MIMO Rate Adaptation in 802.11n Wireless Networks

Ioannis Pefkianakis[1], Yun Hu[2], Starsky H.Y. Wong[3], Hao Yang[4], Songwu Lu[1]
UCLA Computer Science[1], USTC Computer Science[2], IBM Research[3], Nokia Research[4]
{pefkian, slu}@cs.ucla.edu[1], geajy@mail.ustc.edu.cn[2],
hwong@us.ibm.com[3], hao.2.yang@nokia.com[4]

ABSTRACT

This paper studies MIMO based rate adaptation (RA) in 802.11n wireless networks. Our case study shows that existing RA algorithms offer much lower throughput than even a fixed-rate scheme. The fundamental problem is that, all such algorithms are MIMO oblivious; they do not consider the characteristics of diversity-oriented, single-stream mode and the spatial multiplexing driven, double-stream mode. We propose MiRA, a novel MIMO RA scheme that zigzags between intra- and inter-mode rate options. Our experiments show that MiRA consistently outperforms three representative RA algorithms, SampleRate, RRAA and Atheros MIMO RA, in static, mobility and collision settings.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

MIMO, Rate Adaptation, IEEE 802.11n

1. INTRODUCTION

Rate adaptation (RA) in the recent IEEE 802.11n systems dynamically adjusts the Modulation Coding Scheme based on the runtime channel quality. It is critical to network throughput, yet unspecified by the standard. What makes RA in the 802.11n setting different from the legacy 802.11a/b/g systems is its new Multiple-Input Multiple-Output (MIMO) technology. With multiple transmit and multiple receive antennas and channel bonding, the 802.11n standard allows for two operation modes. One is the diversity-oriented single-stream (SS) mode. The other is the spatial multiplexing driven, double-stream (DS) mode. Using both SS and DS, it thus offers a wide range of rate options, starting from 6.5Mbps and reaching 600Mbps at the maximum. MIMO RA is likely to be important in 802.11n systems due to the wider span and larger number of rate options.

In this paper, we identify issues and propose solutions for MIMO-based RA in 802.11n systems. Our work started from a case study of existing RA algorithms using prototypes and experiments on standard-compliant 802.11n AP platform. The case study asked a simple question: Can we simply apply these RA algorithms, which have been shown to work well for the legacy 802.11a/b/g networks, to the MIMO setting? To this end, we chose two representative RA algorithms: RRAA [2], and SampleRate [4]. We also evaluated another algorithm, Atheros MIMO RA [3], used in 802.11n Atheros chipsets. To our surprise, all three algorithms did not do well, offering 28% to 44% lower goodput, defined as effective throughput by excluding protocol overhead, than the best fixed-rate scheme. The fundamental problem is that, all such algorithms do not properly consider SS and DS modes, thus not exploiting the inherent MIMO characteristics, which exhibit very different loss patterns across SS and DS modes.

Our extensive experiments further discovered that there exhibits a non-negligible, non-monotonic relation between loss and rate in 802.11n MIMO scenarios, when considering all rate options and ignoring operation modes. As the rate value increases, loss does not monotonically grow with rates in different modes. A higher rate may consequently lead to lower loss and larger goodput. This is the reason why existing RA algorithms got stuck at lower rates that yield much smaller goodput. However, the good news is that within each SS/DS mode, the monotonic behavior between loss and rate still largely holds.

In this paper, we design, implement and evaluate MiRA, a new RA algorithm for 802.11n MIMO systems. MiRA uses a novel zigzag RA scheme, which opportunistically zigzags between intra-mode RA and inter-mode operations. When performing rate update, it first retains its current SS/DS mode but adapts the rate upward/downward. This intra-mode RA exploits the monotonicity between loss and rate in the same mode. When it cannot further improve goodput in its current mode, MiRA performs inter-mode RA by exploring the other DS/SS mode. MiRA further uses prioritized, adaptive probing to reduce the penalty incurred by probing bad rates and swiftly identify the best rate. Finally, MiRA exploits features of frame aggregation and BlockAck in 802.11n to detect collisions from channel errors.

Our experiments in both controlled testbed settings and

MobiCom'10, September 20–24, 2010, Chicago, Illinois, USA.

field trials confirm the performance gain of MiRA in all cases of fading, hidden terminal-induced collisions, and mobility. Our experiments show that MiRA outperforms all three RA algorithms with goodput gains up to 124.8% in static settings, 182.2% for mobile clients, and up to 1094% in intense interference scenarios. In the field trials, MiRA achieves goodput gains from 19.4% up to 67.4% over the three existing algorithms.

The rest of this paper is organized as follows. Section 2 introduces the background, and describes our experimental setting. Section 3 studies a simple case of applying existing RA algorithms in the 802.11n scenario, and Section 4 reports the findings on characteristics of SS and DS modes. Section 5 presents the design of MiRA, and Section 6 describes its implementation and evaluation. Section 7 reviews the related work. Section 8 concludes the paper.

2. BACKGROUND

This section introduces the 802.11n standard and describes our experimental setting.

2.1 IEEE 802.11n Standard

The 802.11n standard incorporates three new features to boost performance.

Multiple-input Multiple-output (MIMO) MIMO uses multiple transmit and receive antennas to achieve spatial multiplexing and spatial diversity. Spatial Multiplexing transmits independent and separately encoded, spatial streams, from each of the multiple transmit antennas. The current standard allows for maximum four spatial streams. Spatial diversity (often implemented via maximal ratio combining) transmits a single stream from each transmit antenna. It leverages the independent fading over multiple antenna links to enhance signal diversity.

Channel-bonding/40MHz operation This operation simultaneously uses two separate channels to transmit data, thus doubling the rate in principle. The legacy 802.11a/b/g systems use a single 20MHz channel, but 802.11n can operate in the 40MHz mode over two adjacent channels, one as the control and the other as the extension.

Frame aggregation Frame aggregation amortizes protocol overhead over multiple frames. It packs several data frames, called Mac Protocol Data Units (MPDUs), into an aggregate frame (called A-MPDU).

The current 802.11n standard-compliant systems implement most, but not all functionalities of the standard. They can operate on both 2.4GHz and 5GHz bands with both 20MHz and 40MHz modes. The number of supported antennas is up to three and the antenna configurations are 2x2, 2x3, and 3x3. Current systems support up to two spatial streams. 20MHz rates go up to 135Mbps while 40MHz rates go up to 300Mbps. SS rates are implemented using diversity, and DS rates are implemented using Spatial Multiplexing. The 802.11n standard also supports fast MCS feedback mechanism, but current systems do not implement this feature.

2.2 Experimental Setting

We next describe the experimental platform and setup for our MIMO rate adaptation study.

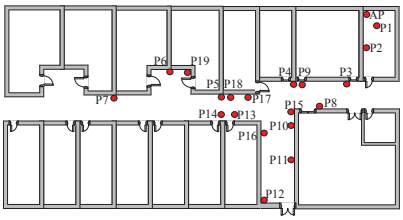


Figure 1: Experimental floorplan.

Experimental Platform We conduct all experiments on a programmable AP platform, which uses Atheros AR5416 2.4/5 GHz MAC/BB MIMO chipset and supports both SS and DS modes. It supports up to 130Mbps and 300Mbps data rates for 20MHz and 40MHz channel operations, respectively. The clients in our experiments are using either Broadcom or Marvell 802.11n chipsets. Frame aggregation and BlockAck (i.e., ACK for A-MPDU) are also available. Both AP and clients have 3 available antennas.

Our platform provides per-frame control functionalities, and we have implemented mutliple RA algorithms in the AP firmware. Upon receiving a BlockAck, the RA module gets feedback including the number of MPDUs in the transmitted A-MPDU (called as $nFrames$) and the number of MPDUs received with errors (called as $nBad$). If the entire A-MPDU is lost, the number of hardware retries (called as $retries$) is also available. We can then compute *Sub-Frame Error Rate* (SFER) as $SFER = \frac{nFrames \times retries + nBad}{(retries+1) \times nFrames}$. Per-antenna SNR information is available to the RA module as well.

Experimental Setup We conduct our experiments in both a campus setting and a RF chamber. Figure 1 shows the floorplan of the campus building we run the experiments. Spots P1 to P19 represent different locations where the clients are placed. The RF chamber offers an RF shielded room isolated from external RF noises and interferences. We run the same set of experiments with three different 802.11n clients: Buffalo WLI-CB-AG300NH 802.11a/b/g/n wireless adapter based on Marvell 802.11n chipset, Linksys WPC600N 802.11a/b/g/n and Airport Extreme wireless adapters using Broadcom chipset. The results presented in the paper are from Airport Extreme adapter, which supports up to 270Mbps rates.

3. A CASE STUDY

We started our work by examining how well the existing RA algorithms work in the 802.11n MIMO setting. The goal is to understand which factors in these RA schemes lead to their performance gain or loss and which MIMO characteristic is the root cause.

To illustrate our findings, we first present a case study with one static client located at P4 (see Figure 1). We studied three representative RA algorithms: RRAA [2], SampleRate [4], and Atheros MIMO RA [3]. RRAA and SampleRate have been shown to work well in the legacy 802.11a/b/g scenarios, and Atheros MIMO RA is a new algorithm used in 802.11n Atheros chipsets. We also conducted fixed-rate experiments at every 802.11n rate option.

Table 1 summarizes the results of these experiments. Unfortunately, all three RA algorithms perform worse than the best fixed-rate scheme, with 28% to 44% lower goodput. The goodput at the best fixed rate is $128.5Mbps$, while

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

Table 1: Rate distribution, Goodput and SFER of existing RA algorithms at P4.

Atheros RA gives 71.4$Mbps$, RRAA offers 85.4$Mbps$, and SampleRate gives 91.9$Mbps$. These results clearly indicate that the existing RA algorithms cannot be effectively applied in 802.11n networks.

It turns out that, all three RA algorithms were stuck at rates much lower than the best rate choice. Table 1 states that the goodput at 162DS is 128.5$Mbps$, while the goodput at 108SS, 108DS, 121.5SS and 135SS are only 96.5$Mbps$, 96.3$Mbps$, 74$Mbps$ and 36.6$Mbps$, respectively. Obviously, a good RA should transmit most of its frames at 162DS rather than at other rates. However, as illustrated in Table 1, the rate distribution of each RA, which provides the percentage of data frames transmitted at a given rate, shows the opposite results. SampleRate transmits 89% of frames at 108DS, RRAA transmits 53% and 47% at 121.5SS and 108SS. The Atheros MIMO RA is even worse, transmiting 51% at 108SS and 49% at 54SS, and not using 162DS at all.

We next examine what happens at rate 162DS and other rates. Our experiments, plotted in Figure 2(a), reveal that two factors play a critical role: non-negligible, non-monotonic relation between SFER and rate, and frame aggregation. Figure 2(a) shows that, SFER does not monotonically increase as the rate grows from 121.5 to 162 Mbps. The frame loss SFER is only 4.3% at 162DS, but is 54.6% at 135SS and 17.9% at 121.5SS. This finding in 802.11n MIMO settings is clearly different from that in the legacy 802.11a/b/g systems. Aggregation level is another factor that affects goodput. Figure 2(a) states that, the average aggregation level is 27 MPDUs at 162DS but is 15 MPDUs at 121.5SS. This (11.3MPDU) larger aggregation level also leads to goodput improvement as the amortized per-frame overhead is smaller. With both low SFER and high aggregation level, 162DS significantly outperforms other rates.

Once we discovered the two factors of non-monotonic SFER and frame aggregation level, we further look into why existing RA designs have difficulty in identifying and staying at the best rate that offers highest goodput. The RRAA algorithm [2] assumes that SFER monotonically increases with rate. Therefore, RRAA assumes that higher rates would yield higher losses when evaluating the rate 121.5SS. This is true for 135SS but not true for 162DS. As a result, it never probes 162DS that has smaller SFER and highest goodput. Atheros MIMO RA also assumes monotonicity in that all rates above the current rate R have no smaller SFER. When probing, it upper bounds the candidate rates for selection (maxRate) by probing one rate higher than the current best-goodput rate R. By analyzing actual packet traces, we observe that probing fails at 135SS and maxRate is set at 121.5SS for most transmissions. Consequently, Atheros MIMO RA transmits almost all of the frames at rates lower than 121.5Mbps. SampleRate [4] randomly samples diverse rates via probing, but suffers from stale statistics on the goodput and SFER at a rate as it updates statistics only by probing these rates. It consequently gets stuck at rates below 135Mbps as shown in Table 1. Moreover, the SampleRate implementation bounds sampling to at most 2 rates higher than the current transmission rate. It thus does not update stale statistics for rates greater than 135Mbps and transmits most data at 108DS[1].

4. UNDERSTANDING MIMO CHARACTERISTICS IN 802.11N SYSTEMS

The above case study shows that the fundamental reason for RA under-performance is due to inherent MIMO characteristics [11,15,16]. We now seek to conduct a more thorough study on such 802.11n characteristics.

4.1 SFER Non-Monotonicity in SS and DS

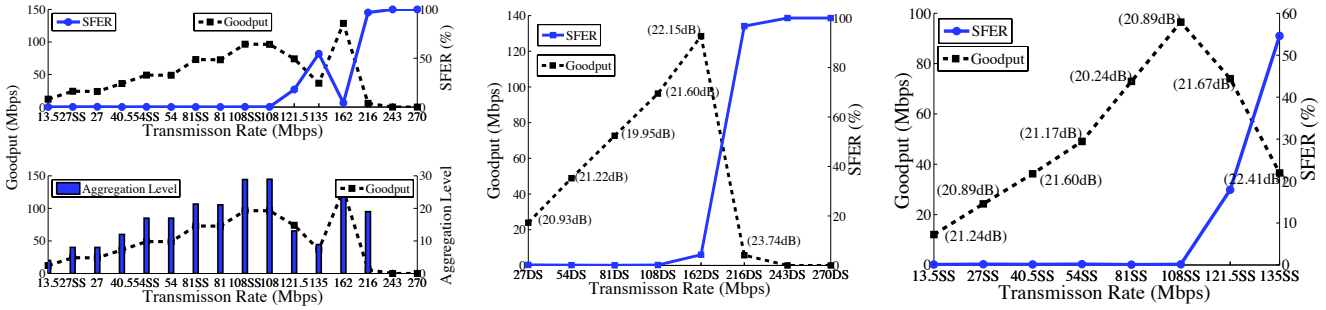
Our experimental results show that, different from the legacy 802.11a/b/g systems, *there exhibits a non-negligible, non-monotonic relation between the rate option and SFER in 802.11n MIMO settings* when considering all rates in both SS and DS modes. SFER does not monotonically increase when the transmission rate increases. The non-monotonicity appears more distinctive under two scenarios: (i) in the high-rate region (e.g., $\geq$ 121.5SS), and (ii) at same rates in different modes (e.g., 54SS and 54DS). Representative examples of these two cases are illustrated in Tables 2 and 3. Table 2 shows that the non-monotonicity in SFER is particularly severe between three adjacent cross-mode rates (i.e., 121.5SS, 135SS, 162DS). In four locations P3, P4, P8, and P10 (We only show a subset of results due to space constraints), SFER increases as the rate increases from 121.5SS to 135SS, but drops significantly as the rate further moves to 162DS. SFER drops 50.3% at P4 when switching from 135SS to 162DS. Similar results are also observed in the RF chamber, where 121.5SS and 135SS have up to 6.4% and 8.1% higher SFER than 162DS, respectively. Non-monotonicity also exhibits in the same-rate pairs. The SFER difference can be as large as 28.2% (location P10), as shown in Table 3. Note that this non-monotonic behavior is not caused by SNR variations. Both Table 2 and 3 show that the SNR values only exhibit minor differences at a given location.

The root cause for the behavior is that SS and DS are based on different communication approaches [11]. Thus it is unlikely that they will exhibit similar loss-rate relations by simply merging them together via the numerical value of the transmission rate.

In contrast, the monotonicity between SFER and rate still holds well in each single-mode case. Our extensive experiments show that, in either SS or DS mode, the SFER does monotonically increase as the rate grows (see Figures 2(c) and 2(b) for an illustrative example).

The non-monotonicity between SFER and rate has profound implications for 802.11n rate adaptation design. Many existing RA algorithms implicitly assume the monotonicity between SFER and rate. For example, one popular mechanism is to sequentially probe upward/downward the rates,

[1]We also tuned SampleRate by relaxing this sampling bound, but it still suffers from stale statistics and probing overhead in many cases, as we shall discuss in Section 6.3.



(a) SFER non-monotonicity in cross modes. (b) SFER monotonicity in DS mode. (c) SFER monotonicity in SS mode.

Figure 2: SFER vs. Transmission Rate in 802.11n MIMO settings (measured at location P4).

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

Table 2: SFER non-monotonicity w.r.t. rate in cross modes.

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

Table 3: SFER w.r.t. different cross-mode rate pairs.

and adjust the rate based on the probing result. Its underlying premise is that, the packet error rate goes higher as the rate increases, and there is no need to probe/use higher rate if the current one performs poorly. While this mechanism works reasonably well in the legacy system, it does not work in the dual-mode MIMO settings.

4.2 SS/DS Mode Selection

The above findings indicate that MIMO RA design should differentiate the two MIMO modes. The next issue is to identify possible conditions under which SS underperforms or outperforms DS. Several theoretical studies [11, 15, 16] have shed lights on it via examining the tradeoff between Diversity and Spatial Multiplexing gains. Our goal is to find the answer via experiments in the 802.11n setting.

The comparison between SS and DS mode in various locations is summarized in Tables 2 and 3. Our results show that, SNR can serve as a coarse-grained indicator to decide which mode is more likely to be the winner. In low-SNR regions (say, $< 13dB$ in our experimental settings), SS is more likely to outperform DS. In these low-SNR, far-away locations, SS is a winner over DS with 5% or more goodput

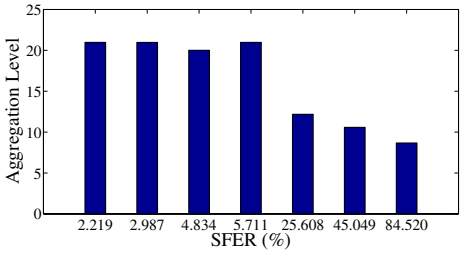


Figure 3: SFER vs. aggregation level.

gain in 85.7% of locations tested, while its goodput and loss are similar to DS in the remaining locations. The winning SS rates span the broad set of 13.5SS, 27SS, 40.5SS, 54SS, and 81SS. The average goodput gain is 15.6% but varies from 6% to 40.2%. In high-SNR regions (say, $> 16dB$ in our setting), DS is more likely to outperform SS. In fact, in almost all cases, DS is the winner over SS, with the average goodput gain being 33.2%. The actual goodput gain varies from 17% to 60.4%. The winning DS rates span the broad set of 108DS, 162DS, 243DS, 270DS, and 300DS.

One should be cautious in applying the above findings, because they simply show the general trend rather than claim which specific mode is the winner in all cases. In fact, there is always the gray area where either can be the likely winner. Moreover, there are several non-trivial challenges in finding a good SNR threshold to be used to select between SS and DS modes. We will elaborate on these issues in Section 6.3.

4.3 On Frame Aggregation

We next study the behavior of frame aggregation and its impact on RA designs. We used the frame aggregation algorithm currently implemented in the Atheros chipset. An in-depth study of different frame aggregation algorithms is beyond the scope of this work. Our experiments reveal interesting findings due to the interplay of rate adaptation and frame aggregation. First, since higher aggregation can lead to higher goodput due to amortized overhead, the RA designs may naturally try to maximize the *aggregation level*, defined as the number of data frames (MPDUs) in an aggregate frame (A-MPDU). However, our experiment shows that this is not always the best strategy because high aggregation level makes RA less responsive to fast channel dynamics thus reducing the effective goodput. Table 4 shows such an example. In the experiments, we ran two algorithms at location P6. One was RRAA, and the other was RRAA

Rates (Mbps)	RRAA (%)	RRAA-Limited (%)	Aggregation Bound
13.5SS	2	1	4
27SS	3	0.5	8
27DS	1	0.5	8
40.5SS	8	14	13
54SS	23	42	17
54DS	29	26	17
81SS	11	1	26
81DS	23	15	26
108SS			35
108DS			35
Goodput (Mbps)	24.22	35.60	
SFER (%)	46.61	24.83	
Avg. Aggr. level	19.36	11.81	

Table 4: Rate distribution (%) and Goodput performance for RRAA and RRAA Limited at P6.

with upper bound on the maximum aggregation level (i.e., RRAA-limited). The maximum aggregation level is set in proportion to the rate (shown in the last column of Table 4), thus seeking to achieve equal air time among transmissions at each rate. Table 4 shows that the average aggregation in RRAA is about 7.6 MPDUs larger than that in RRAA-limited. However, RRAA-limited offered 46.9% goodput gain over RRAA, even with smaller aggregation. Our analysis of the traces showed that, RRAA experienced 21.8% higher loss, and higher aggregation level during lossy transmissions led to more frame losses, thus reducing goodput. Table 4 indicates that RRAA transmits 34% of frames at 81 Mbps, which gave 86.3% SFER. Higher aggregation at this lossy rate hurts goodput.

Second, SFER can affect frame aggregation along both positive and negative directions. On one hand, higher loss may increase the aggregation level. This is because the lost MPDUs are put back into the software queue for retransmission, thus higher loss can result in larger queue size and more aggressive aggregation. To verify this hypothesis, we fixed the transmission rate at 135SS and the data source at 60Mbps, and varied the SFER by switching to different locations. When the loss is small (say, 5.3%), the average aggregation level is 3.0 MPDUs. When the loss is medium (say, 29.2%), the aggregation level is 10.5 MPDUs. When the loss is excessive (say, 99.5%), the aggregation level grows to 18.9 MPDUs. On the other hand, loss may also have negative impact. Figure 3 plots how the aggregation level evolves with SFER in one experiment, where the transmission rate was set at 81SS and the data source was aggressive enough to ensure full software queue. We see that high SFER dropped the average aggregation level from 21 MPDUs to 8.7 MPDUs in the experiment. It turns out that, the limiting factor here is the Block ACK Window (BAW) specified by the 802.11n standard. BAW moves forward as long as MPDUs with sequence numbers inside the BAW are acknowledged, similar to the sliding window scheme in TCP. However, if the first MPDU with sequence number Seq within BAW is lost and to be retransmitted, then all followup aggregate frames can only aggregate frames within the window of BAW, i.e., with sequence numbers less than $Seq+64$, where 64 is maximum number of frames aggregated in a single frame in 802.11n. If there are four followup aggregate frames, the aggregation level is only 16 MPDUs on average. Therefore, the position of the lost sub-frame affects the aggregation level for the followup frames.

5. DESIGN

MiRA seeks to identify and set its transmission rate to the best rate option. Unlike other RA algorithms, MiRA uses a novel zigzag scheme, which opportunistically switches between intra- and inter-mode RA operations, to address the 802.11n MIMO characteristics. It further uses two-level prioritized probing to reduce the penalty of excessive probing at bad rates. Finally, MiRA detects collisions from channel errors based on the loss pattern learned from the 802.11n frame aggregation and BlockAck, without using the RTS/CTS mechanism. We now elaborate on each operation in details.

Procedure 1 ZigZagRA: Input (BlockAck), Output (r)

```
1: update-stats( BlockAck, r)
2: collision-detection-and-reaction( BlockAck, r)
3:
4: //zigzag RA: intra- and inter- probing
5: //isProbe: a variable indicating whether the last frame is a
   probe
6: //probeSeq: a list of rates already probed
7: if isProbe = true then
8:    update-priority-probing-timer( BlockAck, r)
9:    if intra-mode-RA-finished(probeSeq) = false then
10:      (r, isProbe, probeSeq) ← next-intra-rate(r, probeSeq)
11:   else if inter-mode-RA-finished(probeSeq) = false then
12:      (r, isProbe, probeSeq) ← next-inter-rate(r, probeSeq)
13:   else
14:      //finish probing, select the best rate among the probes
15:      (r, isProbe, probeSeq) ← best-rate(r, probeSeq)
16:   end if
17:   return r
18: end if
19:
20: if probe-timer-expired() = true then
21:   //adaptive probing timer expires
22:   (r, isProbe, probeSeq) ← timer-expired-rate()
23: else if G_r(t) ≤ Ḡ_r(t) − 2 · σ_r(t) then
24:   //channel becomes good
25:   (r, isProbe, probeSeq) ← next-higher-intra-rate(r)
26: else if G_r(t) ≥ Ḡ_r(t) + 2 · σ_r(t) then
27:   //channel becomes bad
28:   (r, isProbe, probeSeq) ← next-lower-intra-rate(r)
29: else
30:   //remain in current rate
31:   isProbe ← false
32:   probeSeq ← r
33: end if
34: return r
```

5.1 Zigzag RA: Intra- and Inter-mode RA

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

The pseudo code is shown in Procedure 1. It decides, for each aggregated frame, which rate is used and whether this frame serves as a probe. There are three kinds of probes: same-mode downward, same-mode upward and cross-mode.

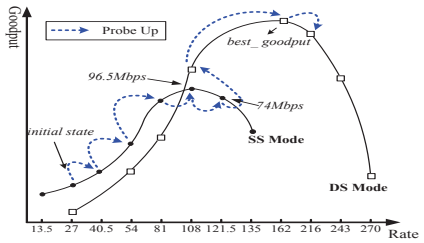


Figure 4: Example for Zigzag RA: Rate upward trajectory upon better channel.

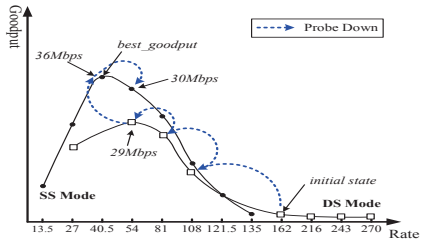


Figure 5: Example for Zigzag RA: Rate downward trajectory upon worse channel.

Based on the transmission result, it updates the statistics and then triggers further probes if necessary. Its operations can be illustrated by the examples in Figures 4 and 5.

Suppose the starting rate is 27SS at time t_0, again at location P4. Upon detecting a better rate, MiRA moves upward in the SS mode. It continues to probe upward as long as the estimated goodput keeps on increasing, thus going through the probing sequence at rates of 40.5SS, 54SS, 81SS, 108SS. When it further probes 121.5SS that gives the goodput $74Mbps$, it does not see a higher or equal goodput than 108SS (offering $96.5Mbps$ in Table 1). MiRA thus completes the intra-mode RA operation within SS mode. Subsequently, MiRA zigzags to the DS mode by first probing at 108DS, which is the lowest DS rate whose loss-free goodput is higher than $96.5Mbps$. Within the DS mode, It further probes upward to 162DS and 216DS. It finally sets the transmission rate at 162DS since 216DS delivers lower goodput than 162DS, thus completing the upward operations.

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

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

5.1.1 Prioritized Probing

Similar to other RA algorithms, MiRA uses actual data frame transmissions to probe other rates and identify the best rate that offers highest goodput. Different from other solutions, MiRA devises a novel, prioritized probing scheme to address MIMO related cross-mode characteristics. Moreover, it uses adaptive probing to dynamically adjust the probing interval based on the measured SFER and recent probing history, in order to reduce excessive probing to bad rates. MiRA addresses four issues in its probing scheme: (1) When to initiate probing? (2) What rates to probe? (3) How to probe the candidate rates in both modes? and (4) How to avoid excessive overhead?

Probing triggers MiRA triggers probing and subsequent goodput estimation using both event-driven and time-driven mechanisms. It starts probing whenever it observes significant change in the measured goodput at the current rate. Specifically, it probes downward (to a lower rate) when $G_r(t) \leq \overline{G_r}(t) - 2 \cdot \sigma_r(t)$, where $G_r(t)$ is the measured goodput for rate r at time t, $\overline{G_r}(t)$ is the moving average of the goodput, and $\sigma_r(t)$ is the standard deviation of the goodput[2]. Similarly, it probes upward (to a higher rate) when $G_r(t) \geq \overline{G_r}(t) + 2 \cdot \sigma_r(t)$. Alternatively, when the probing timer for a given rate option expires, MiRA initiates probing at that given rate. In essence, MiRA uses time-driven probing to update stale information on goodput statistics, and event-driven probing scheme to quickly track sudden channel variations. Once the probe starts, MiRA uses a single A-MPDU to probe the selected rate. Since each aggregate frame typically carries tens of frames, this probe is typically sufficient to collect loss statistics.

Candidate rates for probing MiRA opportunistically selects the candidate set of rates to probe at a given time. When probing upward, it first starts from the immediate, higher rate option within the same mode. It sequentially goes to each higher rate option. Note that SFER also increases with each higher rate in the same mode. The intra-mode probe stops at the highest rate option if its next higher rate has a goodput estimate (using its measured SFER) smaller than the highest goodput estimate obtained so far. It then selects rates for inter-mode probing, starting from the lowest rate, which loss-free goodput is higher than the highest goodput estimate so far. In the example of Figure 4, the candidate rate set is {40.5SS, 54SS, 81SS, 108SS, 121.5SS, 108DS, 162DS, 216DS} when the upward probing starts from 27SS. Note that in inter-mode probing, the goodput estimate at 108SS is about $96Mbps$, higher than the loss-free goodput at 81DS. Therefore, the inter-mode probing in DS mode starts from 108DS. When probing downward, it also starts from the immediate lower rate within the same mode. It sequentially goes to each lower rate until its highest goodput estimate so far is larger than the next lower rate. This implies the best goodput estimate so far is larger than the loss-free goodput that the lower rate may offer. In the example of Figure 5, the candidate rate set is {108DS, 81DS, 54DS, 40.5SS, 54SS} when the downward

[2]More precisely, it is the moving average of the standard deviation, as we will describe in Section 5.1.2.

probing starts from 162DS. Note that the goodput estimate at 54DS is about $30Mbps$, so MiRA does not probe 27SS whose loss-free goodput will be lower than the goodput estimate at 54DS. Therefore, MiRA initiates inter-mode probing. To this end, 40.5SS is chosen first since its loss-free goodput is better than $30Mbps$. It then probes upward at 54SS which offers lower goodput estimate, so it finally identifies the best rate as 40.5SS.

Two-level probing priority MiRA ranks the sequence of rates to be probed within each mode and across modes using a two-level priority scheme.

The first-level priority addresses intra-mode and inter-mode probing. In MiRA, intra-mode probing is always given higher priority and takes precedence over inter-mode probing. Therefore, probing in MiRA always starts to probe other rates in the same mode (SS or DS). The second-level priority manages probing order among candidate rates in the same mode. MiRA always gives higher priority to the rate option closer to the current rate. Therefore, it always probes the adjacent rate first, and then the next higher/lower rate in the same mode when probing upward/downward. In the example of Figure 4, at the initial rate 27SS, the probing priority for upward probe within SS follows the order: $40.5SS > 54SS > 81SS > 108SS > 121.5SS$. When probing downward from 162DS in Figure 5, the probing priority within DS follows the sequence: $108DS > 81DS > 54DS$.

When switching to the other mode (say, from SS to DS), the lowest rate in the new mode which loss-free goodput is closest to the best-goodput rate in the old mode receives highest priority. Consequently, in the example of Figure 4, when probing upward from the SS mode to the DS mode, 108DS, which loss-free goodput is the closest to the best goodput estimate in SS (i.e., 108SS that has goodput estimate of 96.5Mbps), has the highest probe priority among all candidate rates within DS. The upward probing order in DS is $108DS > 162DS > 216DS$. In the same example of downward probing from DS to SS, 40.5SS is the highest priority probe rate since it is the lowest rate whose loss-free goodput is closest to the best goodput rate 54DS in the DS mode. After switching the probe to SS, the probing order follows $40.5SS > 54SS$.

Once probing is done, MiRA leap-jumps to the best rate after probing. In a sense, MiRA stays in the middle between sequential rate adaptation (e.g., RRAA) and best-rate RA (e.g., SampleRate): It differs from RRAA in that it may leap to the best rate nonsequentially; it differs from SampleRate in that it still probes sequentially among rate candidates.

Adaptive probing interval MiRA uses two mechanisms of *loss-proportional* and *binary exponential growth* to adaptively set the probing intervals for three eligible rates: the two adjacent intra-mode rates and one inter-mode rate. These three rates are used for probing upward and downward in the current mode, and probing in the other mode. The inter-mode rate is the smallest rate in the other mode which loss-free goodput is larger than the goodput at the current rate. Consider the current rate 54SS at time t_2 in Figure 4, the adaptive probing intervals are set for three rates: 81SS and 40.5SS used for intra-mode, and 54DS which loss-free goodput is larger than the goodput 30Mbps at 54SS. As MiRA adapts its rate upward or downward, these three rates are also changed accordingly.

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

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

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

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

5.1.2 Goodput Estimation

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

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

where $\alpha = \frac{1}{8}$ and $\beta = \frac{1}{4}$ are two parameters.

Note that the instantenous goodput depends on the aggregation level, which may vary a lot from one aggregate frame to another. Using the aggregation level measured from the current probe may lead to fluctuating and inaccurate estimation. To address this issue, we use the moving average of the aggregation level:

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

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

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

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

5.2 Handling Hidden Terminals

Recent studies [2,5] have shown that interference-induced data losses can adversely affect the rate adaptation operations. In such cases, reducing the rate upon losses may exacerbate collisions since the transmission takes long time at lower rates. Thus, a good RA design should differentiate between channel fading losses and collision losses.

Collision Detection MiRA takes a novel approach to collision detection by exploiting the unique MIMO features of frame aggregation and BlockAck. During our extensive experiments, we have observed that channel fading losses and collision losses tend to exhibit very different patterns (uniform and near-binary, respectively). As an illustrative

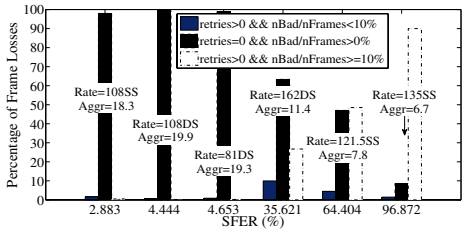


Figure 6: Loss patterns w/o interference

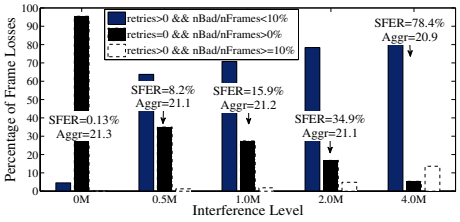


Figure 7: Loss patterns with interference

example, Figure 6 shows the loss patterns without interference at location P15, while Figure 7 presents the loss patterns under a hidden terminal setting. We categorize the frame losses into three types, based on the number of retries and the loss rate in the last retry. These results (and similar ones at other locations) reveal a distinct pattern of collision losses:

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

That is, the last aggregate frame experienced at least one retry, yet in the last retry, it was received with very mild subframe loss. The root cause of the above interference loss pattern can be attributed to the corruption of the PHY header upon collisions, thus causing the entire A-MPDU to be lost [17].

These findings provide us a simple heuristic to infer the possible occurence of collisions, by checking the above condition against each aggregate frame transmission. While this heuristic is shown to be quite effective in our experiments (detailed in Section 6), it may lead to incorrect detection results occasionally (categorizing fading losses as collisions, or vice versa). To improve the detection accuracy, MiRA relies on repeated collision indications during a short timespan, rather than a single instance. To this end, MiRA maintains a dynamic interference observation window ($IFWnd$), which is normally set to 0. Whenever an aggregate frame satisfies Condition (1), MiRA suspects collisions and thus initializes $IFWnd$ to a pre-defined value (say, 3 in our implementation). For the subsequent $IFWnd$ aggregate frames, if any of them exhibits the collision pattern again, MiRA will confirm the collisions and trigger the reactions, as described below. Otherwise, $IFWnd$ decrements by one for each frame not satisfying Condition (1), until $IFWnd$ reaches 0.

Two alternatives to collision detection, using adaptive RTS filter [2] and SNR [5], both have downsides in the 802.11n MIMO case. An MIMO device typically operates at much higher rates than the legacy 802.11b/a/g device, thus the relative overhead of RTS/CTS grows much larger. Because the adaptive RTS/CTS scheme turns on RTS/CTS regardless of date rate or frame size, it introduces significant overhead with high rates and/or small frames. On the other hand, the SNR-triggered approach requires the sender to obtain fine-grained, per-frame accurate SNR information from the receiver, which is not available in current 802.11n systems. Moreover, 802.11 systems only measure SNR for successfully received, not collided frames at the receiver.

Cost-Effective Collision Reaction MiRA takes a cost-effective approach to whether to turn on RTS/CTS protection. Note that blindly using RTS/CTS upon collisions is not a viable strategy, because the RTS/CTS overhead can be prohibitively expensive, especially at high rates. Thus, we turn on RTS/CTS *o*nly when the potential gain outweighs the overhead.

Specifically, we estimate the RTS/CTS overhead as T_{RCTS}, the channel time used by transmitting RTS/CTS signaling messages. On the other hand, given the current aggregate frame $nFrames$ to be transmitted at rate R, the potential saving is estimated as $\frac{nFrames}{R}$. The logic here is that, without RTS/CTS, this aggregate frame needs at least one retry to get through after collisions. Now the condition to turn on RTS/CTS is given by,

$$\frac{nFrames}{R} \geq k \cdot T_{RCTS} \tag{2}$$

where k is a benefit/cost ratio (say, 1.5 in our prototype). If this condition is not met, MiRA simply resets $IFWnd$, without turning on RTS/CTS.

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

5.3 Other Design Options for MIMO RA

Now we discuss several alternative design approaches to MIMO RA. While seemingly simple at the conceptual level, they all have obvious downsides that pose non-trivial challenges to make them work consistenly well in practice. The first approach searches for the best rate option across both SS and DS modes within a pre-specified range of rates. It is mode oblivious in that it does not explicitly consider the SS/DS mode characteristics. However, it is nontrivial to properly pre-select the range: the smaller the rate range, the higher probability the optimal rate is missed; the larger the rate range, the bigger the probing overhead. We will further examine an algorithm *tuned-SampleRate* in this category in Section 6.3. The second approach exploits the observation made in Section 4.2 and takes a SNR-based mode selection. When the measured SNR is lower (or higher) than a threshold, it chooses the SS mode (or DS mode) and uses the conventional RA within the mode[3]. One challenge for this approach is how to set the thresholds, which change with different operation environments, as we demonstrate in Section 6.3. Last, it seems that, the fast MCS feedback approach measuring SNR at the receiver and sending back

[3]Another design variant makes use of two SNR thresholds in selecting the mode.

	Atheros RA	RRAA	SampleRate
Static UDP	(5.6-82.3)%	(5.3-71)%	(5.6-104.5)%
Static TCP	(9.1-107.9)%	(5.9-37.5)%	(14.7-124.8)%
Mobility UDP	116.1%	30.2%	182.2%
Mobility TCP	72.5%	4.9%	94%
Hidden Terminal	(79.4-1094)%	up to 6.5%	(33.8-983)%
Field Trial	67.4%	20.6%	19.4%

Table 5: Goodput gains of MiRA over existing RAs.

to the sender, which is included in the 802.11n standard yet implemented by the current chipsets, can address most problems. However, as our experiments show in Section 6.3, the measured SNR can have large fluctuations, thus making this approach not as effective as it appears.

6. IMPLEMENTATION AND EVALUATION

In this section, we describe our implementation of MiRA on a programmable AP platform and evaluate its performance using both controlled experiments and field trials.

6.1 Implementation

We implemented MiRA in the firmware of a programmable AP platform (about 900 lines of C code). Compared with other RA algorithms, MiRA poses two implementation challenges. First, its probing mechanism requires frame transmission and rate control, which are two separate modules in the driver, to be synchronized on a per-AMPDU basis. We maintain an additional binary state for each client (other states kept at AP are per-client statistics), which is set upon collision losses and checked for each AMPDU transmission. The second challenge is that, the NAV for RTS cannot be directly set by the transmission module of the driver. To reserve the wireless channel, we use Atheros' *Virtual more Fragment* interface, which consists of a virtual more-fragment bit (vmf) and a *burst_duration* parameter. Atheros uses this interface to enable frame bursts. Upon collision losses, if channel reservation is possible we set the vmf bit as 1 and *burst_duration* as the transmission time of the aggregated frames that NAV in RTS protects (the reception time of BlockAck is also included in *burst_duration*). The virtual more-fragment bit goes down to the hardware queue together with the burst of aggregate frames.

6.2 Performance Evaluation

In this section, we compare MiRA with RRAA [2], SampleRate [4] and Atheros MIMO RA [3]. For RRAA, we disabled its adaptive RTS/CTS filter, except in the hidden terminal settings, to avoid goodput degradation which was observed to be up to 12.2% during our experiments. These experiments were conducted in various scenarios with varying client locations, mobility, hidden terminal stations, and different MIMO channel settings. The traffic was both UDP and TCP from our AP to one client. All the algorithms were implemented on AP side. The results show that MiRA consistently outperforms existing alogithms in all scenarios, with goodput improvement up to 124.8% for static clients, 182.2% for mobile clients, and 67.4% in field trials. The performance gains of MiRA over existing RAs are summarized in Table 5.

6.2.1 Static Clients

We first evaluate the RA algorithms with static clients at multiple locations. The goal is to assess how well each algorithm handles random channel losses, e.g., due to path loss, shadowing or multi-path effect. We conduct these experiments during midnight and, in the 5GHz band cases, we select clean channels with no background traffic (as verified by the sniffer). We perform tests with both high-volume UDP traffic and a single TCP flow with various MIMO configurations, e.g., 5GHz vs. 2.4GHz bands, 3×3 vs. 2×2 antennas. The channel bandwidth is set to 40MHz in all experiments unless explicitly specified.

UDP/3×3 Antennas/5GHz Case: Figure 8 plots the UDP goodput measured at 6 different locations (as marked in Figure 1) with 3×3 antennas at 5GHz band and the maximum MiRA goodput gains over the other designs. We see that MiRA performs better than other algorithms at all locations, with goodput gains up to 70.7% over Atheros RA, 54.2% over RRAA, and 68.9% over SampleRate. Except from the closest client-to-AP location where all RAs tend to transmit at high rates, MiRA delivers gains from 49.4% (location P5) to about 70.7% (location P4).

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

TCP/3×3 Antennas/5GHz Case: We also conduct experiments with a TCP flow. Figure 9 shows that MiRA gives significant TCP goodput gain over others, up to 107.9% over Atheros MIMO RA, 37.5% over RRAA, and 124.8% over SampleRate. Similar to the UDP scenario, MiRA offers high gains in all locations, starting from 24.1% (location P5) to 124.8% (location P6).

UDP/3×3 Antennas/2.4GHz Case: We also test 2.4GHz channels. Setting the channel to 40MHz in 2.4GHz results in partially overlapping channels. During this experiment, we set our AP on Channel 1. We sniff many other APs on other channels: twelve on Channel 1, two on Channel 4, eight on Channel 6, six on Channel 9 and nine on Channel 11. The goodput performance and gains of MiRA vary from 9.6% to 57.7% at five locations, as shown in Figure 10. We see that losses and goodput degradation are significant compared with the 5GHz band due to highly uncontrolled interference.

UDP/3×3 Antennas/2.4GHz/20MHz Case: We finally repeat the experiments of the 2.4GHz band setting using 20MHz channel. For the 20MHz channel case, MiRA gives also significant gains which are up to 36.9% over Atheros RA, 70% over RRAA, and 44.5% over SampleRate. Even with 20MHz channel the highest goodput observed was 43Mbps because of the intense interference.

By analyzing the traces from the above experiments, we have identified additional aspects that contribute to the performance gains of MiRA.

Effective probing: Most existing RAs only use pre-defined periods/steps to determine when to probe for better channel, which may lead to high losses in scenarios where SFER difference between adjacent rates is big. We verify this in

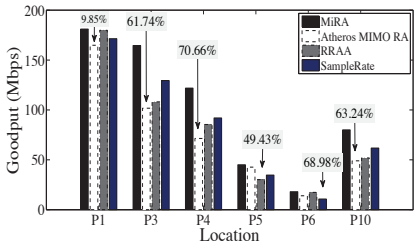


Figure 8: 3×3 **/5GHz/UDP static setting.**

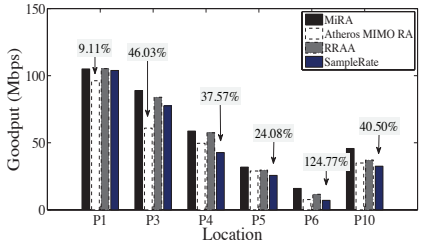


Figure 9: 3×3 **/5GHz/TCP static setting.**

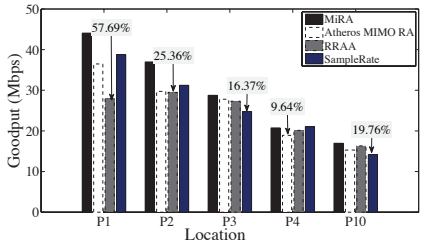


Figure 10: 3×3 **/2.4GHz/UDP static setting.**

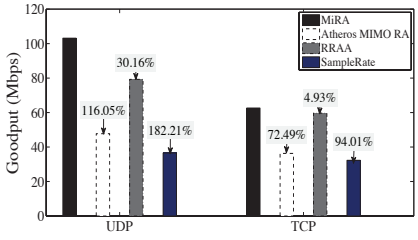


Figure 11: Goodput in mobility setting.

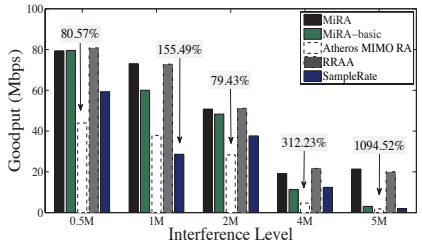


Figure 12: Goodput in hidden terminal.

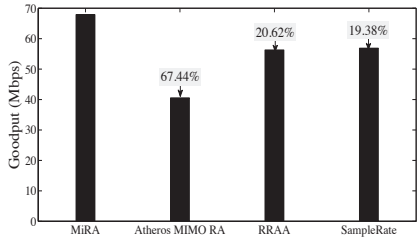


Figure 13: MiRA performance in field trials.

our experiments. For example, at location P4, RRAA sends 42.5% of MPDUs at 121.5Mbps, which exhibits significant loss ($SFER_{121.5} = 17.9\%$ in Table 1). Similar behavior is observed in other scenarios such as P10, where RRAA repeatedly probes 108SS (31.7% MPDUs are sent at 108SS) that consistently suffers from more than 34% loss. Similar to RRAA, SampleRate also uses non-adaptive probing, despite less aggressive than RRAA. In contrast, the *adaptive probing mechanism* of MiRA prevents it from excessively transmitting at lossy rates. MiRA transmits only 2% and almost 0% probes at low-goodput rates, at locations P4 and P10, respectively.

Handling SFER non-monotonicity: MiRA is not trapped at lower rates in the scenarios where SFER non-monotonicity exhibits across SS/DS rates. It switches to the best rate even though it belongs to the different mode. Using P4 as an example, MiRA transmits 96% of frames at 162Mbps that is the best-goodput rate most of the time. In contrast, other algorithms transmit more than 90% of their data at rates lower than 162DS.

We also observed that Atheros MIMO RA may occasionally get stuck at a lower rate because of SFER same-rate-pair non-monotonicity (say 54SS/DS) in some locations. The Atheros algorithm *a priori* ranks all rates to be probed in the particular order, say 54DS has higher ranking than 54SS but lower than 81SS in the implementation. Consider that the current probe upper-bound rate is set as 54DS in the Atheros algorithm. However, 54SS has higher goodput than 54DS in locations P3, P4, and P10. Then the algorithm has no way to skip 54DS and probe higher rate 81SS. Therefore, the algorithm may get stuck at 54SS and can send up to 50% of frames at 54SS in these three locations.

6.2.2 Mobile Clients

In order to gauge the responsiveness of MiRA to fast channel dynamics, we carry a client and walk from P1 to P6 and then come back at approximately constant speed of 1m/s. Figure 11 plots the goodput of the four RAs for both UDP and single-flow TCP traffic. MiRA offers goodput gain up to 116.1% over Atheros RA, up to 30.2% over RRAA, and up to 182.2% over SampleRate. As discussed in Section 5, MiRA uses (i) *moving average* to detect significant channel changes, (ii) only one AMPDU to probe, which is transmitted in a relatively short period and typically contains enough samples, and (iii) resetting statistical history upon rate changes. Consequently, MiRA quickly adapts to channel dynamics due to mobility.

6.2.3 Setting with Hidden Terminals

We next evaluate whether MiRA can successfully infer collision losses and adjust the rate accordingly in the presence of hidden terminals. We also compare MiRA with MiRA-basic (MiRA without interference module) to evaluate the performance of our interference module. In the hidden terminal setting, we also turn on RRAA's Adaptive RTS filter. Figure 12 presents the gains of MiRA at five interference levels by varying the traffic intensity of the hidden terminal. We observe that MiRA is very effective in intense interference scenarios (4Mbps and 5Mbps interference) where it can gain up to 1094.5% over Atheros MIMO RA, and up to 599.9% over SampleRate. MiRA performs similar to RRAA, without having to pay the RTS/CTS overhead of the adaptive RTS filter in RRAA. MiRA can give significant gains over MiRA-Basic from 5.1% to 599.9%.

We attribute the gain to both the selective RTS mechanism and the decision that MiRA does not probe down upon detecting hidden terminals. MiRA yields up to 32.5% smaller average loss compared with Atheros MIMO RA and SampleRate. By avoiding probing down in high interference scenarios, MiRA still transmits at high rates under heavy collisions. With 4Mbps interference, Atheros MIMO RA

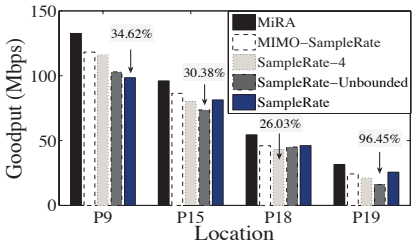


Figure 14: Comparing MiRA with Tuned-SampleRate.

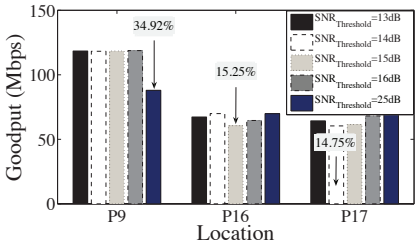


Figure 15: MIMO-SampleRate vs. different SNR thresholds.

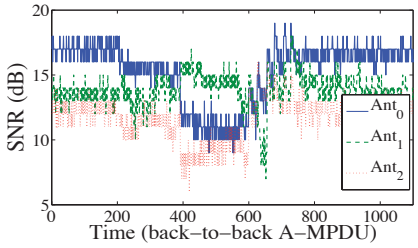


Figure 16: Per-antenna SNR of control channel in a static setting.

transmits 94% of frames at the lowest rate 13.5, while MiRA only transmits 6% at this rate.

6.2.4 *Field Trials*

We also conduct uncontrolled field trials under realistic scenarios, where various sources of dynamics coexist in a complex manner. In our field trial, we use 3 static clients, at locations P4, P10, and P17, and we move an 802.11n client on a regular basis based on the mobility scenario of Section 6.2.2. We use TCP traffic to evaluate each RA for about an hour. During our experiments, the physical environment was highly dynamic as people walk back and forth.

Figure 13 shows that MiRA gives 67.4% goodput gain over Atheros MIMO RA, 20.6% goodput gain over RRAA, and 19.4% goodput gain over SampleRate.

6.3 Assessing MIMO RA Alternatives

We next assess three alternative design guidelines via experiments.

Tuned SampleRate Algorithm A best-goodput RA algorithm as SampleRate, can directly jump to the best-goodput rate independent of its MIMO mode. However by upper-bounding sampling up to 2 rates higher than the current rate, SampleRate has limitations to address SFER non-monotonicity as stated in Section 3. To address this issue we implement SampleRate-4 that enlarges the sampling bound to 4, and SampleRate-Unbounded that allows for search among all the rates larger than the current rate.

Figure 14 plots the goodput results of all three algorithms and the respective goodput gain of MiRA at four locations P9, P15, P18, and P19. It indeed shows that expanding the search scope helps to obtain a better rate at P9, where SampleRate-4 achieves goodput gain of 18% over SampleRate. However, SampleRate-4 has lower or similar goodput in the other three locations. At P19, SampleRate delivers 21.2% goodput gain over SampleRate-4. SampleRate-Unbounded is even worse, incurring goodput reduction up to 37.3%. Trace analysis reveals that SampleRate transmits 87% of frames at the high-goodput rates (40.5Mbps, 54Mbps), whereas SampleRate-4 transmits only 50% at these rates. SampleRate-Unbounded transmits 9.5% of frames at almost 100% loss. Sampling of these expanded rates consequently incurs higher probe penalty.

SNR-based Mode Selection RA The next algorithm MIMO-SampleRate selects the SS/DS mode based on a preselected SNR threshold, say, 14dB in our experiments[4]. In our implementation, SNR is measured from the received ACK frames and averaged over all antennas. It exploits the findings in Section 4.2: When SNR is smaller than the threshold, then SS is more likely to be the winning mode; when SNR is larger, then DS is inferred as the winning mode. Once the mode is chosen based on SNR, then the SampleRate algorithm is used within the mode for actual RA.

The results of MIMO-SampleRate are plotted in Figure 14. MIMO-SampleRate achieves higher or similar goodput compared with tuned-SampleRate, with goodput gain up to 20%. However, MiRA still performs best, 11% to 30% higher than MIMO-SampleRate. These results seem to indicate that the SNR-based MIMO-SampleRate can achieve pretty good performance, while retaining its operation simplicity. However, our experiments show that, there may not be an optimal SNR threshold to give the highest goodput in all the settings. The best SNR threshold values may also depend on the operation environment.

Figure 15 shows the goodput results using different SNR thresholds at locations P9, P16, and P17. By choosing a high SNR threshold, say 25dB, at location P9, we exclude DS rates (including the highest-goodput rate 162DS), thus incurring goodput degradation up to 34.9% compared with using [13dB, 16dB] thresholds at P9. However, choosing SNR thresholds in [13dB, 16dB] does not guarantee the best performance in other locations. At P16 and P17, 25dB threshold outperforms other choices up to 15.3% and 14.8% respectively. This is attributed to the algorithm's fluctuation between SS/DS modes when using other threshold values. For example, at P17 (where SNR is 14.5dB), while 25dB threshold transmits more than 98% of the frames at 81SS (which is the best-goodput rate), other threshold values give sub-optimal rate distributions.

RA Based on Fast MCS Feedback When the fast MCS feedback becomes available in 802.11n commodity devices, we can use receiver-initiated rate adaptation based on per-frame feedback of the best rate computed via the measured SNR. While this approach seems quite promising, it also poses challenges. Our experiments show that, even in a static setting, the SNR may fluctuate within a large range in the MIMO channel. Figure 16 plots the per-antenna SNR evolution for the control channel (the primary 20MHz channel) in static 3×3 antenna setting. Even in a static setting, the measured SNR can vary from 6dB to 19dB. Similar results also hold for the extension channel, fluctuating from 4dB to 19dB. The large SNR fluctuations make accurate rate selection based on SNR values challenging. We note that similar observations have been made for the commodity 802.11a/b/g systems due to hardware calibrations and inter-

[4]A more sophisticated algorithm can use two thresholds, but the results are similar.

fering transmissions [14]. Moreover, the SNR-BER relations also vary with different propagation environments. Consequently, SNR-based solutions require in-situ training to perform well across different propagation environments [5].

7. RELATED WORK

There have been a number of rate adaptation (RA) proposals [2,4–6,12,13]. They either target the legacy 802.11a/b/g networks [2, 4–6], or take a cross-layer approach [12, 13] by using Signal-to-Noise ratio and loss information to select the best-goodput rate. These algorithms are not designed for MIMO systems. They do not consider SS and DS operation modes and 802.11n frame aggregation.

The early work on MIMO RA [8, 9, 16] takes the receiver-based approach by exploiting the fast MCS feedback mechanism. In addition to the issues raised in Section 6.3, these algorithms require PHY feedback, not available in the current 802.11n driver [16]. Another sender-based RA proposal [18] ARFHT selects the appropriate MIMO mode based on SNR differences among receive antennas. It assumes MIMO channel reciprocity as it measures SNR at the sender based on received frames. Moreover, challenges using SNR are discussed in Section 6.3. Atheros MIMO RA [3] is implemented in 802.11n chipsets. It selects the best-goodput rate based on SFER statistics, but the candidate rates for sampling and selection are upper-bounded by *maxRate*.

There are also a few experimental studies relevant to this work. In [10], authors report experimental findings on 802.11n, but they do not focus on RA related issues. In [7], experiments are based on a testbed that supports only a limited set of 802.11n features.

There are extensive theoretical studies on MIMO communications (e.g., [15, 16]). These analytical results seek to characterize the theoretical features and tradeoffs of MIMO systems, often in the limiting cases. In contrast, our study uses experiments to examine the behavior of 802.11n MIMO devices in the real-world settings.

8. CONCLUSION

In this paper, we use an IEEE 802.11n compliant, programmable AP platform to study MIMO rate adaptation. Our research shows that, an 802.11n RA scheme must address MIMO related characteristics to do well. To this end, we propose MiRA, a new RA algorithm that explicitly adapts to the SS and DS modes in 802.11n MIMO systems. The key insight is that diversity-oriented SS mode and spatial multiplexing-driven DS mode exhibit different features and cannot be managed indistinctly. Existing RA solutions do not properly consider characteristics of SS and DS, thus suffering from severe performance degradation. In fact, even a fixed-rate scheme may outperform them. In a nutshell, our work is among the first to examine MIMO RA in a practical setting using programmable 802.11n commercial hardware. We expect that our effort will stimulate more community effort on MIMO RA to exploit the full capacity of MIMO communication.

9. ACKNOWLEDGMENTS

We appreciate the constructive comments by our shepherd, Dr. Srikanth V. Krishnamurthy, and the anonymous reviewers. We also thank Drs. Babak Daneshrad and Lixia Zhang, Mr. Suk-Bok Lee and Ms. Chunyi Peng for helpful discussions.

10. REFERENCES

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

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

[3] M. Wong, J. M. Gilbert and C. H. Barratt. Wireless LAN using RSSI and BER parameters for transmission rate adaptation. *US patent* 7,369,510, 2008.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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