Open-Loop Link Adaptation for Next-Generation IEEE 802.11n Wireless Networks

Qiuyan Xia, *Student Member, IEEE*, Mounir Hamdi, *Senior Member, IEEE*, and Khaled Ben Letaief, *Fellow, IEEE*

***Abstract*—In this paper, we propose a novel open-loop autorate fallback algorithm (i.e., the AutoRate Fallback for High-Throughput (ARFHT) algorithm) for the emerging high-throughput (HT) IEEE 802.11n wireless networks. ARFHT extends the legacy link adaptation algorithms for single-input–single-output (SISO) wireless networks to be applicable in the context of multiple-input–multiple-output (MIMO)-based 802.11n wireless networks. It adapts the MIMO mode in terms of spatial multiplexing and spatial diversity, which are the two fundamental characteristics of MIMO technology. It also modifies the link estimation and probing behavior of legacy SISO algorithms. The combined adaptation to the appropriate MIMO mode and the modulation coding scheme together achieves more efficient channel utilization. We will present in detail the design guidelines and key ideas for the ARFHT algorithm. A comprehensive simulation study using ns-2 demonstrates that ARFHT achieves excellent throughput and packet error-rate performance in diverse environments and is highly responsive to time-varying link conditions with minimum computational complexity and protocol overhead.**

***Index Terms*—Link adaptation, modulation and coding schemes (MCSs), multiple-input–multiple-output (MIMO), open loop, SDM, STBC, 802.11n.**

I. Introduction

With the growing demands for faster and higher-capacity wireless local area networks (WLANs), the IEEE 802.11 Task Group n (TGn) seeks to achieve higher physical (PHY) layer data rates and improved medium access control (MAC) efficiency in the next-generation WLAN standard termed 802.11n. It is designed to be backward compatible with 802.11a/b/g and will improve the peak throughput to at least 100 Mb/s, which is measured at the MAC data Service Access Point (SAP) [1]. The Draft Specification 2.0 and 3.0 [2] were approved in March and November 2007, respectively, incorporating or responding to the thousands of technical changes noted for Draft 1.0. The finally ratified standard is expected to come by 2009. It defines an HT-PHY based on multiple-input–multiple-output (MIMO) orthogonal frequency-division multiplexing (OFDM) technologies. The HT-MAC enhancements seek to improve the application layer throughput by mitigating the medium wastage due to excessive contention, preamble overheads, and interframe spacing.

Similar to the single-input–single-output (SISO) WLAN standards, 802.11n also specifies a set of modulation and coding schemes (MCSs), each regulating the modulation, coding, and number of spatial channels transmitted on the channel. The actual MCS can adaptively be selected to suit the MIMO channel conditions. It is known that higher level modulation and coding require higher signal-to-noise ratios (SNRs) to maintain a small bit error rate (BER). In addition, the MIMO PHY supports two MIMO modes of operation: *spatial diversity* for better signal quality and *spatial multiplexing* (SM) for higher throughput [3]. These two modes can be adjusted by sending data over a variable number of spatial streams (N_{SS}). The general observation is that at high SNR regions, we need to use multiplexing as long as the packet error rates are small to achieve better throughput. In contrast, we need to have mechanisms to detect persistent channel deteriorations and switch to diversity to increase the transmission reliability at low SNR regions.

Link adaptation in a SISO system is used to dynamically adjust the modulation and coding under the time-varying link qualities to maximize certain performance metrics (e.g., throughput). As for a MIMO-based wireless system, the selection of a suitable combination of modulation, coding, and MIMO mode (SM, spatial diversity, or a hybrid one) should more carefully be investigated. The 802.11n draft standard supports both open-loop and closed-loop link adaptations. The closed-loop operation assumes that perfect channel knowledge is available at the transmitter, either through explicit feedback from the receiver using specific control frames or through channel sounding and calculation between the transmitter and the receiver. However, this approach incurs a lot computation complexity and communication overhead.

In this paper, we focus on developing an open-loop link adaptation algorithm implemented at the MAC layer based on the most recent PHY draft for 802.11n [2]. The algorithm should highly be adaptable to a variety of wireless environments that may change in a very short time. It requires no complicated channel state calibration at the transmitter, and no communication overhead is involved since only the transmitter's statistics are utilized. Moreover, the algorithm should be versatile to diverse traffic conditions and act in an application-aware manner. Finally, this open-loop extension is particularly important in heterogeneous 802.11 WLANs to maintain seamless interoperability and coexistence with legacy devices, which typically utilize open-loop link adaptation.

Manuscript received June 27, 2008; revised December 3, 2008. First published January 19, 2009; current version published August 14, 2009. This work was supported by the Research Grants Council under Contract HKUST610307. The review of this paper was coordinated by Prof. M. Guizani.

Q. Xia and M. Hamdi are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: xiaqy@cse.ust.hk; hamdi@cse.ust.hk).

K. B. Letaief is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: eekhaled@ece.ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2013234

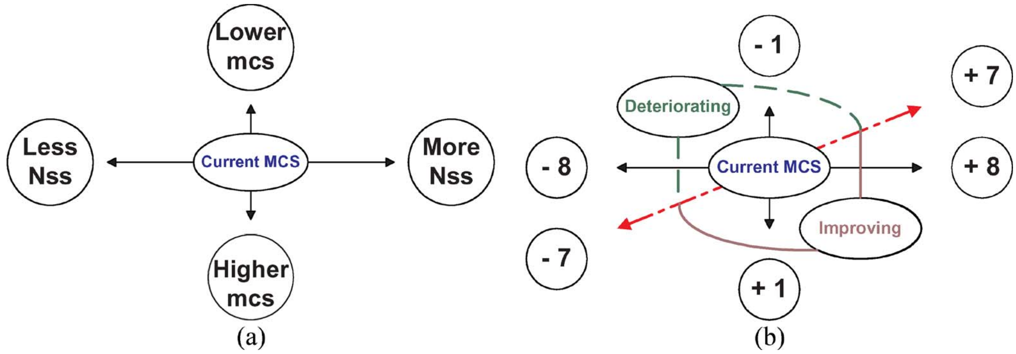


Fig. 1. MCS search directions under changing link qualities. (a) Two-dimensional MCS search process. (b) Actual probing directions.

In the proposed AutoRate Fallback for High-Throughput (ARFHT) algorithm, a transmission scheme is determined according to the previous transmission history. Different from the SISO case, it is now a 2-D adaptation: the search for an appropriate modulation and coding of all spatial streams and for a MIMO mode that leverages spatial diversity and/or SM. Specifically, we first derive a relationship that allows the transmitter to estimate the channel quality dynamics by observing the link layer acknowledgement (ACK) and the receive signal strength indicator (RSSI). Intuitively, this relationship should account for the transmitter's "credits" accumulated in the previous transmissions for its future probing. To do so, we maintain several statistic counters that are dynamically updated. The search behavior is determined based on the predicted channel quality dynamics, namely, the vertical search regarding the modulation and coding adjustment and the horizontal search for N_{SS} adjustment [refer to Table II and Fig. 1(a)]. To make the search processes efficient, we propose a simple and novel link-probing method that accounts for both dimensions. We then design the open-loop link adaptation rule, which is basically a threshold-based scheme with the goal of throughput maximization [or. alternatively, minimizing the expected transmission time (ETT)]. The MCS is changed when one or more counters exceed the corresponding thresholds. The main concern here is to make all the parameters adaptive to instantaneous channel conditions.

This paper is organized as follows. In Section II, a brief introduction to the 802.11n draft standard is provided. The related work is summarized in Section III. Section IV first presents the link quality estimation (LQE) and link-probing schemes and then describes the proposed open-loop link adaptation algorithm (i.e., ARFHT) in detail. Section V shows the simulation methodology and implementations of the abstract MIMO PHY model and the ARFHT algorithm in ns-2 simulations, followed by the performance evaluation. The conclusion and future work are given in Section VI.

II. IEEE 802.11n

A. IEEE 802.11n MIMO-OFDM PHY

The main PHY layer enhancement proposed for IEEE 802.11n is the use of multiple-transmit and multiple-receive antennas at the stations. The basic MIMO channel model is a matrix of transfer functions between antennas, which is represented by (relevant MIMO/802.11n terms are shown in Table I)

TABLE I
MIMO/802.11n TERMS

N_{TX} (N_{RX})	Number of transmit (receive) antennas
STBC	Space-Time Block Coding
SDM	Spatial Division Multiplexing
MCS	Modulation Coding Scheme
HT-SIG	High Throughput Signal Field

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_{\mathrm{RX}}} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{21} & \cdots & h_{N_{\mathrm{TX}},1} \\ h_{12} & h_{22} & \cdots & h_{N_{\mathrm{TX}},2} \\ \vdots & \vdots & \vdots & \vdots \\ h_{1N_{\mathrm{RX}}} & h_{2N_{\mathrm{RX}}} & \cdots & h_{N_{\mathrm{TX}},N_{\mathrm{RX}}} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_{\mathrm{TX}}} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{N_{\mathrm{RX}}} \end{bmatrix} \tag{2}$$

where

$$\mathbf{x} = [x_1 \; x_2 \; \ldots \; x_{N_{\mathrm{TX}}}]^T$$

is the input symbol vector

$$\mathbf{y} = [y_1 \; y_2 \; \ldots \; y_{N_{\mathrm{RX}}}]^T$$

is the output symbol vector, and

$$\mathbf{n} = [n_1 \; n_2 \; \ldots \; n_{N_{\mathrm{RX}}}]^T$$

is the complex additive white Gaussian noise (AWGN) vector with zero mean and variance σ^2. Here, the superscript "$[\;]^T$" stands for the transpose operation, and

$$\mathbf{H} = [h_{ij}]_{N_{\mathrm{RX}} \times N_{\mathrm{TX}}} \; (1 \leq i \leq N_{\mathrm{TX}}, \; 1 \leq j \leq N_{\mathrm{RX}})$$

is the channel matrix, where each coefficient is a complex Gaussian random variable that models the fading gain between the ith transmit and jth receive antenna. OFDM directly extends to MIMO channels, with the inverse fast Fourier transform (IFFT)/fast Fourier transform (FFT) and cyclic prefix (CP) operations being performed at each of the transmit and receive

TABLE II
HT BASIC MCS FOR MANDATORY 20-MHz MODES (800-ns GUARD INTERVAL)

MCS Index (Mbps) Modulation & coding $\backslash N_{SS}$	1	2	3	4
$BPSK, 1/2$	0 (6.5)	8 (13.0)	16 (19.5)	24 (26.0)
$QPSK, 1/2$	1 (13.0)	9 (26.0)	17 (39.0)	25 (52.0)
$QPSK, 3/4$	2 (19.5)	10 (39.0)	18 (58.5)	26 (78.0)
$16QAM, 1/2$	3 (26.0)	11 (52.0)	19 (78.0)	27 (104.0)
$16QAM, 3/4$	4 (39.0)	12 (78.0)	20 (117.0)	28 (156.0)
$64QAM, 2/3$	5 (52.0)	13 (104.0)	21 (156.0)	29 (208.0)
$64QAM, 3/4$	6 (58.5)	14 (117.0)	22 (175.5)	30 (234.0)
$64QAM, 5/6$	7 (65.0)	15 (130.0)	23 (195.0)	31 (260.0)

antennas. The combined MIMO–OFDM solution decouples the wideband frequency-selective MIMO channel into a set of $N_{\mathrm{RX}} \times N_{\mathrm{TX}}$ parallel MIMO channels. Each subcarrier can be characterized at discrete frequencies $k\Delta f$ by $\mathbf{H}(k)$. The elements of $\mathbf{H}(k)$ are complex channel coefficients representing the gains between each pair of transmit and receive antennas. Therefore, the received signal in a MIMO–OFDM system in subcarrier k can be represented by $\mathbf{y}(k) = \mathbf{H}(k)T[\mathbf{x}(k)] + \mathbf{n}(k)$, where $T[\cdot]$ is a transmit spatial processing (if any) on the transmitted symbol vector [4].

TABLE III
ENCODING RULES WHEN ANTENNAS OUTNUMBER SPATIAL STREAMS

N_{TX}	N_{SS}	First SS	Second SS	Third SS
2	1	Coded across antennas 1 and 2	N/A	N/A
3	2	Coded across antennas 1 and 2	Transmitted on antenna 3	N/A
4	2	Coded across antennas 1 and 2	Coded across antennas 3 and 4	N/A
4	3	Coded across antennas 1 and 2	Transmitted on antenna 3	Transmitted on antenna 4

TABLE IV
DETERMINING THE NUMBER OF SPACE–TIME STREAMS

N_{SS} from MCS field	STBC field	N_{STS}
1	0	1
1	1	2
2	0	2
2	1	3
2	2	4
3	0	3
3	1	4
4	0	4

B. Link Adaption for 802.11n

Previous work [5] showed that sending the maximum number of independent data streams (i.e., SM) is rarely the best strategy for maximizing the achievable data rate for a target BER and received Signal-to-Noise-and-Interference Ratio (SNIR). Instead, the mode of operation that maximizes the data rate involves a tradeoff between SM and diversity. The optimal diversity–multiplexing tradeoff has theoretically been investigated in [3], [6], and [7]. Different from this work, here, we take a practical approach based on instantaneous channel quality estimations. In most cases, an antenna will be used for each spatial stream. There may be cases when the number of antennas is greater than the number of spatial streams. Transmitting a single spatial stream across multiple antennas using Space-Time Block Code (STBC) can greatly improve the reliability, particularly in small SNR regions.

Similar to the IEEE 802.11a/b/g standards, the 802.11n draft provides multiple data rates by using different modulation, coding schemes, and modes of multiple antennas. For a *basic* MIMO operation, the same modulation and coding methods are used for all spatial streams of an MCS. Table II shows the basic MCS set with a maximum of four spatial streams ($N_{\mathrm{SS}} = 1, 2, 3, 4$) for mandatory 20-MHz modes. When the number of transmit antenna N_{TX} is larger than the transmitted spatial streams N_{SS}, the additional transmit antenna can be used for diversity gain. In the 802.11n draft, a set of optional robust transmission rates that are only applicable when the number of the space–time streams N_{STS} is greater than the number of spatial streams N_{SS}. Then, N_{SS} spatial streams are mapped to N_{STS} space–time streams, which are again mapped to N_{TX} transmit chains (see Table III). These rates are based on either STBC or hybrid STBC/Spatial Division Multiplexing (SDM) schemes. The number of space–time streams N_{STS} can be determined from the MCS and STBC HT-SIG fields (see Table IV). Therefore, link adaptation for 802.11n is more complicated than those designed for 802.11a/b/g, as we need to determine both the MIMO mode (STBC, SDM, or hybrid STBC/SDM) and the modulation coding scheme.

While the 802.11 a/b/g standards only support open-loop link adaptation, the 802.11n draft specification supports both open-loop and receiver-assisted closed-loop link adaptations. The closed-loop method provides signaling, training, and feedback (e.g., MCS and channel state information (CSI) feedback) mechanisms that are carried in control frames. It also supports transmit beamforming (BF) training through sounding packet and calibration exchange. On the other hand, the open-loop method does not require any explicit feedback from the receiver to the transmitter; instead, it is based on implicit feedbacks by observing the ACK packets. However, the ACK packets only provide binary information to the transmitter whether its choice of the MCS was supported by the channel and does not provide any information to help perform spatial shaping. In this paper, we address these challenges by focusing on the open-loop link

TABLE V
COMPARISONS BETWEEN STATISTIC-BASED AND SIGNAL-MEASUREMENT-BASED SCHEMES

Statistic-based	Signal-measurement-based
ARF [8], AARF [9], CARA [12], RRAA [13], SampleRate [14]	RBAR [10], OAR [11]
collect transmission statistics	measure the signal strength
no requirement for RTS/CTS and no changes to the standard	may require RTS/CTS and entail changes to standard
have to be carefully designed to achieve good performance	good performance (if neglect the overhead)

adaptation, which is easy to deploy and can straightforwardly be used when transmitting to legacy devices.

III. RELATED WORK

A. Link Adaptation for SISO WLANs

A number of link adaptation algorithms for the 802.11a/b/g WLANs have been proposed in the literature. Based on the CSI used for channel quality estimation, they can roughly be divided into two categories, i.e., *statistic-based* and *signal-measurement-based* schemes. Table V gives the comparisons between these two approaches. We briefly review the AutoRate Fallback (ARF) and Adaptive Auto Rate Fallback (AARF) algorithms here, and for the others, see relevant papers and our previous paper [15] for details. The ARF algorithm [8] uses a success threshold $ST = 10$ and a failure threshold $FT = 2$ to decide the rate increment/decrement behavior. However, it cannot quickly react to fast channel variations. In addition, it may overreact when the channel quality is stable over a relatively long period of time. In [9], AARF was proposed to alleviate the regular failure problem of ARF. It adapts ST by using a binary exponential backoff procedure to increase the period between successive failed attempts that are mostly experienced in stable channel conditions, which can better reflect the channel conditions.

B. Closed-Loop Link Adaptation for MIMO WLANs

Most of the existing work assumes a closed-loop operation. In [5], the authors proposed a joint PHY and MAC strategy. The PHY scheme maximizes the data rate for a target BER, given a MIMO channel instance. It selects a subset of the total number of transmit antennas and chooses the best constellation that can be supported on each of the selected antennas. The selected rate setting is then fed back from the receiver to the transmitter in the MAC design. Although this protocol is compatible with 802.11a/g, it still requires changes to the current IEEE 802.11 standards for providing seamless compatibility with 802.11 legacy devices. Moreover, there are control messages (RTS/CTS/ACK) sent using a single antenna before data transmissions, which may introduce considerable overhead.

In [16], the proposed approach evaluates the link quality based on SNR and spatial selectivity information to decide between different diversity and multiplexing modes, i.e., BF, double space–time transmit diversity (D-STTD), and SM. The transmission scheme (i.e., BF, D-STTD or SM) that provides the highest throughput for the predefined fixed error rate is selected for a given link. In [17], the authors proposed a low-complexity technique that uses the soft output (i.e., the reliability values) from the space–time decoder for an efficient link adaptation, taking into account the interference between subchannels. They built up a lookup table offline, and by estimating the channel quality, the most appropriate transmission mode can be selected. Although this approach achieves higher throughput than using other channel quality indicators when the average SNR is larger, it performs slightly worse for lower average SNRs.

Recently, cross-layer approaches, such as those in [18] and [19], have been proposed for MIMO link adaptation, which require MAC and PHY to cooperatively work to take full advantage of the MIMO technologies adopted for 802.11n. The CSI is periodically exchanged between the sender and the receiver. Nevertheless, these designs may lead to costly overheads associated with separate PHY and MAC layer signaling, as well as closed-loop feedback.

C. Open-Loop Link Adaptation for MIMO WLANs

Very few MIMO open-loop link adaptation algorithms have been proposed in the literature. In [20], the authors designed new combinations of STBC and SDM solutions for the MCS set of 802.11n, targeting either an increase of the peak data rate (by SDM) or enhancement of the link (by STBC) or a mixture of the two using a hybrid approach. It shows that the combinations of open-loop multiantenna approaches can benefit the system performance a lot, while avoiding the protocol overhead consumed in the feedback signalization and calibration process. However, this previous work does not specify how to switch among different STBC/SDM schemes.

IV. ARFHT OPEN-LOOP LINK ADAPTATION

The basic idea of link adaptation is to estimate the link quality and dynamically choose the most appropriate transmission scheme. A transmission scheme in this paper comprises two elements (i.e., the MIMO mode with/without space–time code, and the modulation coding scheme). Nevertheless, other parameters can also be considered. The goal of link adaptation is to maximize the throughput or stabilize/upperbound the Frame Error Rate (FER) to some target value (for example, 10%). It decides which MIMO mode and modulation coding scheme should be used and how long the transmitter should stay at the current transmission scheme.

The choice of the transmission scheme has a direct impact on the fundamental properties of a WLAN, e.g., the throughput. A static transmission scheme cannot maximize the network's capacity due to the time-varying channel characteristics and rapid change of traffic conditions. It has been shown in [21] that a dynamic transmission scheme is needed to maximize the utilization in wireless networks to take advantage of the spatial diversity and increased capacity. Controlling the data rate by adjusting the MIMO mode and the modulation and coding can be used to better exploit the benefits of independent fading of multipath propagation. This is accomplished by collecting statistics at the sender such as RSSI, ETT, average retries, frame loss rate, etc.

We propose an open-loop (i.e., without perfect channel knowledge at the transmitter) approach for 802.11n link adaption as there are some basic characteristics of the MIMO mode, modulation, and coding selection using open loop. It uses the same channel coding, the same TX power, and the same modulation and coding for each spatial channel. The feedback consists of only the binary ACK, and there is no sounding, TX BF, TX mode recommendation, or link quality matrix feedback overhead. It can also offer the ability to spread a single encoded stream across multiple antennas without using a closed-loop operation. Finally, it is easy to implement since this strategy can be formulated using the extension of SISO link adaptation while taking into account the new capabilities of 802.11n MIMO PHY. Moreover, this open-loop extension is particularly important in heterogeneous 802.11 WLANs for maintaining seamless interoperability and coexistence with legacy devices, which typically only utilize open-loop link adaptation. To the best of our knowledge, most of the present work on link adaptation is not specially proposed for 802.11 WLANs. In addition, little research has investigated the open-loop link adaption for 802.11n. Our paper serves this purpose to first offer an effective yet simple scheme based on open-loop approaches.

We present our ARFHT open-loop link adaptation algorithm in detail in the following sections. Specifically, the objectives of ARFHT are as follows.

1) Derive a relationship that allows the sender to estimate the channel dynamics based on the ACK and RSSI measured at each RX antenna. Intuitively, this relationship should account for the sender's "credits" accumulated in the previous transmissions for future probing. We are considering maintaining several statistic counters that are dynamically updated.
2) Determine the probing behavior based on the predicted link quality dynamics: the vertical search regarding the modulation and coding adjustment and the horizontal search for $N_{\mathrm{SS}}/N_{\mathrm{STBC}}$ adjustment.
3) Design the open-loop link adaptation rule. This is a threshold-based scheme with the goal of throughput maximization (or alternatively, minimizing the ETT). The Quality-of-Service (QoS) requirements can be added by predefining the FER and access delay restrictions. The main concern is to make all the parameters adaptive to instantaneous channel conditions.

A. Link Quality Estimation

Wireless links are error prone due to interference, noise, fading, mobility, and so on. One of the design goals of ARFHT is to provide efficient LQE in the presence of random wireless errors. In practice, modern devices typically support "multirate retry" to resolve short-lived bursty errors by retransmitting the lost packet at possibly different rates until a success or exceeding the retry limit. Usually, there are four retry series $(r_0/c_0, r_1/c_1, r_2/c_2, r_3/c_3)$, each pair meaning that the attempts at rate r_i are at most c_i times. As soon as the medium is available, the Head-Of-the-Line (HOL) frame is sent at the rate r_0; other retries are automatically carried out if necessary, as specified by the multirate retry series. Finally, the transmission status is returned to the transmission descriptor for the reference of users.

TABLE VI
PARAMETERS USED IN MIMO OPEN-LOOP LINK ADAPTATION

$rate/rate_{cur}$	long-term/current transmission rate
$success/failure$	receive complete ACK/receive partial ACK
$error$	packet dropped
$successV/successH$	Vertical/Horizontal Success Counter
$failureV/failurH$	Vertical/Horizontal Failure Counter
F_{sv}/F_{sh}	Update Function of $successV/successH$
F_{fv}/F_{fh}	Update Function of $failureV/failureH$

The ARFHT also includes multirate retry-based error recovery and distinguishes between different types of ACKs. A "partial ACK" is an ACK received after retransmissions of some lost packet. It can be used for the detection of multiple bursty losses. A "complete ACK" is an ACK received immediately after the first attempt of a packet transmission. Whenever the transmissions of a packet are finished, the transmission status is reported to the MAC high layer in the descriptor. These states include the subfields of "ok," "retries," "ACK signal strength," and so on. Then, the sender invokes the LQE algorithm, which updates the node's various counters of transmission history. Similar to the ARF algorithm in the SISO case, if a row of partial ACKs is received, then LQE classifies the wireless losses to be sustained, which means the channel is deteriorating, and downscale adaptation may be performed. On the other hand, when continuous complete ACKs are received, LQE classifies the wireless link to be in good state or improving, and upscale adaptation may be carried out. Normal transmission/no link adaptation behavior is followed if the LQE indicates stable link qualities.

The LQE algorithm is shown in Algorithm 4, and the parameters used in the open-loop link adaptation are listed in Table VI. Before each packet's transmission, $rate_{\mathrm{cur}}$ is initialized to *rate*, which is the long-term transmission rate dynamically adjusted by ARFHT. During the transmission of a packet, $rate_{\mathrm{cur}}$ is adjusted according to the multirate retry mechanism by checking the function *Lookup (multirate_retry)*. The parameter *rates* records the multiple data rates used for transmitting the last packet if there are more than one attempts. The parameters *success*, *failure*, and *error* are the counters associated with *rate* and are updated after receiving/missing an ACK. They are reset whenever *rate* is changed. Different from these counters, which are simply increased by one or reset to zero, there are still other counters, such as *successV*, *succssH*, *failureV*, and *failureH*, which are updated according to specific functions. For example, the vertical counters are set in consideration of the difference between the perfect and the last transmission time, and the horizontal counters are updated as a function of the RSSI differences across the RX antennas. The reasons are that the vertical counters are used to simulate the behavior in the SISO case, and the horizontal counters are used to measure the diversity of signal strength across the RX antennas. If the last transmission time is close to the perfect transmission time, it indicates the existence of a good channel since the packet can easily get through at the first attempt; on the other hand, it means that the sender is struggling to transmit the packet as the channel cannot support the rate that was last used. In addition, note that using the transmission time to update the counters inherently takes the packet size and transmission rate into account. Similarly, if the RSSI values across different RX antennas are quite different, then it is unjustified to extend the spatial streams by multiplexing. We give the update functions as follows:

$$F_{sv} = perfect\_tx(rate)/last\_tx(rate) \tag{3}$$

$$F_{sh} = \max\left(3/(maxRSSI - minRSSI + 1),\ 1\right) \tag{4}$$

$$F_{fv} = last\_tx(rate)/perfect\_tx(rate) \tag{5}$$

$$F_{fh} = \min\left((maxRSSI - minRSSI + 1),\ 3\right). \tag{6}$$

B. Link Probing

As long as the output of the LQE indicates changing channel qualities, the Link-Probing algorithm is invoked to decide an MCS that may improve the performance. In the SISO case, this decision is simply to upscale or downscale by one. However, in the MIMO case, it is now a 2-D search process characterized by a search for an appropriate modulation and coding of spatial streams, and a MIMO mode that leverages spatial diversity and/or SM; in this case, the vertical search regarding the modulation and coding adjustment, and the horizontal search for $N_{\mathrm{SS}}/N_{\mathrm{STBC}}$ adjustment [see Table II and Fig. 1(a)].

There are three MIMO modes, i.e., SDM, hybrid STBC/SDM, and STBC. Optional robust transmission rates are achieved when $N_{\mathrm{STS}} > N_{\mathrm{SS}}$: N_{SS} spatial streams are mapped to N_{STS} space–time streams, which are mapped to N_{TX} transmit chains, based either on STBC or hybrid STBC/SDM schemes. For the basic MCS set, all of the spatial streams are encoded with the same modulation and coding; for the optional txBF when channel knowledge is available, spatial streams can use different modulation and coding schemes. We only consider the basic streams shown in Table II in our algorithm and simulations. The default configuration is a hybrid STBC/SDM or SDM scheme with moderate N_{SS} and modulation and coding levels.

Various MIMO mode, modulation, and coding combinations are possible. Fig. 1(b) shows the probing directions that we propose. The dash–dot line denotes an intermediate adjustment of the above two probing directions: The upper right direction means decreasing the modulation and coding by one while increasing the N_{SS} by one, and the lower left direction means increasing the modulation and coding by one while decreasing the N_{SS} by one. The main consideration behind this is to make the MCS adjustment achieve a good balance of sensitiveness (more adaptive) and conservativeness (safer).

The detailed probing process and the MCS update are described in Algorithm 2. After a packet's transmission, the *nextProbe()* function is invoked, which outputs a probing direction, if necessary, or no probing at all. As previously presented, there are six probing directions in total, which correspond to the six functions: *mayUpProbe()*, *mayDownProbe()*, *mayLeftProbe()*, *mayRightProbe()*, *mayLeftUpProbe()*, and *mayRightDownProbe()*. We give the details of the *mayUpProbe()* function, whereas the others are neglected here due to page limitation. Nevertheless, we believe that this is enough since all of these probing functions are of the same framework. As we can see from Lines 1–15 in Algorithm 2, the up-probing behavior is threshold based, which requires that a probing is possible only if the *successV* counter exceeds the success threshold *STV*, and the ETT of the new rate is smaller than that of the current rate, or the *successV* counter exceeds the maximum success threshold *maxSTV*. After calling all these probing functions in the *nextProbe()* function, the possible directions are returned and compared, which further decides the final probing directions. The rule is to select the probing direction that would lead to the minimum data rate difference. The intuition is to avoid that the rate changes too drastically. We also distinguish the "recovery failure," which represents a retransmission immediately after a rate increase, from a "normal failure" in the *needRecoveryFallback()* function. In this case, the long-term rate will fall back to the previous rate. Finally, if the long-term rate needs to be adjusted, then the sender calls the *rateUpdate()* function, which updates the long-term rate as well as the rate-relevant parameters.

TABLE VII
THRESHOLDS AND UPDATE RULES

Thresholds	Update interval	Update rule
STV	[minSTV(8), maxSTV(20)]	A-LILD
STH	[minSTH(10), maxSTH(25)]	A-LILD
FTV	minFTV(3), maxFTV(5)	N/A
FTH	minFTH(6), maxFTH(8)	N/A
timeout	40	N/A
enough	10	N/A
ssThresh[4]	3, 8, 16, 31 [dB]	N/A

C. Threshold Update

Several thresholds are maintained at the sender, e.g., the success thresholds *STV* and *STH*, and the failure thresholds *FTV* and *FTH*. *STV* and *FTV* are related to the MCS index adjustment without changing N_{SS} and N_{STBC}. On the other hand, *STH* and *FTH* are referred to change the MCS index together with N_{SS} and N_{STBC}. Their values and update rules are illustrated in Table VII. Here, we use Adaptive Linear Increase Linear Decrease (A-LILD) to make the threshold settings more meaningful. The update details are introduced in Algorithm 3. Intuitively, if the current modulation and coding level is high (indicated by *rate*%8), then the *STV* should be increased more to discourage further increment of the modulation and coding level. As for the *STH*, the current N_{SS} is used in the adjustment process: The higher the N_{SS}, the larger the *STH* value.

D. ARFHT Algorithm

The ARFHT algorithm (see Algorithm 1) is comprised of three components, namely, Link Probing, THreshold

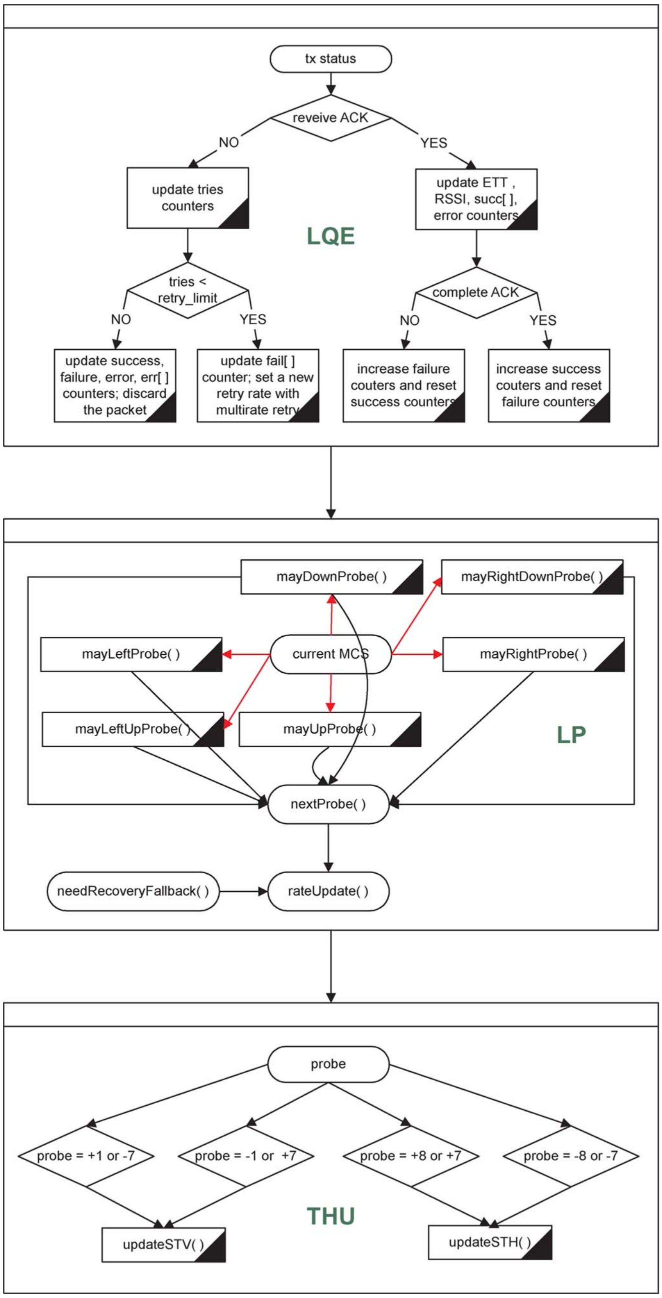


Fig. 2. Flowchart of the ARFHT algorithm.

Update, and LQE, as described in Algorithms 2–4. The flow of the ARFHT algorithm is illustrated in Fig. 2, where the input–output relationship of the various modules is clearly demonstrated.

Algorithm 1 ARFHT Algorithm

1: call LQE to update the statistic counters after each packet transmission;
2: call Link-Probing to decide whether to probe a new MCS, or need recovery fallback if there is a failure immediately after a MCS adaptation;
3: if the MCS is adapted, update the success thresholds by THU;

Algorithm 2 Link-Probing Algorithm

1: *mayUpProbe()*:
2: **if** $(rate+1)\%8! = 0$ **then**
3: **if** $successV >= STV$ **then**
4: $newrate = rate + 1;$
5: **if** *!err[newrate]* **then**
6: **if**$isenough(newrate)\&\&fail[newrate]*2 <= succ[newrate] \;\&\;\& \; ETT[newrate] \; < \; = ETT[rate] * timePercentS$ **then**
7: upProbe $= 1;$
8: **else if** $!isenough(newrate) \;\&\;\& \; perfectETT[newrate] \; < \; = \; ETT[rate] * timePercentS$ **then**
9: upProbe $= 1;$
10: **else if** $successV >= maxSTV$ **then**
11: upProbe $= 1;$
12: **end if**
13: **end if**
14: **end if**
15: **end if**
1: *mayDownProbe()*;
1: *mayLeftProbe()*;
1: *mayRrightProbe()*;
1: *mayLeftUpProbe()*;
1: *mayRightDownProbe()*;
1: *nextProbe()*:
2: calculate all possible link probing directions by calling the *may[.]Probe()* functions;
3: select the probing direction that would lead to the minimum data rate difference;
4: return the MCS difference before and after the rate switch (possible values are $probe = +1/-1/+7/-7/+8/-8$).
1: *needRecoveryFallback()*:
2: **if** $failure > 0,\ \&\&\ recovery\ \&\&\ rate$ increased **then**
3: fallback to the previous *rate*;
4: **end if**
5: $recovery = 0;$
1: *rateUpdate()*:
2: update *rate*, N_{SS} and N_{STBC}:
3: $recovery = probe;$
4: $rate+ = probe;$
5: **if** $probe == -8\| -7$ **then**
6: $N_{\mathrm{SS}} --; \quad N_{\mathrm{STBC}} ++;$
7: **else if** $probe == +8| +7$ **then**
8: $N_{\mathrm{SS}} ++; \quad N_{\mathrm{STBC}} --;$
9: **end if**

Algorithm 3 Threshold Update Algorithm

1: update *STV* and *STH*:
2: **if** $probe == +1\| -7$ **then**
3: $STV+ = max(rate\%8, 4);$
4: $STV = min(STV, maxSTV);$
5: **else if** $probe == -1\| +7$ **then**
6: $STV- = max(rate\%8, 4);$
7: $STV = max(STV, minSTV);$

end if
if $probe == +8 \| +7$ **then**
  $STH+ = N_{\mathrm{SS}}; \quad STH = min(STH, maxSTH);$
else if $probe == -8 \| -7$ **then**
  $STH- = N_{\mathrm{SS}}; \quad STH = max(STH, minSTH);$
end if

Algorithm 4 LQE Algorithm

if receiving an ACK **then**
  update $\overline{ETT}[\,]$ at the *rates* used for tx last packet;
  update $\overline{RSSI}[\,]$, the average RSSIs of recent ACKs measured on RX antennas;
  $timer++; \quad error = 0; \quad succ[rate_{\mathrm{cur}}]++;$
  if receiving a "complete ACK" **then**
    $success++; \quad failure = 0;$
    $failureV = 0; \quad failureH = 0;$
    $successV += F_{sv}(perfect\_tx(rate), last\_tx(rate));$
    $successH+ = F_{sh}(minRSSI, maxRSSI);$
  else if receiving a "partial ACK" **then**
    $success = 0; \quad failure++;$
    $successV = 0; \quad successH = 0;$
    $failureV += F_{fv}(perfect\_tx(rate), last\_tx(rate));$
    $failureH+ = F_{fh}(minRSSI, maxRSSI);$
  end if
else if missing an ACK but $tries < retry\_limit$ **then**
  $fail[rate_{\mathrm{cur}}]++;$
  $rate_{\mathrm{cur}} = Lookup(multirate\_retry);$
else if missing an ACK and $tries >= retry\_limit$ **then**
  $timer++; \quad failure = 0; \quad success = 0;$
  $error++; \quad err[rate]++;$
end if

V. Simulation Methodology

A. Building Modules

To the best of our knowledge, no network simulator has a built-in 802.11n PHY model, as the standard itself is in progress. Therefore, we need to build our own abstract PHY model and incorporate the PHY layer empirical data into the network simulator (i.e., ns-2). Several modules are used to build the abstract PHY model.

1) MATLAB TGn channel models [22]: It contains the MATLAB scripts to generate the MIMO channel matrix H for channel models A to F [23], [24]. There are two main parts. The first consists of scripts to compute a set of correlation matrices for uniform linear array (ULAs), and the other can be used to embed the generated MIMO channel into a broader link-level simulation. This is exactly what we utilize in our simulation.
2) ns-2 802.11 support [25]: This package develops a new 802.11 module for ns-2 with support for ET/SNRT/BER-based PHY models, 802.11a multirate, and 802.11e Hybrid Coordinator Function (HCF) Controlled Channel Access (HCCA) and Enhanced Distributed Channel Access (EDCA). It also contains implementations for the ARF and AARF rate adaptations. Our implementation of ARFHT is within the same framework of these two algorithms.
3) Packet error probability (PER) prediction model for MIMO–OFDM WLAN systems [4]: This method uses postdetection SNRs as an abstraction of the PHY layer, which is sufficient for generating error processes in the system simulations that can accurately reflect the interaction between the MIMO–OFDM PHY layer and the underlying wireless channel. In our simulation, we adopt this abstract PHY model (with some slight modifications) to predict PERs for spatial-spreading MIMO processing. We cite some important results in this paper; see the details in [4]. The effective SNR for spatial stream i is

$$\mathrm{SNR}_{\mathrm{eff}}(i) = 10^{X(i)} \tag{7}$$

where

$$X(i) = \frac{1}{N}\sum_{k=0}^{N-1} \log\gamma(k,i) - a \cdot var\left(\log\gamma(k,i)\right). \tag{8}$$

$\mathrm{SNR}_{\mathrm{eff}}(i)$ is simply the geometric mean of the subcarrier SNRs in stream i, adjusted by the variance. The constant a is used to fit the model to the simulation results obtained with the actual PHY link simulator. $\mathrm{SNR}_{\mathrm{eff}}(i)$ is then used to calculate the coded bit error probability $P_b(i)$ for the data rate used in stream i, which is given in the next section. To calculate $\gamma(k,i)$, we use it in a similar way (with some small changes) as introduced in [4], i.e., spatial spreading.

B. Model Implementation

We build an abstract PHY model (i.e., PER versus SNR relationship) by including the results of the TGn MIMO channel models [22] into the MAC simulator (ns-2-80211 [25]). This is accomplished by modifying the existing modules in Section V-A, as well as adding our own modules. Fig. 3 illustrates the model structure. The detailed steps are as follows.

1) We first configure the MATLAB scripts for TGn channel models to generate the various realizations of the channel matrix $\mathbf{H}$, which are sampled at specific time intervals. Based on $\mathbf{H}$, we implement our own MATLAB scripts for the calculation of the RX power of each spatial stream, and the RX SNR measured on each RX antenna, based on the parameters for 802.11n WLAN. Table VIII summarizes the PHY model parameters used. Then, the RX power and the RX SNR results are incorporated into the MAC (ns-2-80211) simulator. Specifically, the RX power of each spatial stream is used to calculate the postdetection SNR, and the RX SNR at each RX antenna is used as the RSSI reported by the PHY layer to the MAC layer.
2) In the ns-2-80211 simulator [25], a PHY-BER receive model is implemented. Specifically, using the current RX power, the SNIR is calculated against the interference power and the noise power. The SNIR, together with

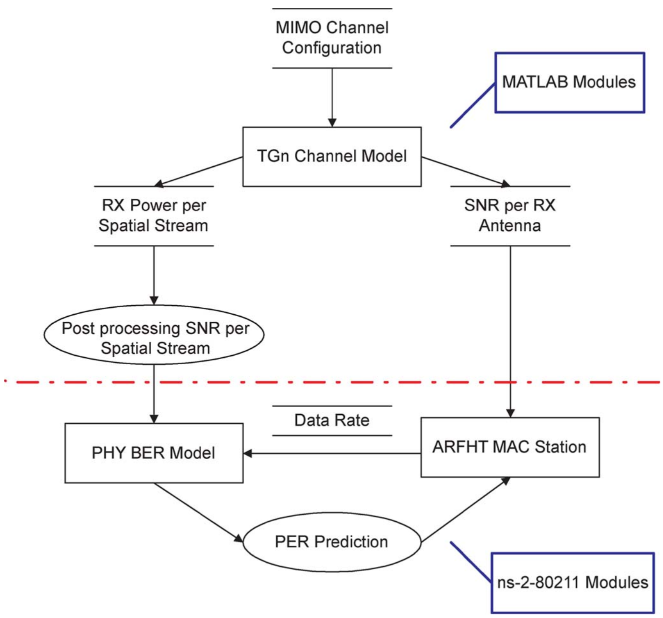


Fig. 3. Module structure in the simulator.

TABLE VIII
PHY MODEL PARAMETERS

Carrier frequency	5.25 [GHz]
Signal Bandwidth	20.0 [MHz]
Data subcarriers	52
Tracking pilots	4
GI	4 [μsec]
Sampling Rate	80 MHz in 20 MHz mode
Receiver Type	MMSE
MIMO processing	Spatial Spreading
Multi-path fading	Enabled
Shadow fading	Disabled
Transmit Power	50.0 [mW]
Noise figure	10.0 [dB]
Antenna Gain (TX/RX)	0.0 [dBi]
Antenna Pattern (TX/RX)	Omni-Directional

the current data rate (modulation coding scheme) and the packet size, is used to calculate the BER and then the PER. In the SISO case, for example, the 802.11a PHY, multiple data rates are supported by using binary phase shift keying (BPSK) and M-ary quadrature amplitude modulation (QAM). Convolutional coding and data interleaving techniques are also used for forward error correction (FEC). The PER can be calculated according to the following steps [26]. First, calculate the BER for different modulations with SNR $\mathrm{SNR}_{\mathrm{eff}}$. For BPSK modulation, the BER P_b is

$$P_b = Q(\sqrt{2 \cdot \mathrm{SNR}_{\mathrm{eff}}}). \tag{9}$$

For M-ary QAM $(M = 4, 16, 64)$, the symbol error probability P_s is

$$P_s = 1 - \left\{1 - \left[2 \cdot \left(1 - \frac{1}{\sqrt{M}}\right) \cdot Q\left(\sqrt{\frac{3}{M-1} \cdot \mathrm{SNR}_{\mathrm{eff}}}\right)\right]\right\}^2 \tag{10}$$

and therefore, with Gary-coded assignment, the BER P_b for M-ary QAM is

$$P_b = P_s \cdot \frac{1}{\log_2 M}. \tag{11}$$

In (9) and (10), the $Q(\cdot)$ function is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt. \tag{12}$$

Then, assuming binary convolutional coding and hard-decision Viterbi decoding, the BER of the decoded bit for a modulation-coding scheme (with FEC) P_μ has a union bound that is

$$P_\mu < \sum_{k=d_{\mathrm{free}}}^{\infty} a_k P_k \tag{13}$$

where d_{free} is the free distance of the convolutional code, a_k is the total number of error events with k bit errors, and P_k is the probability of an incorrect path at distance k from the correct path being chosen by the decoder. P_k can be calculated by

$$P_k = \begin{cases} \frac{1}{2}\binom{k}{k/2}(P_b)^{k/2}(1-P_b)^{k/2} \\ \quad +\sum_{i=1+k/2}^{k}\binom{k}{i}(P_b)^i(1-P_b)^{k-i}, & k \text{ is even} \\ \sum_{i=(1+k)/2}^{k}\binom{k}{i}(P_b)^i(1-P_b)^{k-i}, & k \text{ is odd.} \end{cases} \tag{14}$$

Next, with P_μ, the probability of correctly receiving a chunk of data P_j is given by

$$P_j = (1 - P_\mu)^l \tag{15}$$

where l is the number of bits in the chunk over which the SNIR and the modulation coding scheme are constant. Finally, the PER for the SISO case P_i is calculated by

$$P_i = 1 - \prod_{j=1}^{n} P_j \tag{16}$$

where j is the index of the chunk, and n is the total number of the chunks in a packet. Note that over each chunk of data, both the SNIR and the modulation coding scheme are constant. However, for different chunks in this packet, the SNIR and/or the modulation coding scheme are different. In the MIMO case, we generalize the steps listed above for SISO. P_b and P_j are now calculated for each spatial stream i. Then, the overall PER with N_{ss} spatial streams P_e is calculated as [4]

$$P_e = 1 - \prod_{i=1}^{N_{\mathrm{ss}}}(1 - P_i). \tag{17}$$

3) The ARFHT MAC station also records the RSSI information. Specifically, the PHY layer reports the measured RSSI value from each RX antenna to the MAC layer. This parameter is a measure by the PHY sublayer of the energy observed at the antenna, which is used to

TABLE IX
SIMULATION PARAMETERS

Channel model	802.11n Channel Model B (NLOS)
System configurations	4x4
PPDU length	1000 [bytes]
Target PER	10%
Throughput	$TP = (1 - PER) \times dataRate$
Simulation length	>= 4 [s]

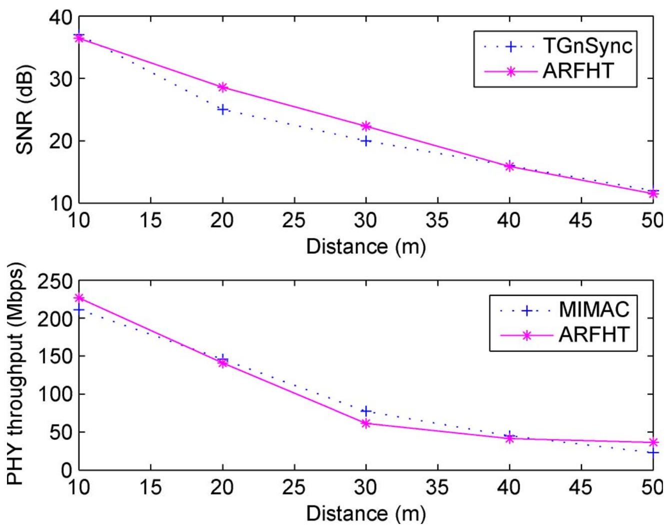


Fig. 4. RX SNR and PHY throughput at different distances.

receive the current frame. The RSSI is measured during the reception of the Physical Layer Convergence Protocol (PLCP) preamble. It is intended to be used in a relative manner and is a monotonically increasing function of the received power. The MAC layer may decide the MCS based on this information. However, the PHY layer is assumed to provide RSSI value of the PLCP header part with inaccuracy. Therefore, we do not use RSSI as the only metric to select the MCS in ARFHT.

4) A new link adaptation algorithm, i.e., ARFHT, is developed and implemented at the MAC layer, as outlined in Section IV.

C. Simulation Results

In this section, we provide the simulation results for basic MIMO operations with configurations, as shown in Table IX. The MCS definitions and indexing for the basic MIMO modes can be found in Table II.

To validate our PHY model, we first provide the received SNR curves at different distances. For each distance, the simulations were run 50 times, and the RX SNR was measured as the average over all receive antennas. Fig. 4 shows that our calculated RX SNRs match well the RX SNRs in TGnSync's simulation results [27]. Later, we can use this RX SNR as the RSSI reported by the PHY layer to the MAC layer in the ns-2-80211 simulator. In the next step, we test the PHY throughput at different distances to validate the PHY-BER receive model. The same settings were used as in the preceding test. We then compare the obtained PHY throughput with the throughput in MIMO-MAC (MIMAC) [5], which is a simplified closed-loop approach. Again, Fig. 4 demonstrates that the open-loop ARFHT algorithm can actually provide very high throughput (HT).

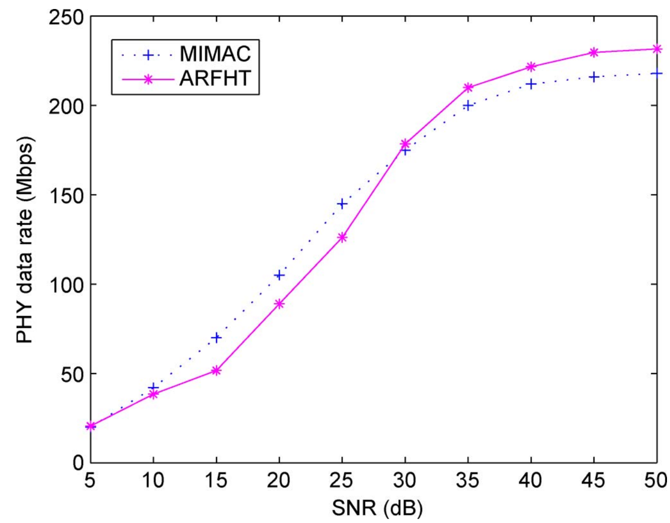


Fig. 5. PHY average data rate measured at different postdetection SNRs.

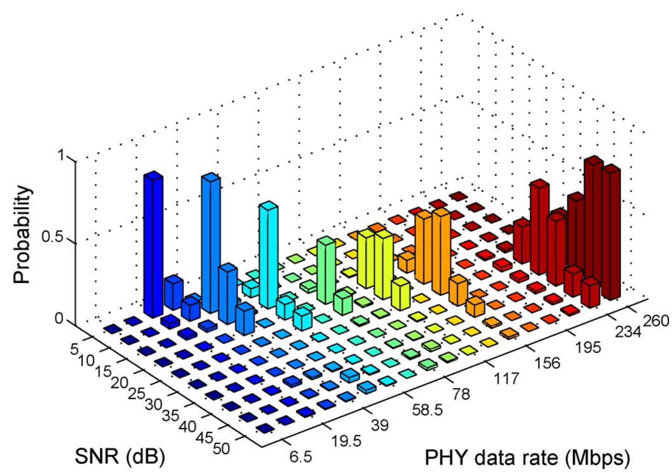


Fig. 6. PHY data rate selection probability.

Fig. 5 shows the average PHY data rates measured in our simulation as a function of the postdetection SNRs, which are obtained by varying the TX–RX distances. At lower SNRs, ARFHT achieves smaller data rates than MIMAC. This is mainly due to the presence of a lossy link, and ARFHT relies on the arrival of ACKs to adapt the data rate as well as requiring not to exceed the target PER. As the SNR improves, the ARFHT can quickly switch to a higher MCS since ACK continuously arrives. Finally, ARFHT arrives at the highest MCS in the basic MCS set.

Fig. 6 shows the rate selection distribution as a function of postdetection SNRs. We get similar observations like in MIMAC: Higher rates can easily be achieved at higher SNRs, since more multiplexing gain is utilized. In addition, for a given SNR level, the selected rate is not fixed. Compared to the results obtained in MIMAC, our algorithm more evenly distributes the selected rates. This is because no exact CSI is available at the sender; therefore, it is more difficult for the sender to fix a proper MCS (hence, the data rate).

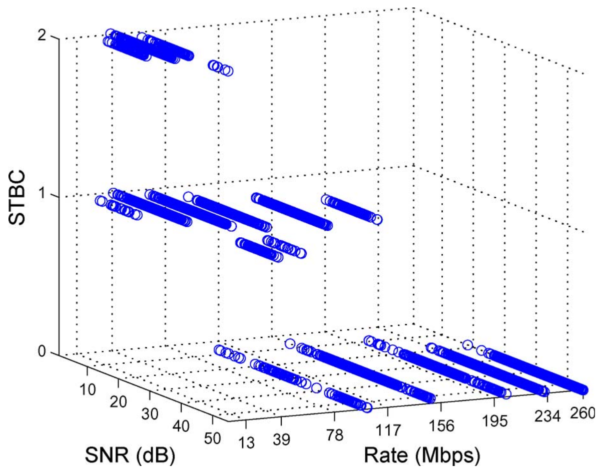


Fig. 7. Example of SM and diversity tradeoff.

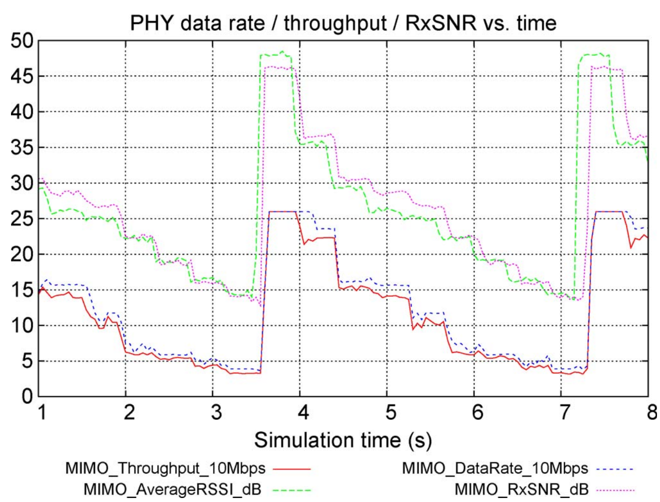


Fig. 8. RX SNR/PHY data rate/throughput under decreasing link quality.

We further study the effect of multiplexing and diversity tradeoff, as shown in Fig. 7. As expected, the open-loop transmit diversity method (i.e., STBC in our case) provides diversity gains (lower BER) at lower SNRs. When the SNR improves, multiplexing provides higher throughput gains with more spatial streams.

Figs. 8 and 9 reflect the adaptiveness of the ARFHT algorithm. In both cases, ARFHT quickly changes the data rates with the variations in link quality. At the same time, ARFHT still maintains a small PER (around 10%). This demonstrates that the performance of the link adaptation is efficient and highly responsive.

D. Further Discussions

The 802.11n draft standard supports the Block ACK (BA) mechanism. There are two variants, i.e., Immediate BA and Delayed BA. For the Immediate BA mechanism, typically, the block size is several packets (4–8), and the transmission time of such a block is within the coherence time of the channel. Therefore, we can still apply our open-loop link adaptation. Whenever a BA is received, since we make the packet length as one of the parameters to update the success/failure counters, these counters will be increased according to the actual successful/failed packet length within the data block. Therefore, it will not affect a lot of the adaptation process. For the Delayed BA mechanism, although the BA is delayed, a normal ACK is sent by the receiver after the BA request. If the normal ACK is correctly received, then we assume a stable link quality and delay the process of updating the counters until the BA is received. Otherwise, if the normal ACK is missing, we temporally retreat to the normal ACK policy and probe the link with a single data packet until we find a proper data rate for the transmission of the next block. Whenever the delayed BA is finally received, we update the counters just like the way we did it in the case of Immediate BA.

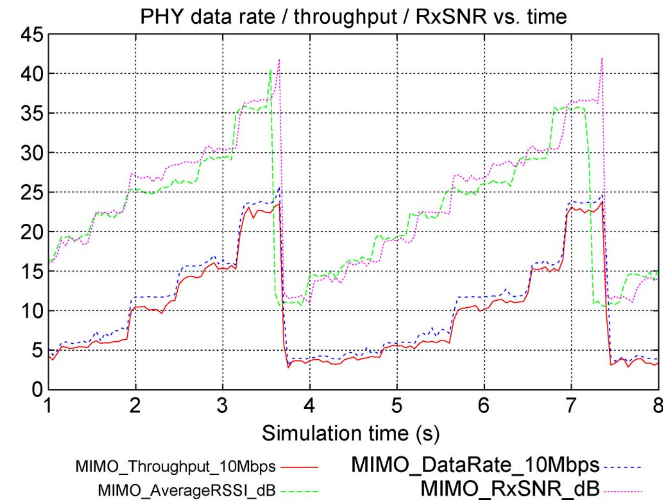


Fig. 9. RX SNR/PHY data rate/throughput under increasing link quality.

VI. Conclusion and Future Work

Link adaptation algorithms are extremely important for WLANs with multirate capabilities. The previously proposed solutions are either SISO WLANs or closed-loop methods for MIMO WLANs. In this paper, we have proposed and evaluated an open-loop algorithm (i.e., ARFHT) that extends the legacy ARF scheme with novel link estimation and probing methods suitable for MIMO WLANs. It maintains the advantages of requiring no changes to the IEEE 802.11 standards and requiring little protocol overhead. ARFHT can adapt to a variety of link conditions and can easily be adopted for future wireless hardware based on the emerging 802.11n standard. Our future work will investigate the following questions: How the errors in the LQE may impact ARFHT's performance; how to extend our scheme to be used with nonbasic MCSs; how to incorporate the MAC enhancements into our scheme, such as BA and frame aggregation; how to design a cross-layer resource allocation framework [28] that jointly considers packet scheduling and link adaptation to maximize the system throughput while satisfying the QoS and fairness requirements of each user; and finally, how to implement our algorithm in state-of-art 802.11n devices and conduct extensive experimentation for performance evaluation in practical scenarios.

REFERENCES

[1] *Draft PAR for High Throughput Study Group*, IEEE 802.11 WG, May 2002.

[2] *Draft Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks-Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 4: Enhancements for Higher Throughput*, IEEE Draft Std. P802.11n/D3.01, Nov. 2007.

[3] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

[4] B. Bjerke, J. Ketchum, R. Walton, S. Nanda, I. Medvedev, M. Wallace, and S. Howard, "Packet error probability prediction for system level simulations of MIMO–OFDM based 802.11n WLANs," in *Proc. IEEE ICC*, Seoul, Korea, May 2005, pp. 2538–2542.

[5] G. Kulkarni, A. Nandan, M. Gerla, and M. Srivastava, "MIMAC: A rate adaptive MAC protocol for MIMO-based wireless networks," Comput. Sci. Dept., Univ. Calif., Los Angeles, CA, Tech. Rep. 040 035, Dec. 2004.

[6] R. Narasimhan, "Spatial multiplexing with transmit antenna and constellation selection for correlated MIMO fading channels," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2829–2838, Nov. 2003.

[7] S. Sfar, L. Dai, and K. B. Letaief, "Optimal diversity-multiplexing tradeoff with group detection for MIMO systems," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1178–1190, Jul. 2005.

[8] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Tech. J.*, vol. 2, no. 3, pp. 118–133, Summer 1997.

[9] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: A practical approach," in *Proc. ACM MSWiM*, Venice, Italy, Oct. 2004, pp. 126–134.

[10] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. ACM MOBICOM*, Rome, Italy, Jul. 2001, pp. 236–251.

[11] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "OAR: An opportunistic auto-rate media access protocol for ad hoc networks," *Wirel. Netw.*, vol. 11, no. 1/2, pp. 39–53, Jan. 2005.

[12] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–11.

[13] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006, pp. 146–157.

[14] J. C. Bicket, "Bit-rate selection in wireless networks," M.S. thesis, Mass. Inst. Technol., Cambridge, MA, Feb. 2005.

[15] Q. Xia and M. Hamdi, "Smart sender: A practical rate adaptation algorithm for multi-rate IEEE 802.11 WLANs," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1764–1775, May 2008.

[16] A. Forenza, M. Airy, M. Kountouris, R. W. Heath, D. Gesbert, and S. Shakkottai, "Performance of the MIMO downlink channel with multimode adaptation and scheduling," in *Proc. IEEE SPAWC*, New York, Jun. 2005, pp. 695–699.

[17] M. Sandell, "Link adaptation for MIMO systems using reliability values," in *Proc. IEEE WCNC*, Las Vegas, NV, Apr. 2006, pp. 1608–1613.

[18] S. S. Parthasarathy and Q.-A. Zeng, "A novel adaptive scheme to improve the performance of the IEEE 802.11n WLANs," in *Proc. AINAW*, May 2007, vol. 2, pp. 334–339.

[19] W. H. Xi, A. Munro, and M. Barton, "Link adaptation algorithm for the IEEE 802.11n MIMO system," in *NETWORKING Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, vol. 4982/2008. Berlin, Germany: Springer-Verlag, May 2008, pp. 780–791.

[20] S. Rouquette-Léveil, A. R. Dias, M. Muck, S. Simoens, M. de Courville, K. Gosse, J.-N. Patillon, K. Blankenship, and B. Classon, "MIMO-based PHY layer techniques for IEEE802.11n," in *Proc. 12th Wireless World Res. Forum Meeting*, Toronto, ON, Canada, Nov. 2004.

[21] Enhanced Wireless Consortium, *HT PHY Specification v1.27*, Dec. 2005. [Online]. Available: http://hydrus.et.put.poznan.pl/~rkotrys/802_11/EWC_PHY_spec_V127.pdf

[22] L. Schumacher, *WLAN MIMO Channel Matlab Program*, May 2004. [Online]. Available: http://www.info.fundp.ac.be/~lsc/Research/IEEE_80211_HTSG_CMSC/distribution_terms.html

[23] *802.11 TGn Channel Models*, IEEE 802.11WG, May 2004.

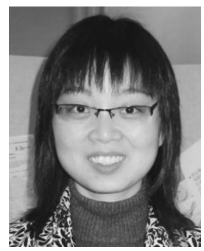
[24] *Description of a MATLAB Implementation of the Indoor MIMO WLAN Channel Model Proposed by the IEEE 802.11 TGn*, May 2004.

[25] M. Lacage, *ns-2 802.11 Support*, Sep. 2005. [Online]. Available: http://code.nsnam.org/mathieu/ns-3-nam

[26] R. Zhu and M. Ma, "Adaptive rate selection scheme based on intelligent learning algorithm in wireless LANs," in *Advances in Computation and Intelligence*, vol. 4683/2007. Berlin, Germany: Springer-Verlag, Aug. 2007, pp. 529–538.

[27] S. A. Mujtaba, *TGnSync Proposal PHY Results*, May 18, 2005. [Online]. Available: https://mentor.ieee.org/802.11/file/04/11-04-0891-06-000n-tgnsync-proposal-phy-results.doc

[28] Y. J. Zhang and K. B. Letaief, "Cross-layer adaptive resource management for wireless packet networks with OFDM signaling," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3244–3254, Nov. 2006.



Qiuyan Xia (S'05) received the B.S. degree in computer science from Nanjing University, Nanjing, China, in 2003 and the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2008.

She is currently with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Her research interests include cross-layer design and optimization, adaptive algorithms, link adaptation, resource management for the IEEE 802.11 WLANs, and quality-of-service provisioning.

Mounir Hamdi (S'90–M'91–SM'06) received the B.S. degree in electrical engineering from the University of Louisiana, Lafayette, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from the University of Pittsburgh, Pittsburgh, PA, in 1987 and 1991, respectively.

Since 1991, he has been a faculty member with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, where he is currently the Head of the department, the Director of the Master of Science in Information Technology, and the Director of the Computer Engineering Laboratory. From 1999 to 2000, he held visiting professor positions with Stanford University, Stanford, CA, and the Swiss Federal Institute of Technology, Lausanne, Switzerland. His general area of research is in high-speed wired/wireless networking, in which he has published more than 250 research publications, received numerous research grants, and graduated more 30 Ph.D./Master students. In addition, he has frequently consulted for companies in the U.S., Europe, and Asia on high-performance Internet routers and switches as well as high-speed wireless local area networks.

Dr. Hamdi is a Member of the University Council and Senate. He is a Senior Member of the Association for Computing Machinery. He is/was on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, *IEEE Communications Magazine*, *Computer Networks*, *Wireless Communications and Mobile Computing*, and *Parallel Computing*. He was a Guest Editor of three special issues of *IEEE Communications Magazine* and two special issues of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He has chaired/cochaired more than ten IEEE international conferences and workshops including the IEEE International High Performance Switching and Routing Conference, the IEEE Global Telecommunications Conference/International Conference on Communications (GLOBECOM/ICC) Optical networking workshop, the IEEE ICC High-Speed Access Workshop, and the IEEE IPPS HiNets Workshop. He is/was the Chair of IEEE Communications Society Technical Committee on Transmissions, Access, and Optical Systems, and Vice-Chair of the Optical Networking Technical Committee, as well as member of the ComSoc technical activities council. He is/was on the technical program committees of more than 150 international conferences and workshops. He has been a frequent keynote speaker to international conferences and workshops. In addition to his research and professional activities, he is also dedicated to university education. He heads The Hong Kong UGC task force on outcome-based education. He also received the Best Ten Lecturers Award, the Distinguished Engineering Teaching Appreciation Award from the Hong Kong University of Science and Technology, and various grants targeted toward the improvement of teaching methodologies.

Khaled Ben Letaief (S'85–M'86–SM'97–F'03) received the B.S. (with distinction), M.S., and Ph.D. degrees from Purdue University, West Lafayette, IN, in December 1984, August 1986, and May 1990, respectively, all in electrical engineering.

In January 1985 and as a Graduate Instructor with the School of Electrical Engineering, Purdue University, he has taught courses in communications and electronics. From 1990 to 1993, he was a faculty member with the University of Melbourne, Melbourne, Australia. Since 1993, he has been with the Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong, where he is currently a Chair Professor and the Head of the Electronic and Computer Engineering Department. He is also the Director of the Hong Kong Telecom Institute of Information Technology. In these areas, he has published over 300 journals and conference papers and given invited keynote talks as well as courses all over the world. His current research interests include wireless and mobile networks, broadband wireless access, orthogonal frequency division multiplexing, cooperative networks, cognitive radio, multiple-input–multiple-output, and beyond third-generation systems.

Dr. Letaief served as a Consultant for different organizations and is the founding Editor-in-Chief of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He served as an elected member of the IEEE Communications Society Board of Governors and an IEEE Distinguished Lecturer. He also served as the Chair of the IEEE Communications Society Technical Committee on Personal Communications and the Chair of the Steering Committee of the IEEE Transactions on Wireless Communications. He has served on the editorial board of other prestigious journals, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS—WIRELESS SERIES (as Editor-in-Chief). He has been involved in organizing a number of major international conferences and events. These include serving as the Co-Technical Program Chair of the 2004 IEEE International Conference on Communications, Circuits, and Systems (ICCCS'04); General Co-Chair of the 2007 IEEE Wireless Communications and Networking Conference (WCNC'07), as well as the Technical Program Co-Chair of the 2008 IEEE International Conference on Communication (ICC'08). In addition to his active research and professional activities, he has been a dedicated teacher committed to excellence in teaching and scholarship. He received the Mangoon Teaching Award from Purdue University in 1990; the Teaching Excellence Appreciation Award from the School of Engineering, HKUST (four times); and the Michael G. Gale Medal for Distinguished Teaching (highest university-wide teaching award and only one recipient per year is honored for his/her contributions). He is the recipient of the 2007 IEEE Communications Society Publications Exemplary Award.