TOWARDS A HOLISTIC RATE ADAPTATION FOR 802.11n/ac MIMO SYSTEMS

ADITYA KULKARNI
Bachelor in Computer Science 2011

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
December 2015

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Aditya Kulkarni
17 December 2015

Acknowledgments

First and foremost, I would like to express my deepest and sincerest gratitude to my supervisor Prof. Ben Leong. Without his unwavering support and encouragement this work would not have been possible. I thank him for the great guidance he has granted me for more than three years', and I thank him for all I learned from him, a learning which was not just limited to academic but also the philosophy of life. I believe that these instructions (and even scoldings) will forever be a source of guidance and encouragement.

My deepest gratitude also goes to Wang Wei, who offered me motivation when I needed it the most in my initial phase, and also extended help and guidance for my work. He helped me greatly and selflessly to get through those tough times. I thank Anirudh and Raj, who worked with me for most of the time and helped me so much in my experiments and project implementation. I also thank Wang Zixiao, Meng Xiangyun, Wai Kay Leong, Xu Yin, Shuaizhao Jing, Wang Qiang, Li Yi, Soedarsono Lim, Nikita Bhargava and Daryl Seah who have spent a great time together as lab mates.

I was also grateful to have worked for Prof. Roger Zimmermann as his TA for one semester. Roger is an outstanding, most humble teacher I have ever seen and he will always be a source of inspiration for me.

I also thank my friends in SoC for their company. I thank Abhay, Kuldeep, Ankit, Vishal, Malay, Narendra, Mobashir, Dipanjan, Priyanka and Rajiv for being with me all these years. I have really enjoyed our company and I'll always cherish these moments. Special thanks to Ashok, who shared a pleasant stay with me in the UTown apartment for almost three years. I promise that I will miss every single one after my graduation from NUS.

I would like to thank my few best friends back at home, namely Mandar, Rajan, Swapnil, Gaurav, Akash, Rohit, Pritesh, Neha, Pankaj, and Sonakshi who have been so tolerant for me over the years. I would especially like to thank Ankita for giving me a strong motivation to go for higher studies and for making me strong enough to face the harsh phases of life.

I want to thank the whole team of Science For Society (S4S) and CareNX namely Shantanu, Gopal, Vaibhav, Ganesh^2, Vinayak, Shital^2, Swapnil, Anurag, Avinash and Harshad. Without their support, I would never have applied and continued for abroad studies.

Finally, I would like to remember my loving grandparents for taking care of me as a child. I believe I stand here today only because of them. I will never ever forget the learnings I received from them on how to take on the hard life. I also owe my greatest thanks to my parents. I thank them for being strong and supportive over the years. I thank my sisters Vaishali and Dipali, who have always cared about me since the very beginning. I thank my friend Prof. Amol and other school friends from India, for being so supportive to my family while I was far away. I also thank my cousins and other relatives for encouraging me time-to-time to work harder.

Words are not enough for me at this very moment to thank all of you for your most heartfelt love and the warmest support. I feel the luckiest human being alive to have all of you around. I love you all, and this thesis is truly dedicated to all of you.

Table of Contents

List of Figures

Abstract

A large number of rate adaptation (RA) schemes have been developed for modern 802.11n MIMO systems. Surprisingly, we found that, in the existing literature there has not been any systematic evaluation of the impact of all the available 802.11n MIMO features, namely frame aggregation (FA), channel bonding (CB), guard interval (GI) and stream modes (SM). In this thesis, we study the impact of these features systematically in order to set or adapt them in a more principled manner and propose a new RA algorithm called *Holistic Rate Adaptation* (or HiRA) for 802.11n/ac systems. We found that, in general, by fixing FA to 'enabled', CB to 40 MHz and GI to 400 ns, we can significantly reduce the search space in doing rate adaptation without loss of performance. HiRA also incorporates *goodput-based* probing to improve probing accuracy, hysteresis to improve stability, and a number of minor innovations. We show that HiRA is able to achieve goodput gains up to 37% in an interference-free scenario, 26% with adjacent channel interference, 27% with co-channel interference, 18% with hidden terminal interference, 11% in a real world mobility scenario and 13% "in the wild" over the next best performing state-of-the-art RA scheme.

Chapter 1

Introduction

Rate Adaptation (RA) is a very well studied problem for single antenna Single-Input-Single-Output (SISO) 802.11a/b/g systems [9, 10, 19, 20, 23]. With the proliferation of Multiple-Input-Multiple-Output (MIMO) technology, many RA schemes have also been proposed for multi-antenna 802.11n MIMO systems [4, 11, 12, 14, 15, 22, 24, 26, 27, 28].

The transition from 802.11a/b/g to 802.11n standard introduced many more Modulation and Coding Schemes (MCS) for the bitrates at the physical layer. In addition, there is new Channel Bonding (CB) feature, where two 20 MHz channels of the frequency spectrum can be 'bonded' together into a single wider 40 MHz channel to double the PHY bitrate. In the new standard, it is also possible to adjust the Guard Interval (GI) between the symbols in order to further trade-off higher bitrate against reliable transmission. To reduce ACK overheads, 802.11n also supports Frame Aggregation (FA) which combines multiple MAC frames into a single aggregated frame (A-MPDU). These new features allow 802.11n systems to achieve a 12-fold increase (up to 600 Mbps) in the maximum theoretical transmission rate compared to legacy 802.11a/b/g systems (54 Mbps).

However, due to the several new features, the number of possible setting combinations have increased significantly to 192 for a 3×3 antenna 802.11n system. This in turn has significantly increased the search space for 802.11n MIMO RA schemes. While investigating existing algorithms (shown in Table 1.1), we observed that most schemes fix some of the features, while dynamically adapting to the rest, and there seems to be no common agreement in the literature on which MIMO features should be fixed, and which should be adapted. We thus adopted a clean slate approach to systematically determine which of the MIMO features could be fixed. The results of our measurement study conducted over a 802.11n indoor

Table 1.1: Existing 802.11n MIMO RA schemes.

RA Scheme	**Year**	**FA**	**CB**	**GI**	**#SM**
Minstrel_HT [4]	2010	ON	Adapt	Adapt	Adapt 3
MiRA [28]	2010	ON	40 MHz	800 ns	Adapt 2
MISS [11]	2011	ON	–	–	Adapt 2
RAMAS [26]	2011	ON	Adapt	Adapt	Adapt 4
Deek et al. [14]	2011	OFF	Adapt	Adapt	Adapt 2
Lakshmanan et al. [24]	2011	–	–	–	Adapt 2
ARAMIS [15]	2013	OFF	Adapt	–	Adapt 2
WRA [27]	2013	ON	40 MHz	800 ns	Adapt 2
ORS [12]	2014	–	–	–	Adapt 4
SampleLite+ [22]	2015	ON	Adapt	800 ns	Adapt 2

testbed with diverse links under different operating scenarios suggest that a good policy would be to enable FA, to set CB to 40 MHz and GI to 400 ns.

By fixing FA, CB and GI, we effectively reduce the search space by 87.5%, which makes simple *hill-climbing* probing strategy plausible for the remaining 24 setting combinations. We also made the following three observations in our investigation of existing RA schemes:

1. Goodput does not always decrease when loss rates increase. It is sometimes possible that when the MCS rate is increased, both loss rate and goodput might increase. This is possible because the higher data rates could potentially more than offset the increased loss rate. This suggests that a *goodput-based* probing strategy instead of the traditional *loss-based* probing strategy would be more accurate.

2. Existing RA schemes are often overly aggressive in probing for better data rates. We can improve performance by introducing some amount of hysteresis and reducing the aggressiveness of probing. We can also improve the probing efficiency using a single-A-MPDU-based probing strategy, which surprisingly is effective in practice.

3. Hidden terminal interference can be easily inferred from repeated A-MPDU retransmissions and can be handled either by increasing the MCS rate, or by enabling RTS/CTS, depending on the circumstances.

We incorporated these insights into a new RA algorithm, which we call *Holistic Rate Adaptation* (or HiRA).

Our evaluation of HiRA on a real 802.11n testbed shows that HiRA consistently outperforms existing state-of-the-art MIMO RA schemes over a wide range of controlled and real world operational scenarios. HiRA achieves goodput gains by up to 37% in an interference-free scenario, 26% with adjacent channel interference, 27% with co-channel interference, 18% with hidden terminal interference, 11% in a real world mobility scenario and 13% “in the wild” over the next best performing state-of-the-art RA scheme.

The rest of the thesis is organized as follows. In Chapter 2 we provide an overview of the existing 802.11 SISO and MIMO RA schemes. In Chapter 3, we present an extensive measurement study that investigates which MIMO features can be fixed without loss of performance. Chapter 4 then describes the proposed RA scheme (HiRA) which is based on the insights from the measurement study. Later, the evaluation results of our scheme are presented in Chapter 5. We discuss the future scope of this work and finally conclude in Chapter 6.

Chapter 2

Related Work

In this chapter, we provide an overview of the prior art in RA schemes for WiFi systems. RA in Single-Input-Single-Output (SISO) 802.11a/b/g systems is a very well studied problem [20]. The problem of RA in SISO 802.11a/b/g systems has been focused on the selection of the best physical bitrate or modulation scheme (among total 8 rates described in legacy 802.11a/b/g drafts) based on the time-varying channel quality [6]. However, the problem of RA for modern WiFi networks with Multiple-Input-Multiple-Output (MIMO) technology is more complicated. In addition to more physical bitrates or modulation schemes, 802.11n/ac systems offer many more features (like FA, CB, GI, SM, MCS etc.) to exploit PHY and MAC layer diversity, which in turn increases the inherent channel capacity for high speed data transfers. Because of these advancements, the search space for PHY bitrates is increased exponentially. Considering the 3×3 antenna setup for 802.11n, there are total 192 ($2 \times 2 \times 2 \times 3 \times 8$) bitrates to choose from. For 802.11ac, this space is even larger, including extensions to all the above features and addition of MU-MIMO or Beamforming techniques. Hence, existing SISO RA schemes can no longer be applied directly to leverage all feature enhancements for RA in MIMO, as they were designed to work with only 8 bitrates. Depending upon the design decisions, we classify the most relevant works into two broad categories: (a) RA schemes designed for the legacy 802.11a/b/g SISO systems and (b) RA schemes designed for the modern 802.11n/ac MIMO systems.

2.1 RA Schemes for 802.11a/b/g SISO Systems

Over the years, a large number of *open-loop* schemes have been proposed for Single-Input-Single-Output (SISO) 802.11a/b/g systems [9, 10, 19, 20, 23, 30, 31]. Open-loop schemes are generally more popular because they rely only on sender-side channel information to estimate link quality and to select the best rate. *Closed-loop* schemes like Receiver Based Auto Rate Adaptation (RBARA) [18] are also available. Closed-loop schemes are generally unpopular because they rely on explicit feedback from the receiver to estimate link quality to select the best rate and are much harder to deploy in practice. The design principles behind most of the RA schemes for legacy 802.11 systems are based on the loss observed during the process of rate selection, although there are few other schemes that prefer sampling to select the best performing rate.

2.1.1 Loss-based SISO RA Schemes

Auto Rate Fallback (ARF) was the first RA scheme which took into account the multi-data rate aspect of the wireless channel [19]. At the beginning the frames are sent at a basic rate of 2 Mbps and after every n successful transmissions (default 10 for ARF), the rate is increased to the next higher data rate. The data rate is lowered to the next lower data rate if 2 consecutive acknowledgements are lost. The sequential increase or decrease of rate based upon the loss happen to be the reason of unnecessary frame loss by constantly trying to increase the data rate even when the channel is stable.

Adaptive Auto Rate Fallback (AARF) [23] is an extension to the ARF scheme to address the issue of unnecessary frame loss due to sequential increase in the rate. Similar to the ARF scheme, the rate is increased after every n successful transmissions. If the first frame in the sent data is lost then AARF takes two actions: (a) Lower the rate to the earlier rate and (b) Increase the next transmission window to $2n$ frames. These actions ensure that the AARF performs better than ARF during the stable channel, however this performance gain also makes RA a bit slower as it takes longer time to settle at the best rate in the dynamic environment.

Collision Aware Rate Adaptation (CARA) differentiates between frame losses due to channel degradation and collisions [20]. If a frame is lost at a certain rate then a RTS is sent at that rate. If the RTS is also lost then the loss is said to be because of collision, since RTS is assumed to be resilient to channel fading. Otherwise the loss is supposed to be happened because of collision. The data rate is reduced only if the frame loss occurs due to channel degradation and not due to collisions. This policy further helps to reduce the number of collisions due to the selection of the higher lossy data rate. The drawback of this approach often stalls CARA because if the channel condition is extremely poor then RTS cannot be sent and CARA stays in that specific rate.

ONOE is a credit based rate adaptation algorithm [5]. It finds the data rates that showed less than 50% loss rate. The data rate selection is based on statistics collected from the transmissions and is adjusted at the end of every 1 second. ONOE is not very sensitive to bursty losses and unresponsive to fast changes since the rate is adjusted only at the end of every 1 second.

Loss Differentiation Rate Adaptation (LDRA) predicts the reason for the frame loss [9]. Whenever a frame is lost, then it is retransmitted at the basic rate. If the frame retransmitted at the basic rate fails to be delivered then the loss is said to be due to channel fading. If it is delivered successfully then the earlier loss was due to collisions. The rate is increased or decreased based upon the estimate of the portion of the sent frames that are predicted to be lost due to channel fading.

Robust Rate Adaptation Algorithm (RRAA) calculates the frame loss ratio at the end of every short time cycle and uses it to adapt between different rates [30]. RRAA has introduced two thresholds for the loss rate in order to determine the data rate for next transmission. If a frame is lost due to a collision, RRAA enables RTS/CTS to eliminate further collisions due to hidden terminals.

2.1.2 Sampling-based SISO RA Schemes

SampleRate scheme uses sampling to find the near optimal performing data rate [10]. At every 10 seconds it randomly selects a rate from the bunch of data rates whose average frame transmission time is less than the average lossless frame transmission time. If an acknowledgement is received then the same data rate is used for successive transmissions. Noticeably it does not need to strictly increase the data rate in a sequential manner. If 4 consecutive transmissions fail then the data rate is reduced to the next lower data rate.

Driver `ath9k` [1] is the most commonly used open-source 802.11 driver and it uses Minstrel [31] as the default RA scheme for legacy 802.11a/b/g systems. Minstrel RA scheme is a sampling based method which increments or decrements the rate based upon the current throughput and associated success probability of the current rate. It updates these loss and probability statistics at the least granularity of every 100 ms for the current rate.

2.1.3 Feedback-based SISO RA Schemes

Receiver Based Auto Rate Adaptation (RBARA) scheme is the first to use control frames RTS/CTS [18] for the closed-loop type feedback. In RBARA, a sender selects a data rate and sends the RTS along with the frame. The receiver on receiving the RTS, obtains the SNR and translates (maps) it to the data rate as a measure of estimation of the channel quality. Receiver then informs about this mapped/chosen data rate to the sender by embedding this information in the CTS frame. On receiving the CTS frame, sender then extracts the data rate information and sends the next frame at that rate. Since the RTS/CTS frames have to be modified in the implementation part, RBARA is not compliant with existing 802.11 drivers.

2.2 RA Schemes for 802.11n/ac MIMO Systems

With the recent development of 802.11n/ac Multiple-Input-Multiple-Output (MIMO) systems, a large number of new RA schemes for MIMO systems have been developed [4, 11, 12, 14, 22, 24, 26, 27, 28]. Many of the schemes for SISO systems have been adapted for the new operating environment and they generally work well for MIMO. However, we found that there was no systematic evaluation of the impact for all the available 802.11n MIMO features, namely frame aggregation (FA), channel bonding (CB), guard interval (GI) and stream modes (SM), together, in the existing literature. Some algorithm ignore some features, others seem to set some of the features in a rather haphazard way. Our measurement study in §3, is to the best of our knowledge, the first systematic investigation of these features. Also most existing RA schemes are not specifically designed to handle hidden terminal interference scenarios. We systematically study hidden terminal problem from the perspective of MIMO features and design techniques to improve the performance.

2.2.1 Loss-based MIMO RA Schemes

MiRA [28] RA scheme is based on the key insight that within a stream mode there exists a monotonic relationship between loss and bitrates which we also leveraged in our design of intra-mode and inter-mode probing (§4). It is also one of the few RA schemes that can handle hidden terminals, but it uses a simple loss-based scheme to decide whether to turn on RTS/CTS. MiRA however, does not try to estimate goodput directly, and also uses LGI by default and thus misses out on the benefits of SGI. WRA is a variant of MiRA that opportunistically chooses the rates to exploit the short-term gains observed over a window of packets in the same stream mode [27].

ORS is the family of RA algorithms which learns the optimal RA settings as fast as possible based upon theoretical calculation of the success probability of the rate [12]. Although this approach might help in avoiding exhaustive probing and still allowing adaptation to all the features, it makes unrealistic assumptions about having the prior knowledge of the speed at which environment is changing and that makes it more of a theoretical interest.

2.2.2 Sampling-based MIMO RA Schemes

Minstrel_HT [4] is probably the most significant state-of-the-art MIMO RA scheme because it is the default link layer RA scheme for the latest versions of the `ath9k` driver that support 802.11n. Minstrel_HT performs exhaustive sampling across all 802.11n setting combinations. However, we showed, with a comprehensive measurement study in §3, that an exhaustive search is unnecessary and that it is possible to fix certain MIMO features in practice. Also, if the current choice of the rate turns out to be extremely lossy, Minstrel_HT tends to lower the rate by reducing the number of streams instead of reducing the MCS first.

The key insight behind SampleLite+ [22] is that the RSS of the received beacon frames can be used as a channel quality indicator to choose the rate at the sender. This insight was based on the observed monotonic relationship between best feature setting and RSS of the link. A major drawback of the scheme is that it works only in one direction, i.e. when a client has traffic to send to the AP and not vice versa. Also, the RSS thresholds used may not be applicable to all hardware platforms (c.f. Table 3.2 and Figure 3.10).

2.2.3 Limited-feature-based MIMO RA Schemes

The common drawback of many other MIMO RA schemes [11, 14, 24, 26] is that they focus on optimizing individual MIMO features, such Stream Modes or MCS rates independently, and thus often fail to converse at the optimal MIMO settings. Chen et al. proposed a loss-based SM switching technique called MISS [11] but did not consider adapting to CB or GI. Deek et al. proposed guidelines to choose a CB setting in different interference scenarios but miss out on the simultaneous adaptation with FA or GI [14]. Lakshmanan et al. proposed a metric called Median Multiplexing Factor (MMF) to choose a SM but never adapts to FA, CB and GI [24]. Nguyen et al. proposed a credit-based RA scheme called RAMAS [26]. The idea is to divide the 802.11n features into two groups: (a) The modulation group with different MCS values and (b) The enhancement group that consists of SM, CB and GI. The scheme adapts to these group independently and finally combine the results together to decide the overall best feature configuration.

2.2.4 Feedback-based MIMO RA Schemes

Closed-loop MIMO RA schemes [16, 32] require hardware modifications to retrieve the information about the channel quality from the receiver. Sending Channel State Information (CSI) back to the sender using a special feedback frame turns out to be expensive in terms of air-time overhead [13]. Thus Adaptive Feedback Compression (AFC) scheme proposed a technique to reduce this overhead by compressing the frame size [32]. As CSI is still not widely supported across all chipsets, schemes like ARAMIS [16] rely on universally obtainable channel quality metric such as diff-SNR as a feedback information. ARAMIS continuously profiles and learns the diff-SNR and best rate relationship over the time from each feedback frame. It then leverages this relationship to map the recently obtained diff-SNR value against the best rate from the sample space of rates to make a selection.

Chapter 3

Which MIMO Features Matter?

In this chapter, we take a principled approach to determine which of the MIMO features could be set to a fixed setting and which need to be adapted as per the operating environment. We carried out an extensive measurement study to investigate the impact of all possible combinations of MIMO feature settings in a 3×3 antenna setting. The measurement study encompassed a wide variety of links in three different operating scenarios.

Experimental setup. Our measurements were conducted in a large (28 m × 34 m) indoor hall with several pillars within our university campus (see Figure 3.1). We placed a custom-built 802.11n access point (AP) close to the wall at location P. After surveying the space, we found 9 representative locations that exhibited different combinations of three spatial properties: (i) proximity, (ii) line-of-sight (LOS) and (iii) signal strength w.r.t the AP (as shown in Table 3.1). Using a spectrum analyzer, we found that in the less crowded 5 GHz band no other AP or 802.11 device was operating over channels 149, 153 & 157 and hence we chose these channels for our controlled experiments. All our experiments were performed past midnight during university vacations to further avoid external interference due to user traffic. Our custom-built AP and client were built with commodity off-the-shelf hardware: Atheros AR9580 802.11n MIMO chipset with ath9k [1] open source driver and 3×3 MIMO antenna setting.

Methodology. In each experiment, the client was placed at 9 different locations in turn (Figure 3.1). At each location, we collected traces when the AP sent saturating UDP traffic to the client over all 192 setting combinations of {FA, CB, GI and SM} for 10 s each. We repeated the same measurements for two interference scenarios, adjacent and co-channel interference, where the interferer placed at location A sent interfering traffic to a receiver at

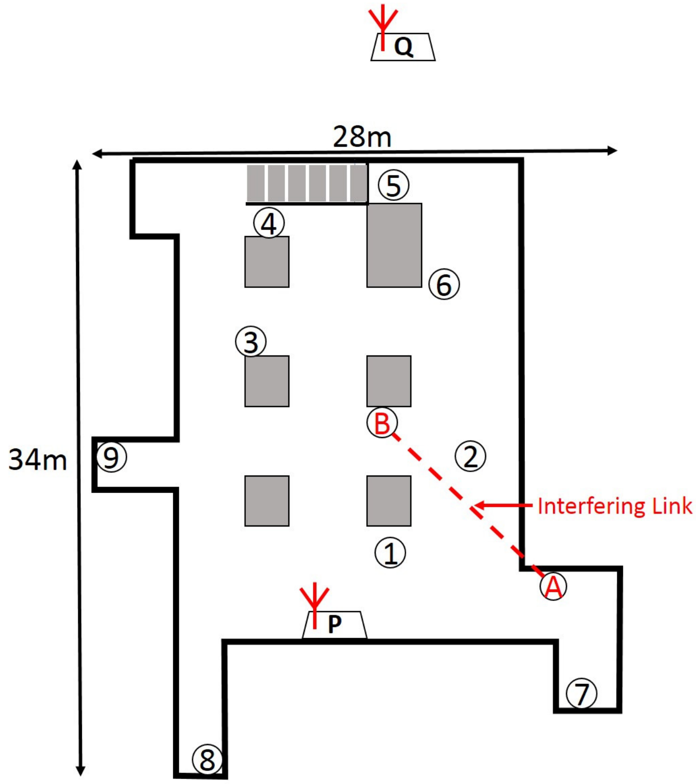

Figure 3.1: Static indoor experimental setup with 9 representative locations. The grey blocks are pillars or obstacles.

location B. The interfering link operated at 52 Mbps (MCS 5, 20 MHz CB, LGI) which is about 4000 MPDUs per second. Figure 3.2 shows the channel configurations for the adjacent and co-channel interference scenarios respectively. In order to correctly characterize the effect of interference, the Message in Message (MIM) mechanism was disabled. Note that we repeated our experiments with the AP at a different location but we found that the results were similar and dependent on the spatial properties described in Table 3.1. Thus for the remainder of this section, we only present the results for the 9 locations with AP at location P.

3.1 Frame Aggregation (FA)

To investigate the impact of Frame Aggregation (FA) on goodput, we divide the achieved goodput for all 192 setting combinations into two sets of 96 setting combinations each: one set with FA enabled and the other set with FA disabled. We plot the results for the three interference scenarios in Figure 3.3. The end points of the line represent the minimum and maximum goodput values achieved by some setting within the set and the boundaries of the box represent the lower and upper quartile values of the goodput. The horizontal line within

Table 3.1: Spatial properties of links in indoor testbed.

Location	**Proximity**	**LOS?**	**Signal Strength**
1	Near	Yes	Strong
2	Near	Partial	Medium-Strong
7	Near	No	Weak
3, 6	Moderate	Partial	Medium
9	Moderate	No	Weak
4	Far	Partial	Medium
5	Far	No	Weak
8	Far	No	Weak

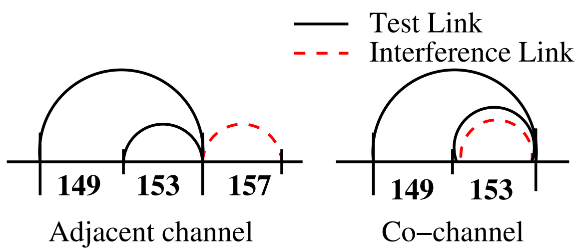

Figure 3.2: Channel configurations for adjacent and co-channel interference scenarios in 5 GHz band

each box shows the mean value. It is clear that enabling FA is favorable for all operating scenarios and the difference in goodput is very significant (i.e. on average a 5-fold increase). This is not surprising because FA reduces the overhead of the MAC header by aggregating multiple MAC frames in a single giant frame (A-MPDU) which in turn reduces the overall data transmission air-time. This suggests that 802.11n should generally be operated with FA enabled. The goodputs for locations 7, 8 and 9 are minimally affected in the interference scenarios because they are occluded from the interfering sender.

3.2 Channel Bonding (CB)

In §3.1, we concluded that FA can be fixed to be always enabled and thus we are left with 96 setting combinations. To investigate how to we should set the Channel Bonding (CB), we again divide the goodputs of the remaining 96 setting combinations (that have FA enabled) in two sets: one set with 20 MHz CB and the other set with 40 MHz CB. In Figure 3.4, we plot the goodputs of the two sets across 9 locations for the three interference scenarios. We found that 40 MHz CB was favorable in every scenario, which was not surprising because this would allow the link to exploit a wider channel capacity. This therefore suggests that CB

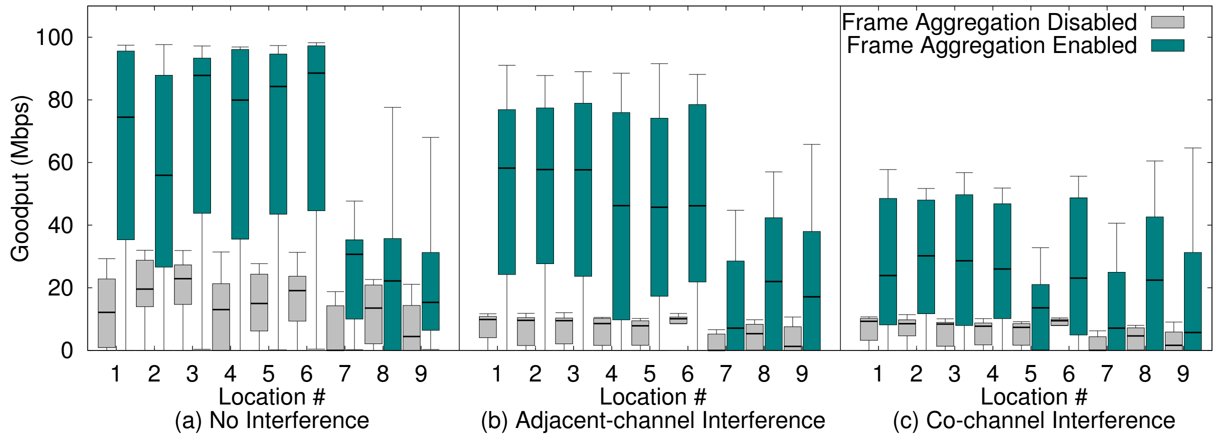

Figure 3.3: Comparison of impact of frame aggregation (FA).

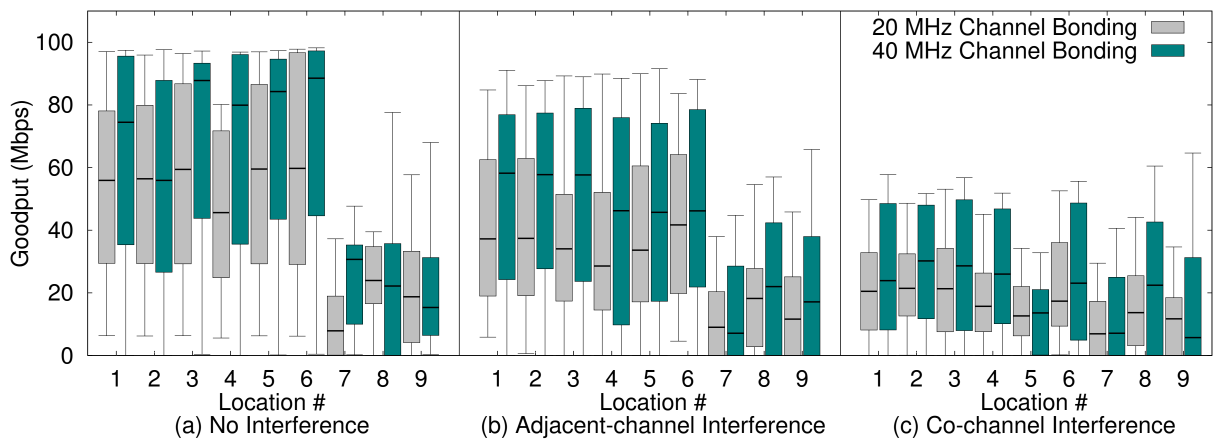

Figure 3.4: Comparison of impact of channel bonding (CB).

should in general be set to 40 MHz.

3.3 Guard Interval (GI)

After fixing FA to be always enabled (§3.1) and CB at 40 MHz (§3.2), the search space is reduced to 48 setting combinations. Now in order to investigate the impact of the GI setting, we again divided the goodputs of the remaining 48 setting combinations (that have FA enabled and 40 MHz CB) in two sets: one set with Short (400 ns) Guard Interval (SGI) and the other set with Long (800 ns) Guard Interval (LGI). In Figure 3.4, we plot the goodputs of the two sets across 9 locations for three interference scenarios. We found that SGI was favorable in every scenario, which was not surprising because SGI was expected to achieve a goodput improvement by up to 12% over LGI [33] by packing more data bits due to reduced inter-symbol spacing. Also SGI can reliably counter the inter-symbol interference in multi-path environments, as long as the difference between the longest and the shortest path does not

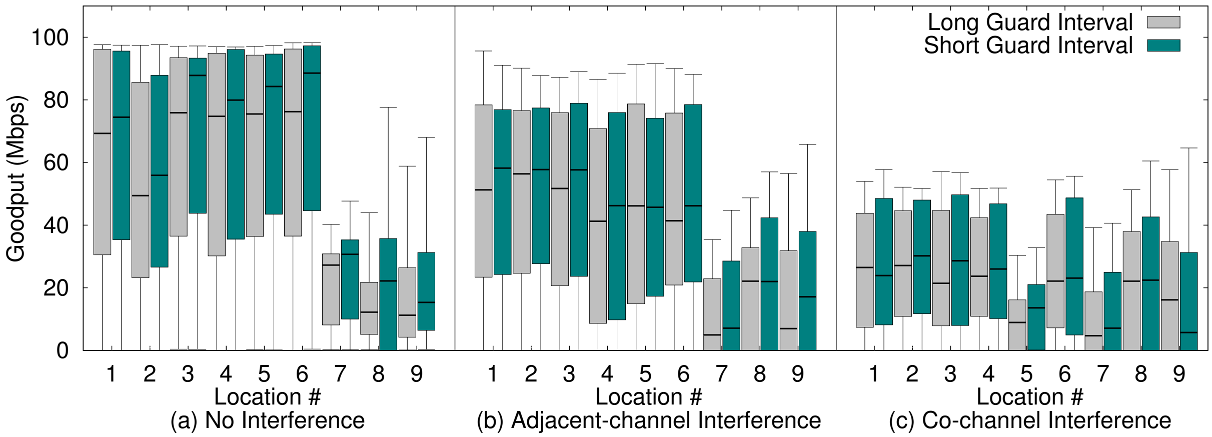

Figure 3.5: Comparison of impact of guard interval (GI).

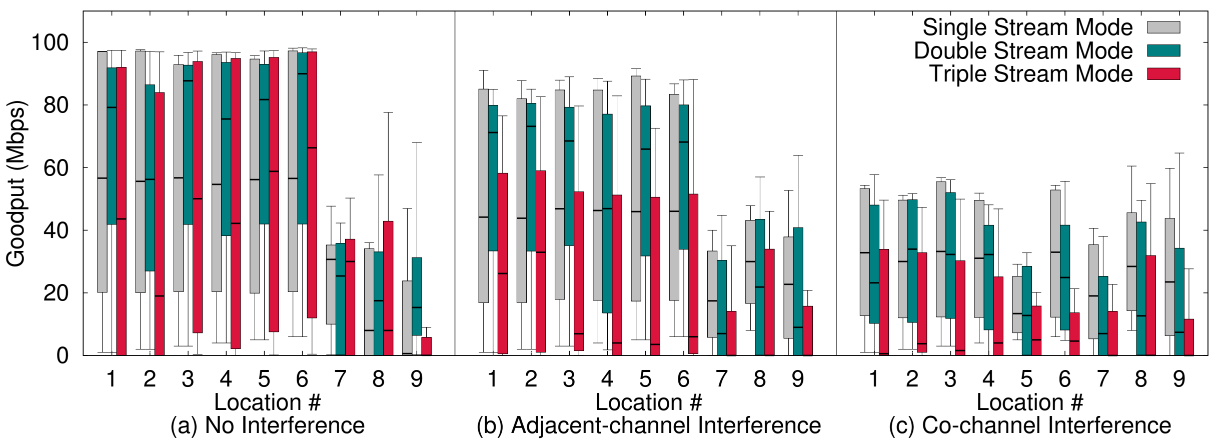

Figure 3.6: Comparison of impact of different stream modes (SM).

exceed 120 m [33] (which is very unlikely for indoor deployment).

3.4 Stream Mode (SM)

After fixing FA (§3.1), CB (§3.2) and GI (§3.3), we are left only with the 24 setting combinations arising from the 8 different MCS rates for each of the three stream modes (SMs). So, we divide the goodputs of the remaining 24 setting combinations into three sets for the three different stream modes. In Figure 3.5, we plot the goodputs of the three sets across 9 locations for the three interference scenarios. We found that for each interference scenario, different stream modes perform well at different locations. In particular, for the case of no interference, the occluded locations 7, 8 & 9, multiple streams perform better because the multipath environment allowed multiple transmission streams to partially overcome the losses arising form occlusion. Single stream transmission seemed to be sufficient for achieving optimal performance when there was line of sight. Interference seemed to degrade the performance

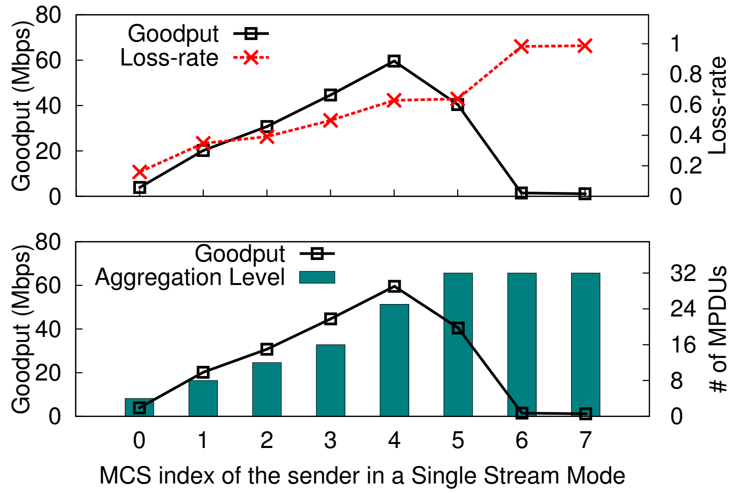

Figure 3.7: Goodput and loss characteristics of Single Stream MCS rates of the sender under hidden terminal interference.

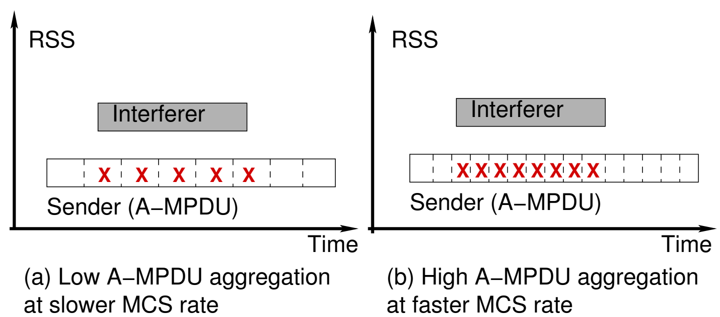

Figure 3.8: Increased MCS rate results in a higher A-MPDU aggregation per unit time.

of multiple stream transmissions much more than single stream transmissions. In other words, depending on the operating environment the choice of stream mode matters and where interference is significant, it might be more advantageous to use single stream transmission.

3.5 Hidden Terminal Interference

With the proliferation of private WiFi access points, hidden terminals are increasingly common [7, 29] and so we also investigated the hidden terminal scenario for 802.11n MIMO rate adaptation. We used the same experimental setup as that in Chapter 3, but we added an additional sender at location Q and a common receiver at location 4 (see Figure 3.1). A hidden terminal scenario is created because the sender at location Q (which we refer to as the *HT interferer*) is not within the carrier sensing range of P. In this experimental setup, the test link suffers from degradation due to Physical Layer Capture (PLC) at the common receiver if it has comparable or lower signal strength [25]. To emulate this scenario, we set the transmission power for the two links accordingly.

When performing rate adaptation, the typical response when losses are detected is to reduce the data rate. However, we found that with losses arising from hidden terminal interference, it is often possible to improve the sender's goodput in the presence of bursty losses. This is illustrated in Figure 3.7, where the sender was set at 40 MHz CB and Short GI in single stream mode, while the HT interferer operated at a constant MCS 13 with 20 MHz CB and Long GI. We note that the loss is significant even at low MCS settings. Furthermore, as we increase the data rate from MCS 0 to MCS 4, the goodput increases, even though the loss rate also increases. This observation was true for all other setting combinations and so we omit presenting the results because of space constraints. We investigate this rather curious phenomenon in detail and call it the *Baazigar Phenomenon*.

Baazigar Phenomenon. Bursty losses are known to be common when there is hidden terminal interference [8]. Consider Figure 3.8 which shows the A-MPDU for a range of MCS rates at the sender and a constant rate of the HT interferer. For the same amount of A-MPDU air-time, more MPDUs can be packed together at a higher MCS rate (Figure 3.8(b)). While this leads to more MPDUs being lost for the same amount of interference, more MPDUs are also being delivered per unit time, which effectively increases the goodput. This *Baazigar Phenomenon* suggests that if we are able to detect that a link is suffering from bursty losses due to hidden terminal interference, it is sometimes possible to increase the throughput by increasing the data rate in spite of higher loss rates, and that RA schemes that reduce rate in response to loss, either directly (RAMAS [26]) or indirectly (MiRA [28], WRA [27]) will miss out on the opportunity to maximize goodput in case of bursty losses due to hidden terminal interference. While high loss rates might cause significant degradation in TCP-based flows, most of these losses in the A-MPDU will be retransmitted by the driver with feedback from the A-MPDU block-ACK. Note that such a phenomenon is possible only with an A-MPDU (i.e. with FA enabled) as the frames in an A-MPDU that are not *hit* can be still recovered.

3.6 Understanding Receiver Sensitivity

It is known that for single stream 802.11 transmissions, there is a relatively steep minimum receive signal strength (RSS) threshold below which frames cannot be decoded, and above which loss rates rapidly fall to zero [14]. Also, for a given MCS rate, the use of multiple streams will result in weaker received signals at the receiver [17]. What is not known is whether such a threshold exists for multiple stream transmissions and how this threshold

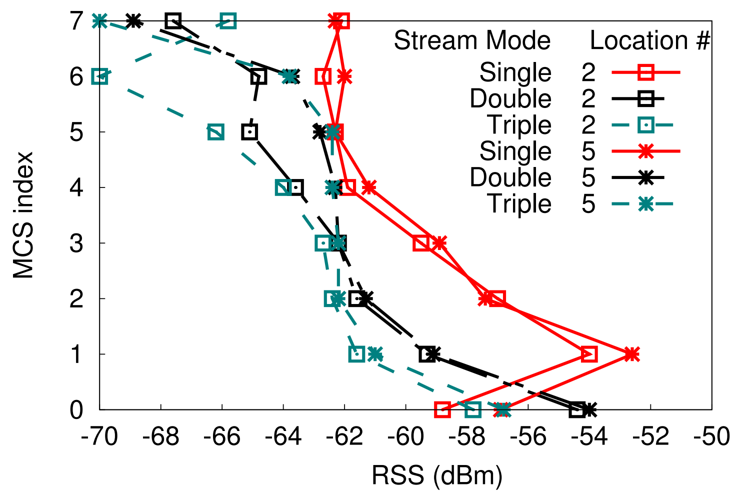

Figure 3.9: MCS index of rates in different stream modes and locations against RSS.

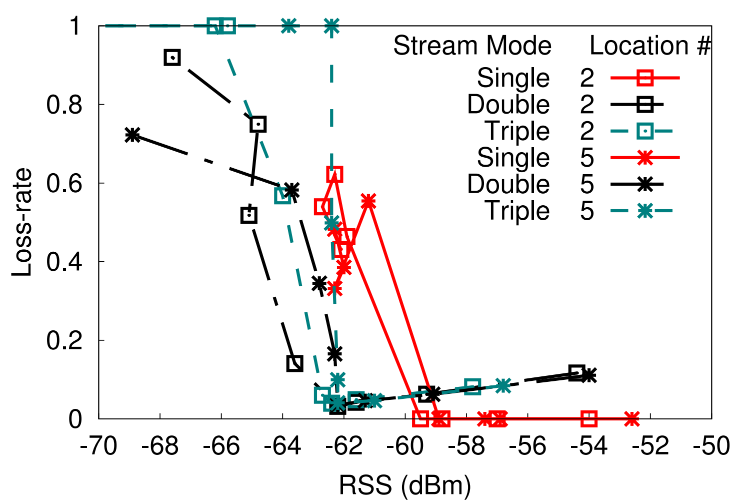

Figure 3.10: Receiver RSS Sensitivity with 40 MHz CB and SGI for MCS rates in different stream modes and locations.

would vary.

We investigated this issue by plotting in Figure 3.10 the loss rate versus RSS for all MCS rates in the three stream modes, for links between the AP and the client at locations 2 and 5. We observed that for all stream modes with 40 MHz CB and SGI, the loss rate increased rapidly to 1 when RSS falls below -62 dBm. When we repeated the same measurements for different combinations of CB and GI, we found that the threshold was lower (-65 dBm) for 20 MHz CB and that it seemed independent of GI. When we repeated these measurements with different hardware, we found that the thresholds were different too. Table 3.2 summarizes the threshold values at different CB settings for the two chipsets that we tested. What we found was that multiple stream transmissions also have a sharp RSS threshold similar to single stream transmissions, but it was not easy to predict the threshold.

Table 3.2: Receiver Sensitivity for different chipsets.

Chipset	**CB**	**Receiver Sensitivity**
Atheros AR9580	20 MHz	-65 dBm
	40 MHz	-62 dBm
Atheros AR5BXB72	20 MHz	-81 dBm
	40 MHz	-78 dBm

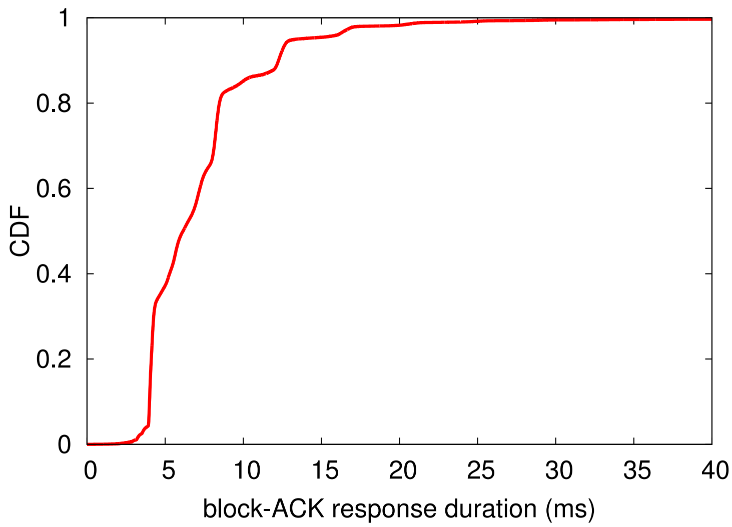

Figure 3.11: CDF of block-ACK response duration across the entire measurement study.

When an RA scheme probes a new MCS setting, it will need to use packet losses to decide that a new MCS setting is bad. Packet losses cannot be detected directly but are deduced with timeouts. Existing RA schemes [4, 26, 27, 28] all use a fixed timeout to detect this situation of no block-ACK. In Figure 3.11, we plot the CDF of the block-ACK response time (which is the time between when an A-MPDU is sent and that when the corresponding block-ACK is received by the sender) for all the experiments in our entire measurement study (~20 billion A-MPDUs). We found that the block-ACK response duration is no longer than 30 ms, which suggests that the timeout values adopted by existing schemes (typically in the range of hundreds of ms) are too large.

Chapter 4

Holistic Rate Adaptation (HiRA)

We developed a new MIMO RA scheme, which we call Holistic Rate Adaptation (HiRA). The design principles behind HiRA are built upon the following insights from our measurement study described in Chapter 3:

1. We can fix FA to 'enabled', CB to 40 MHz and GI to SGI. This significantly reduces the search space for probing to just 24 bitrates, with 8 MCS rates distributed equally across three different stream modes.

2. We employ a goodput-based probing strategy instead of traditional loss-based probing (like RAMAS [26]). Goodput-based probing provides us with a more direct indication of the performance gain achieved by a rate compared to a loss-based probing. Also, a goodput-based strategy would allow us to handle the *Baazigar Phenomenon* by increasing the rate based upon measured goodput when there is hidden terminal interference.

3. We need to detect and react to hidden terminal losses somewhat differently from losses due to signal degradation, because when there is hidden terminal interference, it might be possible to improve goodput either by increasing the MCS rate, or by enabling RTS/CTS.

4. The block-ACK timeout employed by existing RA schemes is too conservative (hundreds of ms) whereas in practice the block-ACK response duration is no more than 30 ms. Reducing the block-ACK timeout to 30 ms can help RA schemes to react faster to the receiver sensitivity losses.

HiRA is implemented as two modules: (i) the *Probing Module* performs inter-stream and intra-stream *goodput-based* rate probing at the 40 MHz CB and SGI setting combination

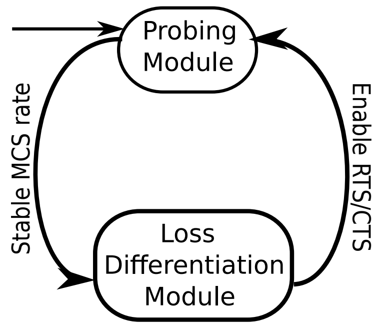

Figure 4.1: HiRA scheme state machine

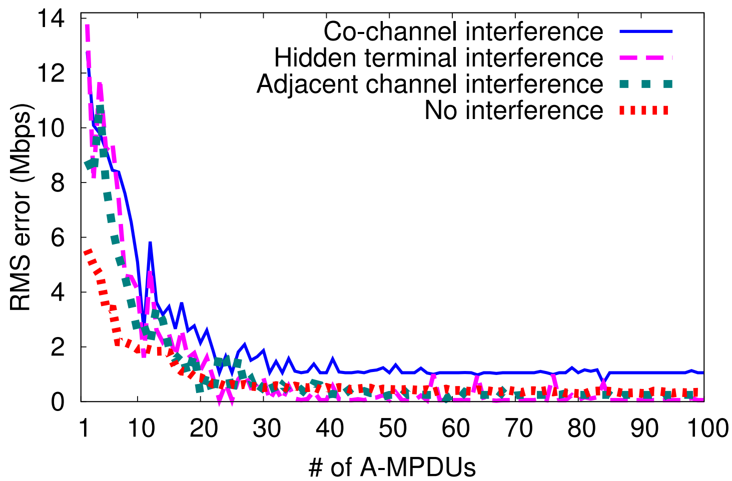

Figure 4.2: RMS error in goodput estimation by considering different # of A-MPDUs while estimation.

and reacts conservatively to loss; and (ii) the *Loss Differentiation Module* attempts to identify hidden terminal interference and helps to decide whether to enable RTS/CTS as shown in Figure 4.1.

4.1 Estimating Goodput

Before we describe how HiRA is implemented, we briefly discuss the challenges of *goodput-based probing*. We obtain the instantaneous goodput G_r of rate *r* at time *t* by:

$$G_r(t) = \frac{[\delta_{Seq} + \delta_{ACKs}] \times MPDU\_size}{TX_{\text{A-MPDU}}} \tag{4.1}$$

where δ_{Seq} is the difference between the sequence numbers and it gives the amount by which the block-ACK window (BAW) has slided since the last block-ACK. δ_{ACKs} is the difference between the number of acknowledged frames and it gives the additional frames that are delivered since the last block-ACK, apart from the slided window. $MPDU\_size$ is the size of a

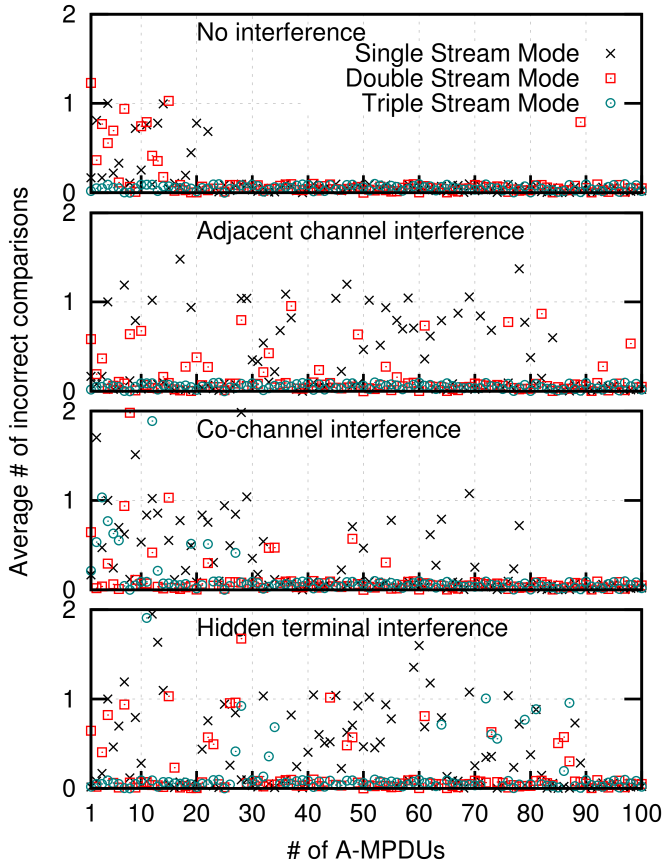

Figure 4.3: Average # of comparison errors within each stream mode due to different # of A-MPDU based goodput estimation under various interference scenarios.

MPDU frame, and TX_{A-MPDU} is the total air-time for the delivery of an A-MPDU and that basically includes the corresponding block-ACK response duration, other MAC layer overheads like DIFS (34µs), SIFS (16µs), PLCP header (20µs) etc.

The key drawback of goodput estimation is that it can take quite a lot of data packets (and time) to obtain an accurate estimate. In fact, in Figure 4.2 we plot the root-mean-square (RMS) error of estimated goodput against the number of A-MPDUs used for estimation. We see that it might take up to 40 A-MPDUs to achieve an accuracy close to the actual goodput. What we realized however was that there was no need for us to achieve absolute accuracy in the goodput estimates. For RA, an algorithm works well as long as the error is systematic and we are able to compare accurately between two MCS settings in relative terms, even if there is some error in the estimates. In Figure 4.3, we plot the average comparison errors w.r.t. the ground truth when we used different numbers of A-MPDUs to obtain the goodput estimates. The error is measured as the number of incorrect comparisons when all MCS rates in a stream mode are compared pair-wise for their relative goodputs. We can see that the average errors are generally small (i.e. less than 1) and even if we used 40 A-MPDUs,

Algorithm 1 HiRA Rate Adaptation

```
▷ SM: current stream mode
▷ rate: current rate
procedure PROBINGMODULE()
    SM, rate ← init()                                    ▷ Initialization phase
    for each A-MPDU sent do                              ▷ Maintenance phase
        if block-ACK timeout then
            intraModeProbing(down)
        else                                             ▷ block-ACK was received
            if G_r(t) ≥ \overline{G_r}(t) + \overline{σ_r}(t) then
                intraModeProbing(up)
                ▷ Link quality improved
            else if G_r(t) ≤ \overline{G_r}(t) − \overline{σ_r}(t) then
                intraModeProbing(down)
                ▷ Link quality degraded
            else
                SM, rate ← SM, rate
                ▷ If no change in the link quality for w block-ACKs,
                ▷ then call the Loss Differentiation Module

procedure INTRAMODEPROBING(direction)
    if direction == up then
        if rate shows highest goodput in SM then
            interModeProbing(up)
        else
            rate ← rate + 1
    else                                                 ▷ direction is down
        if rate is the lowest rate in SM then
            interModeProbing(down)
        else
            rate ← rate − 1

procedure INTERMODEPROBING(direction)
    if direction == up then
        SM ← SM + 1
        rate ← higherTheoretical(rate, SM)
    else                                                 ▷ direction is down
        SM ← SM − 1
        rate ← higherTheoretical(rate, SM)
```

there are still occasional errors in the comparisons. Therefore, we decided to use just a single A-MPDU to obtain goodput comparison estimate between any two MCS rates. It is surprising that this can work, since we would expect a single A-MPDU to be vulnerable to significant channel variations, especially when there is contention for channel access using CSMA.

4.2 Probing Module

There is a trade-off between reactivity and stability. On one hand, we want an RA algorithm that reacts quickly to changes in the operating environment; on the other hand, rapid oscillations will likely cause degradation in performance.

Even after reducing the search space to 24 MCS rates, probing all the rates is not feasible because it takes too long and will reduce network utilization. Furthermore, the variation of goodput for different rates across different stream modes is not easily predictable and we need to do a combination of probing within the same stream mode and probing across stream modes.

To converge to the optimal MCS rate rapidly and to avoid probing suboptimal rates, when we probe across stream modes, we always probe the minimal rate in the next higher stream mode whose theoretical goodput value is more than the actual goodput obtained by the current. The rationale is to avoid probing rates that would definitely not offer higher than the current goodput.

We implement probing in two phases – `init` phase and `maintenance` phase. The `init` phase attempts to rapidly determine the MCS rate that achieves close to maximum goodput across all three stream modes. Thereafter, the `maintenance` phase will periodically adapt to changes in the network and incrementally maintain a rate that is close to the maximum goodput across all stream modes. The procedure is described in Algorithm 1.

4.2.1 Initialization Phase

In the initialization phase (shown as *init* procedure in Algorithm 1), we start probing with the second lowest rate (MCS 1), single stream, 40 MHz CB and SGI setting combination. If the instantaneous goodput of the current rate is more than the instantaneous goodput of the previous rate, *init* procedure probes the next faster rate in the same stream mode (recall that the loss is monotonic for the rates within the same stream mode [27]). This incremental probing continues until it comes across a rate whose instantaneous goodput is less than the instantaneous goodput of the previous rate. Once this happens, the probing jumps to a rate in the double stream mode whose theoretical goodput [3] value is immediately greater than the current goodput. The rationale behind this jump is that, we need to find out whether there is any rate in the upper stream that offers better goodput than the current one and there is no point in jumping to a rate whose theoretical goodput value is itself less than the current goodput. The *init* procedure then probes the next faster rates within double stream mode and subsequently also jumps and probes the faster rates in the triple stream mode. Finally it settles down onto a rate that shows maximum instantaneous goodput across all three stream modes corresponding to 40 MHz CB and SGI. At this point, the initialization phase (*init* procedure) ends and we move in to the maintenance phase.

4.2.2 Maintenance Phase

One observation we made was that once an RA scheme finds the optimal rate, the next higher rate can often result in significant loss. While periodic probing of neighboring rates at fixed intervals [27] is a common strategy to detect and adapt to network changes, we found that it is often inefficient to probe the neighboring suboptimal rates. Instead HiRA chooses to detect the network changes while staying at its current rate. The key insight here is that when the network conditions change, the change is also reflected in the performance of the current rate. More so, depending on the nature of the change, it is possible to predict whether the next higher or next lower rate should be probed. HiRA achieves this by maintaining a historical expected goodput for the chosen rate and monitoring if there is any significant positive or negative change in the instantaneous goodput.

In particular, the maintenance phase maintains the history of the performance of rate *r* at instant *t* using an exponentially weighted moving average (EWMA) of the instantaneous goodput ($\overline{G_r}(t)$) and its average absolute deviation ($\overline{\sigma_r}(t)$):

$$\overline{G_r}(t) = (1-\alpha).\overline{G_r}(t-1) + \alpha.G_r(t) \tag{4.2}$$

$$\overline{\sigma_r}(t) = (1-\beta).\overline{\sigma_r}(t-1) + \beta.|G_r(t) - \overline{G_r}(t)| \tag{4.3}$$

$\overline{G_r}(t)$ tracks the “historical goodness” of the rate w.r.t. goodput and $\overline{\sigma_r}(t)$ tracks the historical volatility of it. In our implementation, we set α to 0.25 and β to 0.125.

If the following condition was true, then we assume that the channel conditions are stable and we maintain the rate r:

$$\overline{G_r}(t) - \overline{\sigma_r}(t) < G_r(t) < \overline{G_r}(t) + \overline{\sigma_r}(t) \tag{4.4}$$

On the other hand, if one of the following conditions was to arise, then we will start probing for a new rate:

$$G_r(t) \geq \overline{G_r}(t) + \overline{\sigma_r}(t) \tag{4.5}$$

$$G_r(t) \leq \overline{G_r}(t) - \overline{\sigma_r}(t) \tag{4.6}$$

Clearly, if the instantaneous goodput increases, it implies that the link quality has im-

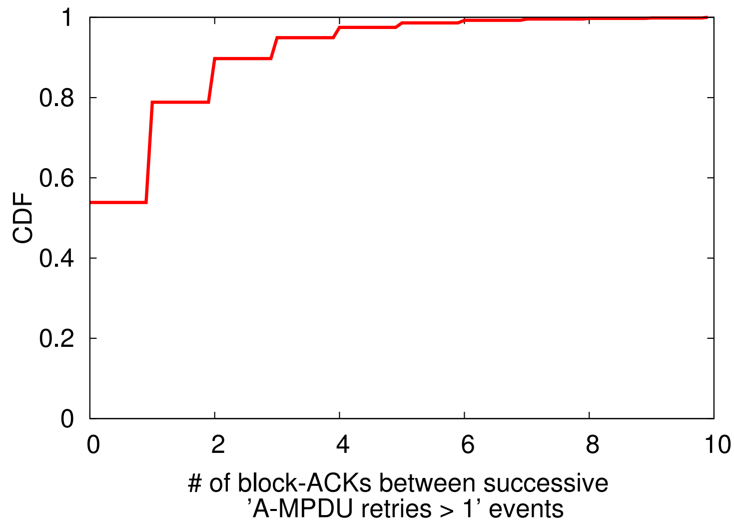

Figure 4.4: CDF of the # of block-ACKs in between successive 'A-MPDU retries > 1' events under hidden terminal interference.

proved and thus HiRA would probe the next higher rates to further improve the goodput. On the other hand, if the instantaneous goodput decreases, it implies that the link quality has degraded thus HiRA would probe the next lower rates which are relatively more robust. In deciding on the next rate, HiRA would probe both the next rate (whether higher or lower) in the current stream mode, as well as the MCS rate in the adjacent stream mode whose theoretical goodput [3] value is greater than the current goodput.

If the same MCS rate is chosen for a window of consecutive A-MPDU transmissions, we conclude that the link quality is relatively stable and that the probing has been stabilized. This is when we activate *loss differentiation*. We enable loss differentiation only when the probing stabilizes because changing the rate while performing a loss differentiation would adversely affect the accuracy of loss differentiation [8]. We found from practical experimentation that a window of 10 consecutive A-MPDUs was sufficiently large to allow us to infer that the probing phase has stabilized, without being too large that *loss differentiation* is hardly ever activated.

4.3 Loss Differentiation Module

The loss differentiation module has two functions: (i) determine whether there is hidden terminal interference; and (ii) if so, whether to increase MCS rate, or to enable RTS/CTS.

When a packet collision occurs under CSMA, the whole A-MPDU is typically lost and the sender will retransmit the whole A-MPDU at most once. This is because the CSMA backoff algorithm will kick in after the collision is detected. If multiple collisions continue to occur, there is too much contention and RTS/CTS will be required. Thus, an A-MPDU retransmission value of at most 1 suggests that the loss was due to CSMA collision. On the other hand, when loss happens because of hidden terminal interference, we observed that A-MPDUs retransmissions generally happen more than once. In Figure 4.4, we plot the CDF of number of block-ACKs (or A-MPDUs) in between two A-MPDUs that were retransmitted more than once in the presence of hidden terminal collisions. These results suggest that we can infer hidden terminal collisions with high probability if at least one A-MPDU is retransmitted more than once in an observation window of 5 block-ACKs. This heuristic is clearly not 100% accurate, but we have found that it works well in practice. Replacing this heuristic with a more sophisticated and accurate loss differentiation scheme remains as future work.

Whither RTS/CTS. From Figure 4.4, we noted that more than 50% of A-MPDUs lost under hidden terminal interference incur more than 1 A-MPDU retransmission. Enabling RTS/CTS would be one strategy to reduce these retransmissions. However, blindly enabling RTS/CTS may not be a good strategy as 802.11n MIMO rates are very high and the RTS/CTS overhead can be prohibitively expensive. We therefore enable RTS/CTS only if the transmission time of an A-MPDU at current rate *r* should be greater than the signalling overhead of RTS/CTS frames. This condition is captured by the following equation:

$$\frac{\text{A-MPDU\_aggr\_level}}{r} \geq (RTS\_TX_t - CTS\_RX_t) \tag{4.7}$$

where A-MPDU_aggr_level is the total size of a given A-MPDU, RTS_TX_t is the time when the RTS frame is sent out by the sender and CTS_RX_t is the reception time of the corresponding CTS frame.

Chapter 5

Performance Evaluation

We implemented HiRA using the `ath9k` driver and `Click` [21] on our custom built 802.11n AP platform. We compared HiRA's performance to existing state-of-the-art RA schemes using the testbed described in Chapter 3. We used driver-level implementations of Minstrel_HT, RAMAS and SampleLite+ schemes. Because we were not able to obtain the source code from the authors of [28], we also implemented the MiRA scheme using `ath9k` driver and `Click`. We refer to our implementation as MiRA*, which differs from MiRA in one regard: MiRA* does not use 4 different timers to periodically probe 3 nearby rates from the current rate. Instead, MiRA* uses a single timer at the current rate to detect expiry of the adaptive probing interval and then probes all 3 nearby rates one at a time. We were careful to ensure that this modification would not affect the performance by ensuring that all the desired 3 nearby rates were probed, although it might take 3 extra A-MPDUs each time when the adaptive timer expires (which is very rare). We could not implement closed-loop RA schemes like ARAMIS [15] because CSI was not available for the Atheros chipset. We used the *IPerf* [2] tool for generating UDP traffic.

5.1 Performance in static scenarios

We first evaluate the performance of RA schemes under the no interference, adjacent channel interference and co-channel interference scenarios. For these evaluations, the AP runs each RA scheme for 5 minutes and clients receive the traffic when they are kept at a stationary locations as shown in Figure 3.1. For robustness, we repeated experiments several times in a round robin way at each location and plot the max, mean and min goodputs in Figure 5.1.

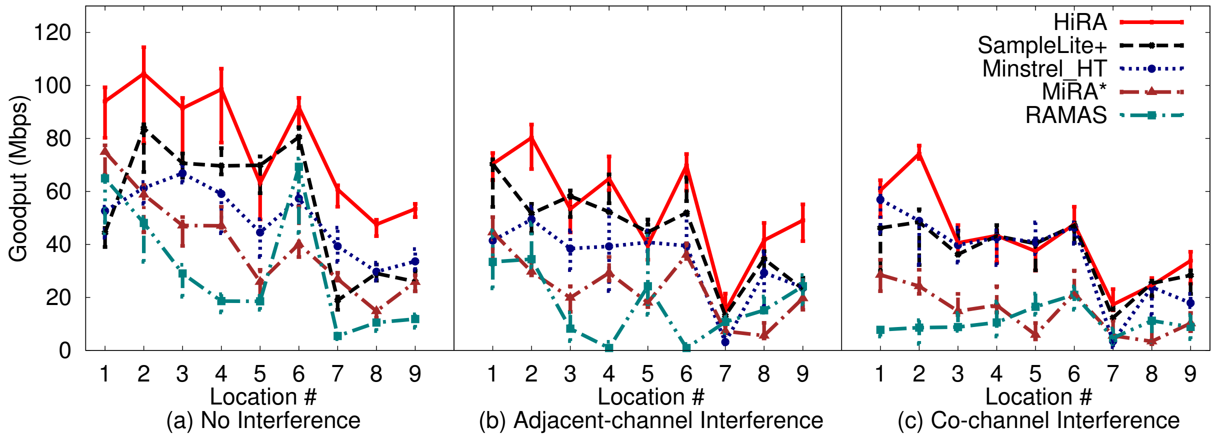

Figure 5.1: Performance of RA schemes under different operating scenarios.

No Interference. We see that HiRA outperforms all other RA schemes when there is no interference. HiRA achieves goodput that is at least 21% better in goodput than the next best performing RA scheme (Minstrel_HT). We note that the improvement over existing schemes depends on the spatial properties of the locations. The locations which are near from the AP and which had moderate or good signal strengths (locations 1, 2, 3, 4, and 6) were likely to have benefited from wider CB of 40 MHz, SGI and stable probing, whereas occluded locations (locations 7, 8 and 9) were likely to have benefited from the use of multiple streams. RAMAS performed well at location 6 because all three stream modes were equally supported at that location (evident from Figure 3.6a) which is a favorable situation for RAMAS as it tends to aggressively probe MCS rates with no regard to the current stream mode.

Interference Scenarios. When there is adjacent channel interference, HiRA still outperformed all other RA schemes at every location by at least 26% than the next best performing RA scheme (Minstrel_HT). When there is co-channel interference, HiRA, Minstrel_HT and SampleLite+ all seemed to have similar performance for most locations, except for locations 2, where HiRA was able to perform noticeably better. We are currently still investigating why there seems to be a common upper bound for the goodput for HiRA, Minstrel_HT and SampleLite+.

5.2 Handling hidden terminals

One of our key insights is to deal specially with hidden terminal interference. To evaluate our effectiveness in handling hidden terminal interference, we adopted the setup from the measurement study as in Figure 3.1. The interference link was operated at different MCS rates with 20MHz CB over channel 149. In Figure 5.2, we plot the goodput achieved versus the

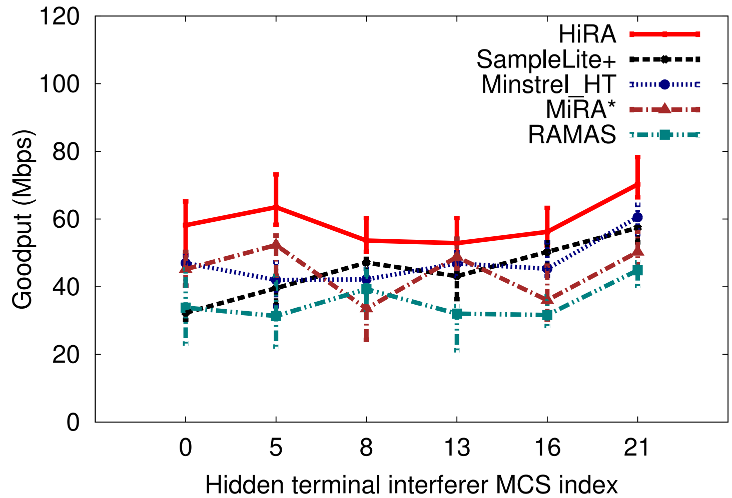

Figure 5.2: Performance of different RA schemes in hidden terminal scenario.

MCS rates of interferer and see that, as expected, HiRA outperformed all other RA schemes for interfering rate by at least 18% than the next best performing scheme. Note that MCS rates 0 and 5 are for single stream, rates 8 and 13 are for double streams and rates 16 and 21 are for triple streams.

Careful analysis showed that the performance gain of HiRA was due to the correct detection of hidden terminal scenario, whereby HiRA would increase the data rates aggressively to obtain higher A-MPDU aggregation during bursty losses, when interferer operated on the higher rates MCS 5 and 13. However, when interferer rate was slow (i.e. for MCS 0, 8, 16) and had a low A-MPDU aggregation, HiRA tended to enable RTS/CTS when signalling overhead was less than the A-MPDU transmission duration to minimize the retransmissions. Subsequently MiRA* also performed well in this case because of loss-differentiation in their algorithm and by enabling RTS/CTS. However MiRA* uses two variables while differentiating loss patterns that are dependent upon the percentage of the loss. Hence whenever rate of the test link rate was lossy even due to path loss and showed scattered loss, it fails to detect the hidden terminal scenario.

5.3 Mobility scenario

Next, we evaluated the performance of HiRA in a mobility scenario. For this evaluation, we carried a client at an approximate constant speed of 1 m/s from location P and visited locations 1 to 6 in sequence before returning to location P. For robustness we repeated the experiment several times. Figure 5.3 shows RSS of the link while client was mobile. We

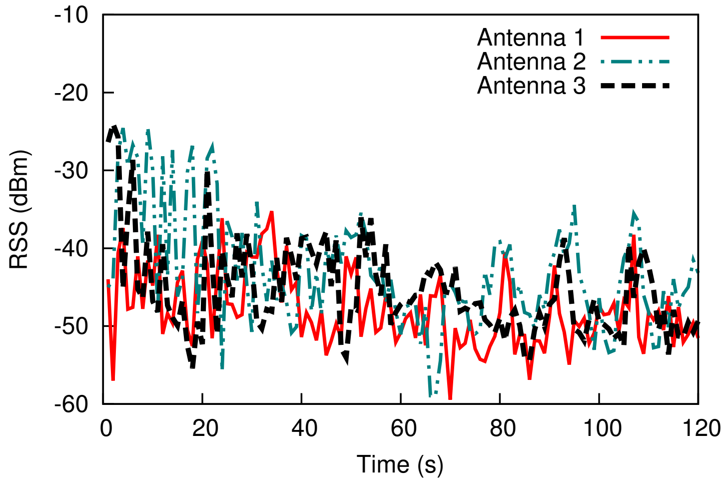

Figure 5.3: RSS of a link during mobility scenario

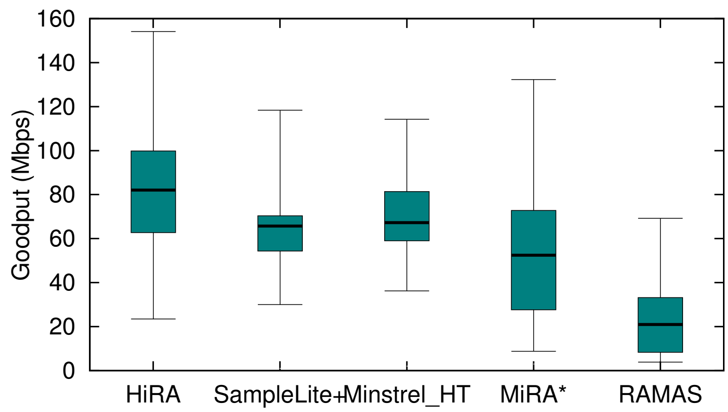

Figure 5.4: Performance of different RA schemes under mobility.

plot the measured per-second goodput for these runs in Figure 5.4 with bars indicating the maximum, upper quartile, mean, lower quartile and minimum values. We see that HiRA outperforms all other RA schemes by up to 11% than next best scheme (Minstrel_HT). We believe that HiRA's good performance was mainly due to its ability to detect channel changes promptly and react quickly.

5.4 Adaptation to changing channel conditions

To investigate how quickly HiRA reacts to changes in channel quality, we performed the following experiment: we set up a test link and an interfering link between two locations operating at MCS 5 with 20MHz CB. The signal strength of the interfering link was strong at around -45 dBm and it was in the CSMA range of the test link. For each RA scheme, we started to send UDP traffic on the test link and after 1 minute, we sent UDP traffic down

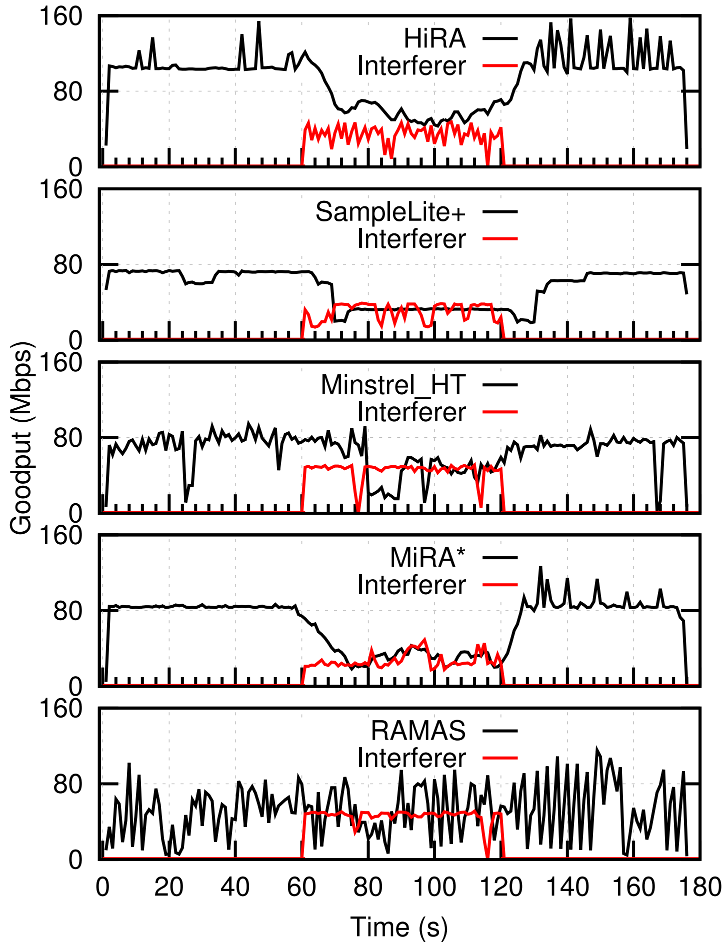

Figure 5.5: Adaptation to the changing channel conditions for different RA schemes.

the interfering link for one minute, which would result in co-channel interference. Once after the traffic on the interfering link is stopped, we continued sending traffic over the test link for another 1 minute to see how RA schemes will react when channel becomes available again. In Figure 5.5, we can see that HiRA, MiRA* and SampleLite+ reacted promptly once the interference starts and when it ends. Minstrel_HT took several seconds to respond. RAMAS did not even seem to respond explicitly to the channel change. It seems however that HiRA is somewhat more aggressive than Minstrel_HT, RAMAS and SampleLite, which all allowed the interfering link a reasonably fair share of the bandwidth. MiRA* seemed to react poorly to co-channel interference.

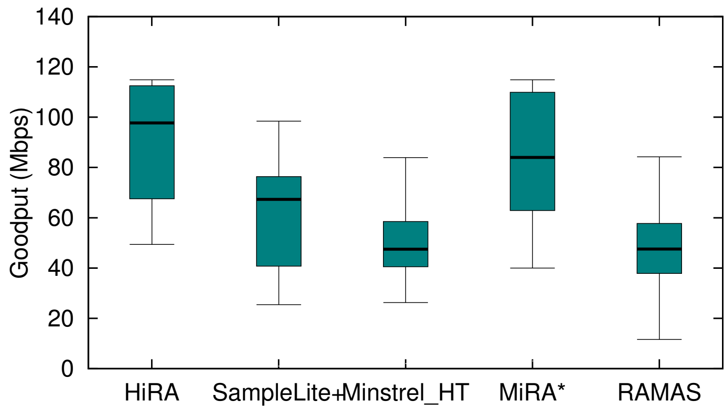

Figure 5.6: Performance of different RA schemes ‘in the wild’.

5.5 Performance ‘in the wild’

Finally, we evaluated the performance of HiRA under realistic scenarios, where various sources of interference coexist in a complex manner. For these experiments, we kept AP and client in an office environment during peak office hours (11am - 5pm) and used a busy channel 1 in 2.4 GHz frequency band when 12 other APs were simultaneously on the same channel. We calculated the per-second goodput and is shown in Figure 5.6 with max, upper quartile, mean, lower quartile and min values. We find that HiRA achieved better goodput than all other schemes by up to 13% more than the next best scheme (MiRA*). Our results were surprising because given our earlier results, we had expected either Minstrel_HT or SampleLite+ to be the best performing algorithm after HiRA. This highlights that there might still be a gap between a controlled evaluation of RA schemes and actual practical performance.

Chapter 6

Conclusion and Future Work

HiRA was not designed to be an optimal MIMO RA scheme. HiRA was developed by putting together some interesting insights from our measurement study of existing 802.11n RA schemes. What we have demonstrated is that there are still gaps in existing RA schemes that can be plugged to improve further RA performance. Neither do we claim that fixing FA to ‘enabled’, CB to 40 MHz and GI to 400 ns is necessarily the best policy under all circumstances. What our work suggests, however, is that this is generally good policy and it remains as future work to investigate the conditions under which these fixed settings would not be optimal for HiRA.

6.1 Future Work

Although HiRA is designed for 802.11n, it also be directly applicable for 802.11ac systems. However, there is still a huge scope to improve HiRA for a better RA scheme for 802.11ac as new features like MU-MIMO, precoding and transmit beamforming etc. can be leveraged. We believe that a new modern RA scheme for 802.11ac can be based upon the principles of HiRA. In the short term, we would like to incorporate our enhancements directly into the `ath9k` driver for 802.11n. Moreover, we would do more in-depth analysis and research on the following aspects of our work.

- Study the impact of all the features in combinations and verify that they have independent impact on the performance gain.
- Evaluate HiRA at different locations and working environments to validate whether the findings about the features are independent of the location.

- Evaluate HiRA over TCP and address the potential performance problems that may arise due to packet re-ordering and retransmissions etc.
- Evaluate HiRA with many users sharing the channel simultaneously and verify if the block-ACK response duration of 30 ms is still applicable.

6.2 Open Issues

We also believe that there are few open problems that are worth addressing while designing a better and faster RA scheme for emerging 802.11 networks.

- Identifying a stream mode that will be favourable at a particular location for a longer period of time. Fixing a stream mode can reduce the sample space exponentially.
- Reducing false positives while identifying link scenarios by designing a better loss-differentiation scheme.
- Investigating HiRA to maximize the coordinated performance of all the nearby users by maintaining a fairness among them.
- Fine tuning parameters α and β to obtain a better trade-off between history (reaction time) and stability (smoothness) of the algorithm.

In future, we look forward to continue working on the same problem to address the above mentioned issues in depth and come up with a more robust and an optimal RA scheme for 802.11ac MIMO systems.

Bibliography

[1] ath9k Linux wireless driver. http://wireless.kernel.org/en/users/Drivers/ath9k.

[2] iPerf - The network bandwidth measurement tool. https://iperf.fr/.

[3] MCS index to rate mapping. http://mcsindex.com/.

[4] Minstrel_HT rate control algorithm for mac80211. https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel.

[5] ONOE MadWiFi. http://sourceforge.net/projects/madwifi.

[6] WLAN Standards 802.11a/b/g Drafts. http://standards.ieee.org/about/get/802/802.11.html.

[7] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in Chaotic Wireless Deployments. In *Proceedings of MobiCom*, 2005.

[8] R. Anwar, K. Nishat, M. Ali, Z. Akhtar, H. Niaz, and I. Qazi. Loss differentiation: Moving onto high-speed wireless LANs. In *Proceedings of INFOCOM*, 2014.

[9] S. Biaz and S. Wu. Loss Differentiated Rate Adaptation in Wireless Networks. In *Proceedings of WCNC*, 2008.

[10] J. C. Bicket. Bit-rate Selection in Wireless Networks. Master's thesis, MIT, 2005.

[11] J. Chen, H. Li, F. Zhang, and J. Wu. MIMO mode switching scheme for rate adaptation in 802.11 n wireless networks. In *Proceedings of GLOBECOM*, 2011.

[12] R. Combes, A. Proutiere, D. Yun, J. Ok, and Y. Yi. Optimal rate sampling in 802.11 systems. In *Proceedings of INFOCOM*, 2014.

[13] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets. CSI-SF: Estimating wireless channel state using CSI sampling and fusion. In *Proceedings of INFOCOM*, 2012.

[14] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth. The Impact of Channel Bonding on 802.11n Network Management. In *Proceedings of CoNEXT*, 2011.

[15] 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 *Proceedings of SECON*, 2013.

[16] 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 *Proceedings of SECON*, 2013.

[17] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. 802.11 with Multiple Antennas for Dummies. *SIGCOMM CCR*, 2010.

[18] G. Holland, N. Vaidya, and P. Bahl. A Rate-adaptive MAC Protocol for multi-Hop Wireless Networks. In *Proceedings of MobiCom*, 2001.

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

[20] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *Proceedings of INFOCOM*, 2006.

[21] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click Modular Router. *Transactions on Computer Systems*, 2000.

[22] L. Kriara and M. K. Marina. SampleLite: A Hybrid Approach to 802.11N Link Adaptation. *SIGCOMM CCR*, 2015.

[23] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 rate adaptation: A practical approach. In *Proceedings of MSWiM*, 2004.

[24] S. Lakshmanan, S. Sanadhya, and R. Sivakumar. On link rate adaptation in 802.11n WLANs. In *Proceedings of INFOCOM*, 2011.

[25] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. An experimental study on the capture effect in 802.11 a networks. In *Proceedings of WiNTECH*, 2007.

[26] D. Nguyen and J. Garcia-Luna-Aceves. A practical approach to rate adaptation for multi-antenna systems. In *Proceedings of ICNP*, 2011.

[27] 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*, 2013.

[28] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu. MIMO Rate Adaptation in 802.11N Wireless Networks. In *Proceedings of MobiCom*, 2010.

[29] W. Wang, Q. Wang, W. K. Leong, B. Leong, and Y. Li. Uncovering a hidden wireless menace: Interference from 802.11 x MAC acknowledgment frames. In *Proceedings of SECON*, 2014.

[30] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proceedings of MobiCom*, 2006.

[31] D. Xia, J. Hart, and Q. Fu. Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs. In *Proceedings of ICC*, 2013.

[32] X. Xie, X. Zhang, and K. Sundaresan. Adaptive Feedback Compression for MIMO Networks. In *Proceedings of MobiCom*, 2013.

[33] A. Zubow and R. Sombrutzki. Evaluation of a lowcost IEEE 802.11n MIMO testbed. *Humboldt-Univ., Berlin, Germany, Tech. Rep. SAR-TR-2011-12*, 2011.