Practical Rate Adaptation in Mobile Environments

Xi Chen, Prateek Gangwal and Daji Qiao
Iowa State University, Ames, IA 50011
{leon6827, prateek, daji}@iastate.edu

***Abstract*—Channel asymmetry and high fluctuation of channel conditions are two salient characteristics of wireless channels in mobile environments. Therefore, when using IEEE 802.11 devices in mobile environments such as vehicular networks, it is critical to have an effective rate adaptation scheme that can deal with these issues. In this paper, we propose a practical rate adaptation scheme called RAM (Rate Adaptation in Mobile environments) and implement it in the Madwifi device driver. RAM uses a receiver-based approach to handle channel asymmetry and a conservative SNR prediction algorithm to deal with high channel fluctuation. More importantly, RAM allows the receiver to convey the feedback information in a creative manner via ACK transmission rate variation, which does not require changes to the device firmware and hence is implementable at the device driver level. The effectiveness of RAM is demonstrated through experimental evaluation in indoor static and mobile environments and outdoor vehicular environments, as well as simulation study based on SNR traces collected from the experiments.**

I. INTRODUCTION

A. *Motivation and Contribution*

The increasing number of IEEE 802.11 devices have been used in various vehicular networking systems. Since most of the resource management schemes for 802.11 devices are designed for static environments, the 802.11-based vehicular networks may experience poor system performance, such as low throughput and high latency. This is due to the salient differences in wireless channel characteristics between static and mobile environments. For example, channel conditions in mobile environments usually exhibit more severe asymmetry and higher fluctuation than those in static environments.

In this paper, we study *rate adaptation* in mobile environments. Rate adaptation is one of the fundamental resource management issues for 802.11 devices. The goal is to maximize the throughput via exploiting the multiple transmission rates available for 802.11 devices and adjusting their transmission rates dynamically to the time-varying and location-dependent wireless channel conditions. From the experiments, we find that most of the existing rate adaptation schemes cannot handle channel asymmetry or high fluctuation of channel conditions well, and hence may not be suitable for mobile environments. A few existing schemes may be able to deal with channel asymmetry but require changes to the CTS or ACK frame formats, which typically are hard-coded in the device firmware. As a result, these schemes do not conform to the 802.11 standard and thus may not be easily implementable with commercial 802.11 devices, which limits their practical applications drastically.

We propose a practical rate adaptation scheme, called RAM (Rate Adaptation in Mobile environments), and demonstrate its effectiveness in both mobile and static environments via experiments and simulations. RAM has the following features:

- RAM is a practical rate adaptation scheme and we have implemented RAM in the Madwifi device driver [1].
- RAM is a receiver-based scheme and can deal with channel asymmetry well. Different from existing receiver-based rate adaptation schemes, RAM uses the variation of the ACK transmission rate to convey the feedback information implicitly. This means that RAM does not require changes to the CTS or ACK frame formats and, hence, can be implemented at the device driver level without modifying the device firmware. To the best of our knowledge, this is one of the first efforts in designing and implementing a receiver-based rate adaptation scheme that works with commercial 802.11 devices.
- RAM is an SNR-based scheme. To deal with high SNR fluctuation, RAM adopts a conservative SNR prediction algorithm to avoid over-estimating future SNR values which may cause unnecessary frame losses and retransmissions.
- RAM uses an RTS window to regulate the usage of RTS frames to deal with hidden nodes. Comparing with RTS adaptation in existing rate adaptation schemes, RTS adaptation in RAM is designed based on a thorough examination of all possible transmission outcomes and the RTS window is updated in a timely manner.

B. *Related Work*

Rate adaptation in static environments has been well-studied in the past [2]–[14]. As shown in Table I, these rate adaptation schemes can be classified in the following ways: *transmitter-based* or *receiver-based*; *packet statistics-based* or *SNR-based*; *window-based* or *frame-based*. In *transmitter-based* schemes, the transmitter makes the rate selection decisions without using any feedback from the receiver. By comparison, in *receiver-based* schemes, the receiver monitors the channel quality, makes the rate selection for the next frame transmission and feeds the selection back to the transmitter.

TABLE I
CLASSIFICATION OF EXISTING RATE ADAPTATION SCHEMES

Schemes	Transmitter or receiver-based	Based on SNR	Window or frame-based	Imple-mented	Deal with hidden
ARF [2]/AARF [3]	transmitter	No	window	Yes	No
CARA-like [4], [5]	transmitter	No	window	No	Yes
RRAA [6]	transmitter	No	window	Yes	Yes
SampleRate [7]	transmitter	No	window	Yes	No
CHARM [8]	transmitter	Yes	frame	Yes	No
Scheme in [9]	transmitter	Yes	frame	No	No
SGRA [10]	transmitter	Yes	frame	Yes	No
ONOE [11]	transmitter	No	window	Yes	No
RBAR [12]	receiver	Yes	frame	No	No
OAR [13]	receiver	Yes	frame	No	No
RARA [14]	receiver	Yes	frame	No	No

The research reported in this paper was supported in part by the Information Infrastructure Institute (iCube) of Iowa State University and the National Science Foundation under Grants CNS 0716744 and CNS 0831874.

978-1-4244-3304-9/09/$25.00 ©2009 IEEE

Based on information used to infer the channel condition, rate adaptation schemes can be classified as *packet statistics-based* and *SNR-based*. In packet statistics-based schemes, ARF, AARF (also known as AMRR for its implementation in the Madwifi device driver) and CARA-like schemes use consecutive frame transmission failure and success counts as indicators of the channel quality. RRAA calculates the frame loss ratio and compares it with some thresholds to make rate updating decisions. SampleRate chooses the rate with the shortest expected frame transmission time. In SNR-based schemes, CHARM, the scheme described in [9] and SGRA are transmitter-based. They use the RSSI (Receive Signal Strength Indictor) values of the ACK frames received by the transmitter to infer the channel condition at the receiver side based on the assumption of a symmetric channel. In comparison, RBAR, OAR and RARA are receiver-based and they use the RSSI values of frames received by the receiver instead.

Based on the rate updating period, rate adaptation schemes can be classified as *window-based* and *frame-based*. ARF, AARF and CARA-like schemes use frame transmission failure and success counts and make rate adjustment when the number of frame transmission failures or successes is above a certain threshold. The window sizes for RRAA and SampleRate are 150 ms and 10 seconds respectively by default. Window-based schemes are reactive in nature as they rely on the past history to predict the channel condition in the future. Moreover, it usually is difficult to determine the optimal window size in dynamic environments when the channel condition varies often. By comparison, frame-based schemes adapt much faster to rapid variations of the channel condition that often are caused by fading and mobility.

In Table I, we also list whether a rate adaptation scheme has been implemented. Note that none of the receiver-based schemes has been implemented yet. In fact, whether these receiver-based schemes can be implemented with commercial 802.11 devices is not clear for the following reasons. RBAR and OAR require modifications to the CTS (and possibly RTS) frame formats, which does not conform to the 802.11 standard, while the variation patterns of the ACK transmission rate proposed in RARA are not supported in Madwifi.

In the presence of hidden nodes, it is difficult for the transmitter to differentiate channel-error-induced frame transmission failures from collision-induced ones, which may lead to pessimistic usage of the transmission rates. Adaptive usage of RTS frames has been recognized as an effective way to deal with hidden nodes, and it has been used in a few rate adaptation schemes such as CARA-like schemes and RRAA.

In [15], the authors propose a rate adaptation scheme for vehicular networks based on the context information such as distance and relative velocity. It is a history-based approach and requires repetitive training before usage. It is designed specifically for vehicles traveling along known routes. Therefore, this scheme may not be suitable for dynamic mobile environments where routes are not known *a priori* and the channel condition is unpredictable. In [16], the authors modify SampleRate for mobile environments by reducing the estimation window size.

C. Organization

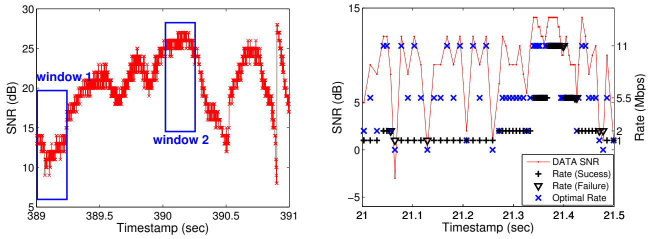
The rest of the paper is organized as follows. A few observations about mobile environments based on experiments are presented in Section II. Section III gives an overview of the proposed RAM scheme and Section IV describes its design and implementation details. Experiment- and simulation-based performance evaluation results are presented in Sections V and VI respectively. The paper concludes in Section VIII.

II. OBSERVATIONS FROM EXPERIMENTS

To design an effective rate adaptation scheme for mobile environments, it is critical to have a good understanding of the characteristics of wireless channels in mobile environments. To do so, we conducted experiments with two laptops equipped with D-Link WNA-1330 802.11b/g cards in various indoor (static and mobile) and outdoor (vehicular) environments. Each laptop is loaded with the Madwifi device driver v0.9.4 to measure and record the channel conditions. In this section, we present the observations and findings from the experiments.

A. Issues with Packet Statistics-based Schemes

1) Window-based rate adaptation schemes: This type of schemes collect packet statistics within a time window (or a window of certain number of packets) and make rate selection decisions at the end of the window. In mobile environments, since the channel condition fluctuates frequently, which will be discussed in Section II-C, packet statistics collected at the current window may become obsolete when making rate selection decisions for future transmission attempts. Fig. 1(a) shows a trace of DATA SNR values in an experimental run for an outdoor vehicular scenario. It can be seen from the figure that it would be too pessimistic or optimistic to use the packet statistics collected from window 1 or window 2 to select rates for future packet transmissions. Another issue with window-based rate adaptation schemes lies in the selection of a proper window size. If the window size is too large, some of the collected information may become outdated at the end of the window, while if it is too small, the collected statistics may not be accurate enough.



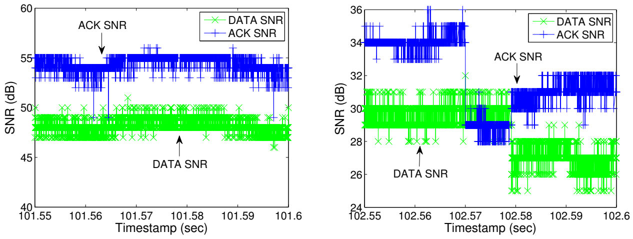
(a) (b)

Fig. 1. Packet statistics-based rate adaption schemes are not suitable for mobile environments. (a) Issues with window-based schemes. (b) Issues with schemes that are based on consecutive frame failure or success counts.

2) Rate adaptation schemes that are based on consecutive frame transmission failure or success counts: ARF and CARA-like schemes use consecutive frame transmission failure or success counts to select the rate for the next transmission attempt. They increase the rate after 10 consecutive transmission successes have been observed and decrease the rate upon two consecutive failures. This type of approaches may not work effectively in mobile environments with high SNR fluctuations. As shown in Fig. 1(b), the success count of 10 may be too conservative for rate increasing, and the failure count of two may be too pessimistic for rate decreasing, which may lead to potential under-utilization of the channel.

B. Severe Channel Asymmetry in Mobile Environments

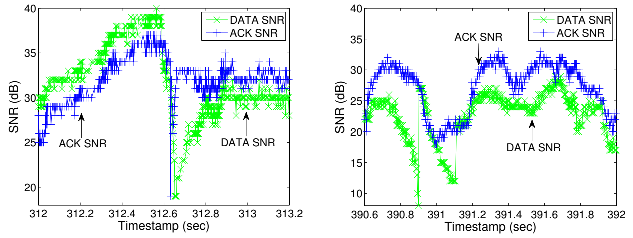
One interesting observation from our experiments is the severe channel asymmetry in practical scenarios. As shown in Figs. 2 and 3, ACK SNR values collected at the transmitter usually differ significantly from DATA SNR values collected at the receiver. The difference is as high as 12 dB in some of the outdoor vehicular scenarios. Since channel symmetry is one of the key assumptions in several existing transmitter-based rate adaptation schemes such as CHARM and SGRA, these schemes may not be suitable for mobile environments. Instead, receiver-based approaches may be a better option.



(a) Indoor static scenario

(b) Indoor mobile scenario

Fig. 2. DATA and ACK SNR differences in indoor scenarios



(a) Outdoor vehicular scenario 1

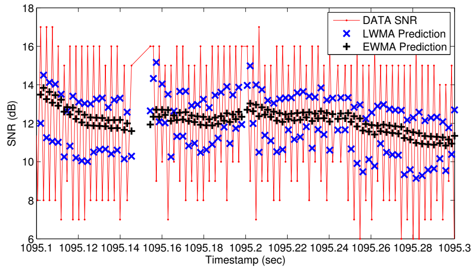
(b) Outdoor vehicular scenario 2

Fig. 3. DATA and ACK SNR differences in outdoor vehicular scenarios

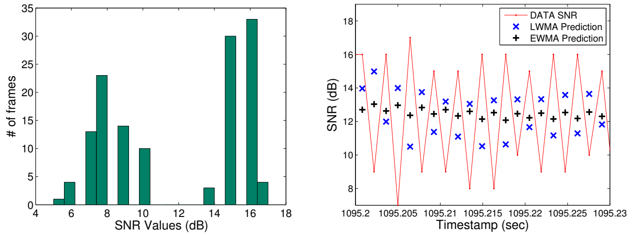
C. High SNR Fluctuation in Mobile Environments

High SNR fluctuation is another important observation from our experiments. In some of the traces such as the one shown in Fig. 4(a), the differences between consecutive SNR values are as large as 10 dB. From the experiments, we notice that high SNR fluctuation usually occurs when the environment suddenly changes, e.g., sudden acceleration of the vehicle, vehicle making a turn, and opening or closing a door of the vehicle. Fig. 4(b) plots the histogram of the SNR values shown in Fig. 4(a) and we can see that the distribution of SNR values is quite irregular.

SNR prediction algorithms used in existing rate adaptation schemes may not be able to handle the high SNR fluctuation properly. As shown in Figs. 4(a) and 4(c), the Light Weighted Moving Average (LWMA) scheme used in CHARM almost always uses the previous SNR value as the prediction for the next SNR value, which results in a large number of over-predictions of SNR values and hence frame transmission failures. Similar problem exists for the simple Exponentially Weighted Moving Average (EWMA) scheme as well.



(a) SNR values and predictions by LWMA and EWMA



(b) Histogram of SNR values

(c) Zoomed view of (a)

Fig. 4. High SNR fluctuation in mobile scenarios

III. OVERVIEW OF THE PROPOSED SCHEME

In the following, we give an overview of the proposed practical rate adaptation scheme, called RAM (Rate Adaptation in Mobile environments), with emphasis on how RAM deals with the issues discussed in the previous section and how RAM may be implemented with commercial 802.11 devices.

1) RAM is a practical receiver-based scheme and can be implemented with commercial 802.11 devices: To deal with channel asymmetry in mobile environments, RAM adopts a receiver-based approach: the receiver makes the rate selection decision for the next frame transmission and feeds such information back to the transmitter. In existing receiver-based rate adaptation schemes, such feedback information usually is conveyed by modifying the format of CTS or ACK frames. Unfortunately, the frame generation process in commercial 802.11 devices is hard-coded in the card firmware and it may need extra efforts to modify it; as a result, such schemes may not be easily implementable with commercial 802.11 devices.[1] Based on this observation, we propose a different approach to convey the feedback information in RAM. Specifically, RAM varies the ACK transmission rate in a controlled manner (via setting different values for a special register using Madwifi) to indicate the rate selection decision for the next frame transmission. This makes RAM practical and implementable with commercial 802.11 devices without modifying the card firmware. Details of this approach will be discussed in Section IV-C.

2) RAM is based on SNR instead of packet statistics: As discussed in Section II-A, packet statistics-based rate adaptation schemes may not work well in mobile environments. Instead, we design RAM to be an SNR-based scheme. Since SNR is a direct measure of the channel condition, RAM

---

[1] The Atheros Hardware Abstraction Layer (HAL) source code was released recently on September 29, 2008. However, according to the Madwifi project [17], the new HAL is not a simple drop-in replacement of the current HAL in Madwifi and it is not yet clear how to integrate the new HAL to make Madwifi a fully open-source driver.

performs well even when the channel condition fluctuates frequently. Moreover, Madwifi reports the SNR value upon each frame reception, which facilitates the deployment and implementation of RAM with commercial 802.11 devices. Details of RAM's rate selection procedure will be discussed in Section IV-B.

3) RAM adopts a conservative SNR prediction algorithm to deal with high SNR fluctuation: To deal with high SNR fluctuation in mobile environments, we propose a conservative SNR prediction algorithm in RAM, which tries to predict the future SNR values as accurately as possible without over-estimating them. Details of this algorithm will be discussed in Section IV-A.

4) RAM uses adaptive RTS to deal with hidden nodes: Adaptive usage of RTS/CTS has been recognized as an effective way to deal with hidden nodes in 802.11 networks. In RAM, we propose a new adaptive RTS approach that uses an RTS window to regulate the usage of RTS frames. Comparing with other adaptive RTS approaches used in existing rate adaptation schemes, our approach is designed based on a thorough examination of all possible transmission outcomes and updates the RTS window in a timely manner. Details of this approach will be discussed in Section IV-E.

IV. DESIGN AND IMPLEMENTATION DETAILS OF RAM

In this section, we describe the design and implementation of the proposed RAM scheme in detail. As shown in Fig. 5, RAM has the following components: at the receiver (i) SNR prediction (ii) rate selection based on SNR prediction (iii) feedback of rate selection to the transmitter; and at the transmitter (i) rate updating and (ii) adaptive usage of RTS.

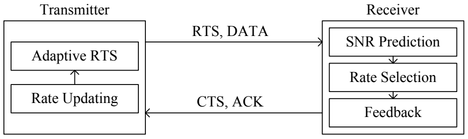


Fig. 5. Overall structure of RAM

A. Receiver: SNR Prediction

To deal with high SNR fluctuation and irregular SNR distribution, we propose a simple conservative SNR prediction algorithm as follows. It maintains the moving averages of the SNR values and the deviations to the average SNR value:

$$\begin{cases} S_{avg} = (1-\delta)\cdot S_{avg} + \delta \cdot S_{curr}, \\ DEV_{avg} = (1-\rho)\cdot DEV_{avg} + \rho\cdot |S_{curr} - S_{avg}|, \end{cases} \tag{1}$$

and predicts the SNR value for the next frame as:

$$S_{est} = S_{avg} - \eta \cdot DEV_{avg}, \tag{2}$$

where S_{curr} is the SNR value reported by Madwifi upon each frame reception[2] and δ, ρ, η are design parameters. We set $\delta = \rho = 0.1$ and $\eta = 1$ in RAM, because from simulations we find that RAM yields better performances when δ and ρ are between 0.1 and 0.3 and η is between 0.5 and 1, and there is no obvious performance variation when the design parameters vary in these ranges.

By considering the deviation of recent SNR values when making the prediction, our algorithm can deal with high SNR fluctuation well. This can be seen from an example shown in Fig. 6 where the predictions by our algorithm follow the lower envelop of the SNR variation closely. This achieves the design goal of predicting future SNR values as accurately as possible without over-estimating them.

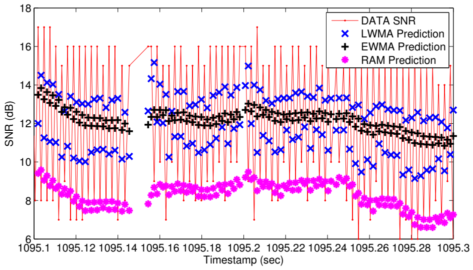


Fig. 6. An example of our SNR prediction algorithm

B. Receiver: Rate Selection based on SNR Prediction

To select the proper rate for the next frame transmission to maximize the throughput, RAM maintains a throughput-vs-(rate, SNR) table. For each (rate = R, SNR = S) pair in the table, we use $G(R, S)$ to denote the expected throughput when the frame is transmitted at rate R and its SNR at the receiver is S, which can be calculated as:

$$G(R, S) = \frac{L(R, S)}{T(R, S)}, \tag{3}$$

where $L(R, S)$ and $T(R, S)$ are the total amount of data received successfully at rate R and SNR of S and the total amount of transmission time for such frames, respectively. The table is updated upon each successfully-received data frame. Based on the predicted SNR value (S_{est}), the receiver looks up the table and selects the rate for the next frame transmission as follows:

$$R^* = \arg\max_R G(R, S_{est}). \tag{4}$$

We implement RAM in the Madwifi device driver, which employs the *multi-rate retry mechanism* to transmit a data frame. In RAM, the receiver is aware of the multi-rate retry mechanism used at the transmitter and may use this information (if needed) to update the throughput-vs-(rate, SNR) table. Before proceeding to the details about how the table is updated, we first give a brief introduction about the multi-rate retry mechanism below.

1) Multi-rate Retry Mechanism: In Madwifi, whenever a frame is ready to send, Madwifi can specify up to four different rates ($r_1 > r_2 > r_3 > r_4$) along with their maximum retry counts (c_i, $i = 1, \cdots, 4$) for the frame and pass these information to the card firmware along with the frame. The frame is discarded after $(c_1 + c_2 + c_3 + c_4)$ unsuccessful transmission attempts, i.e., c_i attempts at the rate of r_i. The card firmware reports the total number of transmission attempts to Madwifi

---

[2] Ideally, the SNR value of each received frame should be used as S_{curr} in the algorithm. Unfortunately, the current Madwifi does not support per-frame-based SNR measurement. It does measure the received signal level for each frame, but only updates the noise level upon each interrupt and usually multiple frames are served between interrupts [18]. Therefore, strictly speaking, the SNR value reported by Madwifi upon each frame reception is *not* the exact SNR value of the frame *but* an approximation to it. Nevertheless, even with such limitation of the current Madwifi, RAM still yields a noticeable performance improvement over existing rate adaptation schemes, which will be shown in later sections via both experiments and simulations.

after the frame has been transmitted successfully or discarded. In RAM, we set $c_1 = 4$, $c_2 = c_3 = 2$, and $c_4 = 2$. The reason for setting a larger retry count ($c_1 = 4$) for the first rate r_1 is because RAM adopts a conservative SNR prediction algorithm and hence r_1 usually is selected conservatively.

2) Updating the Throughput-vs-(Rate, SNR) Table: When the receiver receives a data frame successfully, there are two possible outcomes for each of the unsuccessful transmission attempts (if any) prior to the successful reception of the frame: *frame was corrupted but header can be retrieved by the receiver*, or *frame was completely lost*. For the former case, the information about the unsuccessful transmission attempt such as the transmission rate and the SNR of the frame are reported to Madwifi by the card firmware, while for the latter case, these information are not available. Note that the SNR values reported by the card firmware are integer values, meaning that the number of meaningful entries in the throughput-vs-(rate, SNR) table is a small finite number. There are two ways to update the throughput-vs-(rate, SNR) table: *basic update* which assumes that all unsuccessful attempts belong to the former case, and *delayed update* to handle both cases.

- *Basic Update:* Suppose a data frame is received successfully during the k-th attempt at rate R_k and SNR of S_k, where $1 \leqslant k \leqslant (c_1 + c_2 + c_3 + c_4)$. For each of the unsuccessful attempts, the receiver retrieves the rate and SNR information reported by the card firmware: R_m and S_m where $1 \leqslant m < k$. For each pair of (R_j, S_j) where $1 \leqslant j \leqslant k$, L and T are updated as follows:

$$\begin{cases} L(R_j, S_j) = L(R_j, S_j), & \text{for } 1 \leqslant j < k, \\ L(R_j, S_j) = L(R_j, S_j) + \textit{data\_payload}, & \text{for } j = k. \end{cases} \quad (5)$$

  and

$$T(R_j, S_j) = T(R_j, S_j) + \textit{txtime}(R_j) + \textit{backoff}(j), \quad \text{for } 1 \leqslant j \leqslant k, \quad (6)$$

  where *data_payload* is the payload of the frame, *txtime*(R_j) is the frame transmission duration at rate R_j, and *backoff*(j) is the average backoff time prior to the j-th transmission attempt, which is given by:

$$\textit{backoff}(j) = \min\left[\frac{(\mathrm{CW}_{\min} + 1) \cdot 2^{j-1} - 1}{2},\ \frac{\mathrm{CW}_{\max}}{2}\right] \times \text{aSlotTime}. \quad (7)$$

  Subsequently, the throughput is updated as:

$$G(R_j, S_j) = \frac{L(R_j, S_j)}{T(R_j, S_j)}, \quad \text{for } 1 \leqslant j \leqslant k. \quad (8)$$

- *Delayed Update:* To handle the unsuccessful attempts during which the frame was completely lost, we propose to piggyback the total retry count of the *previous* frame in the data payload of the *current* frame. Since the receiver is aware of the multi-rate retry mechanism used by the transmitter, upon a successful reception of the current frame, the receiver can deduce the transmission rates of all the unsuccessful attempts for the previous frame. Using the most recently predicted SNR to approximate the SNR for all the unsuccessful attempts that were completely lost for the previous frame, the receiver can then update the throughput-vs-(rate, SNR) table in a similar way as in *basic update*. Note that the table is updated using the information about the previous frame upon successful reception of the current frame. This is why this procedure is called *delayed update*.

The current version of RAM implements the basic update procedure. Both update procedures will be implemented in the full version of RAM which we are currently working on. Fortunately, our experiments show that the *"frame was completely lost"* case rarely occurs in practice and the current version of RAM works fine as will be shown in the performance evaluation sections.

C. Receiver: Feedback of Rate Selection to the Transmitter

The 802.11 standard [19] specifies that the an ACK frame should be transmitted at the highest rate in the basic rate set that is less than or equal to the transmission rate of the data frame it is acknowledging. We call such ACK transmission rate the default ACK rate. For example, the 802.11g basic rate set is $\{1,\ 2,\ 5.5,\ 11,\ 6,\ 12,\ 24\}$ Mbps. So if a data frame is transmitted at 18 Mbps, the default rate of the corresponding ACK frame is 12 Mbps. In practice, Madwifi allows two different transmission rates for ACK frames, as listed in Table II for Atheros chipset-based 802.11g cards. Madwifi can specify that an ACK frame is transmitted at a low rate or a high rate (the default rate) via setting different values for a special register [20].

TABLE II

IN MADWIFI: TWO TRANSMISSION RATES AVAILABLE FOR ACK FRAMES

data rate (Mbps)	1	2	5.5	11	6	9	12	18	24	36	48	54
low ACK rate	1	2	2	2	6	6	6	6	6	6	6	6
high ACK rate	1	2	5.5	11	6	6	12	12	24	24	24	24

RAM takes advantage of this Madwifi feature and conveys the feedback information implicitly via the ACK transmission rate variation. Specifically, if the receiver wants to inform the transmitter to transmit the next frame at the same rate as the previously successfully transmitted frame, or at the next higher rate, it transmits the ACK frame at the default high rate or at a low rate, respectively. For example, if the receiver receives a data frame successfully at 36 Mbps, it can signal the transmitter to send the next frame at 36 or 48 Mbps by transmitting the ACK frame at 24 or 6 Mbps, respectively.

Note that for rates of 1, 2, 6, and 9 Mbps, there is only one option for the ACK transmission rate. In RAM, we disable the data transmission rates of 6 and 9 Mbps since it has been observed from experiments that the throughput performances of 6 and 9 Mbps are worse than that of 5.5 Mbps [8]. For rates of 1 or 2 Mbps, rate increasing decisions are made at the transmitter side. Moreover, rate decreasing decisions also are made at the transmitter side. These will be explained in the next section.

D. Transmitter: Updating the Transmission Rate

In Madwifi, the transmitter employs the multi-rate retry mechanism to transmit a data frame. In RAM, the parameters in the multi-rate retry mechanism are set to $c_1 = 4$, $c_2 = c_3 = 2$, $c_4 = 2$, and r_{i+1} is the next lower rate to r_i ($i = 1, 2, 3$). Therefore, once r_1 is decided for a data frame, the multi-rate retry mechanism for the frame is decided. In RAM, we decide r_1 for the next data frame according to the transmission result of the *last* attempt (suppose at the rate of R_{last}) of the previous data frame:

1) If it fails: the transmitter sets r_1 in the multi-rate retry mechanism for the next frame to $r_1 = R_{\text{last}}$.

2) If it succeeds: the transmitter may take the following actions depending on R_{last}:

- If $R_{\text{last}} > 2$ Mbps, the transmitter relies on the feedback from the receiver to set the rate for the next frame. Specifically, if the transmitter receives an ACK frame at the default high rate, it sets r_1 to R_{last}; otherwise, r_1 is set to the next higher rate to R_{last}.
- If $R_{\text{last}} = 1$ or 2 Mbps, the transmitter makes the rate updating decision using the following heuristic. In RAM, the transmitter keeps track of the ACK SNR values. When the current ACK SNR is 5 dB larger than the previous one or when the number of consecutive frame transmission successes is larger than four, r_1 is set to the next higher rate to R_{last}. In an extreme case when $R_{\text{last}} = 1$ Mbps and the current ACK SNR is 9 dB larger than the previous one, r_1 is increased to 5.5 Mbps directly. These thresholds are obtained from the experiments.

By default, Madwifi uses the high ACK rate to calculate the NAV value for a data frame transmission. In RAM, since the receiver may transmit an ACK frame at the low rate to signal rate increasing for the next data frame, we modify the NAV calculation in Madwifi by using the low ACK rate intead. This can be done by modifying the value of a special register [20]. Since ACK frames are short, such modification does not affect the performance much, as will be discussed in Section VII. In addition, since the RAM receiver uses a moving average to update the SNR estimation and feeds back the rate selection decisions to the transmitter on a per-frame basis, the RAM transmitter converges quickly (usually a few frames) to the proper transmission rate in situations when the interval between two frame transmissions is large.

E. Transmitter: Adaptive Usage of RTS

Adaptive usage of RTS/CTS has been recognized as an effective way to deal with hidden nodes in 802.11 networks and it has been used in several rate adaptation schemes such as CARA-like schemes and RRAA. We propose an advanced adaptive RTS scheme in RAM. Similar to the one used in RRAA, our adaptive RTS scheme uses an RTS window (with the size of RTSWnd) to regulate the usage of RTS frames. All data frames within the RTS window shall be transmitted with RTS/CTS support. Moreover, our scheme examines all possible transmission outcomes thoroughly and updates RTSWnd in a timely manner as follows.

Table III lists two ways of attempting a data frame transmission (i.e., with or without RTS/CTS), possible outcomes of each attempt, and the corresponding actions on updating RTSWnd. Initially, RTSWnd is set to zero to disable RTS usage. The basic heuristic behind our adaptive RTS scheme is that RTSWnd should increase more quickly if a shorter frame is lost (which implies a higher collision probability), and decrease more quickly if a longer frame succeeds (which implies a lower collision probability). Since an RTS frame is short, an RTS failure indicates that the collision problem may be severe. Hence, we multiply RTSWnd by three. If an RTS transmission succeeds, we decrease RTSWnd slowly $(\text{RTSWnd} = \text{RTSWnd} - 1)$ regardless whether the subsequent data transmission succeeds or not. This is because an RTS/CTS exchange has already reserved the channel and reduces the probability of collision to the subsequent data transmission. On the other hand, without RTS/CTS support, a successful data transmission (usually with a long transmission duration) indicates a small chance of collision and hence we decrease RTSWnd by half. When a data transmission fails with no preceding RTS, the cause of the failure is not clear. In this situation, we increase RTSWnd slowly $(\text{RTSWnd} = \text{RTSWnd}{+}1)$. We also set a maximum value of 32 for RTSWnd to guarantee stable performance.

TABLE III
DATA TRANSMISSION ATTEMPTS, OUTCOMES, AND CORRESPONDING ACTIONS ON UPDATING RTSWND

Data Frame Transmission Attempt	Outcome of Transmission Attempt	Action on Updating RTSWnd
DATA with RTS	RTS Fail	$\text{RTSWnd} = 3 \times \text{RTSWnd}$
	RTS Succ, DATA Fail	$\text{RTSWnd} = \text{RTSWnd} - 1$
	RTS Succ, DATA Succ	
DATA without RTS	DATA Fail	$\text{RTSWnd} = \text{RTSWnd} + 1$
	DATA Succ	$\text{RTSWnd} = \text{RTSWnd}/2$

Note that in Madwifi, it is impossible to control the RTS usage on a per-transmission-attempt basis. Therefore, when we implement RAM in Madwifi, the RTS usage is controlled on a per-frame basis. In other words, the transmitter updates RTSWnd when it receives the report from card firmware after the frame has been transmitted successfully or discarded.

V. EXPERIMENTAL STUDY

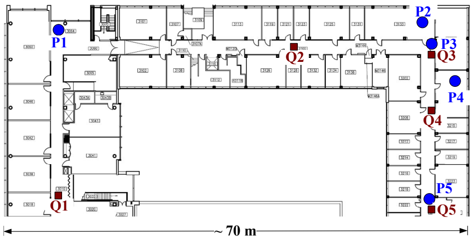
We have implemented all the RAM modules described in Section IV in Madwifi except adaptive RTS. We call this version of the RAM implementation RAM-BASIC. Implementation of the full version of RAM is under progress and will be completed in the near future. In this section, we evaluate the effectiveness of RAM-BASIC using experimental results.

A. Experimental Setup

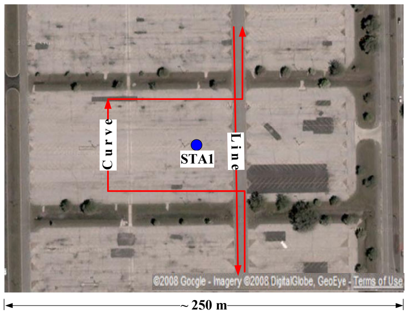
The hardware and software configurations used in our experiments are listed in Table IV. All experiments are performed between two Dell Latitude D620 laptops equipped with D-Link WNA-1330 802.11b/g WLAN adaptors, which embed Atheros 5005G chipsets. We use Iperf [21] as the UDP packet generator. CBR (Constant Bit Rate) traffic is generated at 30 Mbps with packet size of 1470 octets. The results for each scenario are averaged over five experimental runs. In order to minimize potential unexpected performance variation caused by people's movement and interference from other 802.11 devices, indoor experiments are conducted at nighttime

TABLE IV
CONFIGURATION PARAMETERS

Parameters	Values
Computer	Dell Latitude D620 Laptop
Operating system	Linux Kernel 2.6.24-16
WLAN adaptor	D-Link WNA-1330
Device driver	Madwifi v0.9.4
802.11 PHY	802.11g
Transmit Power	14 dBm
CBR packet size	1470 octets
CBR rate	30 Mbps



(a) Venue for indoor experiments: 3rd floor of Coover Hall



(b) Venue for outdoor experiments: a parking lot near Jack Trice football stadium

Fig. 7. Venues for our indoor and outdoor experiments

TABLE V
DESCRIPTION OF EXPERIMENTAL SCENARIOS

Scenarios	Descriptions
Static-1	STA1 at P3, STA2 at P1
Static-2	STA1 at P3, STA2 at P5
Static-3	STA1 at P4, STA2 at P5
Static-4	STA1 at P2, STA2 at P4
Walk-1	STA1 at P3, STA2: Q2→Q1→Q2
Walk-2	STA1 at P3, STA2: Q4→Q5→Q4
Walk-3	STA1 at P4, STA2: Q3→Q5→Q3
Walk-4	STA1 at P2, STA2: Q4→Q2→Q4
SlowDrive-1	STA1 is static, STA2 moves along the line up to 20 MPH
SlowDrive-2	STA1 is static, STA2 moves along the curve up to 20 MPH
FastDrive-1	STA1 is static, STA2 moves along the line up to 35 MPH
FastDrive-2	STA1 is static, STA2 moves along the curve up to 35 MPH

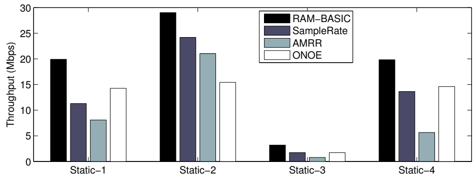
or weekends and outdoor experiments are conducted in the afternoon during weekends.

We conduct experiments in both static and mobile scenarios. Indoor experiments (static and mobile) are performed on the 3rd floor of Coover Hall (our department building), as shown in Fig. 7(a), and outdoor vehicular experiments are performed in a parking lot near Jack Trice football stadium, as shown in Fig. 7(b). We mark several locations and moving trajectories on the figures, based on which we design 12 different experimental scenarios, as described in Table V.

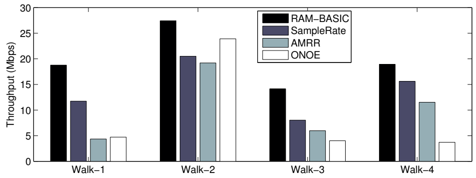
We compare the throughput performance of RAM-BASIC against three rate adaptation schemes that have been implemented in Madwifi: SampleRate, AMRR and ONOE. Note that AMRR is the Madwifi version of AARF which is an adaptive variant of the well-known ARF scheme. SampleRate tries to maximize the throughput by estimating per-frame transmission time at each rate and selecting the transmission rate with the lowest expected per-frame transmission time. ONOE is less sensitive to individual frame failures than ARF and its variants. It basically tries to find the highest rate that has less than 50% frame loss ratio.

B. Experimental Results

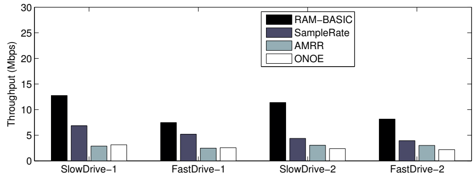
Experimental results are shown in Fig. 8. It is clear that RAM-BASIC outperforms other testing schemes in all experimental scenarios, indoor or outdoor, static or mobile, and the performance gain becomes more significant as the relative speed of two stations goes up. This is because RAM-BASIC is receiver-based. By using the feedback from the receiver, the transmitter can select the proper transmission rate to match the current channel condition. Moreover, RAM-BASIC is frame-based, which can adapt much faster to rapid variations of the channel condition. In comparison, SampleRate, AMRR and ONOE are transmitter-based schemes and based on packet statistics. As a result, they usually are slow in adapting to the channel variation.



(a) indoor static scenarios



(b) indoor mobile scenarios



(c) outdoor vehicular scenarios

Fig. 8. Comparison of experimental results

In most experimental scenarios, AMRR and ONOE perform the worst. As introduced in the previous section, AMRR is an adaptive variant of the well-known ARF scheme. ARF waits for 10 consecutive successes before increasing the rate while AMRR adapts this threshold by using a binary exponential backoff starting with 10. Unfortunately, from our experiments, we find that channel fluctuation is common in practice, even in indoor static environments. So in the presence of channel fluctuation, it is rare to have 10 consecutive frames transmitted successfully. As a result, AMRR almost always chooses a large threshold when making rate increasing decisions, and hence is very slow in increasing the transmission rate when the channel condition gets better. Similarly, ONOE is also a conservative rate adaptation scheme by design: it increases the transmission rate at most once during any one-second period.

VI. SIMULATION STUDY

In this section, we use the ns-2 simulator [22] to further evaluate the performances of RAM-BASIC and the full version of RAM (referred to as RAM) that includes adaptive RTS.

A. Simulation Setup

Instead of using the propagation model given in the ns-2 simulator, we import the SNR traces from our experiments unless specified otherwise. To import the SNR traces, we use a timestamp-based approach. Basically, according to the timestamp of a packet, we set its SNR value based on the collected traces. Moreover, we use the empirical PDR (Packet Delivery Rate) vs. SNR curves (obtained from 10 experimental traces with each lasting for about 10 minutes) instead of the simple 0/1 packet delivery model given in the ns-2 simulator. For the clarity of presentation and explanation, we assume the 802.11b PHY in the simulations. Simulations for other 802.11 PHYs yield similar results.

We compare the throughput performances of RAM-BASIC and RAM against several existing rate adaptation schemes that have not been implemented in Madwifi or whose source codes are not available: CHARM, CARA and RRAA. We also evaluate their performances in a hidden nodes scenario. In the simulations, each transmitter transmits in a greedy mode, i.e., its data queue is never empty, and all data frames are transmitted without fragmentation. We use LLC/IP/UDP as the upper layer protocol suite, and the MAC-layer data payload length is 1500 octets.

B. Simulation Results

1) Indoor Static Scenarios: We first simulate indoor static scenarios and results are plotted in Fig. 9. It can be seen that both RAM-BASIC and RAM show comparable or better performances than other testing schemes in all four scenarios. RAM-BASIC and RAM yield similar performances. As will be shown in Sections VI-B2 and VI-B3, their performances are similar as long as there are no hidden nodes in the network. This is because without hidden nodes, the RTS window of RAM is zero for most of the time, meaning that RTS usage is disabled for most of the time and hence RAM is almost equivalent to RAM-BASIC. For the Static-4 scenario, we observe that the performance of CHARM is significantly worse than others. This is because CHARM is designed based on the assumption of symmetric channel conditions, which does not hold in Static-4. From the SNR traces collected in Static-4, we notice that the channel conditions exhibit severe asymmetry and the difference between DATA SNR and ACK SNR may be as large as 12 dB.

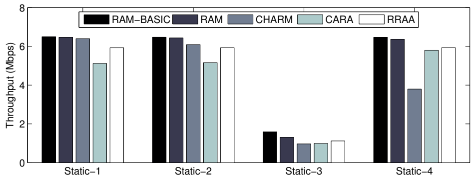
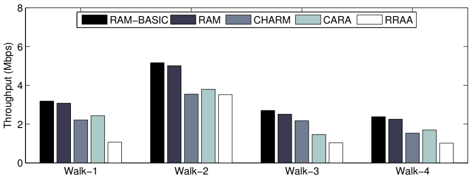


Fig. 9. Simulation results for indoor static scenarios

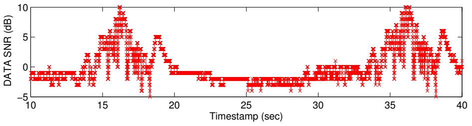
2) Indoor Mobile Scenarios: Simulation results for indoor mobile scenarios are shown in Fig. 10. Throughput comparison is shown in Fig. 10(a). Similar to indoor static environments, RAM-BASIC and RAM yield higher throughput than others for all indoor mobile scenarios. Again, CHARM's performance deficit is due to its inability to handle channel asymmetry, which may cause unnecessary frame losses (as a result of over-estimating the channel condition) or under-utilization of the good channel condition (as a result of under-estimating the channel condition). CARA outperforms RRAA mainly because CARA has a relatively smaller rate adjustment window than RRAA.

In order to have a good understanding on how and why RAM-BASIC and RAM outperform other testing schemes, we investigate the Walk-1 scenario in more depth and study the cause of the observed throughput differences by plotting the instant throughput and the rate usage distribution for each scheme in Figs. 10(c) and (d), respectively. Since the plots for RAM-BASIC and RAM are very similar to each other when there are no hidden nodes in the network, we only show the results of RAM in these figures. Fig. 10(b) plots the trace of SNR values for Walk-1.

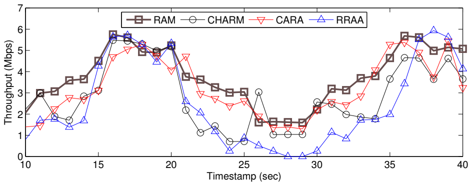
Each point in Fig. 10(c) represents the throughput measured for the one-second period ending at the corresponding time instance, called *instant throughput*, which reflects the rate selections during that one-second period. As shown in the figure, RAM almost always yield higher instant throughput than others except for a very few time instances. RRAA's slow rate adjustment to channel variation can be seen clearly from the figure: slow rate decreasing between 20 and 25 seconds and slow rate increasing between 30 and 35 seconds. In comparison, CARA's rate adjustment tracks the SNR variation fairly well because of its small rate adjustment window, which results in better overall throughput performance than RRAA. It also can be seen from the figure that rate adjustment by



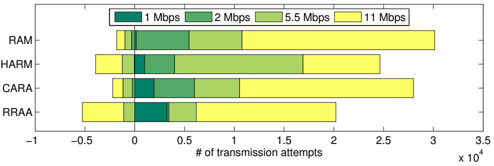
(a) Throughput comparison



(b) SNR variation in Walk-1



(c) Instant throughput comparison for Walk-1. Each point represents the throughput measured for the one-second period ending at the corresponding time instance.



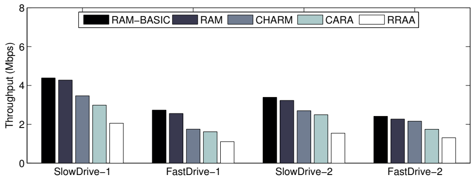
(d) Comparison of rate usage distributions for Walk-1. The number of successful or failed transmission attempts is shown as a positive or negative bar.

Fig. 10. Simulation results for indoor mobile scenarios

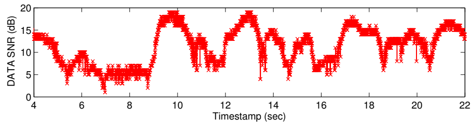
CHARM does not match well with the SNR variation of data frames due to its assumption on a symmetric channel.

In Fig. 10(d), the numbers of successful or failed transmission attempts at different transmission rates are shown as positive or negative bars of different colors. As shown in the figure, RAM makes effective usage of the available transmission rates: (i) a majority of the successfully transmitted frames are attempted at the higher rates of 5.5 or 11 Mbps; (ii) a very few frames are transmitted at the lowest rate of 1 Mbps; and (iii) the frame loss ratio is low (5.67%). In comparison, both CHARM and RRAA suffer a much higher frame loss ratio at 13.75% and 20.64% respectively, while RRAA transmits a significant portion (∼12.78%) of the frames at 1 Mbps. On the other hand, the rate usage distribution for CARA is similar to RAM, which conforms to their comparable throughput performances shown in Fig. 10(a).

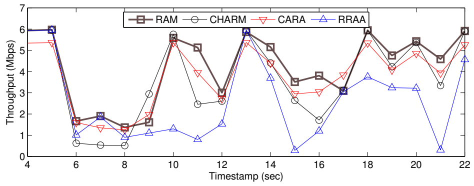
3) Outdoor Vehicular Scenarios: In general, simulation results shown in Fig. 11 for outdoor vehicular environments are similar to those shown in Fig. 10 for indoor mobile environments. In addition to the similar observations discussed in the previous section, we have a few more observations as follows. Firstly, as shown in Fig. 11(b), the channel condition for outdoor vehicular environments (with faster station movement) fluctuates more frequently and at a larger scale than indoor mobile environments (with slower station movement). Secondly, as shown in Fig. 11(d), CHARM experiences an even higher frame loss ratio (22.44% for SlowDrive-1 in comparison to 13.75% for Walk-1). This implies that channel asymmetry is more severe in outdoor vehicular environments. Thirdly, in spite of the high fluctuation of channel conditions and the severe channel asymmetry in outdoor vehicular environments, RAM continues to perform well, thanks to its receiver-based rate selection and conservative SNR prediction. As shown in Fig. 11(d), for SlowDrive-1, around 84.33% of the frames are transmitted successfully at 5.5 or 11 Mbps while the frame loss ratio is less than 5%.



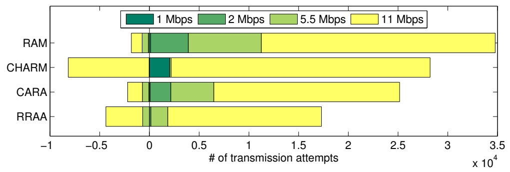
(a) Throughput comparison



(b) SNR variation in SlowDrive-1



(c) Instant throughput comparison for SlowDrive-1



(d) Comparison of rate usage distributions for SlowDrive-1

Fig. 11. Simulation results for outdoor vehicular scenarios

4) Hidden Nodes Scenario: The last part of the simulation study is to investigate the performances of testing schemes in the presence of hidden nodes. Since it is difficult to set up an ideal hidden nodes scenario in the experiments for collecting the SNR traces, we use the models provided in the ns-2 simulator in this simulation. The simulated hidden nodes scenario is shown in Fig. 12, where two transmitters are located at opposite sides of the receiver and they are hidden to each other during the entire trajectory. Both transmitters move at 5 m/s and they start moving at the same time. We simulate a Ricean fading channel with a K-factor of 6 dB and assume a maximum speed of 10 m/s for movement in the environment.

Throughputs of testing schemes are compared in Fig. 13(a). As expected, schemes with RTS capabilities such as RAM, ARF+RTS, CARA and RRAA can deal with hidden nodes well and yield higher throughput. Among them, RAM has the best performance. In comparison, schemes without RTS capabilities suffer significant performance degradation, including RAM-BASIC, CHARM, ARF and SampleRate. ARF performs particularly bad because it cannot differentiate collision-induced losses from channel-error-induced losses and hence transmits at a very low rate. Note that CHARM yields comparable throughput as RAM-BASIC. This is because we use the Ricean fading model to simulate a perfect symmetric channel in the simulation instead of using the DATA and

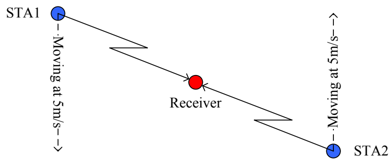
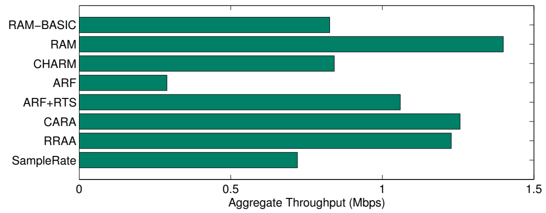
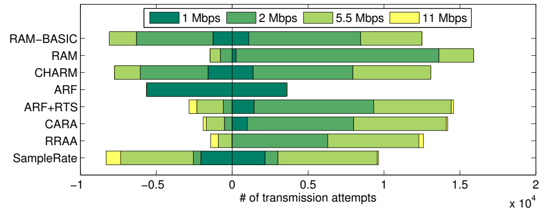


Fig. 12. The simulated hidden nodes scenario



(a) Throughput comparison



(b) Comparison of rate usage distributions

Fig. 13. Simulation results for the hidden nodes scenario

ACK SNR traces collected from experiments which usually are asymmetric. We plot the rate usage distributions of testing schemes in Fig. 13(b). The high frame loss rate caused by hidden nodes can be seen in the figure for RAM-BASIC, CHARM, ARF and SampleRate. Among them, ARF has the highest frame loss ratio of 61%. In comparison, when RTS is used to deal with hidden nodes, the frame loss ratio is reduced drastically for RAM, ARF+RTS, CARA and RRAA. RAM has the largest number of successful transmission attempts and the smallest frame loss ratio, which explains its best throughput performance.

VII. DISCUSSION AND FUTURE WORK

In this section, we discuss a few possible directions to improve RAM further.

A. Interoperability between RAM and Non-RAM Stations

1) non-RAM Transmitter and RAM Receiver: The RAM receiver could check the duration value of the received frame to verify whether the transmitter is a RAM transmitter. If the duration value is calculated using the low ACK rate, the receiver considers the transmitter as a RAM transmitter and replies ACK at the rate determined by the RAM scheme. Otherwise, the receiver considers it as a non-RAM transmitter and replies ACK at the default high rate.

2) RAM Transmitter and non-RAM Receiver: In this case, since the non-RAM receiver always replies ACK at the default high rate, the RAM transmitter will never get the feedback to increase the transmission rate. To deal with this issue, RAM could use a hybrid scheme at the transmitter side as follows. If the RAM transmitter always receives ACKs at the default rate, it considers the receiver as a non-RAM receiver and then switches to a transmitter-based rate adaptation scheme.

B. Effects of ACK Rate Variation on the System Performance

Since we set the duration field using the low ACK rate, when the receiver replies ACK at the high rate, the channel will be idle for a short period of time. To verify that this mechanism will not cause noticeable performance degradation, we simulate multiple Tx-Rx pairs transmitting simultaneously and stations are randomly distributed within a circle with a radius of 40m. Simulation parameters are similar to those in Section VI-B4. We compare RAM with another scheme called Ideal-RAM in terms of the system throughput. Ideal-RAM operates in the same way as RAM except that Ideal-RAM does not use the ACK rate variation to convey the feedback information; rather, the rate selections by the receiver are made available to the transmitter by modifying the ns-2 simulator. So Ideal-RAM is only possible with the simulator but not implementable in practice. We vary the number of Tx-Rx pairs and simulation results (averaged over 20 simulation runs) are shown in Table VI. We can see that even when the network is highly loaded with 16 Tx-Rx pairs, the ACK rate variation used in RAM only results in a small 2.4% degradation of the system throughput.

TABLE VI
THROUGHPUT COMPARISON (in Mbps) WITH MULTIPLE SIMULTANEOUS TX-RX TRANSMISSION PAIRS

# Tx-Rx Pairs	1	2	4	8	16
Ideal-RAM	4.5828	4.0640	3.9815	3.8752	3.1668
RAM	4.5822	3.9989	3.9555	3.7992	3.0887

C. Possible Improvement on the Table Updating Scheme

The current Throughput-vs-(Rate, SNR) table updating scheme in RAM works fine when the interference is weak or stable. However, it may not respond quickly in the presence of bursty strong interferences in the network. This scheme could be improved to deal with channel dynamics better by using a window-based approach or a moving average-based approach to limit the impact of outdated data.

VIII. CONCLUSIONS

From our experiments in indoor static and mobile environments and outdoor vehicular environments, we find that conditions of wireless channels in mobile environments exhibit severe asymmetry and high fluctuation. To deal with these issues, we propose a practical rate adaptation scheme, called RAM, for 802.11 devices in mobile environments. We have implemented the basic version of RAM in the Madwifi device driver and the implementation of the full version will be completed in the near future. Experimental and simulation results show that RAM is able to deal with channel symmetry and adapts quickly to the channel variation. It outperforms existing rate adaptation schemes in both static and mobile environments, particularly in outdoor vehicular scenarios.

REFERENCES

[1] Multiband Atheros Driver for Wifi, http://www.madwifi.org/.

[2] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless lan for the unlicensed band," in *Bell Labs Technical Journal*, vol. 2, no. 3, August 1997, pp. 118–133.

[3] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *ACM MSWiM '04*, 2004.

[4] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware rate adaptation for ieee 802.11 wlans," in *IEEE InfoCom'06*, 2006.

[5] S. Kim, S. Choi, D. Qiao, and J. Kim, "Enhanced rate adaptation schemes with collision awareness," in *IFIP Networking'07*, 2007.

[6] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM MobiCom'06*, 2006.

[7] J. Bicket, "Bit-rate selection in wireless networks," Master's thesis, MIT, 2005.

[8] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *ACM MobiSys'08*, 2008.

[9] J. del Prado Pavon and S. Choi, "Link adaptation strategy for ieee 802.11 wlan via received signal strength measurement," in *IEEE ICC'03*.

[10] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, "A practical SNR-guided rate adaptation," in *IEEE InfoCom Miniconference'08*, 2008.

[11] Onoe Rate Control, http://www.madwifi.org/browser/trunk/ath_rate/onoe.

[12] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *ACM MobiCom'01*, 2001, pp. 236–251.

[13] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *ACM MobiCom'02*.

[14] H. Jung, K. Cho, Y. Seok, T. Kwo, and Y. Choi, "RARA: Rate adaptation using rate-adaptive acknowledgment for ieee 802.11 wlans," in *IEEE CCNC'08*, 2008, pp. 62–66.

[15] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode, "CARS: Context aware rate selection for vehicular networks," in *IEEE ICNP'08*.

[16] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *ACM MobiSys'07*.

[17] Atheros Madwifi HAL Source Code Released, http://madwifi-project.org/wiki/news/20080929/hal-source-code-released.

[18] RSSI in Mafwifi, http://madwifi-project.org/wiki/UserDocs/RSSI.

[19] IEEE 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Standard, IEEE, Aug. 1999.

[20] Atheros Registers, http://madwifi.org/wiki/DevDocs/AtherosRegisters.

[21] Iperf, http://dast.nlanr.net/projects/Iperf.

[22] Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/.