrupts a part of an aggregated packet. Hence, if channel fading causes packet losses, it also affects the FSR, which is computed without the retransmission counts.

After obtaining FSR, AARA follows different procedures according to the current state. Figure 5.12 explains the transition condition between four different states.

- NORMAL: typical transmission status where a transceiver aggregates the maximum possible frames into an A-MPDU.

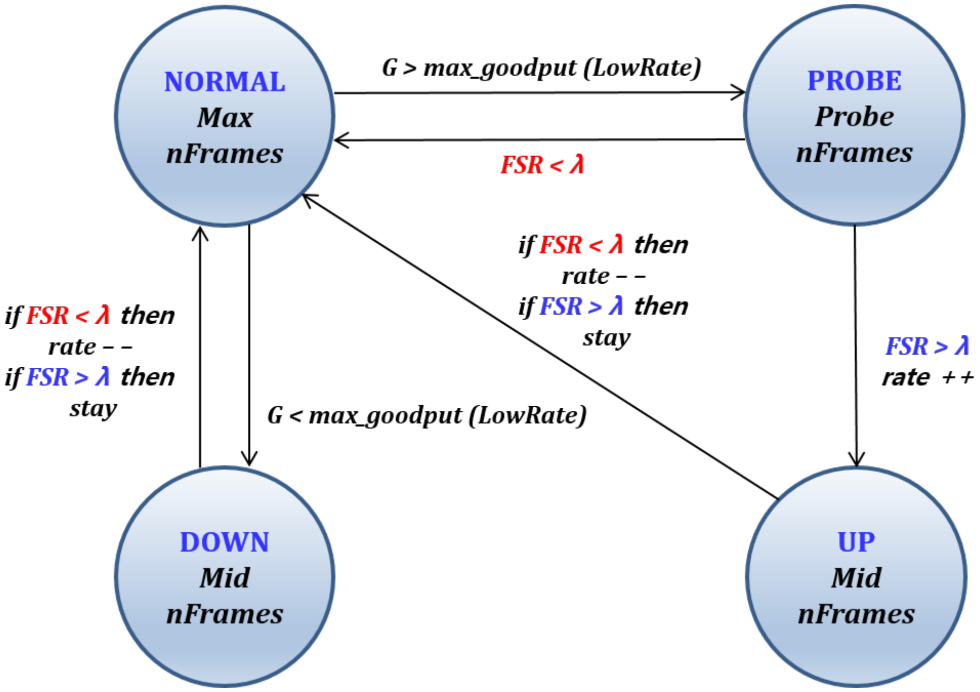
- PROBE: probe frame transmission status where a transceiver limits aggregation level for the probing packet.

- UP: rate increasing status where a transceiver increases data rate by one step and limits aggregation level for a specified interval upon successful probing.

- DOWN: prior phase before decreasing rate where a transceiver stays in the same rate, but limits aggregation level for a specified interval.

Starting from NORMAL state, once the goodput of the last sampling window is greater than maximum potential goodput of the rate one step lower, AARA initiates channel probing, and then FSR of the probe packet is compared to an A-MPDU success threshold (λ) to determine whether the probing is successful. If the probing satisfies the threshold, the current state is switched to UP state, otherwise, it turns back to NORMAL state. When the goodput of the last sampling window is less than maximum potential goodput of the rate one step lower in NORMAL state, the state changes to DOWN state before switching to a lower data rate. The A-MPDU success threshold (λ) is an important parameter to make the final decision on rate increment in the PROBE status. In this dissertation, this value is set to 90% based upon the heuristics.

Figure 5.12: State Transition Diagram



The rationale behind the state transition is that RA should be agile to short-term channel gain while maintaining stability to short-term channel disturbance. This balance is achieved by controlling frame aggregation level of A-MPDU packets before rate change. To this end, the *Min nFrames* is defined as the minimum number of frames in an A-MPDU which can achieve the same throughput which a rate one step lower can accomplish with its *Max nFrames* listed in Table 5.4. *Mid nFrames* is defined as the median value of the *Max nFrames* and *Min nFrames*. This value is used for transmission on UP and DOWN states, while *Probe nFrames* is used for PROBE state transmissions. This *Probe nFrames* is basically the same as *Mid nFrames*, but it has the maximal limit of ten frames.

Using Multi-Rate Retry (MRR) chain [represented by $(r0/c0)(r1/c1)(r2/c2)(r3/c3)$] is another way for AARA to mitigate short-term fluctuation of the link. When the current state is NORMAL, $r0$ is set to the best rate and $r1$ is set to the last data rate right before the current one. On the other hand, a probing rate is inserted into the first slot $(r0)$ for PROBE state, followed by the other two rates used in NORMAL state. Note that AARA

Algorithm 1 AARA Algorithm

```
//Declaring variables
//$MRR$ includes $(r0, c0)(r1, c1)(r2, c2)(r3, c3)$
$nSuccess$ = number of true bits in BlockACK
$nFrames$ = total number of acknowledged bits in BlockACK
$\lambda$ = A-MPDU success threshold
$status$ = current status
$nextLowerRate$ = next lower bandwidth rate
$nextHigherRate$ = next higher bandwidth rate

$FSR = (nSuccess \times 100)/nFrames$

if $status$ == PROBE then
    if $FSR > \lambda$ then
        set_rate ($r0$)
        $status$ = UP
    else
        $status$ = NORMAL
    end if
else if $status$ == NORMAL then
    $goodput$ = max_goodput ($r0$) $\times$ $FSR$
    if $goodput$ < max_goodput ($nextLowerRate$) then
        $status$ = DOWN
    else if $FSR > \lambda$ then
        set_probe ($nextHigherRate$)
        $status$ = PROBE
    end if
else if $status$ == UP or $status$ == DOWN then
    if $FSR < \lambda$ then
        set_rate ($nextLowerRate$)
    end if
    $status$ = NORMAL
end if
```

does not utilize the fourth chain because the aggregation level can be unnecessarily limited as discussed in Section 5.6.3.

The numbers of retransmission counts ($c0 \sim r3$) are also key parameters that will be discussed in more detail in Section 5.7.3. The detailed pseudo code of state transition is presented in Algorithm 1.

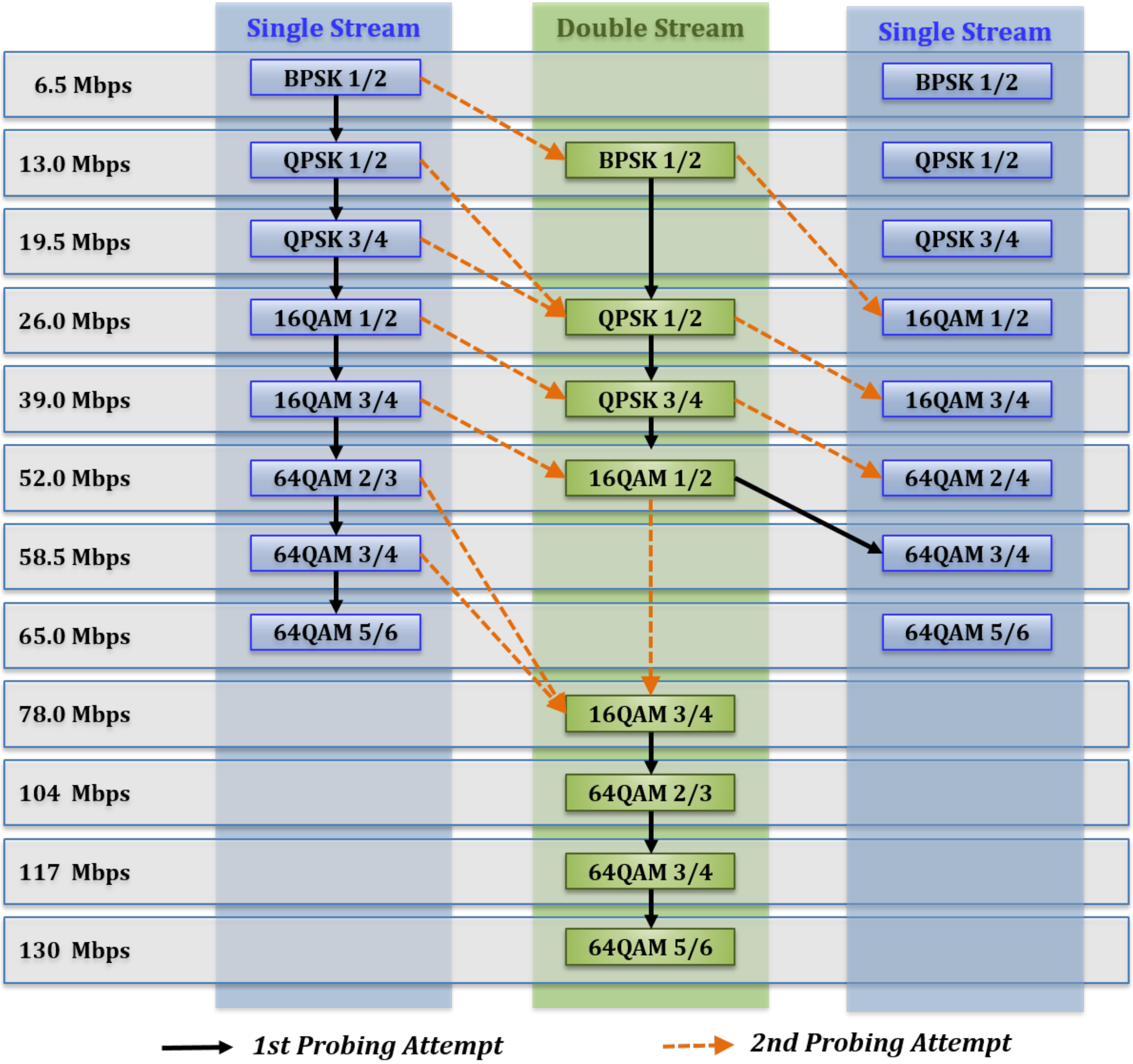
5.7.2 Adaptive Probing

Probing is a key mechanism to track the best performing rate in AARA. First, AARA uses a normal aggregated data packet instead of the single short packet used in Ath9K RA or Minstrel HT, but limits the aggregation level of a probing packet by *Max nFrames* value in order to obtain sufficient statistics while still avoiding the risk of losing the entire aggregated packet. Secondly, AARA exponentially increments probing intervals to a certain limit when there are consecutive identical probing attempts, thus reducing unnecessary probing overhead under stable channel conditions.

As introduced in Section 5.6.5, depending on the existence of multiplexing structure, the channel may have a non-monotonic relationship between rate increment and FSR. In other words, a higher rate option may counter-intuitively lead to less transmission failure than a lower rate option. Therefore, linear choice of candidate rates possibly causes the RA to be stuck in a sub-optimal rate.

Based on in-depth reviews of the current MIMO RAs demonstrating non-monotonic behaviors in MIMO system [1, 30], non-monotonic cases happen only when a high modulation SS rate is used even though a multiplexing structure exists. To avoid such cases, AARA not only tries to probe the next higher rate, but also attempts cross-mode rates. Figure 5.13 illustrates the cross mode probing directions for each MCS rate. The solid black line represents the first attempt. If the first probing fails, the second candidate rate, represented by the dashed orange line, is attempted. The next higher rate within the same MIMO mode is chosen as the first attempting direction, whereas the second direction selected is a higher bandwidth rate with lower modulation scheme among cross-mode rates. Notice that there is an exception on DS 52 Mbps where the cross-mode rate is tried ahead of the same mode higher rate. These probing candidates are set to the *nextHigherRate* used in Algorithm 1 by turns according to priority.

Figure 5.13: Cross mode Rate Probing Directions



5.7.3 Congestion Detection and Response

As discussed earlier, there are two types of contention-induced losses. First, *In-range* collision is caused by limitation of DCF protocol where multiple stations are assigned to the same back-off slot. In this case, due to the timing proximity on transmissions, the headers of packets are corrupted causing A-MPDU retransmission. Since colliding with the same packet in the retransmission attempt is not likely, the number of consecutive retransmissions are limited. At this point, AARA identifies an *In-range* collision. In contrast, hidden station-induced collision may affect a series of A-MPDUs in a row because hidden stations are oblivious of the ongoing transmissions.

The number of retransmissions without partial frame loss is used for detecting congestion in AARA. Since most packet collisions corrupt not only data frames, but also the header of an A-MPDU, the collision destroys the entire A-MPDU rather than exploiting partial data frames, assuming that the working rate is within a reasonable range. Therefore, multiple retransmissions were made before the last successful attempt, and FSR of the last trial is over 90%, AARA judges that the current channel is suffering from congestion.

Upon detecting *In-range* collisions, AARA ignores the incident; however, when AARA identifies a possible hidden station, it initiates the RTS/CTS handshake which is the only available solution for the hidden station problem. The current implementation sets the retransmission threshold to three for hidden station detection according to the heuristics given in Figure 5.9.

5.8 Performance Evaluation

In this section, the performance of AARA is evaluated and compared with various MIMO RAs, Minstrel HT, Atheros RA and RAMAS [30], in diverse settings introduced in 5.5.2. Since performance degradation of MIMO-oblivious RAs with MIMO stations has been sufficiently proven in the literature, only representative MIMO RAs were evaluated in this experiment. AARA was evaluated and compared by measuring the success or failure of the chosen transmission rate as compared to the optimal rate for the current channel quality. Except for the contention scenario, the traffic was sent from the AP to clients.

5.8.1 Implementation

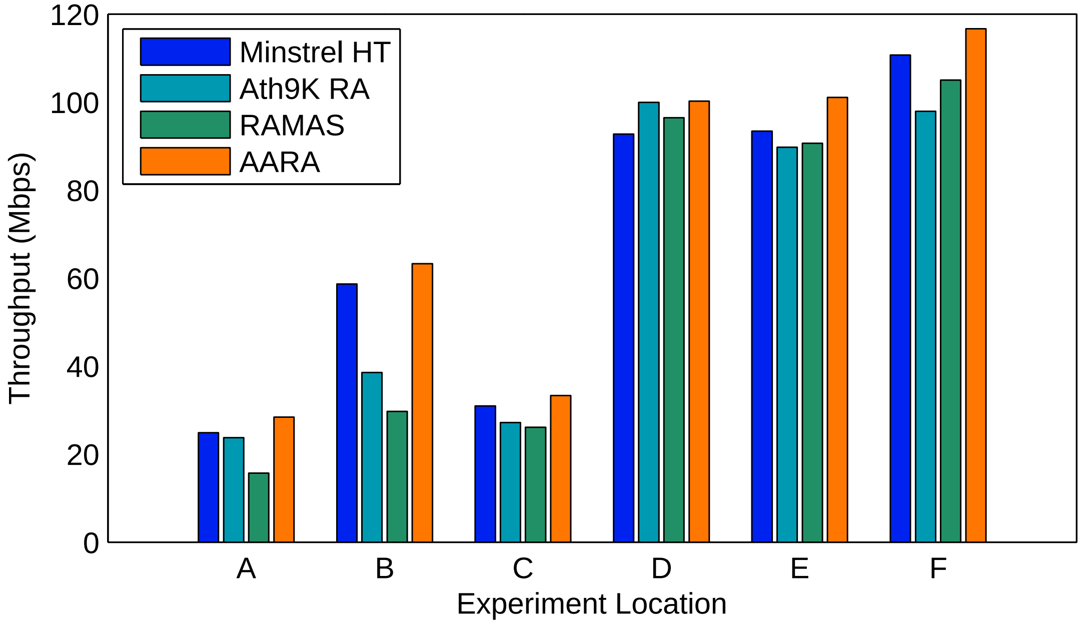
The AARA algorithm was implemented using an open source Compat-wireless network driver. It provides Minstrel HT and Atheros RA, which are both statistic-based algorithm as embedded options. Even when the new rate is assigned to the driver, the change is not immediately applied to the next data transmission. Once the new rate is set, it becomes effective after five to seven A-MPDU packet transmissions. The delay is due to changing and obtaining values from shared memories between the transmission module and the RA module. Since the two built-in RA algorithms gather statistics at a specified interval (100ms), which is relatively longer than the delay in the rate update, the performance of these RAs is not adversely affected. In comparison, AARA employs a faster statistics interval for adapting to the instantaneous channel variation. Consequently, a delay of this magnitude can cause serious distortion. Therefore, the RA module is directly embedded into the transmission module so that the updated rate is immediately applied to the next transmission.

Along with AARA, RAMAS [30] was implemented to compare AARA performance with a representative MIMO RA. Since the algorithm is straightforward and relatively simple, it is not difficult to implement; however, obtaining similar performance in different experiment set-up from the original paper is challenging. In addition, the performance of RAMAS varies according to implementation parameters such as time window, credit threshold, and acceptable error rates. Even though the optimal parameters are different in the experiments, the original values were used as recommended in the paper.

5.8.2 Static Setting

Experiments for static configuration were performed to evaluate whether different RAs efficiently identify the optimal transmission rate over the current link condition. Figure 5.14

Figure 5.14: Throughput in Static Setting



presents the UDP throughput measured at six different places inside a typical office building as shown in Figure 5.3. AARA consistently outperforms the other three algorithms by up to 14.3%, 64.2%, and 113.2% compared to Minstrel HT, Ath9K RA, and RAMAS respectively. Minstrel HT generally follows the AARA except for the location D. RAMAS shows good performance in proximal settings, but suffers in relatively far settings. Table 5.7 presents the percentage of performance gain of the new algorithm over existing RAs.

Figure 5.15 presents a set of rate evolutions in the experiments of location E in order to investigate the factors of performance benefit in depth. As already pointed out, the purpose of RA in static settings is to locate the best rate as quickly as possible while minimizing overhead for channel estimation. The optimal rate, 117 Mbps, can be easily identified in Figure 5.15. All four RAs seemed to be successful in finding it, but except in the case of RAMAS, the difference in throughput gain resulted from the probing overhead. Minstrel HT probes the candidate rates randomly. Even the lowest rate (6.5 Mbps) is occasionally attempted for link quality evaluation and these random attempts degrade performance. In contrast to Minstrel HT, Ath9K RA tries the next higher transmission rate than the current

Table 5.7: Thoughtput Improvement of AARA over RAs in Comparison

	Minstrel HT	Ath9K RA	RAMAS
Static (A)	14.3%	19.6%	81.3%
Static (B)	7.9%	64.1%	113.2%
Static (C)	7.8%	22.8%	27.7%
Static (D)	8.1%	0.2%	3.9%
Static (E)	8.2%	12.6%	11.5%
Static (F)	5.4%	19.1%	11.1%
Mobile	27.0%	23.9%	48.8%
In-range	50.6%	144.8%	473.2%
Hidden	32.2%	149.5%	209.0%

rate, but as discussed, a single short-length packet is used for probing, and the unexpectedly higher success ratio in transmitting at 130 Mbps rate option is attributed to this probing method. Even though the FER for 130 Mbps is around 36.9% in practice, all the probing attempts for the rate were successful. This overestimation requires the Ath9K RA to change the active rate to 130 Mbps causing serious frame losses.

In RAMAS, the normal A-MPDU packet was exploited for the sampling. However, since it drops down transmission rate for every six losses out of 30 frames, RAMAS drastically decrements the rate choice when multiple A-MPDU packets cause high AFER in transmission. For example, two consecutive A-MPDU can carry 64 MPDU frames total, 32 frames for each packet as indicated in Table 5.4, and assuming 50% of the frames are lost, 32 frames can be lost for only two A-MPDU transmissions. This loss can drop down the transmission rate by two steps at once and causes throughput reduction.

As expected, AARA identifies the best rate as quickly as other RAs. Also, it efficiently probes the next higher rate with normal A-MPDU packets, but reduces probing overhead by limiting nFrames when probes fail. Moreover, the exponential probing interval increment works perfectly to further restrict the unnecessary probing attempts as can be found in the rate evolution figure.

Figure 5.15: Rate Evolution in Static Setting

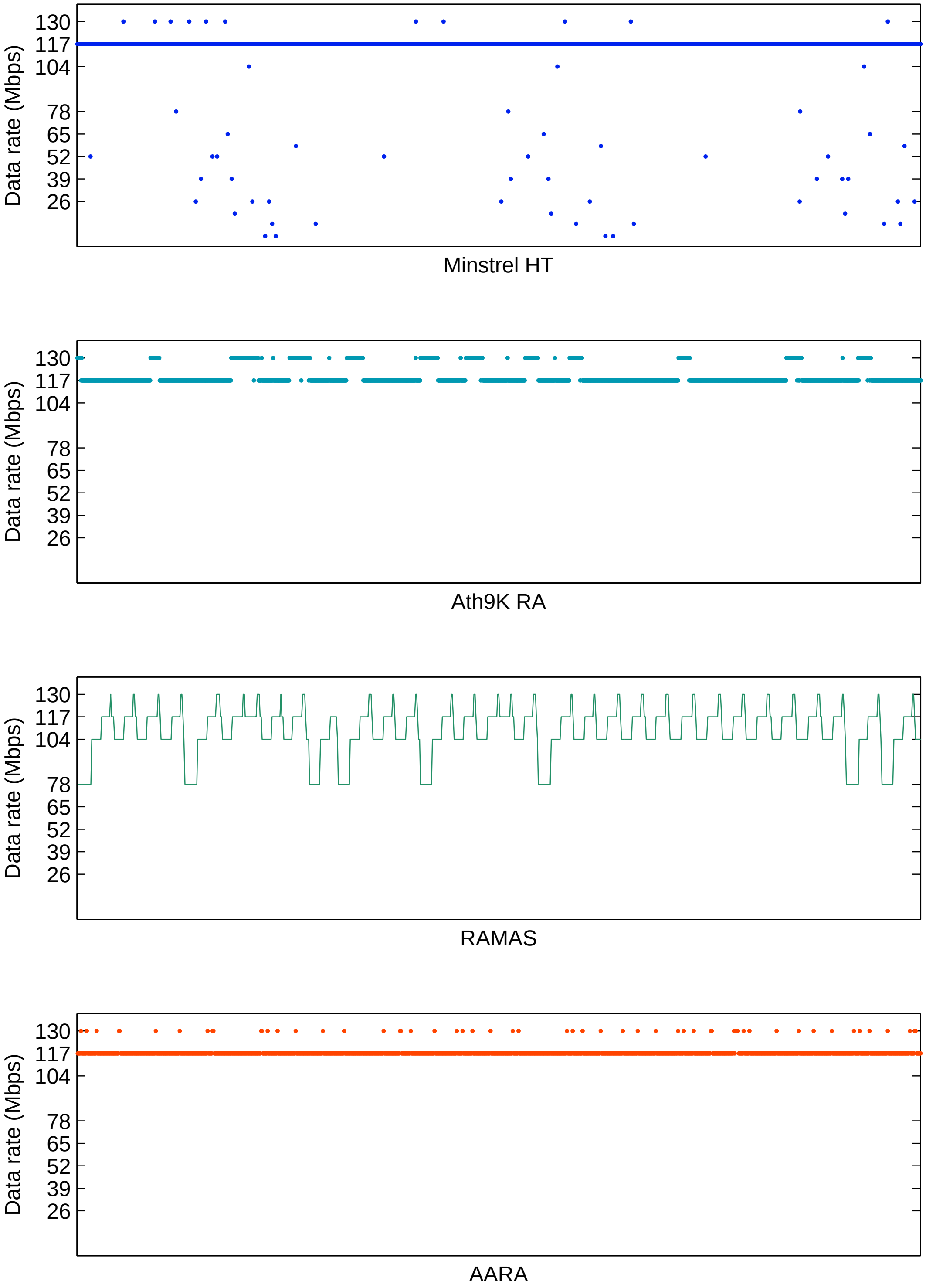


Figure 5.16 shows varying performance statistics at the same location as the evolution graph. The first graph exhibits average number of frames in a single A-MPDU over the transmissions. The reason that RAMAS shows the best aggregation level is that RAMAS does not utilize the MRR scheme which limits the aggregation level. While AARA uses three MRR chains on probing mode and two chains on the other modes, Ath9K RA utilizes all four chains, causing the aggregation density to stay in low levels. Although Minstrel HT sets only three chains, the third chain is required to be one stream rate. Hence, the aggregation level for Minstrel HT cannot exceed 20, which is the maximum nFrames for MCS 7, 65 Mbps rate option as shown in Table 5.4. The enhanced utilization of aggregation helps RAs to reduce the packet transmission attempts and the total number of transmitted packets is inversely proportional to nFrames, as shown in the middle graph. This reduced transmission also implies that there is less protocol overhead required per packet transmission and the channel can be less congested due to the reduced channel occupancy.

The rightmost graph shows the frame error ratio out of all the transmitted MPDU frames. The Ath9K RA exhibits the worst FER due to the misjudgment of the short length probing of the 130 Mbps rate option, while the other RAs perform similarly.

5.8.3 Mobile Setting

In order to observe responsiveness and adaptiveness to agile channel dynamics, four RAs were tested in a mobile environment. Figure 5.18 depicts throughput and FER of all four RAs tested in this setting. As can be noted, AARA performs better than Minstrel HT, Ath9K RA, and RAMAS by up to 51.9%, 53.2%, and 44.3% receptively.

The performance enhancement is attributed to AARA's agility to channel dynamics as explained by the rate evolution in Figure 5.18. AARA stays in a higher rate region when the node is static but moves quickly to lower options when the node starts to move. Minstrel HT suffers from the same drawback as in the static setting due to randomness in search, and Ath9K RA tends to stay in a higher rate region even when the station is moving

Figure 5.16: Performance in Static Setting (Location E)

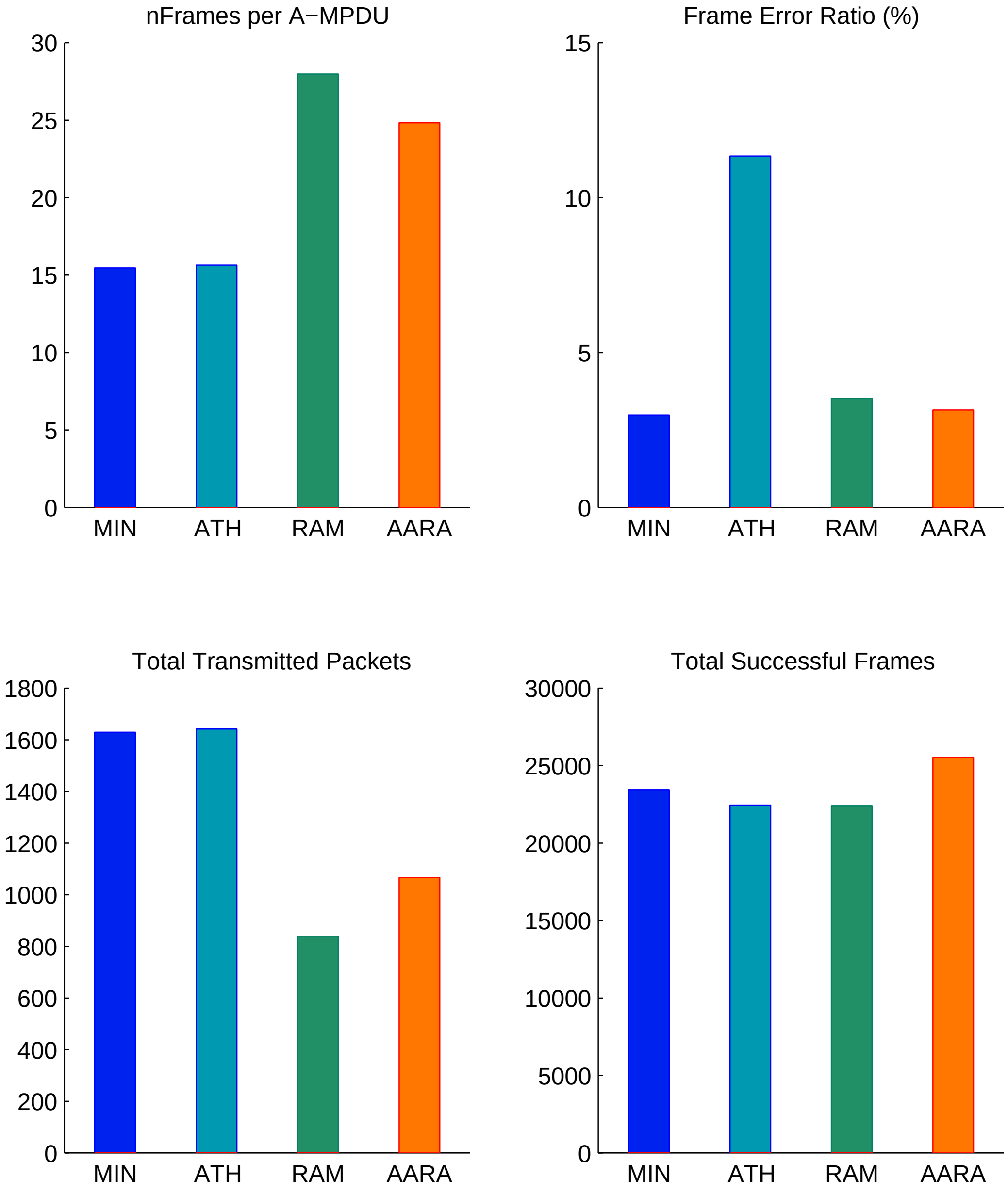
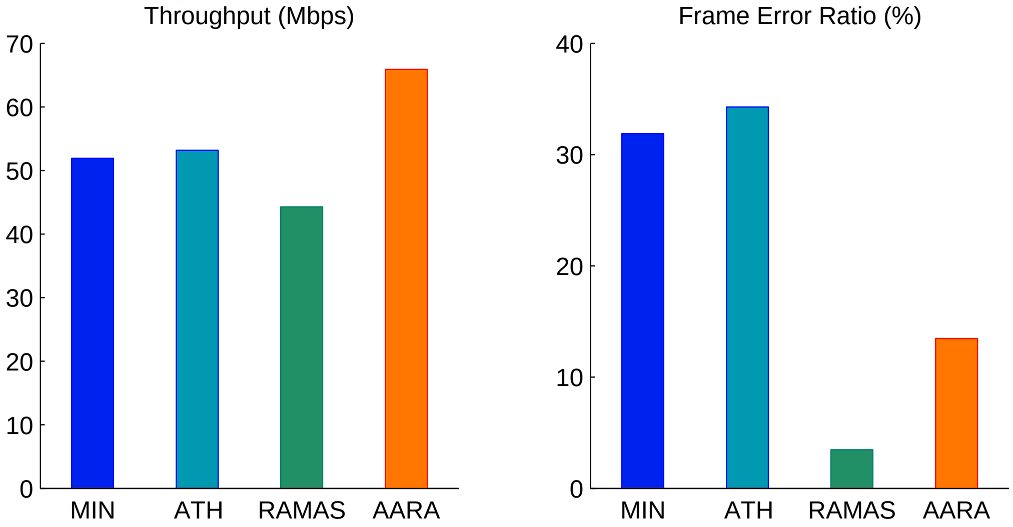


Figure 5.17: Throughput and FER in Mobile Setting



due to the overestimated probing results. Note that Ath9K RA does not use 65 Mbps for transmissions and this limitation affects the performance as well. On the other hand, RAMAS exhibits a similar pattern according to movement, but the serious underestimation of the link quality drastically drops the rate to the lowest option and leads to significant performance degradation.

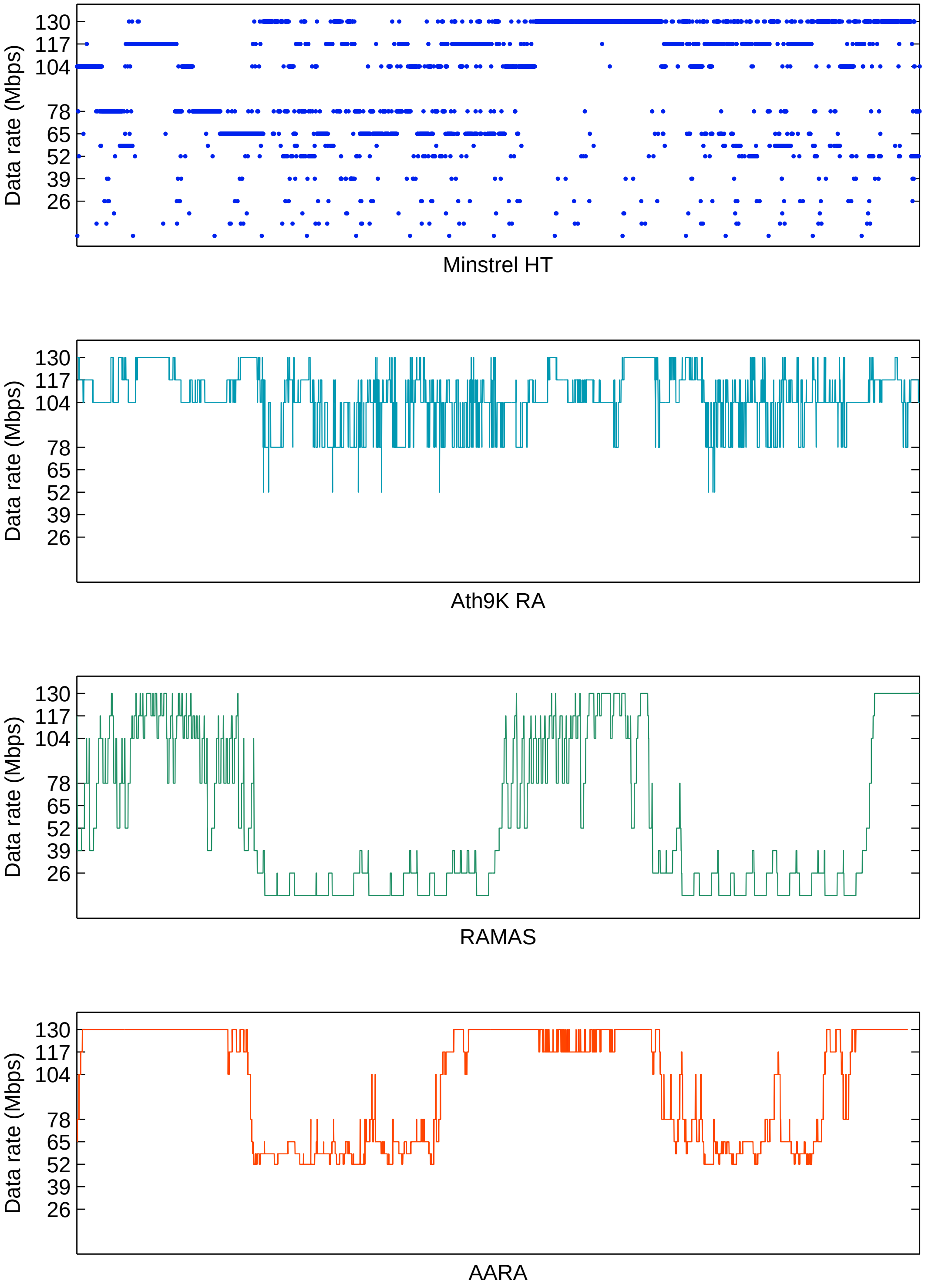
FER is also illustrated in Figure 5.18 to show AARA's low error rate compared to Minstrel HT and Ath9K RA. RAMAS exhibits the lowest FER among all the RAs tested because of inappropriately underestimating the channel and staying in the lower rate region.

5.8.4 In-range Collision Setting

The purpose of this experiment was to test how RAs differentiate and filter out *In-range* collision-induced packet losses from the channel estimation process. The performance gain of AARA for this experiments is the best out of four test-beds and reaches up to 50.6%, 144.8%, and 473.2% respectively over Minstrel HT, Ath9K RA, and RAMAS.

As can be found in Figure 5.20, while other RAs stay in the lower rate region, AARA maintains the higher rate options during the experiment and produces the best performance. This improved performance results from the different channel estimation model. Minstrel

Figure 5.18: Rate Evolution in Mobile Setting



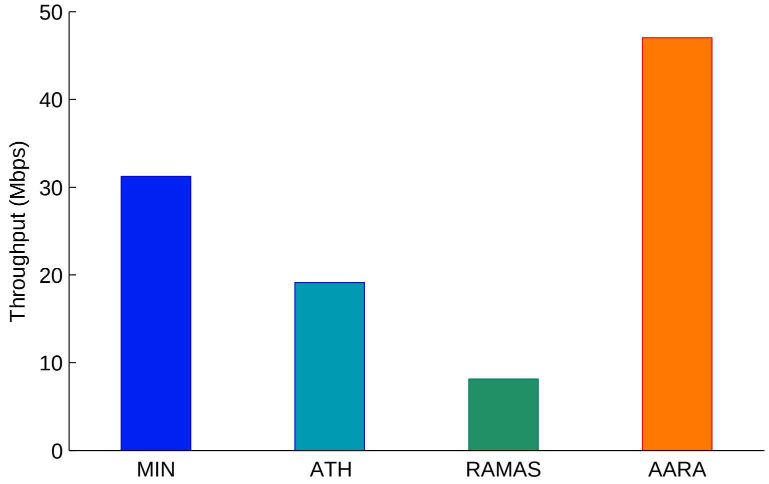
Minstrel HT

Ath9K RA

RAMAS

AARA

Figure 5.19: Throughput in In-range Collision Setting



HT and Ath9K RA include the A-MPDU retransmission counts to compute the goodput of the currently transmitted A-MPDU packet so that the goodput takes account of lost frames upon *A-MPDU failure*. This approach seems intuitive and reasonable; however, because *A-MPDU failure* is caused by collisions, as discussed in Section 5.6.4, and the losses by collisions should be filtered out from channel estimation, considering retransmissions can mislead the RA's decision. Figure 5.21 illustrates A-MPDU Frame Error Ratio with and without retransmission counts in the experiment. While the bars indicate the AFER with retransmission considered, the dots plot the AFER per-A-MPDU without retransmission counts. The majority of packets yield low AFER when retransmissions are excluded in computation. This finding confirms the assumption made in Section 5.6.4 that *In-range* collisions affect the protocol header and cause *A-MPDU failure* rather than partial frame loss within each packet.

5.8.5 Hidden Collision Setting

In the hidden collision setting, it is challenging to replicate the same environment so that each RA experiences the same link condition due to the *capture effect*. When there are multiple signals transmitted, if one of them has enough signal power to be distinguished from

Figure 5.20: Rate Evolution in *In-range* Contention Setting

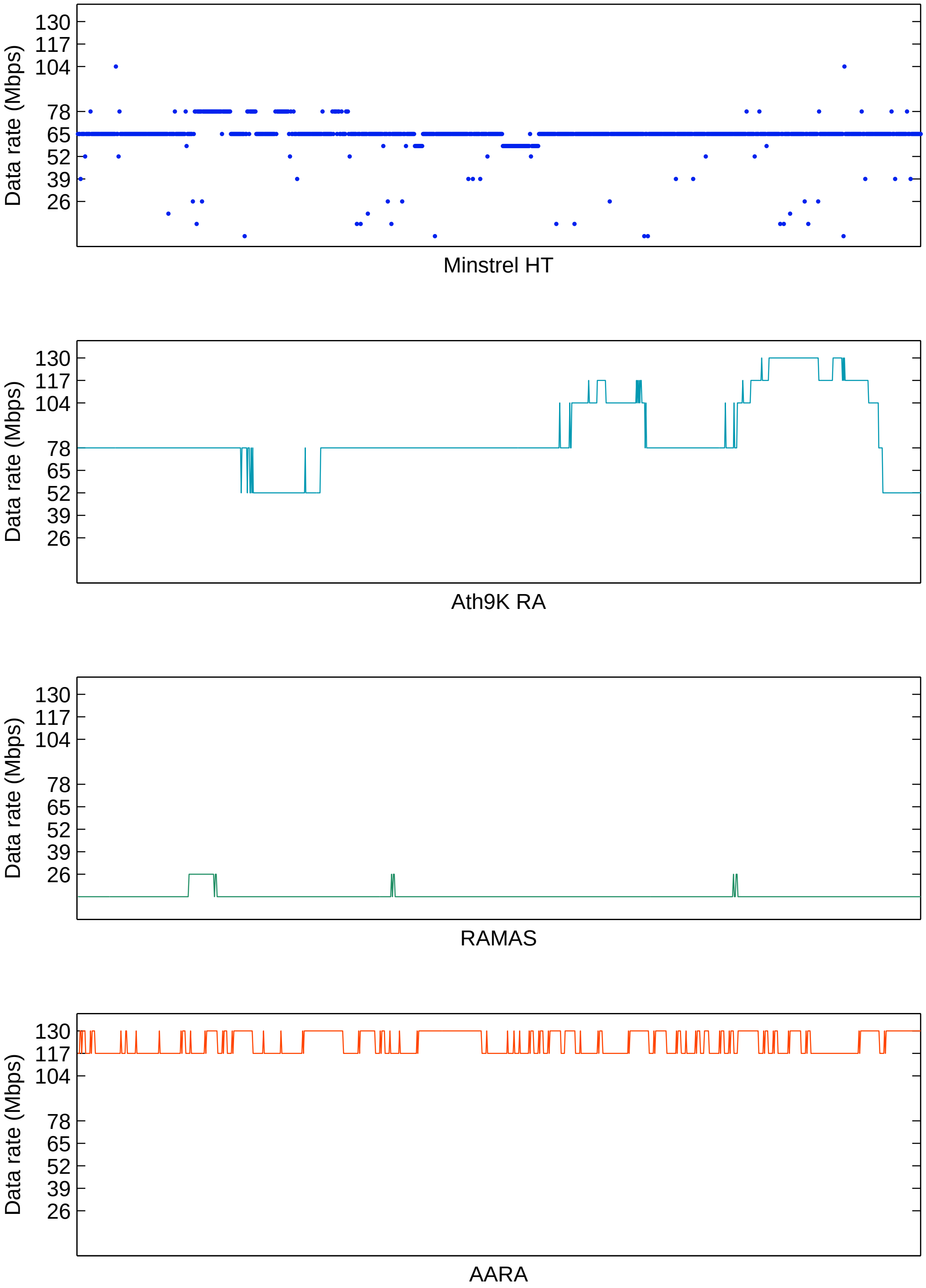
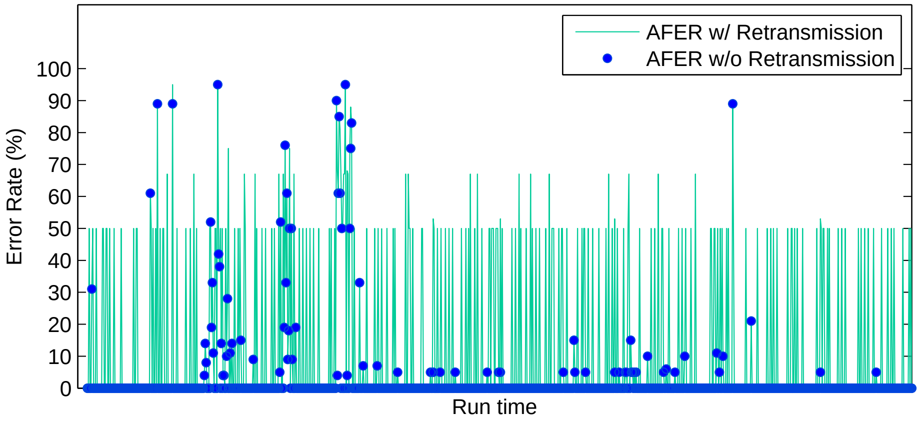


Figure 5.21: AFER w/ and w/o Retransmissions in *In-range* Contention Setting

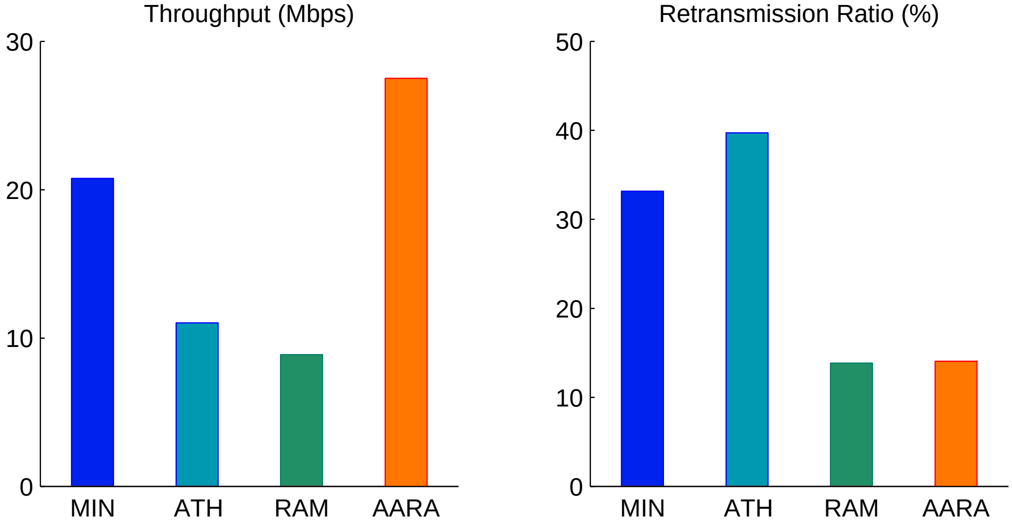


the another weaker signal, the *physical layer capture effect* occurs and hidden collision effect may diminish even with simultaneous transmissions. According to the rate evolutions presented in Figure 5.24, the potential best rate seems to be 78 Mbps. Unlike other algorithms, AARA successfully stays around the possible optimal rate in spite of significant congestion by a hidden station.

Similarly to other settings, AARA provides the best performance by up to 32.2%, 149.5%, and 209.0% over Minstrel HT, Ath9K RA, and RAMAS receptively. While the other algorithms perform poorly because they do not differentiate the causes of collisions, AARA is proved to effectively detect congestion caused by hidden stations and respond appropriately. Based upon the findings in Section 5.6.4, AARA considers the current link state as hidden congestion when there are more than three consecutive *A-MPDU failures* and initiates the RTS/CTS handshake to prevent collision-induced packet losses. Minstrel HT also activates RTS/CTS handshake, but because it initializes the protection from the second rate ($r1$) in the MRR chain and no such case occurred during testing, Minstrel HT never activated the handshake during the experiments.

The question is how long the handshake should be sustained to effectively avoid the congestion. In Figure 5.23, the throughput performance is represented as the duration of RTS/CTS exchange grows by 100ms per unit. Since AARA performs best when the control

Figure 5.22: Throughput and Retransmission Ratio in Hidden Collision Setting



mechanism is activated for 100ms, it has been set as the parameter for RTS/CTS activation duration.

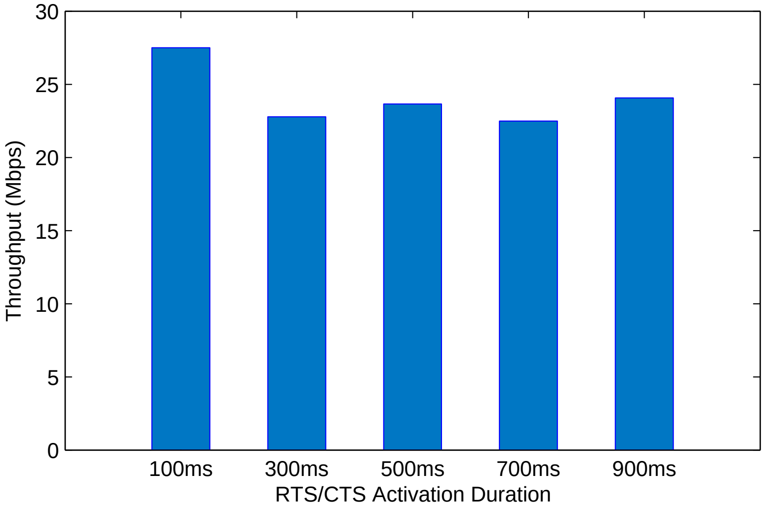
Thanks to the protection mechanism, retransmission ratio dropped by 42.5% and 35.5% compared to Minstrel HT and Ath9K RA respectively. On the other hand, RAMAS exhibited a slightly lower retransmission ratio than AARA because of the *capture effect*. RAMAS transmitted most of the packets at the second lowest rates (13 Mbps) due to the serious losses from collisions. However, since the lower rates have the strongest signal, the receiver could intelligently filter out the weaker signal from the hidden station.

5.9 Summary

In this chapter, based on extensive heuristic analysis of the impact of Frame Aggregation on Rate Adaptation algorithm, a novel rate adaptation scheme called *AARA* is designed and evaluated.

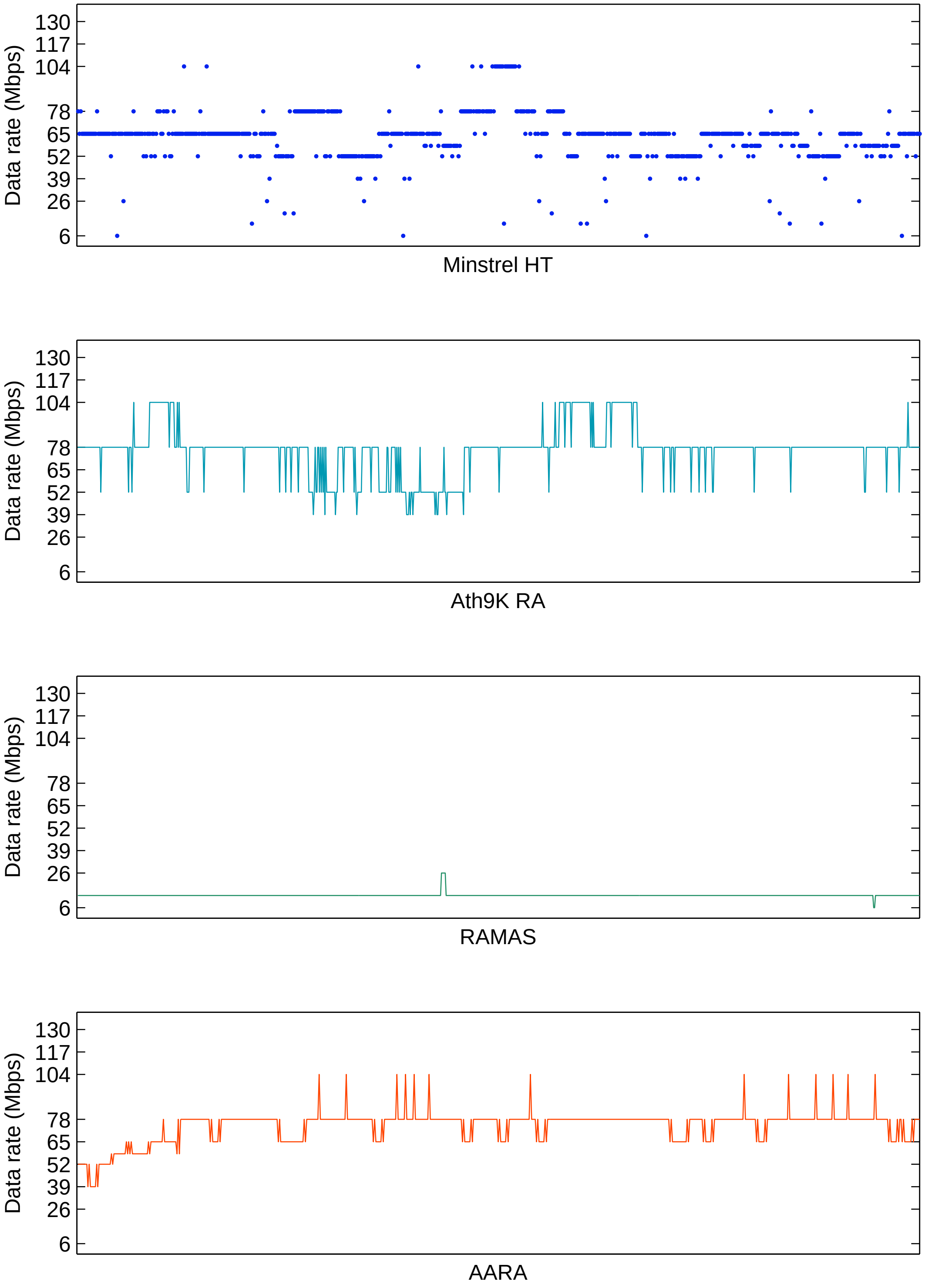
AARA significantly improves network performance over existing RAAs by the following enhancements. First, it estimates the channel using a normal A-MPDU packet with reduced potential risk of excessive frame losses. Second, AARA alleviates drastic rate changing by adopting aggregation level control before and after the shift. Third, based on the case study,

Figure 5.23: Throughput Comparison according to RTS/CTS Operation Duration



AARA efficiently differentiates the collision-induced frame losses based upon the implication of *A-MPDU failure*. The efficacy of the proposed algorithm was proved by extensive tests conducted in various environments.

Figure 5.24: Rate Evolution in Hidden Contention Setting



Minstrel HT

Ath9K RA

RAMAS

AARA

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The goals of this dissertation were to study various design factors required for efficient Rate Adaptation algorithm and to implement agile Rate Adaptation algorithms to optimize wireless link performance. In order to achieve this goal, a comprehensive survey of a wide range of RA schemes is carried out in Chapter 2. Along with categorizing and classifying popular existing RA algorithms, this chapter also summarized related work associated with potential factors of RA design.

Chapter 3 describes design challenges and questions regarding RA. In summary, the proposed RAs should address the following questions: (1) For static environments, how does the proposed solution effectively find and switch to the best rate while remaining adaptable to unpredictable short-term channel fluctuation with minimal overhead? (2) For mobile environments, how does the new algorithm effective adapt to varying link conditions to obtain an instantaneous optimal rate? (3) For congested environments, how does the RA scheme intelligently distinguish the potential reason for the loss to employ adaptable responses?

Two novel RA schemes are proposed to resolve these research challenges with extensive performance evaluation. In Chapter 4, Rate Adaptation with Collision Differentiation (RACD) is proposed to address the question of collision differentiation. RACD appears to be is the first algorithm which attempts to differentiate *In-range* collision from hidden collision by using *CTS-to-self* control packet. After classifying the error sources, RACD responds appropriately by actions such as initiating RTS/CTS handshake and adjusting Contention Window size. Extensive simulation results confirm that RACD significantly improves performance, especially in congestion scenarios.

Chapter 5 addresses the fact that typical RA algorithms do not consider Frame Aggregation in rate decisions even though FA is one of the dominant innovations specified in the new amendment. The impact of Frame Aggregation (FA) on RA algorithm design is closely investigated from various perspectives with a case study. Based on the analysis and findings from the case study, a novel AARA (Aggregation Aware Rate Adaptation) algorithm is proposed and evaluated via exhaustive experiments carried out in controlled and repeatable environments. The experiment results verify that AARA finds the optimal rate with minimized probing overhead in the static setting, swiftly switches to short-term best rate in the mobile setting, and intelligently identifies the congestion-induced frame losses in the congestion setting. The improved performance mainly results from efficiently controlled probing packets, a bumping period during the rate transition, and a new contention-detection metric used for collision differentiation.

6.2 Future Work

Even though the proposed solutions address several challenging problems, drawbacks and limitations remain which are worthy of future research.

First, the proposed RA assumes that the network traffic is saturated. In case of unsaturated traffic scenario, mandating a maximal aggregation level may cause additional delay during which the transmission queue should be filled enough to complete an aggregated packet. This delay may lead to performance degradation due to the time constraints in real time application. In addition, the delay in transmission may affect the TCP performance which mandates the specified arrival time. In addition to the problem caused by the transmission delay, *Coherence time* may affect the efficiency of frame aggregation. *Coherence time* is defined as the time-interval during which the channel stays sufficiently constant for the receiver to be able to decode the transmitted signal with a certain data rate. If the coherence time decreases, meaning that the channel condition changes faster over time (called *Fast fading*), holding the maximum possible aggregation level can lead to higher FER due to the long transmission duration of A-MPDU packets. To address all the issues related to time domain, an interesting topic for future research would be adaptive aggregation level control which adaptively optimizes the level of aggregation according to both channel status and traffic pattern.

Fairness in transmission can be another problem with RA design. Aggregation may impose serious imbalances between potential transceivers so that some of them could starve due to biased opportunity for channel access. As briefly mentioned in Section 5.6.5, some researchers have attempted to develop a metric to identify the multiplexing structure, but these efforts are either too complicated or are incompliant with the standard. Future research could develop a simple and useful metric which conforms to the standard and efficiently identifies the multiplexing structure.

Besides the topics listed above, there are numerous issues regarding to channel adaptation because of the unpredictable nature of the wireless channel medium.

Bibliography

[1] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "Mimo rate adaptation in 802.11n wireless networks," in Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, ser. MobiCom '10. New York, NY, USA: ACM, 2010, pp. 257–268. [Online]. Available: http://doi.acm.org/10.1145/1859995.1860025

[2] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "Joint rate and channel width adaptation for 802.11 mimo wireless networks," in Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on, 2013, pp. 167–175.

[3] S. Biaz and S. Wu, "Rate adaptation algorithms for ieee 802.11 networks: A survey and comparison," in Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, July 2008, pp. 130–136.

[4] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid rate control for ieee 802.11," in Proceedings of the Second International Workshop on Mobility Management &Amp; Wireless Access Protocols, ser. MobiWac '04. New York, NY, USA: ACM, 2004, pp. 10–18. [Online]. Available: http://doi.acm.org/10.1145/1023783.1023787

[5] A. Kamerman and L. Monteban, "Wavelan-ii: a high-performance wireless lan for the unlicensed band," Bell Labs Technical Journal, vol. 2, no. 3, pp. 118–133, 1997. [Online]. Available: http://dx.doi.org/10.1002/bltj.2069

[6] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, ser. MSWiM '04. New York, NY, USA: ACM, 2004, pp. 126–134.

[7] F. Maguolo, M. Lacage, and T. Turletti, "Efficient collision detection for auto rate fallback algorithm," in Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, July 2008, pp. 25–30.

[8] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in Proceedings of the 12th annual international conference on Mobile computing and networking, ser. MobiCom '06. ACM, 2006, pp. 146–157.

[9] I. Pefkianakis, S. Wong, H. Yang, S.-B. Lee, and S. Lu, "Toward history-aware robust 802.11 rate adaptation," Mobile Computing, IEEE Transactions on, vol. 12, no. 3, pp. 502–515, March 2013.

[10] R. T. Morris, J. C. Bicket, and J. C. Bicket, “Bit-rate selection in wireless networks,” Masters thesis, MIT, Tech. Rep., 2005.

[11] “ONOE Rate Control,” http://madwifi.org.

[12] “Atheros ra,” http://wireless.kernel.org/en/users/Drivers/ath9k.

[13] “Minstrel ra specification,” http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel.

[14] D. Nguyen, J. Garcia-Luna-Aceves, and C. Westphal, “Throughput enabled rate adaptation in wireless networks,” in Computing, Networking and Communications (ICNC), 2013 International Conference on, Jan 2013, pp. 1173–1178.

[15] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode, “Cars: Context-aware rate selection for vehicular networks,” in Network Protocols, 2008. ICNP 2008. IEEE International Conference on, Oct 2008, pp. 1–12.

[16] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden, “Improving wireless network performance using sensor hints,” in Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’11. Berkeley, CA, USA: USENIX Association, 2011, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1972457.1972486

[17] G. Holland, N. Vaidya, and P. Bahl, “A Rate-adaptive MAC Protocol for Multi-Hop Wireless Networks,” in Proceedings of the 7th annual international conference on Mobile computing and networking, ser. MobiCom ’01. ACM, 2001, pp. 236–251.

[18] K. Lee, J. Navarro, T. Chong, U. Lee, and M. Gerla, “Trace-based evaluation of rate adaptation schemes in vehicular environments,” in Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st, May 2010, pp. 1–5.

[19] J. Camp and E. Knightly, “Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation,” in Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, ser. MobiCom ’08. New York, NY, USA: ACM, 2008, pp. 315–326. [Online]. Available: http://doi.acm.org/10.1145/1409944.1409981

[20] O. Punal, H. Zhang, and J. Gross, “Rfra: Random forests rate adaptation for vehicular networks,” in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a, June 2013, pp. 1–10.

[21] G. Judd, X. Wang, and P. Steenkiste, “Efficient channel-aware rate adaptation in dynamic environments,” in Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, ser. MobiSys ’08. New York, NY, USA: ACM, 2008, pp. 118–131. [Online]. Available: http://doi.acm.org/10.1145/1378600.1378615

[22] X. Chen, P. Gangwal, and D. Qiao, “Ram: Rate adaptation in mobile environments,” Mobile Computing, IEEE Transactions on, vol. 11, no. 3, pp. 464–477, March 2012.

[23] M. Vutukuru, H. Balakrishnan, and K. Jamieson, “Cross-layer wireless bit rate adaptation,” in Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, ser. SIGCOMM ’09. New York, NY, USA: ACM, 2009, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/1592568.1592571

[24] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, “Frequency-aware rate adaptation and mac protocols,” in Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, ser. MobiCom ’09. New York, NY, USA: ACM, 2009, pp. 193–204. [Online]. Available: http://doi.acm.org/10.1145/1614320.1614342

[25] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, “Accurate: Constellation based rate estimation in wireless networks,” in In USENIX NSDI, 2010.

[26] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Predictable 802.11 packet delivery from wireless channel measurements,” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. –, Aug. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=2043164.1851203

[27] G. Martorell, F. Riera-Palou, and G. Femenias, “Cross-layer fast link adaptation for mimo-ofdm based wlans,” Wirel. Pers. Commun., vol. 56, no. 3, pp. 599–609, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11277-010-9992-9

[28] I. Pefkianakis, Y. Hu, S.-B. Lee, C. Peng, S. Sakellaridi, and S. Lu, “Window-based rate adaptation in 802.11n wireless networks,” Mobile Networks and Applications, vol. 18, no. 1, pp. 156–169, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11036-011-0347-x

[29] Q. Xia, M. Hamdi, and K. Ben Letaief, “Open-loop link adaptation for next-generation ieee 802.11n wireless networks,” Vehicular Technology, IEEE Transactions on, vol. 58, no. 7, pp. 3713–3725, 2009.

[30] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, “Joint rate and channel width adaptation for 802.11 mimo wireless networks,” in Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on, 2013, pp. 167–175.

[31] A. Makhlouf and M. Hamdi, “Practical rate adaptation for very high throughput wlans,” Wireless Communications, IEEE Transactions on, vol. 12, no. 2, pp. 908–916, 2013.

[32] S. Lakshmanan, S. Sanadhya, and R. Sivakumar, “On link rate adaptation in 802.11n wlans,” in INFOCOM, 2011 Proceedings IEEE, 2011, pp. 366–370.

[33] Q. Pang, V. Leung, and S. Liew, “A rate adaptation algorithm for ieee 802.11 wlans based on mac-layer loss differentiation,” in Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on, Oct 2005, pp. 659–667 Vol. 1.

[34] J. Kim, S. Kim, S. Choi, and D. Qiao, “CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs,” in INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006, pp. 1–11.

[35] K. Zhang, A. Lim, S. Wu, and Q. Yang, “A High TCP Performance Rate Adaptation Algorithm for IEEE 802.11 Networks,” International Journal of Computer Networks & Communications (IJCNC), vol. 2, no. 6, pp. 31–44, 2010.

[36] A. Ngugi, Y. Chen, and Q. Li, “Rate adaptation with nak-aided loss differentiation in 802.11 wireless networks,” in Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, Nov 2009, pp. 1–6.

[37] S. Rayanchu, A. MISHRA, D. Agrawal, S. Saha, and S. Banerjee, “Diagnosing wireless packet losses in 802.11: Separating collision from weak signal,” in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, April 2008, pp. –.

[38] S. Biaz and S. Wu, “Loss differentiated rate adaptation in wireless networks,” in Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE, March 2008, pp. 1639–1644.

[39] M. Khan and D. Veitch, “Isolating physical per for smart rate selection in 802.11,” in INFOCOM 2009, IEEE, April 2009, pp. 1080–1088.

[40] H. Jung, T. T. Kwon, K. Cho, and Y. Choi, “React: Rate adaptation using coherence time in 802.11 {WLANs},” Computer Communications, vol. 34, no. 11, pp. 1316 – 1327, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366411000594

[41] P. Acharya, A. Sharma, E. Belding, K. Almeroth, and K. Papagiannaki, “Rate adaptation in congested wireless networks through real-time measurements,” Mobile Computing, IEEE Transactions on, vol. 9, no. 11, pp. 1535–1550, Nov 2010.

[42] K. V. Cardoso and J. F. Rezende, “Increasing throughput in dense 802.11 networks by automatic rate adaptation improvement,” Wirel. Netw., vol. 18, no. 1, pp. 95–112, Jan. 2012. [Online]. Available: http://dx.doi.org/10.1007/s11276-011-0389-9

[43] M. Kim and C.-H. Choi, “Hidden-node detection in ieee 802.11n wireless lans,” Vehicular Technology, IEEE Transactions on, vol. 62, no. 6, pp. 2724–2734, 2013.

[44] P. Chaporkar, A. Proutiere, and B. Radunoviac, “Rate Adaptation Games in Wireless LANs: Nash Equilibrium and Price of Anarchy,” in INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–9.

[45] “The Network Simulator-ns-3,” http://www.nsnam.org.

[46] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, “Aggregation with fragment retransmission for very high-speed wlans,” Networking, IEEE/ACM Transactions on, vol. 17, no. 2, pp. 591–604, April 2009.

[47] D. Skordoulis, Q. Ni, H.-H. Chen, A. Stephens, C. Liu, and A. Jamalipour, “Ieee 802.11n mac frame aggregation mechanisms for next-generation high-throughput wlans,” Wireless Communications, IEEE, vol. 15, no. 1, pp. 40–47, 2008.

[48] T. Selvam and S. Srikanth, “A frame aggregation scheduler for ieee 802.11n,” in Communications (NCC), 2010 National Conference on, 2010, pp. 1–5.

[49] B. Ginzburg and A. Kesselman, “Performance analysis of a-mpdu and a-msdu aggregation in ieee 802.11n,” in Sarnoff Symposium, 2007 IEEE, 2007, pp. 1–5.

[50] Y. Lin and V. Wong, “Wsn01-1: Frame aggregation and optimal frame size adaptation for ieee 802.11n wlans,” in Global Telecommunications Conference, 2006. GLOBECOM ’06. IEEE, 2006, pp. 1–6.

[51] X. He, F. Li, and J. Lin, “Link adaptation with combined optimal frame size and rate selection in error-prone 802.11n networks,” in Wireless Communication Systems. 2008. ISWCS ’08. IEEE International Symposium on, Oct 2008, pp. 733–737.

[52] M. Kim and C.-H. Choi, “Probing-based link adaptation for high data rate wireless lans,” Wireless Communications, IEEE Transactions on, vol. 11, no. 7, pp. 2382–2390, 2012.

[53] “Iperf,” http://iperf.sourceforge.net/.

[54] “Hostapd,” http://wireless.kernel.org/en/users/Documentation/hostapd.

[55] “Msi mini computer,” http://us.msi.com/product/ipc/.

[56] “Compat-wireless,” https://www.kernel.org/pub/linux/kernel/projects/backports/stable/.

[57] “Wireshark packet sniffer,” http://www.wireshark.org.

[58] “AirPcap Nx,” http://www.airpcap.nl/airpcap-nx.htm.