

# EARC: Enhanced Adaptation of Link Rate and Contention Window for IEEE 802.11 Multi-Rate Wireless Networks

Ting-Yu Lin, Ching-Yi Tsai, and Kun-Ru Wu

**Abstract**—IEEE 802.11 wireless network supports multiple link rates at the physical layer. Each link rate is associated with a certain required Signal-to-Interference-and-Noise Ratio (*SINR*) threshold for successfully decoding received packets. On transmission failures, the 802.11 DCF performs a binary exponential backoff mechanism to discourage channel access attempts, hoping to reduce congestion. When traditional link adaptation is applied, both rate reduction and binary backoff represent double penalties for this wireless link, which may cause overly conservative transmission attempts. On the other hand, once transmission succeeds, 802.11 DCF resets the backoff contention window to the minimum value to encourage channel access attempts. At the same time, traditional link adaptation may also decide to increase the data rate, which leads to overly aggressive transmission attempts. We observe this improper interaction of link rate and backoff mechanism that harms the 802.11 system performance, due to separate consideration of those two parameters. In this paper, we propose to jointly adapt the rate and backoff parameters. Specifically, an Enhanced Adaptation of link Rate and Contention window, abbreviated as EARC, is devised. EARC is a closed-loop (receiver-assisted) link rate adaptation protocol that jointly considers the backoff mechanism. With only one extra byte carried by the DATA packet, EARC incurs little controlling overhead despite its receiver-assisted nature. Moreover, since *SINR* information commonly utilized by receiver-assisted protocols is not precisely supported in real devices, we introduce a rate selection reference (RSR) table empirically derived by constantly monitoring the environmental energy level and reception behavior. The RSR table then guides the receiver to select the best sustainable rate for the transmitter. Simulation results demonstrate the RSR table is a practical option for making the rate decision, and the proposed EARC approach is effective in maintaining high system throughput, compared to other link adaptation algorithms.

**Index Terms**—Link adaptation, contention resolution, ARF, BEB, IEEE 802.11, multi-rate.

## I. BACKGROUND

IEEE 802.11 plays an important role in wireless communications. Due to the development of various modulation techniques and coding schemes, multiple transmission rates are now supported by 802.11 physical layers. For example,

Paper approved by B. Sikdar, the Editor for Wireless Packet Access and Cross-Layer Design of the IEEE Communications Society. Manuscript received March 6, 2011; revised January 9 and February 15, 2012.

The authors are with the Institute of Communications Engineering, National Chiao Tung University (e-mail: ting@cm.nctu.edu.tw). T.Y. Lin is the corresponding author.

This research was co-sponsored in part by the NSC of Taiwan under grant number 99-2221-E-009-006, and in part by the MoE Program Aiming for the Top University and Elite Research Center Development Plan (ATU Plan).

Digital Object Identifier 10.1109/TCOMM.2012.071312.110145

802.11b provides 4 data rates (1, 2, 5.5 and 11 Mbps), while 802.11a/g provides 8 rate options (6, 9, 12, 18, 24, 36, 48 and 54 Mbps). Higher transmission rate means higher potential throughput, because it shortens the transmission time in one transmission attempt. However, higher data rate also implies higher packet corruption probability for receiver requires higher Signal-to-Interference-and-Noise Ratio (*SINR*) to successfully decode packets. If the *SINR* perceived at the receiver is lower than required *SINR* threshold, the signal may not be decoded correctly.

Each data rate is associated with a certain *SINR* threshold. The method of selecting an appropriate link rate for transmitting/retransmitting packets is generally comprehended as the link (rate) adaptation mechanism. Various rate-adaptive algorithms have been proposed [1]–[14]. The most commonly used rate adaptation technique is perhaps auto-rate fallback (ARF), which is widely implemented in present wireless devices [6]. Based on ARF, in the literature, plenty of rate-adaptive mechanisms have been proposed to improve the ARF performance [1], [3], [5], [7], [8], [10], [12], [15], [16]. Rate adaptation can also be combined with tuning other physical parameters such as power or carrier sense threshold [1], [9].

In general, rate-adaptive schemes can be classified into two categories: open-loop and closed-loop approaches. Open-loop approaches perform rate adaptations based on the information of whether ACK message is successfully returned or not, which we call *implicit feedback*. ARF is such an open-loop strategy that makes the rate adaptation decision based on ACK information [6]. Adaptive Thresholds (AT) mechanism is an enhanced version of ARF, which dynamically tunes the rate upshift and downshift parameters based on link-layer measurements [15]. On the other hand, closed-loop approaches require the receiver to gather extra information such as *SINR* statistic and inform the sender via control messages, called *explicit feedback*. Consequently, closed-loop approaches may result in better rate predictions, at the expense of controlling overhead. A representative mechanism in this category is the receiver-based auto-rate (RBAR) protocol [4]. ARF with COLLIE (AC) is another closed-loop approach, which tries to improve the ARF performance by distinguishing collision-based packet losses from packet failures due to weak signal according to metrics provided by the receiver [16]. The above rate adaptation mechanisms do not take the 802.11 backoff algorithm into consideration. When the rate and backoff contention window (CW) adjustments are performed separately,

transmission attempts could be made overly conservative or aggressive. Observing this adverse effect on system throughput, authors in [17] propose a joint adaptation of link rate and contention window protocol, named ARC, to address the problem. ARC is an open-loop approach that tries to exercise link adaptations by firstly considering if a proper backoff window has been reached. In symmetric environments (without hidden terminals), ARC performs well, since the transmitter-estimated backoff window adequately reflects the contention status at the receiver. However, in asymmetric environments (where hidden terminals present), ARC performance degrades due to both the inaccurate backoff window and lack of receiver feedback on a proper transmit rate. In addition, since ARC opts to tune contention window (CW) before rate, when the channel condition suddenly becomes good or bad at the receiver, ARC cannot react quickly enough. Several CW adjustments may occur in ARC before reaching a suitable rate and backoff setting.

In this paper, we propose a closed-loop rate adaptive scheme, entitled EARC (Enhanced Adaptation of link Rate and Contention window), that incorporates the backoff CW adjustment. Though sharing the similar concept of jointly adapting link rate and CW parameters, EARC operates differently from ARC. First, with receiver feedback, EARC is able to react to rate changes (due to improved or deteriorated channel condition at the receiver) more quickly than ARC. Second, the transmitter-estimated CW value in EARC can be adjusted based on receiver feedback, further giving EARC the capability of maintaining remarkable system performance even in asymmetric settings (where transmitter and receiver have distinct medium congestion views).

The remainder of this paper is organized as follows. In Section II, we review the binary exponential backoff (BEB) mechanism in 802.11 standard, and five rate adaptation works: ARF, AT, AC, RBAR, and ARC. Section III details our EARC protocol. Extensive simulation results and comparisons with other major multi-rate algorithms are provided in Section IV. Section V validates the EARC performance through mathematical analysis. Finally, in Section VI, we conclude this paper.

## II. PRELIMINARIES

### A. Back-off Mechanism in 802.11 Standard (BEB)

802.11 standard defines two types of media access mechanisms: the Point Coordinate Function (PCF) and the Distributed Coordinate Function (DCF). PCF is a centralized polling-based MAC mechanism, which provides contention-free and time-bounded services. On the other hand, DCF is based on CSMA/CA, mandating stations carrier sense the channel media before transmitting packets. In DCF, every station has a backoff contention window (CW) for collision avoidance. Specifically, at the first transmission attempt, CW is set to the minimum value ( $cw_{min}$ ). A station generates a backoff timer uniformly from  $[0, CW-1]$ , and then starts to count down. When the timer counts down to zero, the station gets the privilege to access the channel. On unsuccessful transmission (ACK not returned), a binary exponential backoff (BEB) mechanism is used to relieve the contention

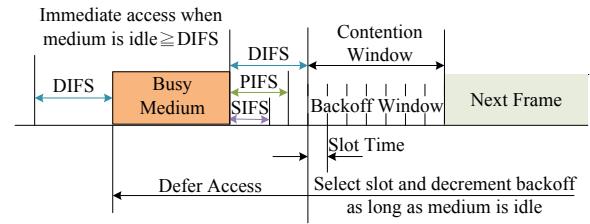


Fig. 1. IEEE 802.11 MAC mechanism.

level. In particular, the station has to double its CW size until CW reaches the maximum CW ( $cw_{max}$ ) value. On successful transmission (ACK returned), DCF resets CW back to  $cw_{min}$ . As several previous works have pointed out, the BEB mechanism in 802.11 DCF does not adapt to the wireless environment wisely [18]–[22]. The 802.11 MAC operations are illustrated in Fig. 1.

### B. Auto-rate Fallback (ARF)

ARF is the most widely implemented rate-adaptive scheme. It was originally used in WaveLAN-II devices, one of the early 802.11 products [6]. The key algorithm of ARF is that sender attempts to upgrade its transmission rate after successfully receiving 10 consecutive ACK frames, whereas the sender switches to a lower rate if it encounters 2 consecutive unsuccessful transmissions (i.e., missing ACK frames or the sender waits longer than timeout). If there is no traffic that has been sent for the present time, then station transmits packet with the highest possible rate. Although ARF is easy to implement, it has one attendant drawback: ARF can not work efficiently under stable or fluctuated channel conditions. That is, either it will constantly try to upgrade the transmission rate (which *SINR* cannot support), leading to unnecessary packet collisions, or can not react quickly enough to match the fluctuated channel conditions.

### C. Adaptive Thresholds (AT)

Observing the problem of using fixed up/down-thresholds without considering time-varying wireless channel characteristics and the impact of link-layer collisions, AT aims to enhance the performance of ARF rate adaptation protocol [15]. Since frame collisions cannot be easily distinguished from channel errors according to missing IEEE 802.11 ACKs, chances are the ARF rate control usually results in unnecessary rate downshifts when channel noise is actually low. To improve the responsiveness of channel variations and mitigate the impact of transmission failures induced by link-layer collisions, the authors in [15] propose a run-time adaptive algorithm to dynamically adjust the up/down-thresholds in ARF based on link-layer measurements. The design philosophy behind AT is to discourage rate downshifts (by setting a higher value for down-threshold) when the number of contending stations increases and reduce rate upshifts (by using a higher value for up-threshold) when contention decreases.

### D. ARF with COLLIE (AC)

Also targeting on improving ARF performance, AC intends to exercise wireless packet loss diagnosis so that transmission

failures caused by link-layer collisions or channel errors (weak signal) can be distinguished [16]. AC statistically analyzes received data through a combination of various metrics such as bit-level and symbol-level error patterns and received signal strength. When the packet loss is diagnosed as an event caused by collision, AC adjusts the contention window (CW) parameter according to the binary exponential backoff (BEB) algorithm used in 802.11 DCF. On the other hand, when the packet loss is determined due to weak signal, AC triggers ARF rate control mechanism. The essential operation of AC greatly depends on the AP module's capability of identifying the true cause of a packet loss and invoking the correct method of adaptation in real-time, which incurs significant per-packet overhead and considerable bandwidth waste when inaccurate diagnosis takes place.

### E. Receiver-based Auto-rate (RBAR)

RBAR is a receiver-based rate-adaptation mechanism [4], which makes the rate adaptation decision based on channel quality estimated at the receiver and informs the sender via RTS/CTS handshaking mechanism. In RBAR, receiver utilizes RTS packet to obtain the RSSI information, and then selects an appropriate data rate provided in CTS to inform the sender. The rate handshaking is confirmed by another Reservation SubHeader (RSH) control message from the sender. Two main drawbacks exist in the RBAR protocol. One is the controlling overhead caused by rate negotiation on a per-packet basis. The other is the fact that RSSI estimation is not precisely supported in most wireless devices, reducing the practical feasibility of RBAR protocol.

### F. Adaptation of Link Rate and Contention Window (ARC)

ARC is an open-loop rate adaptation protocol that jointly considers the contention window adjustment [17]. The ARC protocol estimates the optimal contention window ( $optCW$ ) based on Cali's approximation methods. On transmission successes (failures), the current contention window size  $cw_p$  is compared with  $optCW$ . If  $cw_p > optCW$  ( $cw_p < optCW$ ), then  $cw_p$  is decreased (increased) to perform more aggressive (conservative) transmission attempts while leaving the link rate  $R$  unchanged. Otherwise,  $R$  is upgraded (reduced) to the next higher (lower) rate. Due to its open-loop nature and tuning contention window first, ARC may encounter several transmission failures before reaching a proper backoff and rate setting. In addition, the transmitter-estimated  $optCW$  in ARC does not always reflect the contention level at the receiver under asymmetric networking environments.

## III. OUR EARC PROTOCOL

In this section, we present the operation details of the proposed EARC protocol. Section III-A points out the problem of separately tuning link rate and contention window. Section III-B provides an overview of EARC protocol, and we elaborate on the EARC-related algorithms in Section III-C.

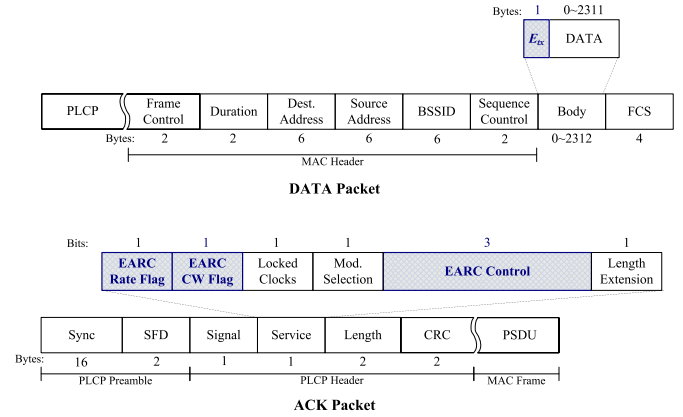


Fig. 2. DATA and ACK packet formats used in our EARC protocol (the shaded areas are specifically utilized by the EARC algorithm).

### A. Problem Statement

In wireless networks, successful data reception is mainly dependent on the Signal-to-Interference-and-Noise Ratio ( $SINR$ ) at the receiver. IEEE 802.11 supports multiple link rates at the physical layer. Each link rate is associated with a certain required  $SINR$  threshold for successfully decoding received packets. Collectively, we define the sum of interference and noise power ( $N + I$ ) as the accumulated environmental energy  $E$ . Suppose no power adjustment exists, apparently  $SINR$  is solely affected by the environmental power level  $E$ . Traditional link rate adaptation approaches try to reduce the transmit rate (hence lower  $SINR$  threshold is required) on transmission failures (potentially due to increased  $E$ ), whereas upgrade the transmit rate (hence higher  $SINR$  threshold is required) on successful transmissions (potentially due to decreased  $E$ ). The accumulated environmental power level  $E$  in some sense indicates the medium congestion status. In 802.11, on transmission failures, the DCF performs a binary exponential backoff mechanism to discourage channel access attempts. When traditional link adaptation is applied, both rate reduction and binary backoff represent double penalties for this wireless link, which may cause overly conservative transmission attempts. On the other hand, once transmission succeeds, 802.11 DCF resets the backoff contention window to the minimum value to encourage channel access attempts. At the same time, traditional link adaptation may also decide to increase the data rate, which leads to overly aggressive transmission attempts. We observe this improper interaction of link rate and backoff mechanism that harms the 802.11 system performance, due to separate consideration of those two parameters.

### B. Protocol Overview

Motivated by the above observations, rather than independently dealing with the two parameters, we propose to jointly consider the link rate and contention window adaptations in a unified framework. In particular, a closed-loop (receiver-assisted) link rate adaptation strategy that also takes contention window adjustment into account, entitled EARC, is developed to improve IEEE 802.11 system capacity. As reviewed in Section II-A, 802.11 DCF is essentially a CSMA scheme,

TABLE I  
*optCW* ESTIMATION FOR IEEE 802.11B

$M$	$p_{opt}^{r_1}$	$optCW^{r_1}$	$p_{opt}^{r_2}$	$optCW^{r_2}$	$p_{opt}^{r_3}$	$optCW^{r_3}$	$p_{opt}^{r_4}$	$optCW^{r_4}$
10	0.0112	177	0.0155	128	0.0243	81	0.0320	61
15	0.0074	271	0.0102	196	0.0159	125	0.0210	94
20	0.0055	364	0.0076	263	0.0119	168	0.0157	127
25	0.0044	458	0.0060	331	0.0094	211	0.0125	159
30	0.0036	551	0.0050	398	0.0078	254	0.0104	191
35	0.0031	645	0.0043	466	0.0067	297	0.0089	224
40	0.0027	738	0.0037	533	0.0059	340	0.0078	256

which mandates a station sense (detect) the wireless channel before attempting to transmit. Only when the sensed (detected) energy is below carrier-sense threshold does a station prepare to carry out its access attempt. EARC taps into this characteristic and lets each station constantly keep track of detected energy levels. In this manner, the environmental energy level  $E$  (as mentioned in Section III-A) can be obtained based on recent energy statistics averaged in a certain time interval<sup>1</sup>. For some communication pair  $tx$  (transmitter) and  $rx$  (receiver), define  $E_{tx}$  and  $E_{rx}$  as the environmental energy level at the transmitter and receiver, respectively. By comparing  $E_{rx}$  to  $E_{tx}$ , a receiver is able to infer the medium congestion difference between the two sides, further utilized to assist in rate selection and contention window adjustment. This energy information is only approximate, yet useful for resolving the problem of asymmetric (different) congestion views comprehended by  $tx$  and  $rx$ . Therefore, we propose to piggyback the  $E_{tx}$  information in DATA packet, as shown in Fig. 2, so that our EARC algorithm at the receiver can utilize this information to perform contention window tuning for the transmitter. Feedback from the receiver is then carried by the ACK packet back to the transmitter. Rather than creating extra overhead, EARC uses the reserved fields (5 bits in total) in PLCP header to carry the feedback, as depicted in Fig. 2. On acquiring the feedback from receiver, our EARC algorithm at the transmitter alters the transmit rate or contention window size accordingly. When performing rate and contention window (CW) adaptations at the receiver, EARC first estimates whether the current transmit rate is the best sustainable choice under the latest observed environmental energy  $E_{rx}$ . A new rate will be suggested if the current rate is not the best one. Otherwise, EARC moves on to evaluate whether the CW needs to be adjusted, based on the difference of  $E_{tx}$  and  $E_{rx}$ . Details on related EARC algorithms are immediately provided in Section III-C.

### C. EARC Algorithms

Suppose  $R$  different rates are supported at the physical layer, denoted as rate  $r_1, r_2, \dots, r_R$  (in the order of increasing rate). Initially, the transmitter uses an estimated optimal contention window ( $optCW^{r_R}$ ) for the highest supported rate  $r_R$ , and transmits DATA at this rate after the backoff

<sup>1</sup>Note that the environmental energy level for a node (obtainable from the device RSSI reading) means the accumulated power strength (transmission energies produced by both successful and lost/corrupted packets over a wireless channel) contributed by surrounding activities (including collision events that lead to lost/corrupted packets) in this node's neighborhood.

mechanism completes. The estimation of  $optCW^{r_i}$  for some transmit rate  $r_i$  is based on an extended model of Cali's analytical approximations [18], presented in Section III-C1. When DATA packet arrives at the receiver, necessary EARC operations are described in Section III-C2. On the other hand, Section III-C3 details the EARC reactions to both cases (whether the transmission succeeds or fails) at the transmitter.

1) *optCW Estimation*: For analytical tractability, the authors in [18] consider a  $p$ -persistent version of 802.11 DCF, where  $p = \frac{1}{E[B]+1}$  and  $E[B]$  is the average backoff slots. Parameter  $p$  is appropriately termed as the *attempt probability*. Define the protocol capacity  $\rho = \frac{m_{r_i}}{t_v}$ , where  $m_{r_i}$  is the packet (message) transmission time at rate  $r_i$  and  $t_v$  is the virtual transmission time (time between two consecutive successful transmissions, as illustrated in Fig. 3). Denote  $E[N_c]$  as the expected number of collisions,  $E[Coll]$ ,  $E[Idle]$ , and  $E[ Succ]$  as the expected time durations of each collision, idle period, and successful transmission, respectively. Now we can express  $t_v$  as follows,

$$t_v = E\left[\sum_{i=1}^{N_c} (Idle_i + Coll_i + t_{DIFS} + t_{SIFS} + t_{ACK})\right] + E[Idle_{N_c+1}] + E[ Succ]$$

$$= E[N_c] \cdot (E[Coll] + t_{SIFS} + t_{ACK} + t_{DIFS}) + (E[N_c] + 1) \cdot E[Idle] + E[ Succ]$$

where  $t_{DIFS}$ ,  $t_{SIFS}$ , and  $t_{ACK}$  represent the time durations of 802.11 DIFS, SIFS, and ACK transmission separately. For some transmit rate  $r_i$ ,  $E[Coll]$  and  $E[ Succ]$  can be approximated as  $E[Coll] = m_{r_i}$  (packet transmission time) and  $E[ Succ] = m_{r_i} + t_{SIFS} + t_{ACK} + t_{DIFS}$  (neglecting propagation delay). Suppose there are  $M$  active nodes in the transmission neighborhood, based on probability computations,  $E[N_c]$  and  $E[Idle]$  can be derived as

$$E[N_c] = \frac{1 - (1 - p)^M}{Mp(1 - p)^{M-1}} - 1, \quad (1)$$

$$E[Idle] = \frac{(1 - p)^M}{1 - (1 - p)^M} \times t_{slot}, \quad (2)$$

where  $t_{slot}$  is the slot time.

Our ultimate goal is to minimize virtual transmission time  $t_v$ , such that the protocol capacity  $\rho$  can be maximized. As illustrated in Fig. 3 (right),  $t_v$  is a function of attempt probability  $p$ , and the optimal  $p$  ( $p_{opt}^{r_i}$ ) at rate  $r_i$  for some given  $M$  exists at the transitional point (bottom) of  $t_v$  curve. The results from [18] suggest that an optimal transmission attempt probability ( $p_{opt}^{r_i}$ ) at rate  $r_i$  can be obtained by observing number of idle slots and active nodes ( $M$ ) within the transmission range. Recall that attempt probability  $p = \frac{1}{E[B]+1}$ , where  $E[B]$  is

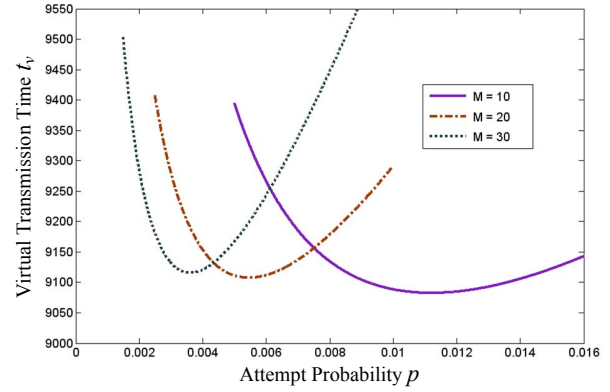
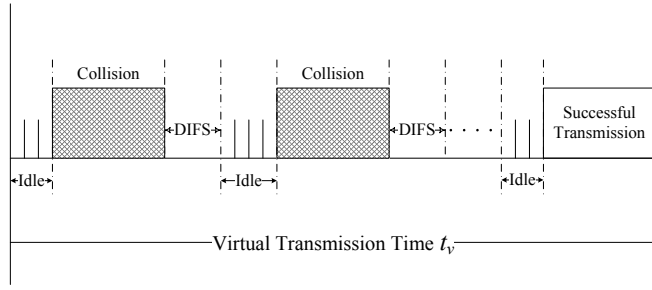


Fig. 3. Definition of the virtual transmission time in  $p$ -persistent IEEE 802.11 protocol, and resultant virtual transmission time from different attempt probability settings under various numbers of active nodes all using transmission rate at 1 Mb/s.

the average backoff slots. Suppose  $E[CW]$  is the expected contention window size, hence  $E[B] = \frac{E[CW]-1}{2}$ . Now the attempt probability can be re-formulated as  $p = \frac{2}{E[CW]+1}$ , or equivalently  $E[CW] = \frac{2}{p} - 1$ . Consequently, for some obtained  $p_{opt}^{r_i}$ , the corresponding optimal contention window size ( $optCW^{r_i}$ ) can be approximated as

$$optCW^{r_i} = \frac{2}{p_{opt}^{r_i}} - 1. \quad (3)$$

We run several simulation experiments to estimate the  $optCW^{r_i}$  for various numbers of active nodes under four 802.11b rates based on this method in ns-2 simulator. Table I shows some of the results, where  $r_1 = 1$  Mb/s,  $r_2 = 2$  Mb/s,  $r_3 = 5.5$  Mb/s, and  $r_4 = 11$  Mb/s.

2) *Receiver Operations:* Given the  $E_{tx}$  information carried by DATA packet and  $E_{rx}$  measured at the receiver, EARC determines how to adjust the rate and backoff parameters. Recall the ACK packet format shown in Fig. 2. Three EARC-related fields (reserved bits in PLCP header) are defined: **EARC Rate Flag**, **EARC CW Flag**, and **EARC Control**. If EARC algorithm at the receiver concludes that the current transmit rate is not the best one, then EARC Rate Flag is set *true*, and EARC Control contains the suggested rate (totally 8 rates can be represented given 3 bits in EARC Control). If there is no need to adjust the rate, EARC Rate Flag is set *false*, and EARC algorithm moves on to evaluate whether the backoff parameter (CW size) should be adjusted. If the perceived environmental energy level  $E_{rx}$  measured at the receiver is different from transmitter-measured  $E_{tx}$ , indicating asymmetric (inconsistent) contention views, EARC sets CW Flag *true* and provides suggested CW adjustment quantity in EARC Control field. Otherwise, EARC CW Flag is set *false*, and the value of EARC Control becomes meaningless (neither rate nor CW need be adjusted). Below we describe the specific process of configuring the three fields executed by EARC algorithm at the receiver.

In order to select the best sustainable rate, we introduce a Rate Selection Reference (RSR) table. Since the *SINR* value is not practically obtainable by current hardware functionality, EARC at the receiver does not perform the rate selection

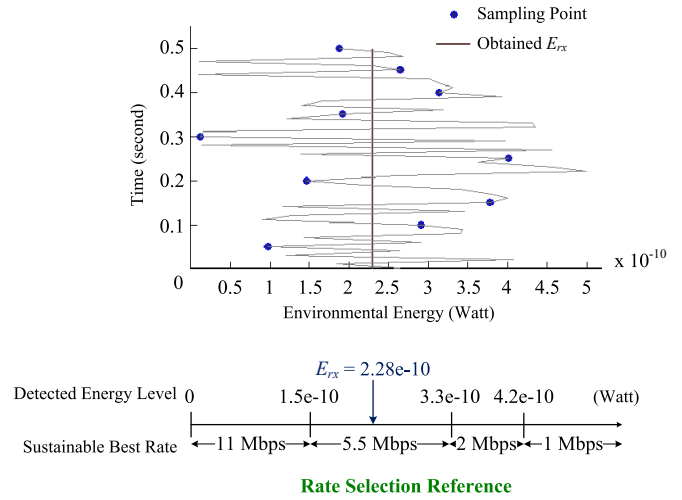


Fig. 4. Illustration of obtaining sensed energy level  $E_{rx}$  and the empirically derived rate selection reference (RSR) table (subject to changes as channel condition varies) at the receiver.

using *SINR*<sup>2</sup>. Instead, the receiver maintains a rate selection reference (RSR) table, illustrated in Fig. 4 (bottom), which is empirically constructed based on recent reception behavior. The RSR is essentially a table that indicates the best suggested rates for different ranges of receiver-measured environmental energy levels ( $E_{rx}$ ). In the example of Fig. 4,  $E_{rx}$  is obtained from 10 sampling values of detected energy averaged in the latest 0.5-second time interval, which turns out to be  $2.28e-10$  Watt. Given the  $E_{rx}$  value, EARC algorithm at the receiver concludes that 5.5 Mb/s is the best rate to use.

The initial RSR table can be constructed experimentally. In our approach, we set up a nominal environment containing a transmission pair with communication distance of 80 meters. By fixing the transmit power at  $9.947e-2$  Watt, we raise the environmental noise/interference power from 0 to  $6e-10$  Watt in the simulator. Given frames (packets) of size equal to 1000 bytes sent continuously at four different transmission rates,

<sup>2</sup>On receipt of a packet, the RSSI (Received Signal Strength Indicator) reading provided by off-shelf IEEE 802.11-based wireless modules contains accumulated energy level (S+N+I). We are unable to discern the difference between received signal strength (S) and interference/noise power level (N+I) from the RSSI reading, making *SINR* (ratio) value difficult/unrealistic to obtain [23], [24].

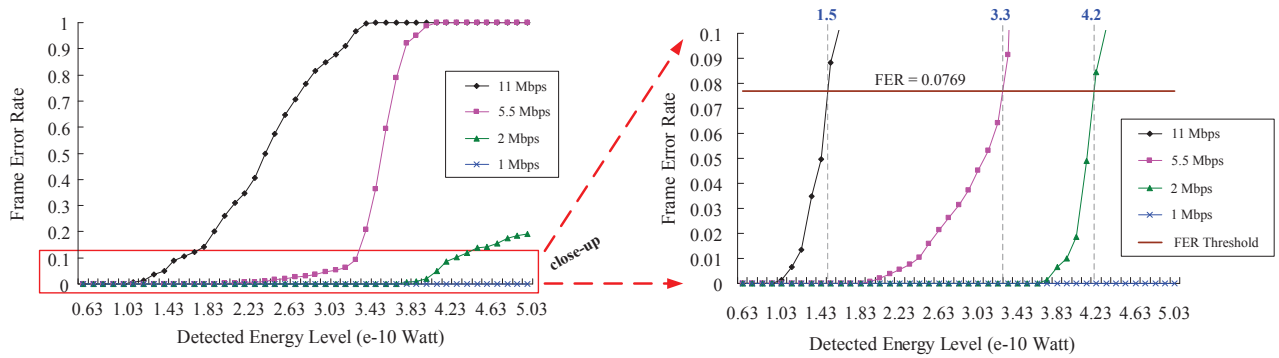


Fig. 5. Relationship of FER against environmental energy level (left) and identifying corresponding energy thresholds in RSR table (right).

the corresponding frame error rate (FER) can be computed at each sampling point with pre-configured noise power level. After combining these measurements and applying appropriate interpolation method, the relationship between FER and environmental energy level can be quantified, as shown in Fig. 5 (left). In order to sustain a bit error rate (BER) below  $1.0e-5$ , or equivalently FER under  $0.0769^3$ , as specified in the IEEE 802.11 standard, we identify the corresponding three energy thresholds for the four rates:  $1.5e-10$ ,  $3.3e-10$ , and  $4.2e-10$  Watt, as illustrated in Fig. 5 (right). The result implies that when the detected energy level is over a certain threshold, the next lower rate should be used to ensure a BER below  $1.0e-5$ . Consequently, the RSR table as displayed in Fig. 4 is generated. Note that the RSR table is subject to changes as the receiver learns and updates by monitoring recent receiving activities from some specific transmitter. Initially, a receiver uses the same default RSR table for all transmitters. Then as time advances, those RSR tables are evolved and customized through receiver's learning mechanism. By constantly measuring/sampling environmental energy levels and recording reception activities into RSR table at the receiver, we tactfully avoid using SINR to infer the best rate. Although the empirically constructed RSR table may not be always accurate in determining a rate, such inaccuracy is allowed and tolerable since rate is not the only parameter in EARC adaptation algorithm. Certain imprecision in rate prediction can actually be fixed by tuning the corresponding CW parameter properly. As we can observe from our simulations (in Section IV), the RSR table indeed provides a good rate selection reference.

Suppose the current transmit rate is  $r_i$  and receiver suggested rate is  $r_j$ . If  $r_i$  is not equal to  $r_j$ , then the receiver sets EARC Rate Flag *true*, and corresponding 3-bit EARC Control, defined as  $b_1b_2b_3$ , with  $value(b_1b_2b_3) = j - 1$ . On the other hand, if  $r_i = r_j$ , meaning that the best rate is already in use, the receiver then looks at the energy difference  $E_{diff}$  between  $E_{rx}$  and  $E_{tx}$ . Define  $E_{diff} = \frac{E_{rx} - E_{tx}}{E_{tx}}$ . Generally speaking, rate adaptation is effective in resolving the collisions due to *concurrent transmissions* (with transmitters locating outside of each other's carrier-sense range), but ineffective in reducing collisions due to *simultaneous transmissions* (inside carrier-sense range). The latter can be alleviated by increasing the contention window size to discourage transmission attempts in

the collision zone. Thus contention window tuning is also critical for system performance. As presented in Section III-C1, an optimal contention window ( $optCW^{r_i}$ ) at rate  $r_i$  can be approximated based on Cali's analytical model. However, the transmitter-estimated  $optCW^{r_i}$  does not necessarily reflect the contention status at the receiver. As a result, the receiver utilizes  $E_{diff}$  to assist in tuning transmitter's CW value to further increase the transmission success probability. Specifically, when  $E_{diff}$  is positive, indicating energy (contention) level at the receiver is higher than that at the transmitter, the CW value should be increased to reduce contention. In contrast, when  $E_{diff}$  is negative, implying contention level is lower at the receiver than that at the transmitter, the CW value can be decreased to encourage more aggressive transmission attempts. Consequently, the receiver sets EARC CW Flag *true* and the first bit ( $b_1$ ) of EARC Control as follows,

$$b_1 = \begin{cases} 1 & \text{if } E_{diff} > 0 \\ 0 & \text{if } E_{diff} < 0 \end{cases}. \quad (4)$$

Next, we utilize the rest two bits ( $b_2b_3$ ) of EARC Control to indicate the CW adjustment quantity for transmitter. Suppose  $K$  values can be represented (in our case  $K = 2^2 = 4$  given two bits  $b_2b_3$  are available). We define  $K$  boundaries  $(0, \frac{1}{K}, \frac{2}{K}, \dots, \frac{K-1}{K})$  for possible  $E_{diff}$  values. When  $|E_{diff}|$  lies between any two boundaries or beyond the largest boundary, the receiver configures the value of  $b_2b_3$  in EARC Control as follows,

$$value(b_2b_3) = \begin{cases} k & \text{if } (\frac{k}{K} < |E_{diff}| \leq \frac{k+1}{K}) \& \& (0 \leq k < K-1) \\ K-1 & \text{otherwise} \end{cases}$$

Table II summarizes all possible cases of  $E_{diff}$  and corresponding settings on EARC CW Flag and EARC Control bits when EARC Rate Flag is set *false*.

We provide the pseudocode for receiver EARC operations in Algorithm 1. Based on the values of EARC Rate Flag, EARC CW Flag, and EARC Control contained in ACK packet, the transmitter is able to perform the rate and contention window adjustment accordingly.

3) *Transmitter Operations*: Once the ACK packet successfully returns from the receiver, the transmitter first checks if EARC Rate Flag is set *true*. If yes, rate and CW are configured to  $r_{b+1}$  and  $optCW^{r_{b+1}}$  respectively, where  $b = value(b_1b_2b_3)$ . If EARC Rate Flag is set *false*, then transmit rate remains at  $r_i$ , and the transmitter moves on to

<sup>3</sup>FER =  $1 - (1 - 1.0e - 5)^{8000} = 0.0769$ , given BER =  $1.0e - 5$  and frame size of 1000 bytes.

TABLE II  
ALL CASES OF  $E_{diff}$  AND CORRESPONDING EARC CW FLAG AND CONTROL BITS WHEN RATE FLAG = *false*

Possible Cases	EARC CW Flag	EARC Control		
		$b_1$	$b_2$	$b_3$
$0 < E_{diff} \leq 25\%$	<i>true</i>	1	0	0
$25\% < E_{diff} \leq 50\%$			0	1
$50\% < E_{diff} \leq 75\%$			1	0
$75\% < E_{diff}$			1	1
$-25\% \leq E_{diff} < 0$		0	0	0
$-50\% \leq E_{diff} < -25\%$			0	1
$-75\% \leq E_{diff} < -50\%$			1	0
$E_{diff} < -75\%$			1	1
$E_{diff} = 0$	<i>false</i>	don't care		

---

**Algorithm 1** EARC Algorithm at Receiver

```

1: while (DATA packet transmitted at rate  $r_i$  received) do
2:   Look up the RSR table and decide a best sustainable rate  $r_j$ 
   based on  $E_{rx}$ ;
3:   if ( $i \neq j$ ) then
4:     EARC Rate Flag set to true;
5:     Set value( $b_1b_2b_3$ ) =  $j - 1$  in the EARC Control field;
6:   else
7:     EARC Rate Flag set to false;
8:     Compare  $E_{rx}$  with  $E_{tx}$  and calculate  $E_{diff}$ ;
9:     if ( $E_{diff} == 0$ ) then
10:      EARC CW Flag set to false;
11:    else
12:      EARC CW Flag set to true;
13:      if ( $E_{diff} < 0$ ) then
14:        Set  $b_1 = 0$ ; // to decrease CW
15:      else
16:        Set  $b_1 = 1$ ; // to increase CW
17:      if ( $(\frac{k}{K} < |E_{diff}| \leq \frac{k+1}{K}) \ \&\& \ (0 \leq k < K - 1)$ ) then
18:        Set value( $b_2b_3$ ) =  $k$ ;
19:      else
20:        Set value( $b_2b_3$ ) =  $K - 1$ ;
21:    Return ACK packet back to transmitter;

```

---

check the EARC CW Flag. If EARC CW Flag is also *false*, then the present CW value, denoted  $cw_p$ , remains. Otherwise, the transmitter should adjust the CW value by setting

$$cw_p = \begin{cases} optCW^{r_i} \left( 1 + \frac{2 \cdot \text{value}(b_2b_3) + 1}{2K} \right) & \text{if } b_1 = 1 \\ optCW^{r_i} \left( \alpha + (1 - \alpha) \left( 1 - \frac{2 \cdot \text{value}(b_2b_3) + 1}{2K} \right) \right) & \text{if } b_1 = 0 \end{cases}$$

where  $\frac{2 \cdot \text{value}(b_2b_3) + 1}{2K}$  is basically the middle value of two boundaries that  $|E_{diff}|$  lies between, and  $\alpha$  is a weighing factor in the range of  $[0, 1)$ . The use of  $\alpha$  allows us the flexibility of bounding CW tuning within a reasonable range, avoiding too aggressive reduction in CW value. For example, if  $\alpha = 0.5$ , the CW reduction quantity will never exceed half of the optimal contention window size ( $optCW^{r_i}$ ) at rate  $r_i$ . On the other hand, from  $cw_p$  formulation, we can observe that the increased CW value will never exceed twice of  $optCW^{r_i}$ . In this manner, we limit the CW adjustment to avoid too drastic changes.

If, unfortunately, ACK does not return (or DATA packet simply fails to reach the receiver), the transmitter has no receiver feedback to assist in the rate and CW adaptation. In this case, the transmitter compares  $cw_p$  with  $optCW^{r_i}$ ,

and increases  $cw_p$  to  $optCW^{r_i}$  if  $cw_p < optCW^{r_i}$ , letting rate stay at  $r_i$ . The design rationale is trying to impose a larger backoff window on future transmission, hoping the next transmission can succeed without the need to decrease rate. However, if  $cw_p \geq optCW^{r_i}$ , then the transmitter should decrease rate to the next lower one (or maintain the rate if it is already the lowest). Meanwhile,  $cw_p$  is set to the optimal CW value at the lower rate.

Algorithm 2 summarizes EARC operations at the transmitter. Note that the feedback from the receiver takes effect on the next DATA packet (including retransmitted packet) to be sent by the transmitter within a certain time interval (*timeout*). In case the next DATA packet arrives after *timeout* expires, the corresponding rate and CW settings become invalid, and the transmitter resets transmit rate to the default rate  $r_R$  (the highest supported rate) and CW at  $optCW^{r_R}$ .

---

**Algorithm 2** EARC Algorithm at Transmitter After DATA Has Been Sent to Receiver Using Rate  $r_i$ 

```

1: Suppose  $cw_p$  is the current CW setting for this particular
   receiver;
2: if (ACK returned) then
3:   if (EARC Rate Flag == true) then
4:     Obtain  $b = \text{value}(b_1b_2b_3)$ ;
5:     Set  $r_{next} = r_{b+1}$ ;
6:     Set  $cw_p = optCW^{r_{b+1}}$ ;
7:   else
8:     Set  $r_{next} = r_i$ ;
9:     if (EARC CW Flag == true) then
10:      switch ( $b_1b_2b_3$ )
11:        case 100: Set  $cw_p = optCW^{r_i} \times (1 + 12.5\%)$ ;
12:        case 101: Set  $cw_p = optCW^{r_i} \times (1 + 37.5\%)$ ;
13:        case 110: Set  $cw_p = optCW^{r_i} \times (1 + 62.5\%)$ ;
14:        case 111: Set  $cw_p = optCW^{r_i} \times (1 + 87.5\%)$ ;
15:        case 000: Set  $cw_p = optCW^{r_i} \times (\alpha + (1 - \alpha)(1 - 12.5\%))$ ;
16:        case 001: Set  $cw_p = optCW^{r_i} \times (\alpha + (1 - \alpha)(1 - 37.5\%))$ ;
17:        case 010: Set  $cw_p = optCW^{r_i} \times (\alpha + (1 - \alpha)(1 - 62.5\%))$ ;
18:        case 011: Set  $cw_p = optCW^{r_i} \times (\alpha + (1 - \alpha)(1 - 87.5\%))$ ;
19:      else
20:        if ( $cw_p < optCW^{r_i}$ ) then
21:          Set  $cw_p = optCW^{r_i}$ ;
22:        Set  $r_{next} = r_i$ ;
23:      else
24:        if ( $i > 1$ ) then
25:          Set  $r_{next} = r_{i-1}$ ;
26:          Set  $cw_p = optCW^{r_{i-1}}$ ;
27:        else
28:          Set  $r_{next} = r_i$ ;
29:          Set  $cw_p = optCW^{r_i}$ ;
30:      if (next DATA packet destined for this particular receiver arrives
       before timeout expires) then
31:        Pick up the backoff timer from  $[0, cw_p - 1]$  and starts to count
       down;
32:        Transmit the DATA packet at rate  $r_{next}$ ;
33:      else
34:        Pick up the backoff timer from  $[0, optCW^{r_R} - 1]$  and starts to
       count down; //  $r_R$  is the highest supported rate
35:        Transmit the DATA packet at rate  $r_R$ ;

```

---

#### IV. SIMULATION RESULTS

In this section, we run simulations in the ns-2 simulator (version 2.34). We add our EARC module in the dei80211mr library that supports 802.11b multi-rate PHY. Four link rates are available: 1, 2, 5.5, and 11 Mbps. Friis radio propagation

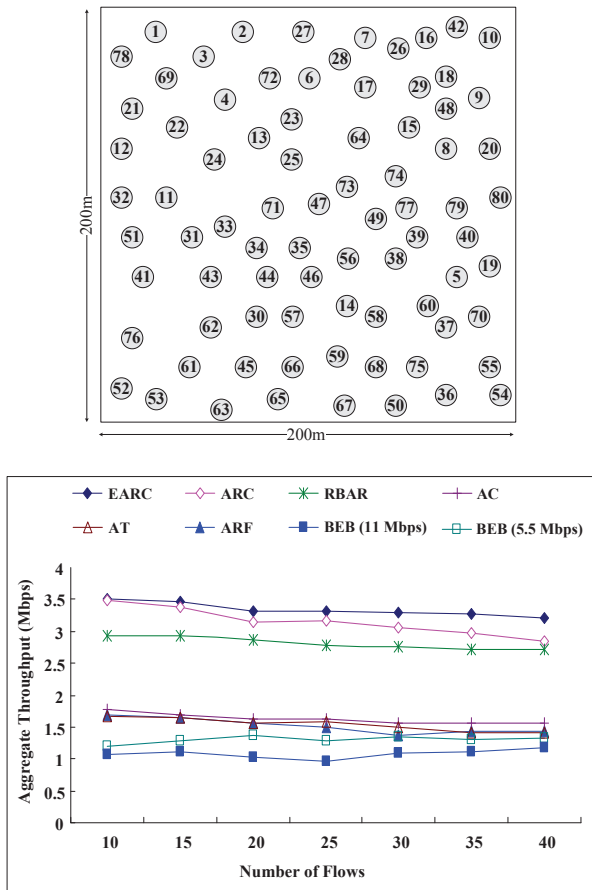


Fig. 6. Performance comparison in a symmetric environment.

model is used. CBR traffic (sending rate = 1 Mbps) is generated with packet size of 1000 bytes. Transmit power is fixed at  $9.947e-2$  Watt. We let every node randomly start transmission within the time range from 0 to 0.2 seconds to reduce initial collisions. MAC parameters  $cw_{min} = 32$  and  $cw_{max} = 1024$  are used. Total simulation time is 20 seconds. Each statistic is obtained from the average of 20 experiments. For comparison purpose, we also implement BEB (with fixed rate at 11 and 5.5 Mbps respectively), ARF, AT, AC, RBAR, and ARC mechanisms. For BEB, ARF, AT, AC, and RBAR, the default binary exponential backoff is used as the CW adjustment strategy. Except for BEB, which has link rate fixed at 11 Mbps.

#### A. Symmetric Environment

Fig. 6 (upper) depicts the simulation environment, where nodes are randomly placed in a 200x200 square-meter rectangular area. A maximum of 40 traffic flows are generated with the average communication distance at 80 meters. In this environment, the contention level at the transmitter is consistent with that comprehended by the receiver (most of the time), to which we refer as a *symmetric environment*. Fig. 6 (lower) shows the system throughput against number of flows for different approaches performed in this symmetric environment. We observe that our EARC outperforms other strategies, while BEB (11 Mbps) produces the worst performance. ARF and

ARF-based approaches (AT and AC) perform slightly better than BEB. By jointly adjusting the rate and CW parameters, ARC yields the second best aggregate throughput. Due to its open-loop nature (without receiver feedback), ARC is unable to react to the varying channel as quickly as EARC does. On the other hand, although RBAR incorporates receiver feedback to assist in rate selection, the system throughput achieved by RBAR is not as high as EARC due to the controlling overhead and binary exponential backoff mechanism used by RBAR. This result demonstrates the importance of designing the rate and CW parameters in a unified framework at the cost of moderate controlling overhead (only one extra byte to carry  $E_{tx}$  in our EARC design). Consequently, EARC improves the performance of ARC and RBAR by 5.7% and 15.6% in average. Compared to ARF (the most commonly implemented rate adaptation mechanism in real devices), EARC produces system throughput around 2.2 times that achieved by ARF-based mechanisms.

In order to have a better understanding of the detailed link rate and CW adaptation process, we provide the link rate utilization and CW statistics, for the case of 40 flows, in Fig. 7. From the statistics, we observe that EARC and ARC use more CW values, while CW used by other strategies only takes on a few values due to binary exponential backoff mechanism. Except for BEB (11 and 5.5 Mbps), all four rates are used. Interestingly, for EARC and RBAR, the largest proportion of rate setting is both at 2 Mbps, despite that the rate selection mechanisms adopted by RBAR and EARC are different. In RBAR, the best rate is selected based on  $SINR$  value, which is obtainable in simulator but not accurately supported by current hardware. In contrast, EARC decides on the best sustainable rate according to the RSR table derived from real reception history, which is practically implementable. This result implies that the RSR table introduced in EARC does good judgement without the need to obtain  $SINR$ , and thus represents a promising option for rate determination. For ARC, the most used rate is 11 Mbps (the highest rate) due to its protocol nature of always tuning CW before rate. For ARF, the majority of rate setting is at 1 Mbps (the lowest rate). Due to the protocol nature of ARF, it is easier to decrease rate (on 2 consecutive failures) than to increase rate (on 10 consecutive successes). Consequently, it becomes harder for ARF to bounce back to a higher rate in heavily contended environment. AT improves this drawback by adaptively using a smaller up-threshold in a contended environment (actually the down-threshold tends to remain larger than up-threshold when the number of contending stations is over 10), leading to the most used rate set at the highest 11 Mbps. However, the improvement over ARF is quite limited, since it is not always advantageous to transmit at fast rates when the contention is high. Instead, by using lower rates and setting appropriate CW values, many collisions between contending stations can be effectively resolved (as in the strategy adopted by our EARC). For AC, the highest 11 Mbps and the lowest 1 Mbps are used with comparatively equal probability. By distinguishing the packet loss causes and taking corresponding actions (tuning CW or rate parameters), AC performs better than ARF but in an insignificant way due to its considerable per-packet diagnosis overhead. Moreover, when the judgement of packet



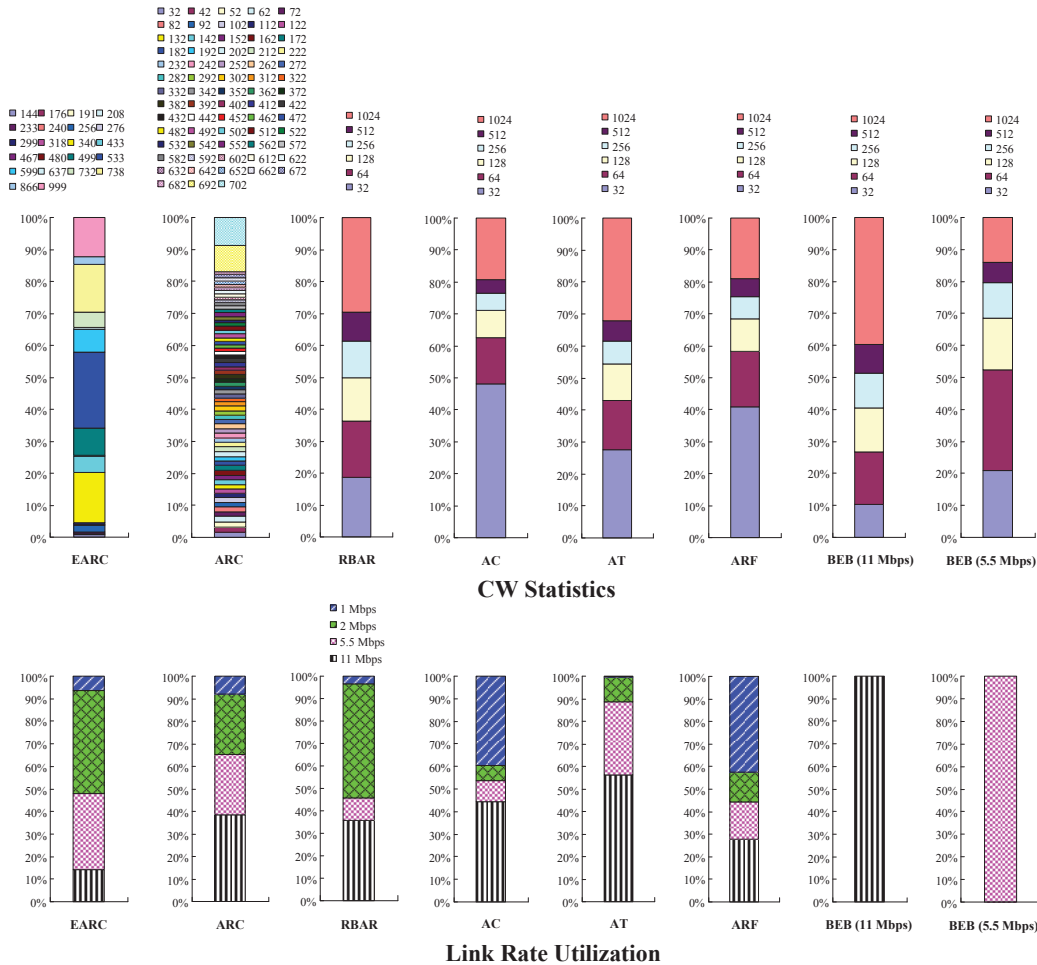


Fig. 7. Investigation of CW settings and link rate utilization when 40 flows are randomly generated in a symmetric environment.

loss causes turns out to be incorrect, extra resource penalty paid by AC is costly and cannot be ignored. The above observations reveal that the rate or CW parameter alone is incapable of handling various failure sources. Both parameters should work cooperatively in order to push the system capacity limit.

**B. Asymmetric Environment**

Now we conduct another set of simulations in an asymmetric environment. Specifically, if hidden terminals exist in the network, the observed contention status at the transmitter is different from that at the receiver, to which we refer as the *asymmetric environment*. Such inconsistent contention comprehension can invalidate the transmitter-estimated CW setting. In Fig. 8, we create two types of asymmetric environments to reflect different contention levels perceived by the transmitter and receiver. We randomly generate 20 flows in such an environment. With inconsistent views, T20R30 is used to express the perceived numbers of active nodes at the transmitter and receiver are 20 and 30, respectively. Six different configurations of the 20 flows are experimented. Fig. 9 shows the system throughput produced by different strategies under the six flow configurations. EARC still performs the best, but ARC degrades due to its inability of properly handling the asymmetric condition, where receiver feedback is essential.

Although having receiver feedback, performance achieved by RBAR is limited by its communication overhead and lack of incorporating appropriate CW adjustment. Collectively, EARC performs better than ARC and RBAR by around 21%, and has the potential to improve the performance of ARF-based approaches by 1.8 times in average.

From the above simulations, we also observe that the joint CW adjustment effectively maintains rate stability, preventing unnecessary rate fluctuations. Specifically, if the medium congestion level can be reduced by imposing a larger backoff window on transmissions, then there may be no need to decrease the link rate. Conversely, if there is extra interference that may be tolerated, a smaller backoff window can be used to encourage more transmission activities while keeping the rate intact.

**V. MATHEMATICAL ANALYSIS**

We build a Markov chain model to evaluate the EARC performance<sup>4</sup>. Similar methodology has been used by [25]–[29]. However, those works deal with CW and link rate independently. In [25], [28], the authors analyze the fixed rate

<sup>4</sup>In the current work, we assume a relatively static environment. We will investigate the mobility-incurred channel dynamics issues when developing our future performance enhancement protocols and report possible interesting findings in a subsequent paper.

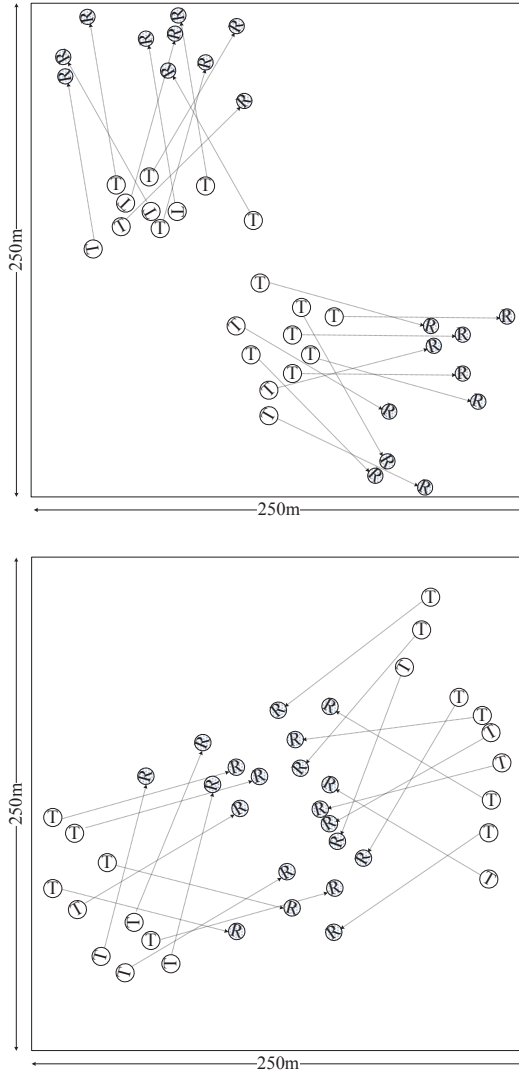


Fig. 8. Asymmetric environments with more contending stations at the transmitter (upper) and more contending stations at the receiver (lower).

802.11 DCF throughput, whereas authors in [26], [27], [29] study the DCF performance under multi-rate environments. Due to the jointly adaptation of link rate and DCF CW size in EARC protocol, we basically extend the Markov chain model from [25] to consider both parameters in transition states. We investigate an IEEE 802.11b network with four rates: 1, 2, 5.5, and 11 Mbps, denoted as  $r_1$ ,  $r_2$ ,  $r_3$ , and  $r_4$  respectively. Suppose  $n$  contending stations exist in the network, and each station always has a packet ready for transmission. Define  $r_i$ ,  $optCW^{r_i}$  and  $r_i, cw_p$  as the states with link rate  $r_i$  (Mbps) using contention window sizes set at  $optCW^{r_i}$  and  $cw_p$  (where  $p = 0, 1, \dots, 7$  indicating eight CW adjustment possibilities in EARC) respectively. Fig. 10 illustrates the simplified Markov chain model on EARC operations (with detailed backoff counting down process omitted).

Let  $b_{r_i, optCW^{r_i}}$  and  $b_{r_i, cw_p}$  (where  $p = 0, 1, \dots, 7$ ) denote the probabilities of states  $r_i, optCW^{r_i}$  and  $r_i, cw_p$ , whereas  $p_{succ}^{r_i}$ ,  $p_{fail}^{r_i}$ , and  $p_{timeout}^{r_i}$  denote the transition probabilities of transmission success, transmission failure, and timeout event when using rate  $r_i$ , respectively. Then we have

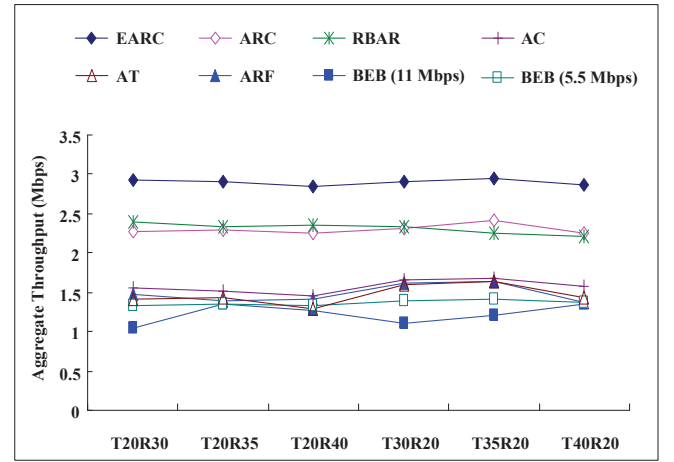


Fig. 9. Throughput performance in an asymmetric environment.

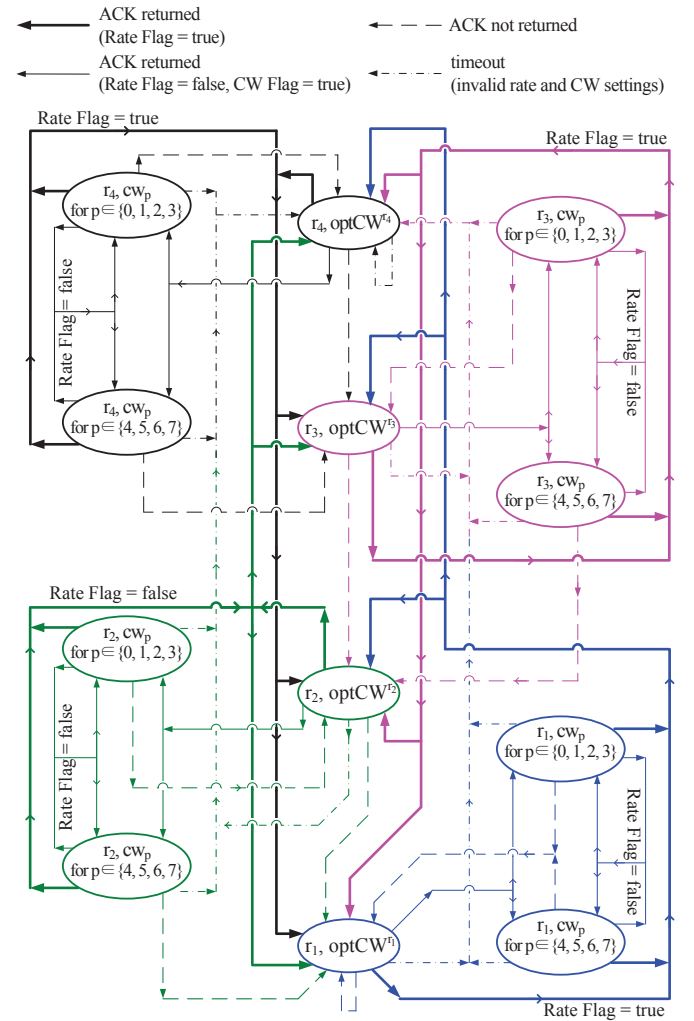


Fig. 10. EARC operations expressed by the Markov chain model (with detailed backoff counting down process omitted).

$$\begin{aligned}
 b_{r_i, optCW^{r_i}} &= \sum_{p=0}^3 b_{r_i, cw_p} \times p_{fail}^{r_i} \\
 &+ \sum_{j=1, j \neq i}^4 \left[ \left( b_{r_j, optCW^{r_j}} + \sum_{p=0}^7 b_{r_j, cw_p} \right) \times \frac{1}{11} \times p_{succ}^{r_j} \right] \\
 &+ \begin{cases} \sum_{k=1}^2 \left[ \left( b_{r_k, optCW^{r_k}} + \sum_{p=4}^7 b_{r_k, cw_p} \right) \times p_{fail}^{r_k} \right] & \text{for } i = 1 \\ \left( b_{r_{i+1}, optCW^{r_{i+1}}} + \sum_{p=4}^7 b_{r_{i+1}, cw_p} \right) \times p_{fail}^{r_{i+1}} & \text{for } i = 2, 3 \\ \sum_{k=1}^4 \left[ \left( b_{r_k, optCW^{r_k}} + \sum_{p=0}^7 b_{r_k, cw_p} \right) \times p_{timeout}^{r_k} \right] & \text{for } i = 4 \end{cases}
 \end{aligned}$$

$$\begin{aligned}
 b_{r_i, cw_0} &= b_{r_i, cw_1} = \dots = b_{r_i, cw_7} \\
 &= \left( b_{r_i, optCW^{r_i}} + \sum_{p=0}^7 b_{r_i, cw_p} \right) \times \frac{1}{11} \times p_{succ}^{r_i} \quad \text{for } i \in \{1, 2, 3, 4\} \\
 1 &= \sum_{i=1}^4 b_{r_i, optCW^{r_i}} + \sum_{i=1}^4 \sum_{p=0}^7 b_{r_i, cw_p}
 \end{aligned}$$

Suppose  $p_{tx}$  denotes the probability that a station attempts transmission and  $p_{tx}^{r_i}$  denotes the conditional probability that rate  $r_i$  is used given the transmission attempt is committed in an observed time slot. Define  $p_{tr}$  as the probability that *at least one* transmission attempt occurs in that slot, which can be expressed as  $p_{tr} = 1 - (1 - p_{tx})^n$  (recall that  $n$  contending stations exist). Then we have

$$p_{succ}^{r_i} = \frac{n \times [p_{tx} \times p_{tx}^{r_i} \times (1 - p_{tx})^{n-1}]}{p_{tx}} \quad \text{for } i \in \{1, 2, 3, 4\}$$

$$p_{succ} = \sum_{i=1}^4 p_{succ}^{r_i}$$

Next, we formulate failure probability  $p_{fail}^{r_i}$  by variables  $p_{tx}$ ,  $p_{tx}^{r_i}$ , and  $p_{tr}$  as follows

$$p_{fail}^{r_4} = \frac{1}{p_{tr}} \times \sum_{i=2}^n \binom{n}{i} \times (p_{tx} \times p_{tx}^{r_4})^i \times (1 - p_{tx})^{n-i}$$

$$p_{fail}^{r_3} = \frac{1}{p_{tr}} \times \sum_{i=2}^n \binom{n}{i} \times \left( \sum_{j=1}^i \binom{i}{j} \times (p_{tx} \times p_{tx}^{r_3})^j \times (p_{tx} \times p_{tx}^{r_4})^{i-j} \right) \times (1 - p_{tx})^{n-i}$$

$$\begin{aligned}
 p_{fail}^{r_2} &= \frac{1}{p_{tr}} \times \sum_{i=2}^n \binom{n}{i} \times \left[ \sum_{j=2}^i \binom{i}{j} \times \left( \sum_{k=1}^j \binom{j}{k} \times (p_{tx} \times p_{tx}^{r_2})^k \times (p_{tx} \times p_{tx}^{r_3})^{j-k} \right) \right. \\
 &\quad \left. \times (p_{tx} \times p_{tx}^{r_4})^{i-j} \right] \times (1 - p_{tx})^{n-i}
 \end{aligned}$$

$$\begin{aligned}
 p_{fail}^{r_1} &= \frac{1}{p_{tr}} \times \sum_{i=2}^n \binom{n}{i} \times \left\{ \sum_{j=2}^i \binom{i}{j} \times \left[ \sum_{k=2}^j \binom{j}{k} \times \left( \sum_{l=1}^k \binom{k}{l} \times (p_{tx} \times p_{tx}^{r_1})^l \right) \right. \right. \\
 &\quad \left. \left. \times (p_{tx} \times p_{tx}^{r_2})^{k-l} \right] \times (p_{tx} \times p_{tx}^{r_3})^{j-k} \times (p_{tx} \times p_{tx}^{r_4})^{i-j} \right\} \times (1 - p_{tx})^{n-i}
 \end{aligned}$$

To obtain the equilibrium state probabilities, we apply the Jacobi Method by initially setting

$$\begin{aligned}
 b_{r_i, optCW^{r_i}} &= \frac{optCW^{r_i}}{\sum_{i=1}^4 (optCW^{r_i} + \sum_{p=0}^7 cw_p^{r_i})}, \\
 b_{r_i, cw_p^{r_i}} &= \frac{cw_p^{r_i}}{\sum_{i=1}^4 (optCW^{r_i} + \sum_{p=0}^7 cw_p^{r_i})}, \quad \text{and } p_{tx}, p_{tx}^{r_i} \text{ as}
 \end{aligned}$$

$$p_{tx} = \sum_{i=1}^4 \left( \frac{b_{r_i, optCW^{r_i}}}{optCW^{r_i}} + \sum_{p=0}^7 \frac{b_{r_i, cw_p^{r_i}}}{cw_p^{r_i}} \right)$$

$$p_{tx}^{r_i} = \frac{b_{r_i, optCW^{r_i}} / optCW^{r_i} + \sum_{p=0}^7 b_{r_i, cw_p^{r_i}} / cw_p^{r_i}}{p_{tx}} \quad \text{for } i \in \{1, 2, 3, 4\}$$

We assume that user traffic arrives constantly and thus the probability  $p_{timeout}^{r_i}$  of timeout event approaches zero for all data rates. During the Jacobi solving process, variables  $p_{tx}$ ,  $p_{tx}^{r_i}$ ,  $p_{tr}$ ,  $p_{succ}^{r_i}$ , and  $p_{fail}^{r_i}$  keep being updated in each iteration until the convergence condition is met.

After solving the equations and obtaining the values for all variables, we can derive the theoretic system capacity  $\rho = \frac{\bar{m}}{t_v}$ , where  $t_v$  is the virtual transmission time and  $\bar{m}$  is the average data bits successfully transmitted in  $t_v$ . Assume link propagation delay is negligible. Let  $t_{IDLE}$ ,  $t_{SIFS}$ ,  $t_{DIFS}$ , and  $t_{PLCP}$  denote the durations of a time slot, SIFS, DIFS, and physical-layer overhead, whereas  $L_{APP}$ ,  $L_{MPDU}$ , and  $L_{ACK}$  indicate the lengths of application payload bits, MAC protocol data unit, and ACK frame, respectively. Define  $T_{succ}^{r_i}$  as the average time a station spends in a successful trans-

TABLE III  
PARAMETER SETTINGS IN OUR ANALYSIS AND COMPARISON OF THEORETIC BOUNDS AND SIMULATIVE RESULTS

Parameter	Value
$r_1/r_2/r_3/r_4/r_{ACK}$	1/2/5.5/11/1 Mbps
$t_{IDLE}/t_{DIFS}/t_{SIFS}/t_{PLCP}$	20/50/10/192 $\mu$ sec
$L_{APP}/L_{MPDU}/L_{ACK}$	8000/8448/112 bits

# flows	Analytic (Mbps)	Simulative (Mbps)
10	3.7974	3.5010
15	3.7801	3.4173
20	3.7691	3.3236
25	3.7710	3.3205
30	3.7707	3.2992
35	3.7657	3.2731
40	3.7661	3.2171

mission using rate  $r_i$ , whereas  $T_{fail}^{r_i}$  as the time spent in a collision with the slowest station using rate  $r_i$ . Then we have

$$T_{succ}^{r_i} = \left( t_{PLCP} + \frac{L_{MPDU}}{r_i} \right) + t_{SIFS} + \left( t_{PLCP} + \frac{L_{ACK}}{r_{ACK}} \right) + t_{DIFS}$$

$$T_{fail}^{r_i} = \left( t_{PLCP} + \frac{L_{MPDU}}{r_i} \right) + t_{DIFS} + T_o$$

where  $r_{ACK}$  is the basic rate used to transmit the ACK frame and  $T_o = t_{SIFS} + t_{PLCP} + \frac{L_{ACK}}{r_{ACK}}$  which indicates the time spent waiting for accessing the channel again. Hence we have  $T_{succ}$  (duration of a transmission success) =  $\sum_{i=1}^4 p_{succ}^{r_i} \times T_{succ}^{r_i}$  and  $T_{fail}$  (duration of a failure) =  $\sum_{i=1}^4 p_{fail}^{r_i} \times T_{fail}^{r_i}$ .

Consequently, we obtain the system throughput  $\rho$  expressed by

$$\rho = \frac{\bar{m}}{t_v} = \frac{p_{tr} \times p_{succ} \times L_{APP}}{(1 - p_{tr}) \times t_{IDLE} + p_{tr} \times T_{succ} + p_{tr} \times T_{fail}}$$

Table III summarizes the parameter settings adopted in our analytic calculations and throughput comparison to the simulative achievements in a symmetric environment (with number of active flows growing from 10 up to 40). The results demonstrate that the simulative data are quite consistent with analytic predictions, hence validating the EARC performance.

## VI. CONCLUSION

We propose a practical EARC protocol for tuning the rate and backoff parameters in an IEEE 802.11 multi-rate environment. By utilizing reserved bits in 802.11 PLCP header and one extra byte carried in DATA packet, EARC incurs little communication overhead despite its closed-loop (receiver-assisted) nature. Instead of trying to obtain the *SINR* value (which is not practically obtainable), EARC decides on the best rate according to an empirically derived rate selection reference (RSR) table at the receiver. Furthermore, the receiver also assists in tuning the transmitter-estimated optimal contention window. With proper interaction of the two parameters,

our simulation results show that the proposed EARC protocol effectively improves the IEEE 802.11 system performance through its unified design intelligence.

## REFERENCES

- [1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *Proc. 2005 ACM MobiCom*.
- [2] J. C. Bicket, "Bit-rate selection in wireless networks," Master's thesis, Massachusetts Institute of Technology, Feb. 2005.
- [3] C.-C. Chen, H. Luo, E. Seo, N. H. Vaidya, and X. Wang, "Rate-adaptive framing for interfered wireless networks," in *Proc. 2007 IEEE INFOCOM*, pp. 1325–1333.
- [4] G. Hollan, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. 2001 ACM MobiCom*, pp. 236–251.
- [5] G. Judd and P. Steenkiste, "Using emulation to understand and improve wireless networks and applications," in *Proc. 2005 USENIX NSDI*.
- [6] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Tech. J.*, vol. 2, no. 2, pp. 118–133, 1997.
- [7] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: collision-aware rate adaptation for IEEE 802.11 WLANs," in *Proc. 2006 IEEE INFOCOM*, pp. 1–11.
- [8] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proc. 2004 ACM MSWiM*.
- [9] T.-Y. Lin and J. C. Hou, "Interplay of spatial reuse and SINR-determined data rates on CSMA/CA-based, multi-hop, multi-rate wireless networks," in *Proc. 2007 IEEE INFOCOM*, pp. 803–811.
- [10] D. Qiao and S. Choi, "Fast-responsive link adaptation for IEEE 802.11 WLANs," in *Proc. 2005 IEEE ICC*, pp. 3583–3588.
- [11] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proc. 2002 ACM MobiCom*, Sept. 2002.
- [12] Y. Xi, B.-S. Kim, J.-B. Wei, and Q.-Y. Huang, "Adaptive multirate auto rate fallback protocol for IEEE 802.11 WLANs," in *Proc. 2006 IEEE MILCOM*, pp. 1–7.
- [13] Q. Xia and M. Hamdi, "Smart sender: a practical rate adaptation algorithm for multirate IEEE 802.11 WLANs," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1764–1775, May 2008.
- [14] H.-Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. 2006 ACM MobiCom*, pp. 146–157.
- [15] J. Choi, J. Na, K. Park, and C.-K. Kim, "Adaptive optimization of rate adaptation algorithms in multi-rate WLANs," in *Proc. 2007 IEEE ICNP*, pp. 144–153.
- [16] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing wireless packet losses in 802.11: separating collision from weak signal," in *Proc. 2008 IEEE INFOCOM*, pp. 735–743.
- [17] A.-C. Li, T.-Y. Lin, and C.-Y. Tsai, "ARC: joint adaptation of link rate and contention window for IEEE 802.11 multi-rate wireless networks," in *Proc. 2009 IEEE SECON*, pp. 1–9.
- [18] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–799, Dec. 2000.
- [19] D.-J. Deng, C.-H. Ke, H.-H. Chen, and Y.-M. Huang, "Contention window optimization for IEEE 802.11 DCF access control," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 5129–5135, Dec. 2008.
- [20] Y. Kwon, Y. Fang, and H. Latchman, "A novel MAC protocol with fast collision resolution for wireless LANs," in *Proc. 2003 IEEE INFOCOM*, pp. 853–862.
- [21] C. Wang, B. Li, and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 1235–1246, July 2004.
- [22] S.-R. Ye and Y.-C. Tseng, "A multi-chain backoff mechanism for IEEE 802.11 WLANs," *IEEE Trans. Veh. Technol.*, vol. 55, no. 5, pp. 1613–1620, Sep. 2006.
- [23] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 121–132, Aug. 2004.
- [24] A. Vlavianos, L. Law, I. Broustis, S. Krishnamurthy, and M. Faloutsos, "Assessing link quality in IEEE 802.11 wireless networks: which is the right metric?" in *Proc. 2008 IEEE PIMRC*, pp. 1–6.
- [25] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
- [26] W. Chu and Y.-C. Tseng, "Performance analysis of IEEE 802.11 DCF in a multi-rate WLAN," *IEICE Trans. Commun.*, vol. E90-B, no. 10, pp. 2836–2844, Oct. 2007.
- [27] H. Park and C.-K. Kim, "Performance analysis of multi-rate IEEE 802.11 WLANs with channel error," in *Proc. 2007 IEEE ICAC*, pp. 1479–1481.
- [28] D. Qiao, S. Choi, and K. G. Shin, "Goodput analysis and link adaptation for IEEE 802.11a wireless LANs," *IEEE Trans. Mobile Comput.*, vol. 1, pp. 278–292, Oct. 2002.
- [29] G. Sharma, A. Ganesh, and P. Key, "Performance analysis of contention based medium access control protocols," *IEEE Trans. Inf. Theory*, vol. 55, no. 4, pp. 1665–1682, Apr. 2009.



**Ting-Yu Lin** (ting@cm.nctu.edu.tw) obtained her Ph.D. degree in Computer Science and Information Engineering (CSIE) from the National Chiao Tung University, Taiwan. When she graduated, she was recipient of The Phi Tau Phi Scholastic Honor award. From Jun. 2003 to Feb. 2004, Dr. Lin was affiliated with Massachusetts Institute of Technology (MIT) as a research scientist. She worked for Industrial Technology Research Institute of Taiwan from Mar. 2004 to Aug. 2005 as a software engineer. Afterwards, she joined the University of Illinois at Urbana-Champaign (UIUC) as a postdoctoral research associate in the period of Sept. 2005 to Aug. 2006 under both the government and university sponsorship. She is currently an assistant professor with the Department of Electrical Engineering, Institute of Communications Engineering, National Chiao Tung University of Taiwan. Her research interests include wireless communications, mobile computing, WLANs/WPANs, wireless sensor/mesh networking, and green computing. Dr. Lin is a member of ACM and IEEE.



**Ching-Yi Tsai** (cytsai.cm96g@nctu.edu.tw) received her M.S. degree in Communications Engineering from the National Chiao Tung University, Hsinchu, Taiwan, in October 2010. She is currently working as a software engineer at Wistron NeWeb Corporation. Her research interests include wireless networking and performance optimization.



**Kun-Ru Wu** (wufish.cm96g@nctu.edu.tw) received his B.S. degree in Electrical Engineering from the National Chung Hsing University, Taichung, Taiwan, in June 2007. He is currently a Ph.D. student with the Institute of Communications Engineering at National Chiao Tung University, Hsinchu, Taiwan. His research interests include wireless mesh networks and linux-based system prototyping. He is a student member of IEEE.