Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs

Daji Qiao, *Student Member*, *IEEE*, Sunghyun Choi, *Member*, *IEEE*, and Kang G. Shin, *Fellow*, *IEEE*

Abstract—Link adaptation to dynamically select the data transmission rate at a given time has been recognized as an effective way to improve the goodput performance of the IEEE 802.11 wireless local-area networks (WLANs). Recently, with the introduction of the new high-speed 802.11a physical layer (PHY), it is even more important to have a well-designed link adaptation scheme work with the 802.11a PHY such that its multiple transmission rates can be exploited. In this paper, we first present a generic method to analyze the goodput performance of an 802.11a system under the Distributed Coordination Function (DCF) and express the expected effective goodput as a closed-form function of the data payload length, the frame retry count, the wireless channel condition, and the selected data transmission rate. Then, based on the theoretical analysis, we propose a novel MPDU (MAC Protocol Data Unit)-based link adaptation scheme for the 802.11a systems. It is a simple table-driven approach and the basic idea is to preestablish a best PHY mode table by applying the dynamic programming technique. The best PHY mode table is indexed by the system status triplet that consists of the data payload length, the wireless channel condition, and the frame retry count. At runtime, a wireless station determines the most appropriate PHY mode for the next transmission attempt by a simple table lookup, using the most up-to-date system status as the index. Our in-depth simulation shows that the proposed MPDU-based link adaptation scheme outperforms the single-mode schemes and the AutoRate Fallback (ARF) scheme—which is used in Lucent Technologies' WaveLAN-II networking devices—significantly in terms of the average goodput, the frame drop rate, and the average number of transmission attempts per data frame delivery.

Index Terms—IEEE 802.11 MAC, IEEE 802.11a PHY, Distributed Coordination Function (DCF), link adaptation, expected effective goodput, system status, dynamic programming.

1 Introduction

The IEEE 802.11 [1] is the first international standard for wireless local-area networks (WLANs) and it has been used widely in most commercial WLAN products available in the market. The IEEE 802.11 MAC specifies two different medium access control (MAC) mechanisms in WLANs: the contention-based Distributed Coordination Function (DCF) and the polling-based Point Coordination Function (PCF). At present, only the mandatory DCF is implemented in the 802.11-compliant products. The 802.11 physical layers (PHYs) provide multiple data transmission rates by employing different modulation and channel coding schemes. For example, the IEEE 802.11b PHY [2] provides four PHY rates from 1 to 11 Mbps at the 2.4GHz band and most 802.11 devices available in the market are based on this PHY. Recently, another emerging high-speed PHY, the IEEE 802.11a PHY [3], has been developed to extend the IEEE 802.11 in the 5GHz Unlicensed National Information Infrastructure (U-NII) band and provides eight PHY modes with data transmission rates ranging from 6 Mbps up to 54 Mbps. While the first generation 802.11a products are available in the market, the 802.11a PHY receives more and more attention thanks to its higher transmission rates as well as the cleaner 5GHz operational frequency band.

The mechanism to select one out of multiple available transmission rates at a given time is referred to as *link adaptation* and the effectiveness of the implemented link adaptation scheme can affect the system performance significantly. For example, due to the heuristic and conservative nature of the link adaptation schemes implemented in most 802.11 devices, the current 802.11 systems are likely to show low bandwidth utilization when the wireless channel presents a high degree of variation. Now, with eight different transmission rates, the 802.11a PHY introduces an even bigger challenge for the link adaptation algorithm design.

In this paper, we present a generic method to analyze the goodout performance of an 802.11a DCF system and derive a closed-form expression of the expected effective goodput. Here, when a wireless station is ready to transmit a data frame, its *expected effective goodput* is defined as the ratio of the expected delivered data payload to the expected transmission time, i.e., the expected bandwidth this station can actually receive after all the overheads are accounted for, including the MAC/PHY overheads, the backoff delay, the interframe intervals, the acknowledgment (Ack) transmission time, and the potential frame retransmission times. Based on the goodput analysis, we propose a novel link adaptation scheme assuming the availability of the wireless channel models. Obviously, in order to deliver a data frame, the higher the PHY rate, the shorter the transmission time in one transmission attempt, but, more likely, the transmission will fail, thus engendering retransmissions. So, there is an inherent tradeoff and our idea is to preestablish a PHY mode table indexed by the *system status* triplet—which consists of the data payload length, the wireless channel

- *D. Qiao and K.G. Shin are with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109. E-mail: {dqiao, kgshin}@eecs.mich.edu.*
- *S. Choi is with the School of Electrical Engineering, Seoul National University, Seoul 151-744, Korea. E-mail: schoi@snu.ac.kr.*

Manuscript received 5 Dec. 2001; revised 18 July 2002; accepted 8 Oct. 2002. For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 1-122001.

1536-1233/02/$17.00 © 2002 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

condition, and the frame retry count. Each entry of the table is the best PHY mode in the sense of maximizing the expected effective goodput under the corresponding system status and is computed by applying the dynamic programming technique. At runtime, a wireless station determines the most appropriate PHY mode for the next transmission attempt by a simple table lookup, using the most up-to-date system status as the index.

Actually, the goodput performance of an 802.11b DCF system can be analyzed in a similar way and the corresponding link adaptation scheme can be designed without much difficulty. In this paper, we focus on the 802.11a systems because the goodput enhancement of an 802.11a DCF system, by using our proposed link adaptation scheme, could be more significant due to the 802.11a PHY's wider range of data transmission rates, i.e., eight different PHY modes.

1.1 Related Work

The commercial WLAN products available in the market have their own proprietary link adaptation schemes. The AutoRate Fallback (ARF) protocol [4], which is used in Lucent Technologies' WaveLAN-II networking devices, is one of the few that are available to the public. It alternates between 1 and 2 Mbps PHY modes, based on the result of keeping track of a timing function and the missed Ack frames. If two consecutive Acks are not received correctly by the sender, the second retry of the current packet and the subsequent transmissions are done at the lower data rate and a timer is started. When either the timer expires or the number of successfully-received Acks reaches 10, the transmission rate is raised to the higher data rate and the timer is cancelled. However, if an Ack is not received for the very next data packet, the transmission rate is lowered again and the timer is restarted. Obviously, this scheme is purely heuristic and cannot react quickly when the wireless channel condition fluctuates.

In recent years, a number of new link adaptation schemes have been proposed for different types of wireless networks. The authors of [5] presented a Receiver-Based Auto-Rate (RBAR) protocol based on the RTS/CTS (Request-To-Send/Clear-To-Send) mechanism by modifying the IEEE 802.11 standard. The basic idea of RBAR can be summarized as follows: First, the receiver estimates the wireless channel quality using a sample of the instantaneously received signal strength at the end of the RTS reception, then selects the appropriate transmission rate based on this estimate and feeds back to the transmitter using the CTS. Then, the transmitter responds to the receipt of the CTS by transmitting the data packet at the rate chosen by the receiver. However, since this protocol requires many changes to the IEEE 802.11 standard, such as the Data, RTS, CTS frame formats, and the physical layer header, it may not be practically useful. Moreover, using RTS and CTS frames all the time is not desirable as it wastes the precious wireless bandwidth when there are no hidden terminals. In [6], two link adaptation schemes were proposed for the General Packet Radio Service (GPRS) development of GSM. One is based on the estimate of the Carrier-to-Interference ratio (C/I) and the other is based on the observation of the block error rate. HIgh PErformance Radio Local Area Network type 2 (HIPERLAN/2) [7] is another wireless broadband access system that has been specified by European Telecommunications Standards Institute (ETSI) project BRAN (Broadband Radio Access Network). Link adaptation is one of the key features of HIPERLAN/2 as it has a PHY that is very similar to 802.11a. The authors of [8] studied the system performance of link adaptation, which uses the C/I as the wireless link quality measurement, for packet data services within HIPERLAN/2. Furthermore, the authors of [9] presented a new algorithm for adaptive modulation and power control in a HIPERLAN/2 network. It first assumes the maximum transmit power and uses the C/I observed at the receiver to determine the proper PHY mode for the next frame transmission to meet the target packet error rate (PER). Then, it reduces the power as much as possible while meeting the target PER.

Note that all the above link adaptation schemes make the PHY mode selection based on monitoring the wireless channel condition. Therefore, they will result in better system performance than those purely heuristic algorithms. However, the common weakness of these schemes is that they provide neither a thorough theoretical analysis on the system performance nor a closed-form relation among the effective goodput, the wireless channel condition, and the PHY mode, which is the most important base for any link adaptation scheme. Moreover, none of these schemes considers how the current frame retry count should affect the PHY mode selection. They all assume implicitly that, once a PHY mode is selected to transmit a data frame, it will remain unchanged for all the potential retransmissions, even when the wireless channel condition changes. In comparison, the authors of [10] analyzed the goodput performance of an IEEE 802.11 WLAN using Lucent Technologies' WaveLAN devices. Since the high-speed 802.11a PHY was not available at that time, the authors assumed the (fixed) Quadrature Phase Shift Keying (QPSK) modulation and analyzed the goodput performance for different sizes of MAC Service Data Units (MSDUs), which are generated by the IEEE 802.2 Logical Link Control (LLC) sublayer. Actually, as indicated in the IEEE 802.11 standard, an MSDU can be fragmented further into smaller MAC frames, i.e., MAC Protocol Data Units (MPDUs), for transmission. In [11], we studied how two adaptive schemes affect the goodput performance of an 802.11a DCF system: dynamic fragmentation of MSDUs and dynamic PHY mode selection for each MPDU transmission. Analysis results show that both adaptive schemes affect the goodput performance in certain variation ranges of the wireless channel condition, but the affecting range of dynamic fragmentation is much smaller than that by changing the PHY mode. In addition, as specified in the IEEE 802.11 standard, once an MSDU has been fragmented, the fragment size will remain unchanged until the end of delivery. For these reasons, we only consider dynamic PHY mode selection in our new MPDU-based link adaptation scheme, which is an important enhancement of the simple MSDU-based adaptive PHY mode selection scheme proposed in [11].

1.2 Organization

The rest of this paper is organized as follows: Section 2 introduces the DCF of the IEEE 802.11 MAC as well as the

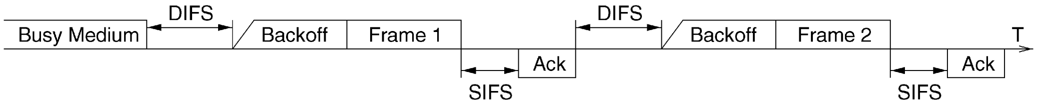


Fig. 1. Timing of successful frame transmissions under the DCF.

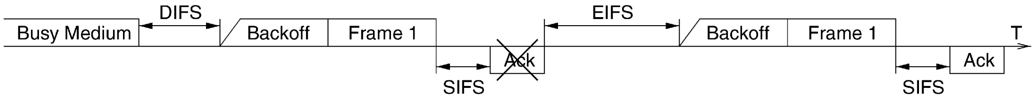


Fig. 2. Frame retransmission due to Ack failure.

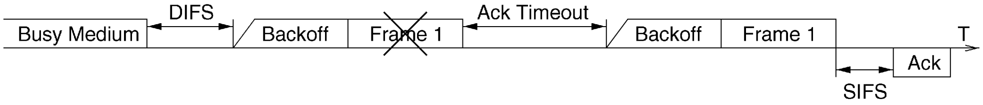


Fig. 3. Frame retransmission due to an erroneous data frame reception.

IEEE 802.11a PHY. The effective goodput analysis of an 802.11a DCF system is shown in Section 3. Section 4 describes our new MPDU-based link adaptation scheme and discusses the implementation issues. Section 5 gives a detailed example on how to establish the best PHY mode table in the MPDU-based link adaptation scheme. Section 6 presents and discusses the evaluation results. Finally, this paper concludes with Section 7.

2 System Overview

2.1 DCF of IEEE 802.11 MAC

The DCF, as the basic access mechanism of the IEEE 802.11 MAC, achieves automatic medium sharing between compatible stations through the use of Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA). Before a station starts transmission, it senses the wireless medium to determine if it is idle. If the medium appears to be idle, the transmission may proceed, else the station will wait until the end of the in-progress transmission. The CSMA/CA mechanism requires a minimum specified gap/space between contiguous frame transmissions. A station will ensure that the medium has been idle for the specified interframe interval before attempting to transmit.

The Distributed InterFrame Space (DIFS) is used by stations operating under the DCF to transmit data frames. A station using the DCF has to follow two medium access rules: 1) the station will be allowed to transmit only if its carrier-sense mechanism determines that the medium has been idle for at least DIFS time, and 2) in order to reduce the collision probability among multiple stations accessing the medium, the station will select a random backoff interval after deferral or prior to attempting to transmit another frame after a successful transmission.

One important characteristic of the IEEE 802.11 MAC is that an acknowledgment (Ack) frame will be sent by the receiver upon successful reception of a data frame. It is only after receiving an Ack frame correctly that the transmitter assumes successful delivery of the corresponding data frame. The Short InterFrame Space (SIFS), which is smaller than DIFS, is the time interval between reception of a data frame and transmission of its Ack frame. Using this small gap between transmissions within the frame exchange sequence prevents other stations—which are required to wait for the medium to be idle for a longer gap (e.g., at least DIFS time)—from attempting to use the medium, thus giving priority to completion of the in-progress frame exchange sequence. The timing of successful frame transmissions is shown in Fig. 1. On the other hand, if an Ack frame is received in error, i.e., received with an incorrect frame check sequence (FCS), the transmitter will recontend for the medium to retransmit the frame after an EIFS (Extended InterFrame Space) interval, as shown in Fig. 2. However, if no Ack frame is received within an SIFS interval, due possibly to an erroneous reception of the preceding data frame, as shown in Fig. 3, the transmitter will contend again for the medium to retransmit the frame after an Ack timeout. Note that, in these figures, a crossed block represents an erroneous reception of the corresponding frame. Moreover, the DCF defines an optional mechanism, which requires that the transmitter and receiver exchange short RTS (Request-To-Send) and CTS (Clear-To-Send) control frames prior to the actual data transmission.

The IEEE 802.11 standard requires that a data frame is discarded by the transmitter's MAC after a certain number of unsuccessful transmission attempts. If the length of a data frame is less than or equal to *dot11RTSThreshold*, the number of transmission attempts is limited by *dot11ShortRetryLimit*, else the maximum number of transmission attempts is set to *dot11LongRetryLimit*. The default values of *dot11ShortRetryLimit* and *dot11LongRetryLimit* are 7 and 4, respectively.

2.2 IEEE 802.11a OFDM PHY

The PHY is the interface between the MAC and the wireless medium, which transmits and receives data frames over the shared wireless medium. The frame exchange between MAC and PHY is under the control of the Physical Layer Convergence Procedure (PLCP) sublayer.

The OFDM has been selected as the modulation scheme for the IEEE 802.11a PHY and the basic principle of OFDM is to divide a high-speed binary signal to be transmitted

TABLE 1
Eight PHY Modes of the IEEE 802.11a PHY

Mode	Modulation	Code Rate	Data Rate	BpS
1	BPSK	1/2	6 Mbps	3
2	BPSK	3/4	9 Mbps	4.5
3	QPSK	1/2	12 Mbps	6
4	QPSK	3/4	18 Mbps	9
5	16-QAM	1/2	24 Mbps	12
6	16-QAM	3/4	36 Mbps	18
7	64-QAM	2/3	48 Mbps	24
8	64-QAM	3/4	54 Mbps	27

over a number of low data-rate subcarriers. There are a total of 52 subcarriers, of which 48 subcarriers carry actual data and four subcarriers are pilots that facilitate phase tracking for coherent demodulation. Each low data-rate bit-stream is used to modulate a separate subcarrier from one of the channels in the 5GHz band. A key feature of the IEEE 802.11a PHY is to provide eight PHY modes with different modulation schemes and coding rates, making the idea of link adaptation feasible and important. BPSK, QPSK, 16-QAM, and 64-QAM are the supported modulation schemes. As listed in Table 1, the OFDM system provides a WLAN with capabilities of communicating at 6 to 54 Mbps. The support of transmitting and receiving at the data rates of 6, 12, and 24 Mbps is mandatory in the IEEE 802.11a PHY. Forward error correction (FEC) is performed by bit interleaving and rate-1/2 convolutional coding. The higher code rates of 2/3 and 3/4 are obtained by puncturing the original rate-1/2 code.

3 Goodput Analysis of an IEEE 802.11a DCF System

When a wireless station is ready to transmit a data frame, its *expected effective goodput* is defined as the ratio of the expected delivered data payload to the expected transmission time. Clearly, depending on the data payload length and the wireless channel conditions, the expected effective goodput varies with different transmission strategies. The more robust the transmission strategy, the more likely the frame will be delivered successfully within the frame retry limit, however, with less efficiency. So, there is a tradeoff and the key idea of link adaptation is to select the most appropriate transmission strategy such that the frame can be successfully delivered in the shortest possible transmission time.

Before proceeding to the details of our new link adaptation scheme, we will first, in this section, analyze the effective goodput performance of an 802.11a DCF system and express the expected effective goodput as a closed-form function of the data payload length (ℓ), the frame retry limit (n_{max})—which can take the value of *dot11ShortRetryLimit* or *dot11LongRetryLimit*, the wireless channel conditions ($\hat{s}$) during all the potential transmission attempts, and the transmission strategy ($\hat{m}$). Here, $\hat{s}$ is a vector of receiver-side SNR (Signal-to-Noise Ratio) values to quantify the wireless channel conditions, $\hat{m}$ is a vector of PHY mode selections, and both $\hat{s}$ and $\hat{m}$ are of length n_{max}.

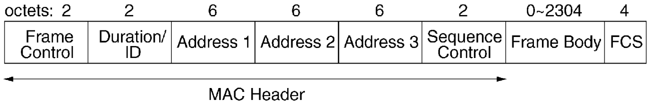


Fig. 4. Frame format of a data frame MPDU.

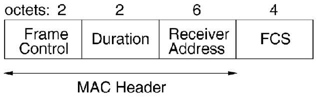


Fig. 5. Frame format of an Ack frame.

3.1 MAC/PHY Layer Overheads

As shown in Fig. 4, in the IEEE 802.11 MAC, each MAC data frame, or MAC Protocol Data Unit (MPDU), consists of the following components:[1] *MAC header*, variable-length information *frame body*, and *frame check sequence* (FCS). The MAC overhead due to the MAC header and the FCS is 28 octets in total. Fig. 5 illustrates the frame format of an Ack frame, which is 14 octets long.

During the transmission, a PLCP preamble and a PLCP header are added to an MPDU to create a PLCP Protocol Data Unit (PPDU). The PPDU format of the IEEE 802.11a PHY is shown in Fig. 6, which includes PLCP preamble, PLCP header, MPDU (conveyed from MAC), tail bits, and pad bits, if necessary. The PLCP preamble field, with the duration of *tPLCPPreamble*, is composed of 10 repetitions of a short training sequence ($0.8\mu s$) and two repetitions of a long training sequence ($4\mu s$). The PLCP header, except the SERVICE field, with the duration of *tPLCP_SIG*, constitutes a single OFDM symbol, which is transmitted with BPSK modulation and rate-1/2 convolutional coding. The six "zero" tail bits are used to return the convolutional codec to the "zero state" and the pad bits are used to make the resulting bit string into a multiple of OFDM symbols. Each OFDM symbol interval, denoted by *tSymbol*, is $4\mu s$. The 16-bit SERVICE field of the PLCP header and the MPDU (along with six tail bits and pad bits), represented by DATA, are transmitted at the data rate specified in the RATE field. Table 2 lists the related characteristics for the IEEE 802.11a PHY, where *tSlotTime*, *aCWmin*, and *aCWmax* will be discussed in the following section.

Note that, while the data frame MPDUs can be transmitted at any supported data rate, all the control frames, including the Ack frames, have to be transmitted at one of the rates in the BSS basic rate set[2] so that they can be understood by all the stations in the same network. In addition, an Ack frame will be transmitted at the highest rate in the BSS basic rate set that is less than or equal to the rate of the data frame it is acknowledging. For example, if

[1] Actually, an additional field of "Address 4" appears in the Wireless Distribution System (WDS) data frames being distributed from one access point to another access point. However, since such WDS frames are rarely used, we do not consider the "Address 4" field in our goodput analysis. Besides, we do not consider the WEP (Wired Equivalent Privacy) option in this paper, which may introduce an extra eight-octet overhead.

[2] Basic Service Set (BSS) is the basic building block of an IEEE 802.11 WLAN. It consists of a set of stations controlled by a single coordination function. BSS basic rate set is the set of data rates that all the stations in a BSS will be capable of using to receive/transmit frames from/to the wireless medium. The BSS basic rate set data rates are preset for all the stations in the BSS.

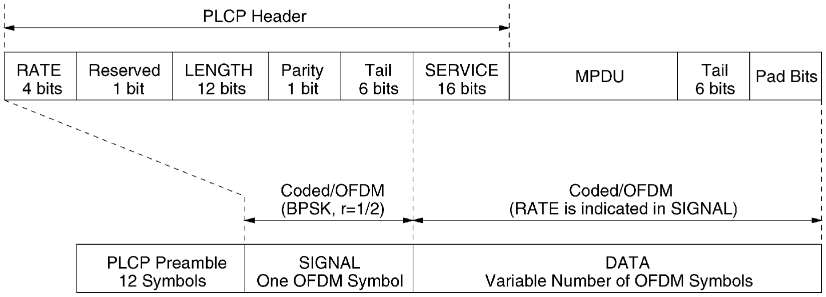


Fig. 6. PPDU frame format of the IEEE 802.11a OFDM PHY.

the BSS basic rate set is {6 Mbps, 12 Mbps, 24 Mbps}[3] and a data frame is transmitted at the rate of 18 Mbps, the corresponding Ack frame will be transmitted at the rate of 12 Mbps.

Based on the above analysis, to transmit a frame with ℓ octets data payload over the IEEE 802.11a PHY using PHY mode m, the transmission duration is:

$$\begin{aligned} T_{data}(\ell, m) &= tPLCPPreamble + tPLCP\_SIG \\ &\quad + \left\lceil \frac{28 + (16+6)/8 + \ell}{BpS(m)} \right\rceil \cdot tSymbol \\ &= 20\mu s + \left\lceil \frac{30.75 + \ell}{BpS(m)} \right\rceil \cdot 4\mu s. \end{aligned} \tag{1}$$

Note that the Bytes-per-Symbol information for PHY mode m, $BpS(m)$, is given in Table 1. Similarly, the transmission duration for an Ack frame using PHY mode m' is:

$$\begin{aligned} T_{ack}(m') &= tPLCPPreamble + tPLCP\_SIG \\ &\quad + \left\lceil \frac{14 + (16+6)/8}{BpS(m')} \right\rceil \cdot tSymbol \\ &= 20\mu s + \left\lceil \frac{16.75}{BpS(m')} \right\rceil \cdot 4\mu s. \end{aligned} \tag{2}$$

3.2 Backoff Delay

The random backoff interval is in the unit of *tSlotTime* and this random integer is drawn from a uniform distribution over the interval $[0, CW]$, where CW is the contention window size and its initial value is $aCWmin$. In the case of an unsuccessful transmission, the backoff procedure will begin at the end of the EIFS interval or the Ack timeout interval and CW is updated to $[2 \times (CW + 1) - 1]$. Once CW reaches $aCWmax$, it will remain at this value until it is reset to $aCWmin$. In the case of a successful transmission, the backoff procedure will begin at DIFS time after receiving the Ack frame and the CW value is reset to $aCWmin$ before the random backoff interval is selected. The average backoff interval before the *i*th transmission attempt or, equivalently, the $(i-1)$th retransmission attempt, denoted by $\overline{T}_{bkoff}(i)$, can be calculated by

$$\begin{aligned} &\overline{T}_{bkoff}(i) \\ &= \frac{\min[2^{i-1} \cdot (aCWmin + 1) - 1, aCWmax]}{2} \cdot tSlotTime. \end{aligned} \tag{3}$$

3.3 Effective Goodput Computation

Assume that a frame with ℓ octets data payload is to be transmitted using PHY mode m over the wireless channel with condition s. Let m' denote the PHY mode used for the corresponding Ack frame transmission and it can be determined based on m according to the rule specified in Section 3.1. In the IEEE 802.11 MAC, a frame transmission is considered successful only upon receiving the corresponding Ack frame correctly. Therefore, the probability of a successful frame transmission can be calculated by

$$P_{s,xmit}(\ell, s, m) = \left[1 - P_{e,data}(\ell, s, m)\right] \cdot \left[1 - P_{e,ack}(s, m')\right], \tag{4}$$

where $P_{e,data}(\ell, s, m)$ and $P_{e,ack}(s, m')$ are the data error probability and the Ack error probability, respectively, and their values vary with different wireless channel models.

Now, let's consider the entire delivery process of the data frame. Since the maximum number of transmission attempts to deliver the frame is n_{max}, we use vector $\hat{m}$ to denote the frame's transmission strategy, i.e., the PHY mode selections for all the potential transmission attempts. Besides, we use m_n to denote the PHY mode selected for the *n*th transmission attempt. Similarly, s_n represents the wireless channel condition during the *n*th transmission attempt and $\hat{s} = \{s_1, \cdots, s_{n_{max}}\}$ is the wireless channel condition vector. The probability of a successful frame delivery within the retry limit can then be calculated by

TABLE 2
IEEE 802.11a OFDM PHY Characteristics

Characteristics	Value	Comments
tSlotTime	$9\mu s$	Slot time
tSIFSTime	$16\mu s$	SIFS time
tDIFSTime	$34\mu s$	DIFS = SIFS + 2 × Slot
aCWmin	15	min contention window size
aCWmax	1023	max contention window size
tPLCPPreamble	$16\mu s$	PLCP preamble duration
tPLCP_SIG	$4\mu s$	PLCP SIGNAL field duration
tSymbol	$4\mu s$	OFDM symbol interval

[3] {6 Mbps, 12 Mbps, 24 Mbps} is the set of the IEEE 802.11a mandatory data rates, and it will be assumed to be the BSS basic rate set in our example and simulation.

$$P_{succ}(\ell, \hat{s}, \hat{m}) = 1 - \prod_{i=1}^{n_{max}} [1 - P_{s,xmit}(\ell, s_i, m_i)]. \tag{5}$$

By referring to Fig. 1, each successful frame transmission duration is equal to a backoff delay, plus the data transmission time, plus a SIFS time, plus the Ack transmission time, and plus a DIFS time. However, whenever the frame transmission fails, the station has to wait for an EIFS interval or an Ack timeout period, and then execute a backoff procedure before the retransmission (see Figs. 2 and 3). According to the *Specification and Description Language* formal description of the IEEE 802.11 MAC operation [1], an EIFS interval is equal to a SIFS time plus a DIFS time plus the Ack transmission time at the most robust 6 Mbps and an Ack timeout is equal to a SIFS time plus an Ack transmission time plus a Slot time. Therefore, the average transmission duration of the data frame, if delivered successfully with the transmission strategy $\hat{m}$, can be calculated by[4]

$$\begin{aligned}\mathcal{D}_{succ|\ell,\hat{s},\hat{m}} = &\sum_{n=1}^{n_{max}} P[n|succ](\ell, \hat{s}, \hat{m}) \\ &\cdot \Bigg\{ \sum_{i=2}^{n} \big[\overline{\mathcal{D}}_{wait}(i) + \overline{T}_{bkoff}(i) + T_{data}(\ell, m_i)\big] \\ &+ \overline{T}_{bkoff}(1) + T_{data}(\ell, m_1) + tSIFSTime \\ &+ T_{ack}(m'_n) + tDIFSTime \Bigg\},\end{aligned} \tag{6}$$

where $P[n|succ](\ell, \hat{s}, \hat{m})$ and $\overline{\mathcal{D}}_{wait}(i)$ are the conditional probability that the data frame is successfully delivered at the *n*th transmission attempt and the average waiting time before the *i*th transmission attempt, respectively, and they can be calculated by

$$\begin{aligned}&P[n|succ](\ell, \hat{s}, \hat{m}) \\ &= \frac{P_{s,xmit}(\ell, s_n, m_n) \cdot \prod_{i=1}^{n-1}[1 - P_{s,xmit}(\ell, s_i, m_i)]}{P_{succ}(\ell, \hat{s}, \hat{m})}\end{aligned} \tag{7}$$

and

$$\begin{aligned}\overline{\mathcal{D}}_{wait}(i) = &\frac{P_{e,data}(\ell, s_{i-1}, m_{i-1})}{1 - P_{s,xmit}(\ell, s_{i-1}, m_{i-1})} \\ &\cdot \big[tSIFSTime + T_{ack}(m'_{i-1}) + tSlotTime\big] \\ &+ \frac{\big[1 - P_{e,data}(\ell, s_{i-1}, m_{i-1})\big] \cdot P_{e,ack}(s_{i-1}, m'_{i-1})}{1 - P_{s,xmit}(\ell, s_{i-1}, m_{i-1})} \\ &\cdot \big[tSIFSTime + T_{ack}(m'_{i-1}) + tSIFSTime \\ &+ T_{ack}(1) + tDIFSTime\big].\end{aligned} \tag{8}$$

$T_{data}(\cdot)$, $T_{ack}(\cdot)$, and $\overline{T}_{bkoff}(\cdot)$ are given by (1), (2), and (3), respectively. On the other hand, the average time wasted to attempt transmission of the data frame n_{max} times in error is given by

$$\begin{aligned}&\mathcal{D}_{fail|\ell,\hat{s},\hat{m}} \\ &= \sum_{i=1}^{n_{max}} \big[\overline{T}_{bkoff}(i) + T_{data}(\ell, m_i) + \overline{\mathcal{D}}_{wait}(i+1)\big].\end{aligned} \tag{9}$$

The expected effective goodput can then be calculated by

$$\begin{aligned}\mathcal{G}(\ell, \hat{s}, \hat{m}) &= \frac{\ell}{\sum_{k=0}^{\infty}\Big[(1-P_{succ}(\ell,\hat{s},\hat{m}))^k \cdot P_{succ}(\ell,\hat{s},\hat{m}) \cdot (k \cdot \mathcal{D}_{fail|\ell,\hat{s},\hat{m}} + \mathcal{D}_{succ|\ell,\hat{s},\hat{m}})\Big]} \\ &= \frac{\ell}{\frac{1-P_{succ}(\ell,\hat{s},\hat{m})}{P_{succ}(\ell,\hat{s},\hat{m})} \cdot \mathcal{D}_{fail|\ell,\hat{s},\hat{m}} + \mathcal{D}_{succ|\ell,\hat{s},\hat{m}}} \\ &= \frac{P_{succ}(\ell,\hat{s},\hat{m}) \cdot \ell}{(1-P_{succ}(\ell,\hat{s},\hat{m})) \cdot \mathcal{D}_{fail|\ell,\hat{s},\hat{m}} + P_{succ}(\ell,\hat{s},\hat{m}) \cdot \mathcal{D}_{succ|\ell,\hat{s},\hat{m}}} \\ &= \frac{E[data](\ell, \hat{s}, \hat{m})}{E[\mathcal{D}_{data}](\ell, \hat{s}, \hat{m})},\end{aligned} \tag{10}$$

which is based on the fact that, with probability,

$$\Big[(1 - P_{succ}(\ell, \hat{s}, \hat{m}))^k \cdot P_{succ}(\ell, \hat{s}, \hat{m})\Big],$$

there is a successful data frame delivery within the retry limit after dropping the previous k frames. It can also be interpreted as follows: The expected effective goodput is equal to the ratio of the expected delivered data payload to the expected transmission time. Note that, under the constraint of the frame retry limit, the successfully delivered data payload is no longer a fixed value of ℓ. It is actually a two-value random variable and can take the value of ℓ—if delivery succeeds, with probability $P_{succ}(\ell, \hat{s}, \hat{m})$—or 0 if delivery fails. So, $E[data](\ell, \hat{s}, \hat{m}) = P_{succ}(\ell, \hat{s}, \hat{m}) \cdot \ell$ is the expected data payload delivered with the transmission strategy $\hat{m}$. Similarly, $E[\mathcal{D}_{data}](\ell, \hat{s}, \hat{m})$ is the expected transmission time spent on the frame delivery attempt, irrespective of whether it is successful or not.

4 The MPDU-Based Link Adaptation Scheme

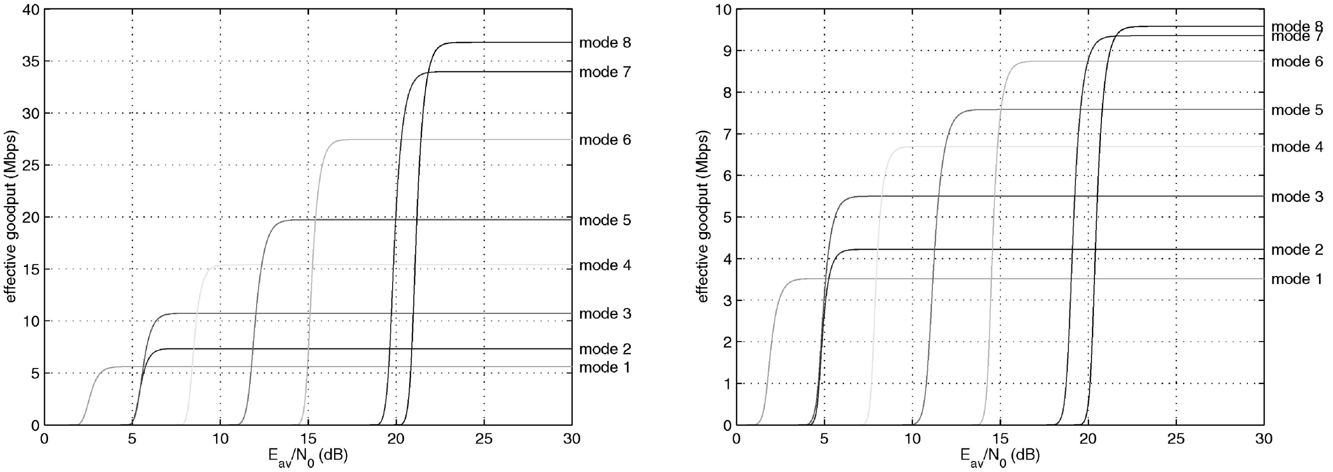
From the above goodput analysis, we observe that, to deliver a data frame over a wireless channel, the higher the PHY modes that are used ($\hat{m} \uparrow$), the shorter the expected transmission time will be ($E[\mathcal{D}_{data}] \downarrow$), but the less likely the delivery will succeed within the frame retry limit ($P_{succ}(\ell, \hat{s}, \hat{m}) \downarrow$). So, for any given set of wireless channel conditions, there exists a corresponding set of PHY modes that maximize the expected effective goodput. Such a set of PHY modes, denoted by $\hat{m}^*$, is called the best transmission strategy for the data frame delivery under the given wireless channel conditions.

Our MPDU-based link adaptation scheme to be presented in this section is an enhancement of the simple MSDU-based adaptive PHY mode selection scheme that we originally proposed in [11]. For completeness, we briefly describe the MSDU-based adaptive PHY mode selection scheme, present some numerical results, and discuss its problems and limitations.

4.1 MSDU-Based Adaptive PHY Mode Selection

The MSDU-based adaptive PHY mode selection scheme is based on a simplified goodput analysis, which assumes the constant wireless channel condition throughout the entire frame delivery period. As the name indicates, one of the key features of this scheme is that, after a wireless station makes

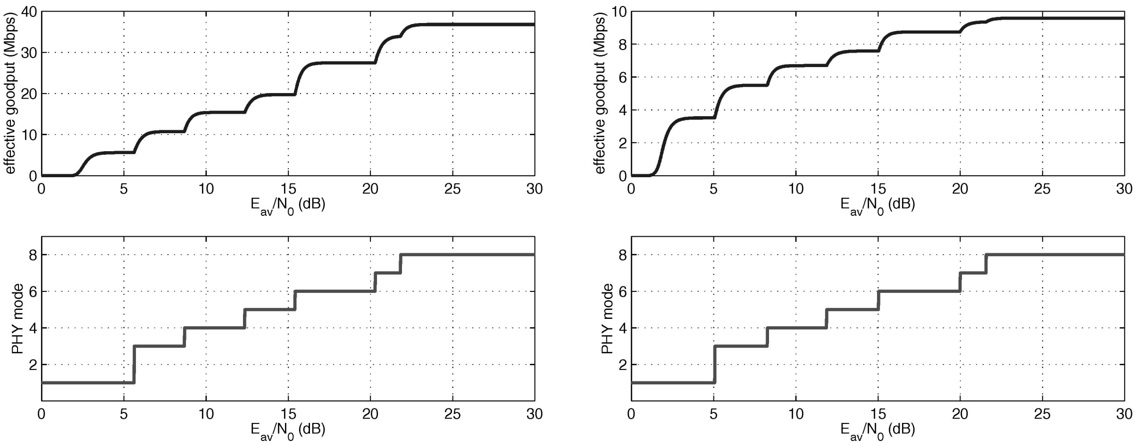
[4] The air propagation delay is neglected in the computations because it is very small (e.g., $\frac{1}{3}\mu s$, assuming 100-meter transmission range) even compared to *tSlotTime* ($9\mu s$), thus having almost no effect on the goodput analysis.



(a)

(b)

Fig. 7. Effective goodput versus SNR. (a) MSDU size: 2,000 octets and (b) MSDU size: 200 octets.



(a)

(b)

Fig. 8. MSDU-based adaptive PHY mode selection to improve the effective goodput. (a) MSDU size: 2,000 octets and (b) MSDU size: 200 octets.

the PHY mode selection and starts transmitting, the selected PHY mode will be used for all the potential retransmissions, i.e., $\hat{m}^* = \{m^*, m^*, \cdots, m^*\}$.

Figs. 7a and 7b show the numerical results of the effective goodput, according to the simplified goodput analysis and assuming the AWGN (Additive White Gaussian Noise) wireless channel noise model, for different PHY mode selections with MSDU size of 2,000 octets and 200 octets, respectively. As expected, the higher rate PHY modes result in better goodput performance in the high SNR range, while the lower rate PHY modes result in better goodput performance in the low SNR range. One interesting observation is that the effective goodput using PHY mode 3 (QPSK modulation with rate-1/2 coding) is always better than that of PHY mode 2 (BPSK modulation with rate-3/4 coding) under all SNR conditions for both frame sizes. The rationale behind this is that, although QPSK has worse error performance than BPSK, the worse performance of the rate-3/4 convolutional code compared to the rate-1/2 convolutional code has more dominant effects. Therefore, in the presence of PHY mode 3, PHY mode 2 may not be a good choice for data delivery services. Another observation from Fig. 7 is that a smaller MSDU size results in lower effective goodputs due to the fixed amount of MAC/PHY layer overheads for each transmission attempt. Fig. 8 shows the maximum effective goodput and the corresponding PHY mode selections for different SNR values. Notice that PHY mode 2 is not part of the selections, which is consistent with the fact that PHY mode 2 results in a smaller effective goodput than PHY mode 3 under all SNR conditions, as shown in Fig. 7.

Although the MSDU-based adaptive PHY mode selection scheme is simple and easy to deploy, it has some limitations. First, since the wireless channel is known to be error-prone and time-varying, it is unrealistic to assume a constant channel condition over the (long) frame delivery period that includes all the retransmissions. Second, since this scheme selects the PHY mode to deliver a data frame before the original transmission starts and sticks with this PHY mode for all the retransmissions, it cannot adapt quickly to the fast-changing wireless channel.

4.2 MPDU-Based Link Adaptation

Recognizing the limitations of the MSDU-based adaptive PHY mode selection scheme, we propose a new MPDU-based link adaptation scheme, which features two major enhancements over the MSDU-based scheme. First, it makes a more realistic assumption on the wireless channel variation that the channel condition remains unchanged over a single MPDU transmission period, which is much shorter than the entire MSDU delivery period. Second, in the middle of an MSDU delivery, the wireless station could adapt the PHY mode for the next transmission attempt if there is any variation of the wireless channel condition, i.e., $\hat{m}^* = \{m_1^*, m_2^*, \cdots, m_{n_{max}}^*\}$, where m_1^*, m_2^*, $\cdots$, and $m_{n_{max}}^*$ could be different. Since there are only finite choices for the PHY mode and the frame retry limit is also finite, we can use exhaustive search to find $\hat{m}^*$ for each set of wireless channel conditions. Then, at runtime, based on the prediction of the channel conditions during each transmission attempt, the best transmission strategy can be determined by a table lookup. This idea may sound attractive, but it works well only with the perfect knowledge on the wireless channel variation throughout the entire MSDU delivery period. Unfortunately, the wireless channel variation is unpredictable. Although there have been many schemes proposed to estimate or predict the wireless channel variation, none of them can guarantee accurate predictions. So, instead, motivated by [12] and [13], we solve the problem of finding $\hat{m}^*$ from a different angle—using the dynamic programming technique.

4.2.1 Preestablished PHY Mode Table

The basic idea of our MPDU-based link adaptation scheme is that the wireless station computes offline a table of PHY modes indexed by the system status and each entry of the table is the best PHY mode in the sense of maximizing the expected effective goodput under the corresponding system status. The *system status* is characterized by a triplet (ℓ, s, n), where ℓ is the data payload length, s is a receiver-side SNR value to quantify the wireless channel condition, and n stands for the nth transmission attempt or, equivalently, the nth frame retry count. This table is used at runtime to determine the best PHY mode for the next MPDU transmission attempt. The table entries are computed as follows.

First, consider the special case when $n = n_{max}$, i.e., when the next transmission attempt is the last chance to deliver the current frame. Obviously, the frame delivery is successful only if both data transmission and Ack transmission are error-free or result in correctable errors. Therefore, similarly to (10), the expected effective goodput, if PHY mode m is used, can be calculated by

$$\mathcal{G}(\ell, s, m, n_{max}) = \frac{E[data](\ell, s, m, n_{max})}{E[\mathcal{D}_{data}](\ell, s, m, n_{max})}, \tag{11}$$

where

$$E[data](\ell, s, m, n_{max}) = P_{s,xmit}(\ell, s, m) \cdot \ell, \tag{12}$$

and

$$\begin{aligned} &E[\mathcal{D}_{data}](\ell, s, m, n_{max}) = \\ &\quad \overline{T}_{bkoff}(n_{max}) + T_{data}(\ell, m) + tSIFSTime + T_{ack}(m') \\ &\quad + P_{s,xmit}(\ell, s, m) \cdot tDIFSTime \\ &\quad + [1 - P_{s,xmit}(\ell, s, m)] \cdot \overline{\mathcal{D}}_{wait}(n_{max} + 1). \end{aligned} \tag{13}$$

$P_{s,xmit}(\cdot)$, $\overline{T}_{bkoff}(\cdot)$, $T_{data}(\cdot)$, $T_{ack}(\cdot)$, and $\overline{\mathcal{D}}_{wait}(\cdot)$ are given by (4), (3), (1), (2), and (8), respectively. Consequently, the best PHY mode to use for the last transmission attempt and the corresponding maximum expected effective goodput are, respectively,

$$m^*(\ell, s, n_{max}) = \arg\max_{1 \le m \le 8} \mathcal{G}(\ell, s, m, n_{max}), \tag{14}$$

and

$$\begin{aligned} \mathcal{G}^*(\ell, s, n_{max}) &= \mathcal{G}(\ell, s, m^*(\ell, s, n_{max}), n_{max}) \\ &= \max_{1 \le m \le 8} \mathcal{G}(\ell, s, m, n_{max}). \end{aligned} \tag{15}$$

Now, let's consider the general case when $1 \le n < n_{max}$. Assume that PHY mode m is selected for the nth transmission attempt. The transmission is considered successful only upon receiving a positive acknowledgment; otherwise, the station has to recontend for the medium to retransmit the frame. We use $f_{R|S}(r|s)$ to denote the conditional probability density function that the wireless channel condition becomes r during the next transmission attempt, i.e., the system status becomes $(\ell, r, n+1)$, given the current wireless channel condition of s. Notice that this density function varies with elapsed time between two transmission attempts and different wireless channel variation models can be characterized by different $f_{R|S}(r|s)$s. Based on the above observations, we can construct the following recursive relation:

$$\mathcal{G}(\ell, s, m, n) = \frac{E[data](\ell, s, m, n)}{E[\mathcal{D}_{data}](\ell, s, m, n)}, \tag{16}$$

where

$$\begin{aligned} &E[data](\ell, s, m, n) \\ &\quad = P_{s,xmit}(\ell, s, m) \cdot \ell + \big[1 - P_{s,xmit}(\ell, s, m)\big] \\ &\quad \cdot \int_{-\infty}^{\infty} f_{R|S}(r|s) \cdot E[data](\ell, r, m^*(\ell, r, n+1), n+1) dr, \end{aligned} \tag{17}$$

and

$$\begin{aligned} &E[\mathcal{D}_{data}](\ell, s, m, n) = \\ &\overline{T}_{bkoff}(n) + T_{data}(\ell, m) + tSIFSTime + T_{ack}(m') \\ &\quad + P_{s,xmit}(\ell, s, m) \cdot tDIFSTime \\ &\quad + [1 - P_{s,xmit}(\ell, s, m)] \cdot \Bigg\{ \overline{\mathcal{D}}_{wait}(n+1) + \int_{-\infty}^{\infty} f_{R|S}(r|s) \\ &\quad \cdot E[\mathcal{D}_{data}](\ell, r, m^*(\ell, r, n+1), n+1) dr \Bigg\}, \end{aligned} \tag{18}$$

respectively. Similarly, the best PHY mode to use for the nth transmission attempt and the corresponding maximum expected effective goodput are

$$m^*(\ell, s, n) = \arg \max_{1 \leq m \leq 8} \mathcal{G}(\ell, s, m, n), \tag{19}$$

and

$$\mathcal{G}^*(\ell, s, n) = \mathcal{G}(\ell, s, m^*(\ell, s, n), n) = \max_{1 \leq m \leq 8} \mathcal{G}(\ell, s, m, n), \tag{20}$$

respectively. Therefore, by using the special case of $n = n_{max}$ as the boundary condition, we have fully specified the computation of the best PHY modes for different system status by (11), (14), (16), and (19).

4.2.2 Runtime Execution

Fig. 9 shows the pseudocoded algorithm of our proposed MPDU-based link adaptation scheme. Before running the program, the wireless station computes the best PHY mode for each set of data payload length (ℓ), SNR value (s), and frame retry count (n). Thus, a best PHY mode table is preestablished and ready for runtime use. The counts *succ_count* for delivered frames and *fail_count* for dropped frames are both reset to 0 and the retry count (n_{curr}) for the frame at the header of the data queue is set to 1. At runtime, the wireless station monitors the wireless channel condition and determines the current system status. Then, the wireless station selects the best PHY mode ($m_{n_{curr}}$) for the next transmission attempt by a simple table lookup. Whenever an Ack frame is received correctly within the Ack timeout, n_{curr} is reset to 1 and *succ_count* is increased; else $n_{curr} := n_{curr} + 1$. As shown in the pseudocode, if a frame cannot be successfully delivered after n_{max} transmission attempts, it will be dropped and *fail_count* is increased, and n_{curr} is reset to 1 for the next frame waiting in the data queue. Notice that, since the best PHY mode table is computed offline, there is no extra runtime computational cost for the proposed MPDU-based link adaptation scheme.

```
⟨ compute $m^*(\ell, s, n)$ for each set of $\ell$, $s$, and $n$ ⟩;
succ_count := 0;    fail_count := 0;
$\mathcal{F}$ := the frame at the header of the data queue;
$\ell_{curr}$ := DataPayloadLength($\mathcal{F}$);    $n_{curr} := 1$;
while (the data queue is non-empty) {
    $s_{curr}$ := the current wireless channel condition;
    $m_{n_{curr}} := m^*(\ell_{curr}, s_{curr}, n_{curr})$ ;
    ⟨ $\mathcal{F}$ is transmitted using PHY mode $m_{n_{curr}}$ ⟩;
    if (an Ack frame is received correctly)    then
        { $n_{curr} := 1$;    succ_count := succ_count + 1; }
    else $n_{curr} := n_{curr} + 1$;
    if ($n_{curr} > n_{max}$)    then
        { $n_{curr} := 1$;    fail_count := fail_count + 1; }
    if ($n_{curr} == 1$)    then {
        ⟨ remove the header frame from the data queue ⟩;
        ⟨ refresh $\mathcal{F}$ and $\ell_{curr}$ ⟩;
    }
}
```

Fig. 9. The MPDU-based link adaptation algorithm.

4.2.3 Implementation Issues

Fig. 10 shows a system architecture to implement the proposed MPDU-based link adaptation scheme. The *link adaptor* module provides two levels of functionality. First, the link adaptor monitors the wireless channel condition, estimates the receiver-side SNR value, and determines the current system status. The SNR value at the receiver side (in dB) is actually equal to the transmit power level (in dBm) minus the path loss (in dB) minus the noise level observed by the receiver (in dBm). Therefore, to estimate the receiver-side SNR value, it is equivalent to estimate the path loss between the transmitter and the receiver and we have developed a simple and novel scheme for this purpose. For details about this path loss estimation scheme, refer to [14]. Second, the link adaptor looks up the prebuilt best PHY mode table to determine the best transmission strategy for the next transmission attempt. The two functionalities are represented as the *SNR estimator* and the *PHY mode selector*, respectively.

One important aspect of this architecture is that the implementation of the link adaptor is transparent to the higher layers, which makes it compatible to the existing network or higher layer applications. Besides, the basic idea of link adaptation is to take advantage of different modulation schemes and FEC capabilities provided by the 802.11a PHY and no additional error correction codes need to be implemented. Therefore, the implementation of the link adaptor module should be fairly simple, thus facilitating its deployment.

5 AN EXAMPLE OF THE BEST PHY MODE TABLE

As described in the previous section, the key idea of our proposed MPDU-based link adaptation scheme is to establish a best PHY mode table indexed by the system status before the communication starts. In order to do so, a wireless station needs the following information a priori: a wireless channel noise model that determines the error performances of the PHY modes and a conditional probability density function, $f_{R|S}(r|s)$, to model the wireless channel variation. There have been many papers [15], [16], [17], [18], [19] dealing with the problem of building accurate

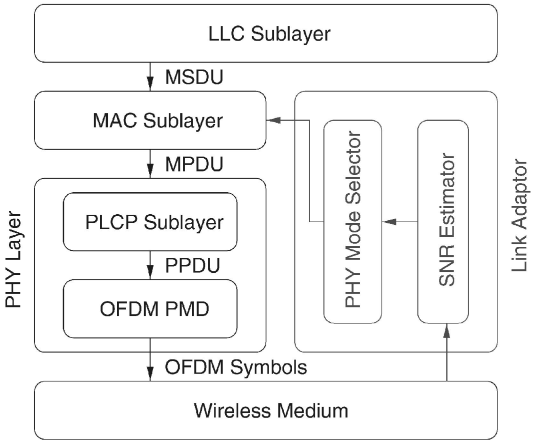


Fig. 10. System architecture for link adaptation.

wireless channel models, which, however, is not the focus of this paper. Our contribution is to propose a simple and effective link adaptation scheme by assuming the availability of such wireless channel models.

In this section, we give a detailed example of how to establish the best PHY mode table and, for simplicity, we assume an AWGN channel noise model and a two-state discrete time Markov chain channel variation model. We can establish the best PHY mode tables with other more realistic channel models as well, although the computations are much more complicated.

5.1 Error Performances of PHY Modes over the AWGN Channel

5.1.1 Bit Error Probability

The symbol error probability for an M-ary ($M = 4, 16, 64$) Quadrature Amplitude Modulation (QAM) [20] with the average SNR per symbol, s, can be calculated by

$$P_M(s) = 1 - \left[1 - P_{\sqrt{M}}(s)\right]^2, \tag{21}$$

where

$$P_{\sqrt{M}}(s) = 2 \cdot \left(1 - \frac{1}{\sqrt{M}}\right) \cdot Q\left(\sqrt{\frac{3}{M-1} \cdot s}\right) \tag{22}$$

is the symbol error probability for the $\sqrt{M}$-ary Pulse Amplitude Modulation (PAM). The Q-function is defined as

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy. \tag{23}$$

With a Gray coding, the bit error probability for an M-ary QAM can be approximated by

$$P_b^{(M)}(s) \approx \frac{1}{\log_2 M} \cdot P_M(s). \tag{24}$$

Note that the 4-ary QAM and Quadrature Phase Shift Keying (QPSK) modulation are identical. For Binary Phase Shift Keying (BPSK) modulation, the bit error probability is the same as the symbol error probability, which is given by

$$P_b^{(2)}(s) = P_2(s) = Q\left(\sqrt{2s}\right). \tag{25}$$

Obviously, the error performance of a modulation scheme varies with different SNR values.

5.1.2 Packet Error Probability

In [21], an upper bound was given on the packet error probability under the assumption of binary convolutional coding and hard-decision Viterbi decoding with independent errors at the channel input. For an h-octet long packet to be transmitted using PHY mode m, this bound is

$$P_e^m(h, s) \leq 1 - \left[1 - P_u^m(s)\right]^{8h}, \tag{26}$$

where the union bound $P_u^m(s)$ of the first-event error probability is given by

$$P_u^m(s) = \sum_{d=d_{free}}^{\infty} a_d \cdot P_d(s), \tag{27}$$

where d_{free} is the free distance of the convolutional code selected in PHY mode m, a_d is the total number of error events of weight d, and $P_d(s)$ is the probability that an incorrect path at distance d from the correct path being chosen by the Viterbi decoder. When the hard-decision decoding is applied, $P_d(s)$ is given by

$$P_d(s) = \begin{cases} \sum_{k=(d+1)/2}^{d} \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k}, & \text{if } d \text{ is odd,} \\ \frac{1}{2} \cdot \binom{d}{d/2} \cdot \rho^{d/2} \cdot (1-\rho)^{d/2} & \\ + \sum_{k=d/2+1}^{d} \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k}, & \text{if } d \text{ is even,} \end{cases} \tag{28}$$

where ρ is the bit error probability for the modulation scheme selected in PHY mode m and is given by (24) or (25). The value of a_d can be obtained either from the transfer function or by a numerical search [22].

Therefore, $P_{e,data}(\ell, s, m)$, the data error probability, and $P_{e,ack}(s, m')$, the Ack error probability, used in Section 3 to compute $P_{s,xmit}(\ell, s, m)$, the probability of a successful frame transmission, can be calculated by

$$\begin{aligned} & P_{e,data}(\ell, s, m) \\ & = 1 - \left[1 - P_e^1(24/8, s)\right] \cdot \left[1 - P_e^m(28 + (16+6)/8 + \ell, s)\right] \\ & = 1 - \left[1 - P_e^1(3, s)\right] \cdot \left[1 - P_e^m(30.75 + \ell, s)\right], \end{aligned} \tag{29}$$

and

$$\begin{aligned} & P_{e,ack}(s, m') \\ & = 1 - \left[1 - P_e^1(24/8, s)\right] \cdot \left[1 - P_e^{m'}(14 + (16+6)/8, s)\right] \\ & = 1 - \left[1 - P_e^1(3, s)\right] \cdot \left[1 - P_e^{m'}(16.75, s)\right], \end{aligned} \tag{30}$$

respectively. Here, $P_e^1(3, s)$ is the packet error probability of the PLCP SIGNAL field because it is 24-bit long and always transmitted with BPSK modulation and rate-1/2 convolutional coding, i.e., PHY mode 1. $P_e^1(\cdot)$, $P_e^m(\cdot)$, and $P_e^{m'}(\cdot)$ are calculated by (26).

5.2 Wireless Channel Variation Model

Fig. 11 shows a two-state discrete time Markov chain to model the wireless channel variation. The wireless channel could be in either a *good* or *bad* state. When the wireless channel is in the *good* state, the corresponding SNR at each time instant is taken from a uniform distribution in the range of 15 to 30 dB, and when the wireless channel is in the *bad* state, the SNR value is drawn from the range of 0 to 15 dB. The time spent in the *good* and *bad* states are taken from exponential distributions with rates $\frac{1}{\mu_g}$ and $\frac{1}{\mu_b}$, respectively. Therefore, the state transition probabilities $t_{g,b}$ and $t_{b,g}$ are equal to $\frac{\mu_b}{\mu_g+\mu_b}$ and $\frac{\mu_g}{\mu_g+\mu_b}$, respectively. Different values of $t_{b,g}$ correspond to different wireless channel variation patterns. For example, if $t_{b,g}$ is close to 0 (1), the wireless channel tends to stay in the *bad* (*good*) state for most of the time.

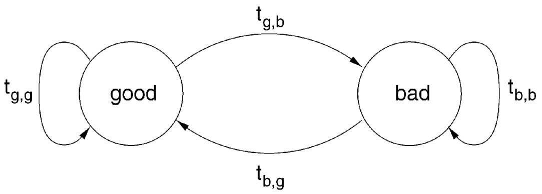


Fig. 11. A two-state discrete time Markov chain to model the wireless channel variation.

5.3 The Best PHY Mode Table

Under the above assumptions for the wireless channel, we can compute the best PHY mode for each set of data payload length (ℓ), SNR value (s), and frame retry count (n), by following the recursive steps specified in Section 4.2.1. Fig. 12 shows an example of the best PHY mode table when $t_{b,g}$ is 0.8 in the wireless channel variation model, the data payload length (ℓ) is 2,000 octets, and the frame retry limit (n_{max}) is seven. The real numbers along the X-axis are the SNR values (s) and the integer numbers along the Y-axis are the frame retry counts (n). Notice that the boundaries between PHY mode selections shift for different frame retry counts and, in general, the smaller the frame retry count, the more aggressive the transmission attempt is. For example, when the SNR is 21 dB, PHY mode 6 is selected for the last transmission attempt, while PHY mode 7 is selected for the first transmission attempt. The shift margins depend on the wireless channel variation model/pattern and the wireless channel noise model. Under certain wireless channel models, the shifts could be more significant.

6 Performance Evaluation and Discussion

In this section, we evaluate the effectiveness of the proposed MPDU-based link adaptation scheme using simulation. As specified in the IEEE 802.11 standard, the length of an MSDU must be less than or equal to 2,304 octets (see Clause 6.2.1.1.2 in [1]), and an MSDU shall be broken into fragments if its size exceeds the fragmentation threshold after adding the MAC overhead—which is 28 octets in total, as shown in Section 3.1. In the simulation, we set both the fragmentation threshold and *dot11RTSShreshold* to be $2,332\ (= 2,304 + 28)$ octets; therefore, each MSDU is transmitted without fragmentation or RTS/CTS support and the maximum number of transmission attempts is seven (the default value of *dot11ShortRetryLimit*). The BSS basic rate set is the set of the IEEE 802.11a mandatory data rates, i.e., {6 Mbps, 12 Mbps, 24 Mbps}. The eight different PHY modes of the IEEE 802.11a PHY and the related parameters used in the simulation are listed in Tables 1 and 2, respectively. Moreover, the wireless channel is modeled by the AWGN channel noise model and the two-state discrete time Markov chain channel variation model, as described in the previous section.

The six testing schemes under consideration are: three Single-Mode schemes using PHY mode 1 (SM-1), mode 5 (SM-5), and mode 8 (SM-8), respectively, the AutoRate Fallback (ARF) scheme used by Lucent Technologies, the simple MSDU-based adaptive PHY mode selection scheme (LA-1), and the proposed MPDU-based link adaptation scheme (LA-2). For each of the testing schemes, an experiment is repeated 100 times to estimate the average goodput, the frame drop rate, and the average number of transmission attempts per MSDU delivery. At each experiment, the receiver requests a data delivery service of 10,000 MSDUs from the transmitter and the length of each MSDU is 2,000 octets. Testing schemes are compared with each other under different wireless channel variation patterns, i.e., different $t_{b,g}$ values.

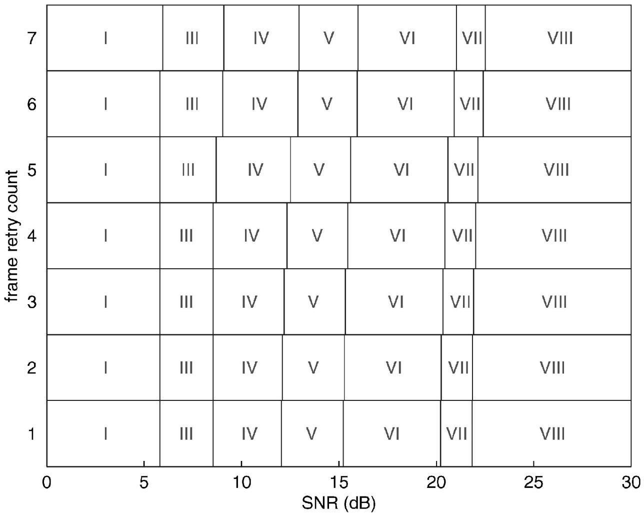


Fig. 12. An example of the best PHY mode table.

6.1 "Timeout" Factor in ARF

In our simulation of ARF, we follow as closely as possible the protocol specifications in [4]. Recall that, in ARF, the PHY rate is raised only when either of the following two conditions holds: The number of consecutive successful frame transmissions reaches 10 or a preset timer expires. Clearly, different values of the timer will certainly affect the performance of ARF and this timeout value was not explicitly specified in [4]. So, similarly to [5], we experiment with several timeout values to determine a reasonable value for our ARF simulation. Instead of setting the timer in seconds (i.e., the absolute time unit), we use a virtual timer, which accounts for the number of transmission attempts, as follows: Each wireless station keeps a count for the number of transmission attempts and this counter is reset to zero after observing two consecutive Ack failures or 10 consecutive Ack successes. When the count reaches a preset threshold, i.e., the "timeout," the PHY rate is raised.

The results of the experiments are plotted in Fig. 13, which shows the average goodput as a function of the timeout value for several different wireless channel variation patterns. It appears that ARF is relatively insensitive to the choice of the timeout and the peak goodput occurs in the timeout range of 5 to 15, beyond which there is little performance change, but below which there is a noticeable drop. The drop can be attributed to the greater frequency at which transmission attempts fail due to aggressive PHY rate increases triggered by shorter timeouts. In general, the timeout should be frequent enough to be responsive to the variations of the wireless channel, but not too frequent because too many transmission failures impact the goodput performance significantly. On the other hand, when the

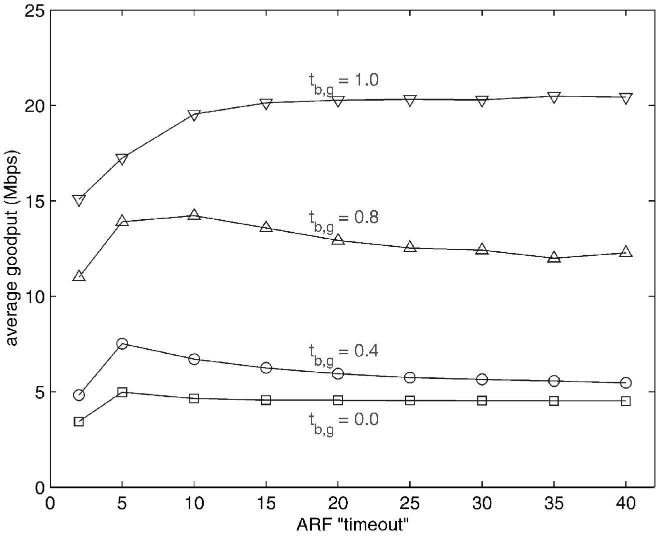


Fig. 13. Goodput comparison of ARF for various "timeout" values.

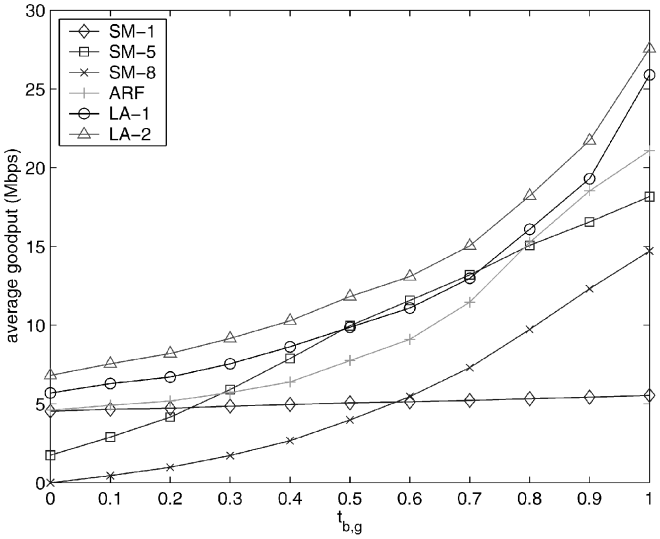


Fig. 14. Goodput comparison of the testing schemes.

timeout is less than (equal to) 10, the PHY rate increases due to 10 consecutive transmission successes, which is one of the kernel ideas of the ARF scheme, never (rarely) occurs. Therefore, it is not appropriate to use a timeout of less than or equal to 10. Based on the above observations, we choose the timeout value to be 15 in our ARF simulation.

6.2 Performance Evaluation

First, the testing schemes are evaluated using the average goodput and the results under different wireless channel variation patterns are plotted in Fig. 14. Clearly, the goodput varies with different channel variation patterns and the link adaptation schemes outperform the single-mode schemes in most cases. We have several observations from this figure.

SM-1 is the most conservative scheme of all. It uses the most robust PHY mode (i.e., PHY Mode 1) for all the transmission attempts. When the wireless channel tends to stay in the *good* state for most of the time, i.e., when $t_{b,g}$ is close to 1.0, SM-1 results in the lowest average goodput because of the limited transmission rate of PHY mode 1. However, since PHY mode 1 is the most robust, most frames are successfully delivered with a very few transmission attempts, regardless of the wireless channel variation pattern. As a result, we observe an almost flat goodput curve for SM-1. On the other hand, SM-8 is the most aggressive scheme, which uses PHY mode 8 for all the transmission attempts. To no one's surprise, SM-8 has the lowest average goodput when the wireless channel tends to stay in the *bad* state for most of the time, i.e., when $t_{b,g}$ is close to 0.0. This is because the poorest error correcting capability of PHY mode 8 results in a large number of frame retransmissions and delivery failures. For the extreme case of $t_{b,g} = 0.0$, all the transmission attempts fail and the average goodput drops to zero. Another single-mode scheme, SM-5, can be viewed as a compromise between SM-1 and SM-8.

The goodput results for three link adaptation schemes are plotted as the cross points (ARF), the circle points (LA-1), and the triangle points (LA-2) in Fig. 14. As expected, all of them result in higher goodputs than SM-1 and SM-8 because of their adaptive use of efficient high PHY modes in the *good* state and robust low PHY modes in the *bad* state. ARF appears to be the most conservative link adaptation scheme of the three. One interesting observation is that, if a link adaptation scheme is not designed carefully—e.g., ARF is purely heuristic and LA-1 cannot adjust the PHY mode between frame retransmissions, then it might even result in a worse performance than some judiciously selected single-mode schemes. For example, when $t_{b,g}$ is between 0.5 and 0.7, SM-5 outperforms both ARF and LA-1. On the other hand, since LA-2 is MPDU-based and is able to adapt the PHY mode for each transmission attempt according to the most up-to-date system status, it is guaranteed to have the best performance. As shown in Fig. 14, LA-2 is significantly better than ARF and outperforms LA-1 by about 10 percent in terms of the average goodput.

Second, the testing schemes are compared using the frame drop rate or, equivalently, the average number of dropped frames out of the 10,000 frames waiting for delivery. The results are listed in Table 3. Notice that SM-1, ARF, and LA-2 are all perfect in terms of frame drop. However, the rationales are different. Unlike SM-1, which sticks with the most robust PHY mode, or ARF, which applies a conservative policy and is reluctant to increase the

TABLE 3
Comparison for Average Number of Dropped MSDUs

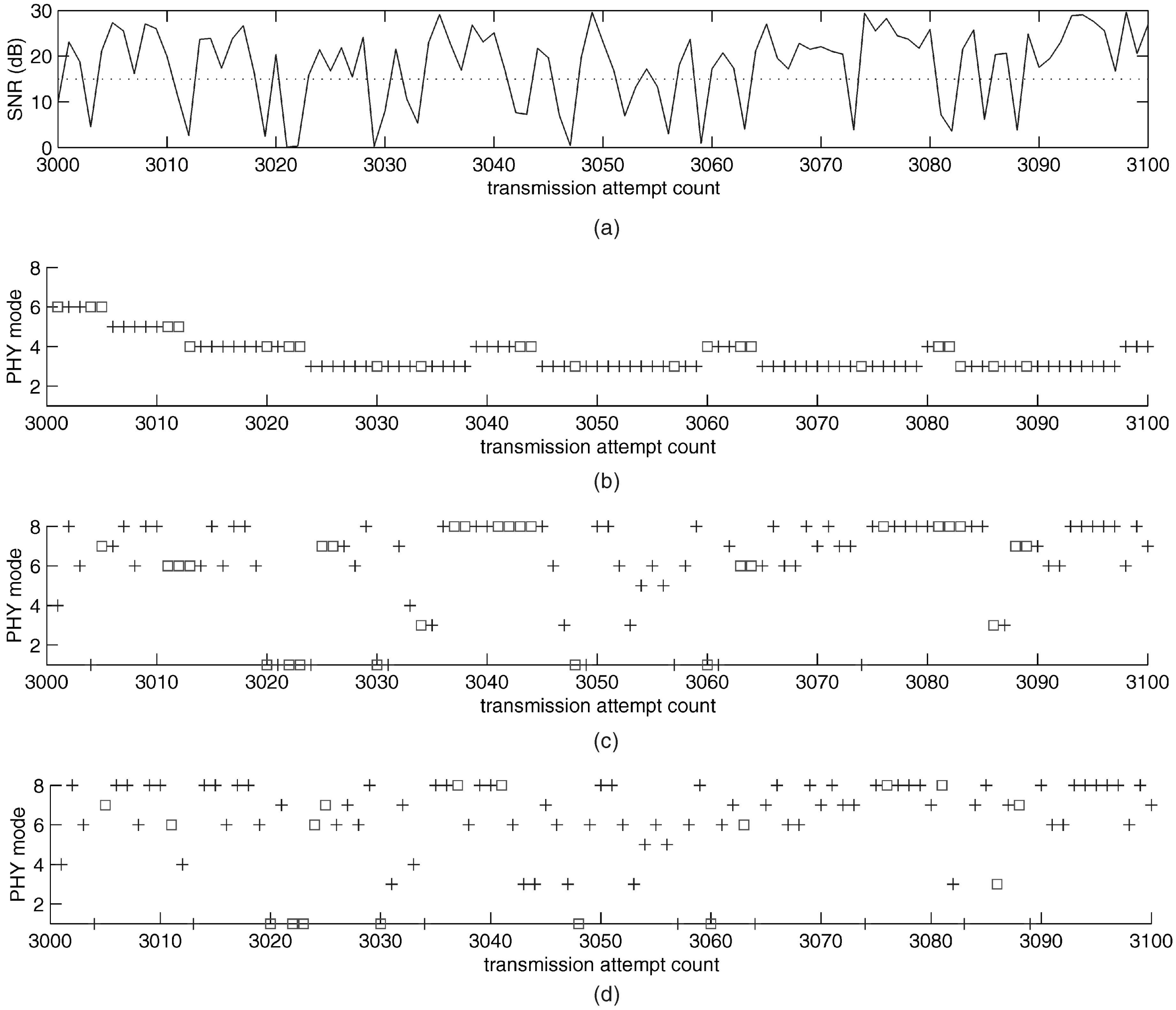
	average number of dropped frames					
$t_{b,g}$	SM-1	SM-5	SM-8	ARF	LA-1	LA-2
0.0	0	2170	10000	1	93	0
0.1	0	1050	6634	0	118	0
0.2	0	535	4461	0	99	0
0.3	0	223	2811	0	81	0
0.4	0	63	1766	0	60	0
0.5	0	21	1002	0	41	0
0.6	0	5	605	0	28	0
0.7	0	0	330	0	17	0
0.8	0	0	170	0	8	0
0.9	0	0	72	0	6	0
1.0	0	0	39	0	2	0

PHY rate, LA-2 presents the excellent frame drop performance due to its quick adaptability to the variations of the wireless channel. We have two observations about SM-5, SM-8, and LA-1. First, in general, the average number of dropped frames increases as the wireless channel gets worse for these three schemes. In particular, when $t_{b,g} = 0.0$, SM-8 results in 10,000 dropped frames, which is consistent with the zero average goodput observed in Fig. 14. Second, LA-1 results in fewer dropped frames when $t_{b,g} = 0.0$ rather than when $t_{b,g} = 0.1$ or 0.2. This counterintuitive observation is surprising at first sight, but rather reasonable. Recall that LA-1 is unable to adjust the PHY mode for retransmissions when the wireless channel condition fluctuates. So, the frame drop performance of LA-1 is affected by not only the wireless channel condition, but the wireless channel variation as well. When $t_{b,g} = 0.0$, since the wireless channel tends to stay in the *bad* state for most of the time, there are relatively fewer SNR variations than when $t_{b,g} = 0.1$ or 0.2. As a result, LA-1 shows better frame drop performance under this wireless channel variation pattern. However, since the wireless channel can get really bad in this case, we still observe a significant number of dropped frames.

TABLE 4
Comparison for Average Number of Transmission Attempts per MSDU Delivery

	average number of attempts (*avg_att*)					
$t_{b,g}$	SM-1	SM-5	SM-8	ARF	LA-1	LA-2
0.0	1.214	4.001	7.000	1.349	1.430	1.279
0.1	1.183	3.275	5.909	1.333	1.387	1.253
0.2	1.166	2.756	5.075	1.327	1.381	1.239
0.3	1.137	2.307	4.326	1.320	1.339	1.210
0.4	1.113	1.987	3.741	1.312	1.300	1.192
0.5	1.090	1.699	3.228	1.307	1.274	1.169
0.6	1.078	1.530	2.835	1.314	1.247	1.154
0.7	1.059	1.383	2.523	1.315	1.213	1.138
0.8	1.037	1.233	2.196	1.316	1.178	1.116
0.9	1.020	1.134	1.984	1.294	1.163	1.101
1.0	1.000	1.040	1.818	1.275	1.134	1.087

Our previous observations and conclusions about the six testing schemes can be further justified by comparing the average number of transmission attempts per MSDU delivery (*avg_att*), as shown in Table 4. Notice that *avg_att*



(a)

(b)

(c)

(d)

Fig. 15. Adaptability comparison of three link adaptation schemes. (a) Wireless channel variation. (b) AutoRate Fallback (ARF). (c) MSDU-based adaptive PHY mode selection. (d) MPDU-based link adaptation.

varies sharply for the single-mode schemes, while link adaptation schemes have smaller *avg_att* values which do not change much with the wireless channel variation pattern. This is reasonable since one of the kernel ideas of link adaptation is to select more robust PHY modes when the wireless channel condition gets worse.

Finally, we use Fig. 15 to illustrate the behavior of the three testing link adaptation schemes and the fast adaptability of LA-2 to the wireless channel variations can be seen more clearly in this figure. The wireless channel variation pattern used to produce the results is $t_{b,g} = 0.8$. Fig. 15a shows the SNR values observed during each of the transmission attempts from #3,000 to #3,100 and the PHY mode selections by ARF, LA-1, and LA-2 are shown in Figs. 15b, 15c, and 15d, respectively. Note that, in these figures, the cross points stand for the successful transmission attempts and a square point represents a transmission failure. We have two observations. First, ARF is slow to adapt to the changes in SNR, as evidenced by the relative dissimilarity between Figs. 15a and 15b. It is clear that ARF is a conservative link adaptation scheme, where PHY modes 3 and 4 are the most popular choices. Second, LA-1 and LA-2 are better at reacting and adapting to the wireless channel variations and their PHY mode selections basically follow the changes in SNR. However, in LA-1, if an MSDU delivery starts right before the wireless channel turns *bad* (e.g., at #3,011 and #3,041), it has to stick with the original PHY mode selection for all the retransmissions, thus resulting in more transmission failures than LA-2.

7 Conclusion

In this paper, we present a generic method to analyze the goodput performance of an IEEE 802.11a DCF system. We express the expected effective goodput as a closed-form function of the the data payload length, the frame retry count, the wireless channel condition, and the selected data transmission rate. Based on the theoretical analysis, we propose a novel MPDU-based link adaptation scheme, assuming the availability of the wireless channel models. It is a table-driven approach and the key idea is to preestablish a PHY mode table indexed by the system status triplet, which consists of the data payload length, the wireless channel condition, and the frame retry count. Each entry of the table is the best PHY mode in the sense of maximizing the expected effective goodput under the corresponding system status and is computed by applying the dynamic programming technique. At runtime, a wireless station determines the most appropriate PHY mode for the next transmission attempt by a simple table lookup, using the most up-to-date system status as the index. Besides, since the best PHY mode table is computed offline, there is no extra runtime computational cost for using our proposed scheme.

We give a detailed example of how to establish a best PHY mode table under the assumptions of the AWGN channel noise model and the two-state discrete time Markov chain channel variation model. Note that our scheme works well with any wireless channel models, although, with more realistic models, the computation of the best PHY mode table is much more complicated. Finally, we compare the performance of the proposed scheme against three single-mode schemes, the AutoRate Fallback (ARF) scheme used by Lucent Technologies, and the simple MSDU-based adaptive PHY mode selection scheme. Simulation results show that the proposed scheme consistently outperforms others in terms of the average goodput, the frame drop rate, and the average number of transmission attempts per MSDU delivery.

Acknowledgments

The work reported in this paper was supported in part by US Air Force Office of Scientific Research under Grant No. F49620-00-1-0327.

References

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

[2] IEEE 802.11b, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 2.4GHz Band,* supplement to IEEE 802.11 Standard, Sept. 1999.

[3] IEEE 802.11a, *Part 11: Wireless LAN, Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5GHz Band,* supplement to IEEE 802.11 Standard, Sept. 1999.

[4] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical J.,* pp. 118-133, Summer 1997.

[5] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," *Proc. ACM MobiCom '01,* pp. 236-251, July 2001.

[6] O. Queseth, F. Gessler, and M. Frodigh, "Algorithms for Link Adaptation in GPRS," *Proc. IEEE Vehicular Technology Conf. (VTC '99),* vol. 2, pp. 943-947, May 1999.

[7] J. Khun-Jush, P. Schramm, U. Wachsmann, and F. Wenger, "Structure and Performance of the HIPERLAN/2 Physical Layer," *Proc. IEEE Vehicular Technology Conf. (VTC '99),* vol. 5, pp. 2667-2671, Sept. 1999.

[8] Z. Lin, G. Malmgren, and J. Torsner, "System Performance Analysis of Link Adaptation in HiperLAN Type 2," *Proc. IEEE Vehicular Technology Conf. (VTC '00),* vol. 4, pp. 1719-1725, May 2000.

[9] J. Habetha and D. Calvo de No, "New Adaptive Modulation and Power Control Algorithms for HIPERLAN/2 Multihop Ad Hoc Networks," *Proc. European Wireless,* Sept. 2000.

[10] P. Lettieri and M.B. Srivastava, "Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency," *Proc. IEEE INFOCOM '98,* vol. 2, pp. 564-571, Mar. 1998.

[11] D. Qiao and S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," *Proc. IEEE Int'l Conf. Comm. (ICC '01),* vol. 7, pp. 1995-2000, June 2001.

[12] M. Elaoud and P. Ramanathan, "Adaptive Use of Error-Correcting Codes for Real-Time Communication in Wireless Networks," *Proc. IEEE INFOCOM '98,* vol. 2, pp. 548-555, Mar. 1998.

[13] D. Qiao and K.G. Shin, "A Two-Step Adaptive Error Recovery Scheme for Video Transmission over Wireless Networks," *Proc. IEEE INFOCOM '00,* vol. 3, pp. 1698-1704, Mar. 2000.

[14] D. Qiao, S. Choi, A. Soomro, and K.G. Shin, "Energy-Efficient PCF Operation of IEEE 801.11a Wireless LAN," *Proc. IEEE INFOCOM '02,* pp. 580-589, June 2002.

[15] P. Bergamo, D. Maniezzo, A. Giovanardi, G. Mazzini, and M. Zorzi, "An Improved Markov Chain Description for Fading Processes," *Proc. IEEE Int'l Conf. Comm. (ICC '02),* vol. 3, pp. 1347-1351, Apr. 2002.

[16] J.-P. Ebert and A. Willig, "A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation," TKN Technical Report TKN-99-002, Technical Univ. Berlin, Telecomm. Networks Group, Berlin, Germany, Mar. 1999.

[17] M. Zorzi, R.R. Rao, and L.B. Milstein, "Error Statistics in Data Transmission over Fading Channels," *IEEE Tran. Comm.,* vol. 46, no. 11, pp. 1468-1477, Nov. 1998.

[18] H.S. Wang and P.C. Chang, "On Verifying the First-Order Markovian Assumption for a Rayleigh Fading Channel Model," *IEEE Trans. Vehicular Technology,* vol. 45, pp. 353-357, May 1996.

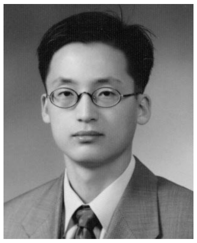
[19] T.S. Rappaport, *Wireless Communications: Principle and Practice.* Upper Saddle River, N.J.: Prentice Hall, 1996.

[20] J.G. Proakis, *Digital Communications,* third ed. New York: McGraw-Hill, 1995.

[21] M.B. Pursley and D.J. Taipale, "Error Probabilities for Spread-Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding," *IEEE Trans. Comm.,* vol. 35, no. 1, pp. 1-12, Jan. 1987.

[22] D. Haccoun and G. Begin, "High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding," *IEEE Trans. Comm.,* vol. 37, no. 11, pp. 1113-1125, Nov. 1989.

Daji Qiao (S'97) received the BE degree in automatic control theory and applications from Tsinghua University, Beijing, China, in 1994, and the MS degree in computer engineering from The Ohio State University, Columbus, Ohio, in 1998. He is currently working toward a PhD degree in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor. His research interests include QoS support, algorithm development, protocol design, and performance evaluation for wireless/mobile networks. He is a student member of the IEEE.



Sunghyun Choi (S'96-M'00) received the BS (summa cum laude) and MS degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1992 and 1994, respectively, and the PhD degree from the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, in September 1999. He is an assistant professor in the School of Electrical Engineering, Seoul National University (SNU), Seoul, Korea. Before joining SNU in September 2002, he was with Philips Research USA, Briarcliff Manor, New York, as a senior member of the research staff for three years. His current research interests are in the area of wireless/mobile networks with an emphasis on the QoS guarantee and adaptation, in-home multimedia networks, wireless LAN and PAN, MAC protocols, connection and mobility management, and multimedia CDMA. He authored/coauthored more than 25 technical papers and book chapters in the areas of wireless/mobile networks and communications. He is also an active participant and contributor of the IEEE 802.11 WLAN standardization committee. Dr. Choi was a recipient of the Korea Foundation for Advanced Studies Scholarship and the Korean Government Overseas Scholarship during 1997-1999 and 1994-1997, respectively. He was also a winner of the Humantech Thesis Prize from Samsung Electronics in 1997. He is a member of the IEEE.

Kang G. Shin (S'75-M'78-SM'83-F'92) received the BS degree in electronics engineering from Seoul National University, Seoul, Korea, in 1970 and both the MS and PhD degrees in electrical engineering from Cornell University, Ithaca, New York, in 1976 and 1978, respectively. He is the Kevin and Nancy O'Connor Professor of Computer Science and founding director of the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan. His current research focuses on QoS-sensitive networking and computing as well as on embedded real-time OS, middleware, and applications, all with an emphasis on timeliness and dependability. He has supervised the completion of 42 PhD theses and authored/coauthored more than 500 technical papers and numerous book chapters in the areas of distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He has coauthored (jointly with C.M. Krishna) a textbook, *Real-Time Systems* (McGraw-Hill, 1997). He received the Outstanding IEEE Transactions on Automatic Control Paper Award in 1987, Research Excellence Award in 1989, Outstanding Achievement Award in 1999, Service Excellence Award in 2000, and Distinguished Faculty Achievement Award in 2001 from The University of Michigan. He also coauthored papers with his students which received the Best Student Paper Awards from the 1996 IEEE Real-Time Technology and Application Symposium and the 2000 USENIX Technical Conference. From 1978 to 1982, he was on the faculty of Rensselaer Polytechnic Institute, Troy, New York. He has held visiting positions at the US Airforce Flight Dynamics Laboratory, AT&T Bell Laboratories, Computer Science Division within the Department of Electrical Engineering and Computer Science at the University of California at Berkeley, and the International Computer Science Institute, Berkeley, California, IBM T.J. Watson Research Center, and Software Engineering Institute at Carnegie Mellon University. He also chaired the Computer Science and Engineering Division, EECS Department, at The University of Michigan for three years beginning in January 1991. He is a fellow of the IEEE and the ACM, and a member of the Korean Academy of Engineering, was the general chair of the 2000 IEEE Real-Time Technology and Applications Symposium, the program chair of the 1986 IEEE Real-Time Systems Symposium (RTSS), the general chair of the 1987 RTSS, the guest editor of the 1987 August special issue of *IEEE Transactions on Computers* on real-time systems, a program cochair for the 1992 International Conference on Parallel Processing, and served on numerous technical program committees. He also chaired the IEEE Technical Committee on Real-Time Systems during 1991-1993, was a distinguished visitor of the IEEE Computer Society, an editor of the *IEEE Transactions on Parallel and Distributed Computing*, and an area editor of the *International Journal of Time-Critical Computing Systems*, *Computer Networks*, and *ACM Transactions on Embedded Systems*.